# Pineview Reservoir Code Challenge

**USU Code Camp**

July 10, 2014

## The Challenge

Pineview reservoir is located on the Ogden River above Ogden, Utah. The reservoir supplies irrigation water, generates hydropower, and protects Ogden residents from floods. We challenge you to develop and code a rule to tell operators how they should store/release water from the reservoir to maximize the benefits (measured in $$) of water use. Your rule must be general and prescribe the volume of water to release each month given the (i) available water in the reservoir, (ii) reservoir capacity, and (iii) maximum water volumes that can be sent through the hydropower turbine and delivered to agricultural users. We will award prizes to the group that codes a release rule that generates the largest benefits. We will also award prizes to the groups that code the most concise release rule (fewest lines of code – while still specifying a valid release rule) and most clearly document their code. Are you ready for this code challenge?

## Background Information on Pineview Reservoir

Figure 1 shows the Pineview reservoir system. Note:

- Water sent through the hydropower turbine is returned to the river below the dam.
- Ogden City is below the dam and sees all flow from the combined releases.
- The irrigation diversion is below Ogden.
- The reservoir capacity is 110,150 acre-feet. The hydropower turbine and irrigation diversion capacities are 18,149 and 11,887acre-feet per month.
- Each month over our 56-year study period, there is inflow to the reservoir and evaporation loss from the reservoir pool surface. Evaporation depends on the month and reservoir surface area. Together, the inflow, evaporation, and water remaining in the reservoir at the end of the prior month determine the water available for release this month.

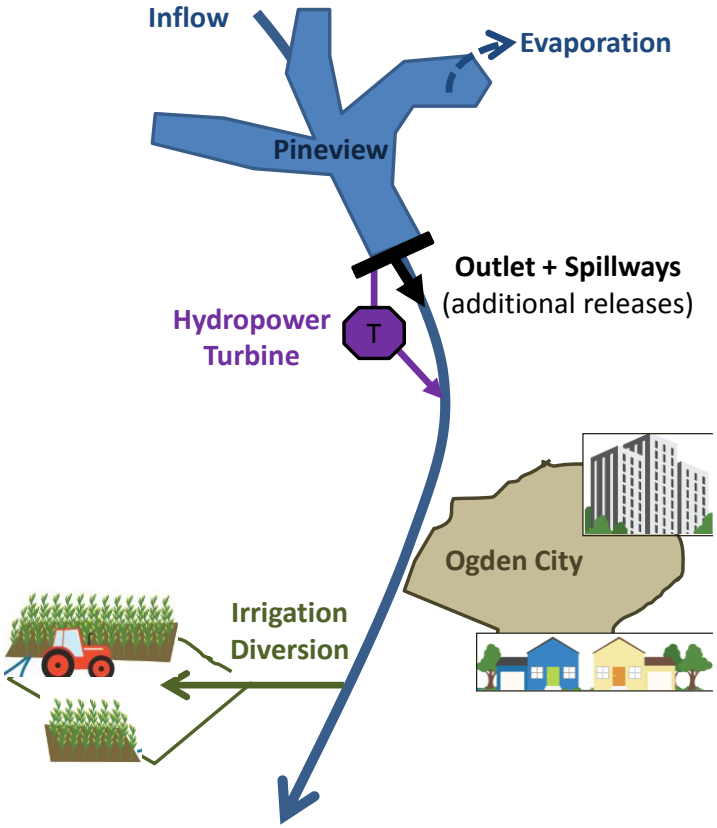


Figure 1. Pineview reservoir schematic

- Your release cannot exceed the available water! But you must release sufficient water to prevent overtopping the dam!

**Benefits of Water Uses ($$$$$)**

Hydropower generation, irrigation, and flood protection have different timings and benefits/damages associated with each water use.

1) Hydropower generation. Dam operators sell the electricity they generate to power companies. These sales revenues are the sale price ($/kW-hr) multiplied by the energy produced (kW-hr). In summer (June to August), the sale price is twice the price in other months. Also, the energy produced depends on both the flow through the turbine and net head on the turbine (water surface in the reservoir). More flow and a higher water surface (more water stored) generate more energy!
2) Irrigation. Farmers require supplemental water from April to September to grow crops that they harvest and sell in late September. Generally, more water increases the crop yield. But, there are also diminishing returns as soils get water logged. Further, if insufficient water is applied in a month, the crops die and there is no harvest!
3) Flood protection. Existing channels and levees protect Ogden residents against some floods. Existing levees can hold back river flows of 85,000 acre-feet per month. Above this amount, only a few adjacent and low-value properties are flooded and damaged. However, flows of 96,500 acre-feet per month flood numerous homes and businesses, and the damages rapidly escalate as flows further increase. Flood damages are subtracted from the other benefits.

**Your Reservoir Release Function**

Below (Figure 2) is the python reservoir release rule function in the file **releaserule.py** you will use and improve upon. The function takes as inputs the current month, reservoir capacity, available water, and capacities of the hydropower turbine and irrigation diversion. The function returns two separate, but related components of the overall reservoir release: the 1) release through the turbine and 2) release through the other outlets. The example function shown in Figure 2 is a very simple rule that never uses the turbine and only releases water to prevent overtopping the dam. It works but incurs many flood damages (negative benefits). Can you improve this rule?

**Test Your Release Rule and View Results**

1. Download and unzip the file **CodeCamp.zip** file from http://cloudshare.iutahepscor.org. You will need to log in to access the file. The username is "codecamp" and the password is "codecamp123!". Hover over the zip file until the "Download" link appears and then download and save the zip file to the desktop of your computer.
2. Once you have saved the zip file to the desktop, double click on it to open the zip file. This will open a program called WinZip. Click the "Extract" button on the toolbar. In the window that pops up, click on "Desktop" in the left column and then click on the

"Extract" button. This should create a new folder called "CodeCamp" on your desktop. Close WinZip when you finish this step.

3. Double click on the "CodeCamp" folder to open a file browser so you can see the contents of the folder.
4. In the folder, open the file **releaserule.py** with the IDLE Python 2.6 editor. Right click on the file "releaserule.py" and in the context menu that pops up, select "Edit with Idle."
5. Program your new release rule as discussed in the previous section.
6. Select **File**=>**Save** (or press Ctrl-S) to save the file **releaserule.py**.
7. Open the file **pineview.py** in the IDLE editor (see step 2 for how to open a file). Select **Run**=>**Run Module** (or press F5 on the keyboard). The model will run and test your release rule over a 56-year period of historical flows (1950 to 2006). When the model finishes, the Idle console window will print a message that the model finished.
8. Double click on the Excel file **ViewPineviewResults.xls** to view results. If you get a warning about Microsoft Office not being activated, just click on the "Close" button. You may also get some additional security warnings.
    a. Just below the Ribbon to the left of the text *Security Warning! Macros have been disabled*, click the button labeled **Enable Content**.
    b. Make the *Overview* worksheet active and click the button labeled **Load Results**. Excel will load results from the file output.csv.
    c. The Excel file shows total revenues over the 56-year period in cells D8 and D9. View plots and tables of results by month and year on the other worksheets.
9. To test a new rule, repeat Steps #2 to 6.
10. To preserve your rule and results for future reference, save the releaserule.py and Excel files under different names. Select **File**=>**Save As**.
11. ***To submit your final release rule***, log into http://cloudshare.iutahepscor.org as in step 1 above. Click on the "GroupReleaseRules" folder. You can drag and drop your release rule file into this folder, but make sure you use a file name that uniquely identifies your group so you don't copy over the files uploaded by other groups (e.g., name your file something like "releaserule_teamaggies.py").

```python
def applyReleaseRule(month, reservoirCapacity, availableWater, turbineCapacity, irrigationCapacity):
    #This function specifies the reservoir release rule.
    #INPUTS
    #    month = the current month of the simulation (a number from 1 - 12)
    #    reservoirCapacity = the maximum capacity of the reservoir (acre-ft)
    #    availableWater = the water volume available for release (acre-ft).
    #            This includes storage from the previous month + inflows - evaporation losses
    #    turbineCapacity = the maximum volume of water that can be sent through the turbine to generate power (acre-f
    #    irrigationCapacity = the maximum volume of water that can be diverted out of the river
    #            for agricultural use (acre-ft)
    #OUTPUTS
    #    hydropowerRelease = amount of water released through the turbine for generating electricity (acre-ft)
    #    additionalRelease = amount of water released from the dam in addition to the hydropower release (acre-ft)
    #
    #In your release rule function, you must specify hydropowerRelease and additionalRelease
    #         -- even if you set them to 0.
    #
    #Start editing below this line to create your release rule
    #-----------------------------------------------------------------------------------------------
    #BEGIN EDITABLE SECTION

    #This is a simple and functional release rule, but it performs poorly
    #It specifies that water is only released to prevent overtopping of the dam
    hydropowerRelease = 0    #No releases for hydropower
    if availableWater > reservoirCapacity:
        #Release water to prevent overtopping the dam
        additionalRelease = availableWater - reservoirCapacity
    else:
        #Don't release any water
        additionalRelease = 0

    #END EDITABLE SECTION - DO NOT EDIT ANYTHING BELOW THIS LINE!
    #-----------------------------------------------------------------------------------------------
    return [hydropowerRelease, additionalRelease]
```

**Figure 2. Example release rule to not generate hydropower and keep the reservoir full.**

## A Few More Notes about the Model

The model file pineview.py calls your release rule function 684 times -- each and every month between January, 1950 and December, 2006! The model starts by assuming the reservoir is full. Then in each month, the model notes the reservoir inflow, evaporation, and storage volume remaining from the prior month. The model calculates the available water, passes this value to your release rule function, runs your function, and checks that your prescribed releases are valid (e.g., it makes sure that your releases are positive values, that you don't release more water than is available, or overtop the dam). If your releases are valid, the model uses your releases to calculate the remaining water stored in the reservoir and associated benefits and damages from your releases. The model then proceeds to the next step. At the end of the 56-year simulation period, the model outputs results by month to the csv file output.csv.

## Other Example Release Rules

A) Keep the reservoir empty

```python
    #-----------------------------------------------------------------------
    #BEGIN EDITABLE SECTION

    #A second simple rule. Release all the available water
    hydropowerRelease = 0 #No hydropower release
    additionalRelease = availableWater #Release all the avaialable water

    #END EDITABLE SECTION - DO NOT EDIT ANYTHING BELOW THIS LINE!
    #-----------------------------------------------------------------------
```

B) Generate as much hydropower as possible. Otherwise, only release water to prevent overtopping the dam.

```
#-----------------------------------------------------------------------------------------------
#BEGIN EDITABLE SECTION

#A third simple rule. Generate as much hydropower as possible.
#Otherwise, only release water to prevent overtopping the dam
if availableWater > turbineCapacity:
    hydropowerRelease = turbineCapacity #Release at turbine capacity
else:
    hydropowerRelease = availableWater #Release all the available water through the turbine

if availableWater-hydropowerRelease > reservoirCapacity:
    #Release aditional water to prevent overtopping the dam
    additionalRelease = availableWater - hydropowerRelease - reservoirCapacity
else:
    #Don't release any additional water
    additionalRelease = 0

#END EDITABLE SECTION - DO NOT EDIT ANYTHING BELOW THIS LINE!
#-----------------------------------------------------------------------------------------------
```

C) When possible, release water to meet the summer irrigation demand, increase releases to prevent overtopping the dam, and do not generate hydropower.

```
#-----------------------------------------------------------------------------------------------
#BEGIN EDITABLE SECTION

#A fourth rule. As much as possibly, only release water to meet the summer full irrigation demand
#Increase releaes to prevent overtopping the dam. Do not generate hydropower

if (month >= 4) and (month <=9):
    #During the irirgation season
    if availableWater >= irrigationCapacity:
        #Release the full monthly irrigation demand
        additionalRelease = irrigationCapacity
    else:
        #Release all the available water to meet as much of the demand as possible
        additionalRelease = availableWater
else:
    #No release during winter months
    additionalRelease = 0

if availableWater - additionalRelease > reservoirCapacity:
    #Increase the additional release to prevent overtopping the dam
    additionalRelease = availableWater - reservoirCapacity

hydropowerRelease = 0 #No hydropower release

#END EDITABLE SECTION - DO NOT EDIT ANYTHING BELOW THIS LINE!
#-----------------------------------------------------------------------------------------------
```

## Additional Resources

- Wikipedia page for Pineview Reservoir
- Documentation and download for Python – http://www.python.org