# Coagulant Adsorption Final Project


## Ken Rivero-Rivera & Catherine Johnson


# Introduction

Water has a lot of important features. Because of these unique features, water is vital to the survival of life on Earth, it is found all over the world, and it is a universal solvent. It's ability to be such a powerful solvent has both helped and hurt. It can carry essential substances to cells around the body of living creatures. But it can also be harmful, as the things it carries can be toxic pollutants, especially in an increasingly polluted world. Thus, there needs to be a way to remove pollutants from water - and one of those ways is through adsorption.

There are different substances that can be used for adsorption - one used in a previous experiment was activated carbon. The team chose to analyze coagulant as an adsorbent. Coagulant is used in the water treatment process to treat turbidity and remove color. Coagulant forms large clumps of solid particles in water called flocs, and these larger particles can be settled out more easily through a filter. Coagulant is also relatively inexpensive, as not much is required for treatment.

# Objectives

The purpose of this experiment was to evaluate the performance of coagulant as an adsorbent. The coagulant was tested at different concentrations and was assessed for two different substances - red dye #40 and red dye #3.

# Procedures

A schematic of the experiment can be seen in the figure below.

1. A ProCoDA calibration curve for the photometer was created for both red dye #40 and red dye #3 using 0, 10, 20, 39, 40, and 50 mg/L standards of each dye.
2. Stock solution (100 mg/L) was made of red dye #40 and red dye #3.
3. The stock solution actual concentration was determined using the calibrated photometer. Red dye #40 was 100mg/L, but red dye #3 was 70mg/L for reasons discussed later.
4. Sand was poured into column to ensure the correct volume of sand was obtained.

5. After a few failed attempts at different flow rates, 1.25 mL/s was determined to work for both red dye #40 and red dye #3 enough to yield useful results and so that sand was not dislodged.
6. Both dyes were run through a sand-only column to obtain a baseline for the experiment (0 coagulant).
7. The sand, coagulant and 100mL of water was added to a beaker for mixing.
8. While mixing, pH was monitored.
9. A basic solution of 0.5M NaOH was added until a pH of at least 6 was reached (Source: Water Treatment Coag). This was a necessary step because coagulant is largely ineffective in acidic conditions, and the aluminum in coagulant makes the mixture acidic. This is also when flocs generally started to form.
10. A small volume of the liquid in the mixture and then the sand portion of the mixture was added to the column. This "wet method" was employed so that there would not be any air pockets in the column.
11. The initial amount of coagulant used was 445 µL, which is equivalent to 22 moles of Aluminum per meter cubed. This concentration was suggested as ideal in “Enhanced Particle Capture through Aluminum Hydroxide Addition to Pores in Sand Media.”
12. The amount of coagulant was varied afterward, as can be seen in Table 1. Both dyes were used for all volumes except the 1500 microliters of coagulant. This is because red dye #3 was removed far more effectively by sand and coagulant than red dye #40 was. Where the red dye #40 experiments took minutes or less, the red dye #3 experiments took a few hours. Using 1500 µL of coagulant with red dye #3 would have taken too much time, and the stock tank for the solution would have run out, thus reducing quality assurance in the experiment.
13. For red dye #3 only, the experimental apparatus was disassembled and cleaned (some parts replaced) between runs because this dye stained the tubing and other equipment.

# Results and Discussion

## Red Dye #3

| Volume of 70.28 g/L PAC as Al | Time to 50% Influent | pH of Sand w/ PAC |
|---|---|---|
| 0 µL | 1.67 min | 0 |
| 445 µL | 7.25 min | 6.9 |
| 745 µL | 102.92 min | 6.7 |
| 945 µL | 127.84 min | 6.6 |

From this data, it is clear that increasing the amount of coagulant increases the effectiveness of it as an adsorbent. However, there is a big jump in the time it takes the dye to return to 50% of the stock solution from 445 µL and 745 µL, whereas the difference between 745 µL and 945 µL is not very large. This would indicate that there is some level where the performance of coagulant begins to plateau, meaning that adding more does not effect how much material is removed from solution.

## Red Dye #40

| Volume of 70.28 g/L PAC as Al | Time to 50% Influent | pH of Sand w/ PAC |
|---|---|---|
| 0 µL | 1.67 min | 0 |
| 445 µL | 2.83 min | 8.0 |
| 745 µL | 5.17 min | 6.0 |
| 945 µL | 5.58 min | 7.5 |
| 1500 µL | 3.17 min | 6.1 |

For red dye #40, the results are more uniform than those of red dye #3. This can be attributed to red dye #40's ability to fully dissolve in water, so the stock solution was uniform. results for this dye also show that increasing the amount of coagulant increases adsorption. Like the results for red dye #3, these experiments show that the performance between 745 µL and 945 µL of coagulant is minimally different. The strangest result was the run with 1500 µL of coagulant, which performed worse than 745 µL and 945 µL. This was surprising, but was probably due to the initial sand-coagulant mixture being too acidic, as the most NaOH was need to counteract the acidity of the largest amount of coagulant, and it may not have been enough.

# Conclusions and Suggestions

Coagulant works very well for materials that precipitate, like red dye #3. Red dye #40 was able to reach 50% of the stock concentration within minutes, whereas red dye #3 took much longer - some runs took hours. Because red dye #3 precipitates easily, it is adsorbed very well.

Additionally, for both dye types there was little difference between the 745 µL and 945 µL runs, indicating that there is a plateau in the performance of coagulant as more is added.

This lab yielded some interesting results, including the adsorption of red dye #40 with 1500 µL, which performed worse than the 745 and 945 µL runs. This could be attributed to a fluke, or a result of human error in measurement or calibrating the instruments. However, the team did note all of the pHs of sand

and coagulant mixture before addition to the column, and the 1500 µL run had the lowest pH at 6.1. The team did not add any more NaOH after this point because nearly 4.5 mL had already been added, the most of any other run, and flocs had begun to form in the mixture. It would be an interesting experiment in the future to see if the performance of coagulant as an adsorbent has correlation with the pH it starts out at.

# Bibliography


- Gebbie, Peter. (July 2006). An Operator's Guide to Water Treatment Coagulants, *Earth Tech Engineering*.
- Lin et. al. (January 2012). Enhanced Particle Capture through Aluminum Hydroxide Addition to Pores in Sand Media, *Journal of Environmental Engineering*, 138, 1.
- Weber-Shirk, Monroe. (December 2018). Adsorption Lab, *CEE4530 GitHub*.


# Appendix

```
from aguaclara.core.units import unit_registry as u
import aguaclara.research.environmental_processes_analysis as epa
import aguaclara.core.physchem as pc
import aguaclara.core.utility as ut
import numpy as np
import matplotlib.pyplot as plt
import collections
import os
from pathlib import Path
import pandas as pd



def adsorption_data(C_column, dirpath):
    """This function extracts the data from folder containing tab delimited
    files of adsorption data. The file must be the original tab delimited file.

    Parameters
    ----------
    C_column : int
        index of the column that contains the dissolved oxygen concentration
        data.
    dirpath : string
        path to the directory containing aeration data you want to analyze
    Returns
    -------
    filepaths : string list
        all file paths in the directory sorted by flow rate
    time_data : numpy array list
        sorted list of numpy arrays containing the times with units of seconds
    Examples
    --------
    """
    #return the list of files in the directory
    metadata = pd.read_csv(dirpath + '/metadata.csv', delimiter=',')
    filenames = metadata['file name']
    #extract the flowrates from the filenames and apply units
    #sort airflows and filenames so that they are in ascending order of flow rates


    filepaths = [dirpath + '/' + i for i in filenames]
    #C_data is a list of numpy arrays. Thus each of the numpy data arrays can have different lengt
    # cycle through all of the files and extract the column of data with oxygen concentrations and
    C_data=[epa.column_of_data(i,epa.notes(i).last_valid_index() + 1,C_column,-1,'mg/L') for i in
    time_data=[(epa.column_of_time(i,epa.notes(i).last_valid_index() + 1,-1)).to(u.s) for i in fil

    adsorption_collection = collections.namedtuple('adsorption_results','metadata filenames C_data
    adsorption_results = adsorption_collection(metadata, filenames, C_data, time_data)
    return adsorption_results
```

```
C_column = 1

#Url to Github folder containing red dye #3 ProCoDA data
dirpath1 = "https://raw.githubusercontent.com/klr227/EnvELab/master/Final_Project/Final_3_Data"

#Measured flow rate
flow_rate = 100*u.mL/(80*u.s)

metadata1, filenames1, C_data1, time_data1 = adsorption_data(C_column,dirpath1)

#Initial concentration of red dye #3 stock
C_0_R3 = 70 * u.mg/u.L

#Zero the concentration data by subtracting the value of the first data point from all data points
#Convert time data to minutes
for i in range(np.size(filenames1)):
  C_data1[i] = C_data1[i] - C_data1[i][0]
  time_data1[i] = time_data1[i].to(u.min)

#Empty array for legend array needed for graphs. Plot red dye #3 concentrations(divide by initial
mylegend1 = []
for i in range(np.size(filenames1)):
  plt.plot(time_data1[i], C_data1[i]/C_0_R3,'-');
  mylegend1.append(str(metadata1['Coagulant added (uL)'][i]) + ' µL')

plt.xlabel(r'Time (minutes)');
plt.ylabel(r'Red dye #3 $\left ( \frac{C}{C0} \right )$');
plt.legend(mylegend1);
plt.savefig('Final_Project/Pictures/Red_Dye_3_graph.png');
plt.show()
#Url to Github folder containing red dye #40 ProCoDA data

dirpath2 = "https://raw.githubusercontent.com/klr227/EnvELab/master/Final_Project/Final_40_Data"
metadata2, filenames2, C_data2, time_data2 = adsorption_data(C_column,dirpath2)

#Initial concentration of red dye #40 stock
C_0_R40 = 100 * u.mg/u.L

#Zero the concentration data by subtracting the value of the first data point from all data points
#Convert time data to minutes
for i in range(np.size(filenames2)):
  C_data2[i] = C_data2[i] - C_data2[i][0]
  time_data2[i] = time_data2[i].to(u.min)

#Empty array for legend array needed for graphs. Plot red dye #3 concentrations(divide by initial
mylegend2 = []
for i in range(np.size(filenames2)):
  plt.plot(time_data2[i], C_data2[i]/C_0_R40,'-');
  mylegend2.append(str(metadata2['Coagulant added (uL)'][i]) + ' µL')

plt.xlabel(r'Time (minutes)');
```

```
plt.ylabel(r'Red dye #40$\left ( \frac{C}{C0} \right )$');
plt.legend(mylegend2);
plt.savefig('Final_Project/Pictures/Red_Dye_40_graph.png');
plt.show()

#Value at which effluent equals half of the initial red dye #3 stock concentration
half_R3 = C_0_R3*0.5
#Empty array for times at which effluent equals 50% influent
half_time1 = np.zeros(np.size(filenames1))*u.s
#For loop to find time at which effluent equals 50% influent at which point it breaks the loop and
for i in range(np.size(filenames1)):
  C = C_data1[i]
  for k in range(np.size(C)):
    if (C[k] >= half_R3):
      break
  half_time1[i] = time_data1[i][k]
#Convert from seconds to minutes
half_time1 = half_time1.to(u.min)

#Value at which effluent equals half of the initial red dye #40 stock concentration
half_R40 = C_0_R40*0.5
#Empty array for times at which effluent equals 50% influent
half_time2 = np.zeros(np.size(filenames2))*u.s
#For loop to find time at which effluent equals 50% influent at which point it breaks the loop and

for i in range(np.size(filenames2)):
  C = C_data2[i]
  for k in range(np.size(C)):
    if (C[k] >= half_R40):
      break
  half_time2[i] = time_data2[i][k]
#Convert from seconds to minutes
half_time2 = half_time2.to(u.min)
```