# CEE 4530 Final Project: Impact of Salinity and Temperature on Oxygen Transfer into Water

**Team 1: Megan McEnaney, Clayton Sasaki, Kristen Ajmo**

**Due: May 14th, 2018**

## Introduction and Objectives

There are many characteristics that define ocean currents and aquatic life, but the two main factors are temperature and salinity. Both of these may affect dissolved oxygen levels, and in turn, aquatic life. At higher temperatures air can hold more water vapor, so the resulting evaporation could lead to higher salinity in the oceans. However, this relationship between ocean temperature and salinity is complicated by other factors such as sea spray, sedimentation, and the melting of ice sheets. The ocean water density and the overall ocean conveyor belt is a function of salinity and temperature gradients. This system is threatened in the face of climate change in which salinity levels are being affected by melting ice and increased UV ray penetration and heat trapping are affecting temperature. The format of this experiment will help us to better understand how changes of temperature and salinity will affect aeration rates, determine which is more impactful, and see how they perform in union.

On a smaller scale, as cities salt roads for winter, this salt enters the environment. During strong storms the salt leaches into waterways and eventually finds its way into larger bodies of water. We want to look at the impact this salt might have on the dissolved oxygen in these bodies of water through possible changes in reaeration rates which can subsequently impact aquatic biomes. Temperature effects on dissolved oxygen will yield meaningful results in terms of how climate change and increasing the global temperature will affect aquatic life and the functioning of aquatic ecosystems. It may provide insight into the level at which temperature increases will severely impact these areas and threaten mortality rates.

In order to perform this experiment, we will be modeling gas transfer with a constant flow rate (250 micromoles/s). We then varied temperature while keeping salinity constant, and vice versa, to see the effects on the gas transfer coefficient, $k_{v,l}$ (found using the provided EPA code and the oxygen saturation concentration). The saturation concentration was found using the equation $C^* = P_{O_2}\, e^{\left(\frac{1727}{T} - 2.105\right)}$. We also linearized the equation $ln\frac{C^*-C}{C^*-C_0} = -k_{v,l}(t - t_0)$ to find $k_{v,l}$, which is the resulting slope. Another variable that allows us to look at the effects of temperature and salinity on dissolved oxygen is the oxygen transfer efficiency, or OTE. This is given by the equation $OTE = \frac{k_{v,l}(C^*-C)VRT}{MW_{O2}Q_{air}P_{air}f_{O2}})^*$ and it is the ratio of the molar rate of dissolution into the aqueous phase to the molar transfer rate of oxygen through the diffuse.

By completing this experiment, we hope to observe the trend between temperature and salinity and oxygen transfer. The results of this could help us make predictions as to the future of our oceans, in addition to smaller effects such as runoff from road salt.

## Procedure

We started our experiment by setting it up exactly like the schematic shown in the Gas Transfer Lab Manual with the addition of a few elements. The set up can be seen in Figure 1 below.

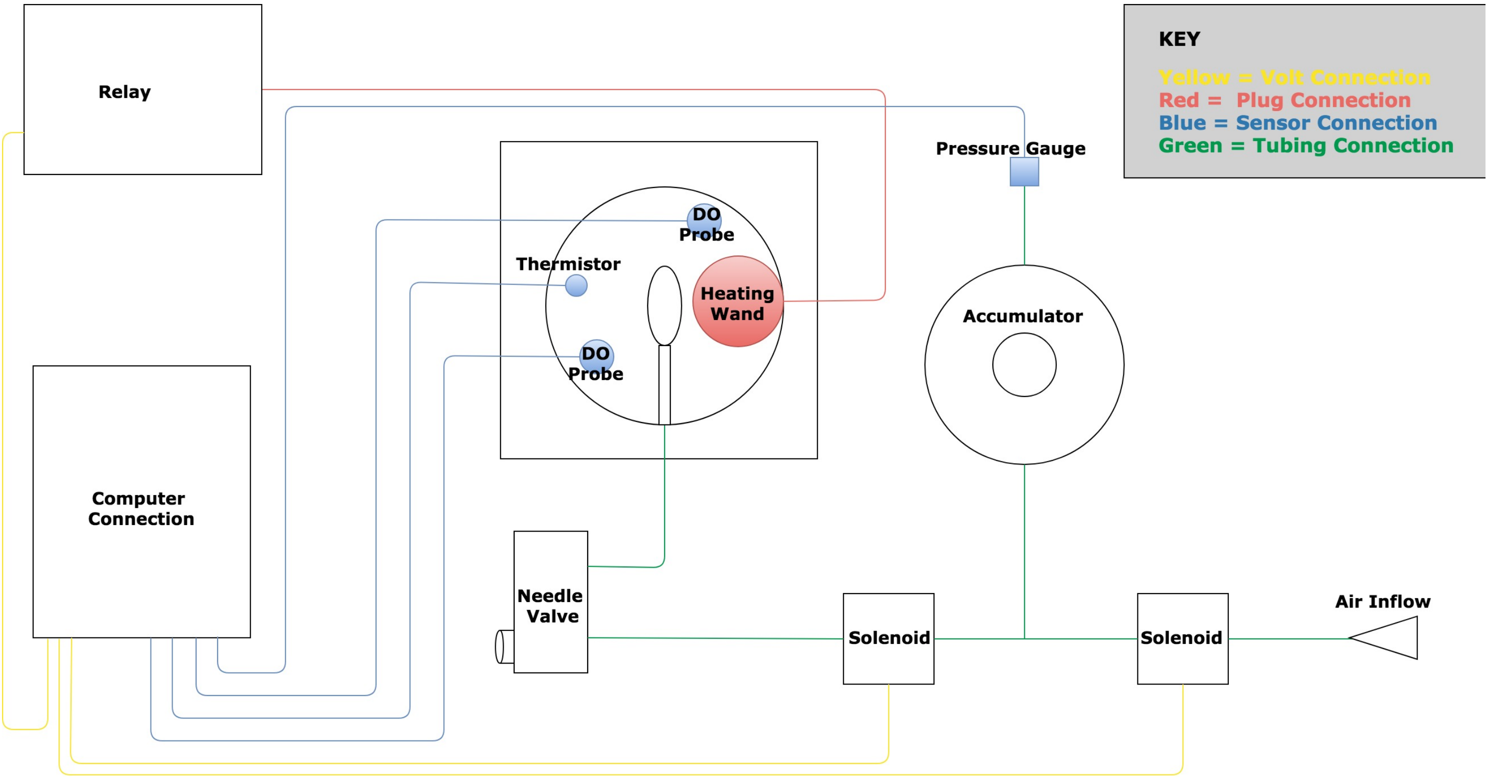

Figure 1: Schematic of the apparatus used to measure Dissolved Oxygen with all connections color coded

For the first experiment, we varied salinity: 0, 10, 20, 30, and 35 mg/L. These values were used as we wanted to look at the differences between freshwater and saltwater. The purpose was to reach salinities close to those found in the ocean (~33-36 mg/L). To achieve each salinity, the amount of salt added to the 0.5 L of water was calculated. Once the salinity experiments were completed the same was done at various temperatures: 24, 28, 32, 36, and 40 degrees Celsius. These temperatures were again chosen to be comparable to current and future ocean temperatures under climate change.

The general procedure is described below: Before running the experiments, the system was set up and calibrated into order to make sure the system outputted the correct air flow rate. 0.5 L of water was added to the container along with 10 microliters of 100g/L cobalt solution (the cobalt acted a catalyst for the deoxygenation). Beginning the main portion of the procedure. the needle valve was closed (because of a leaky solenoid) and stirring was begun. Depending on which experiment was run the next step varied, but either salt was added to the oxygen saturated solution (making sure that all the salt was completely dissolved) or the solution was heated to a specified temperature. The $C_{sat}$ was calibrated during this time based on temperature and salinity (explained more below). 500 microliters of 100 mg/mL NaSO3 was injected to deoxygenate the solution and get the second calibration point. At this point the stirrer was turned off (reasoning elaborated in the hurdles and challenges subsection). Finally, the needle valve was opened, re-aeration was commenced and continued until a linear region was established.

Before each experiment it was necessary to calibrate the DO Probe in ProCoDa. For the salinity experiments, this was carried out using the calculated values for $C_{sat}$ along with varying the pressure inputted into ProCoDa. Because the $C_{sat}$ cannot be directed edited, the pressure was varied until ProCoDa showed the correct $C_{sat}$ value. It was easier to adjust the $C_{sat}$ when only temperature was being varied. This was simply done by inputting the correct temperature of the solution.

Heating of the solution was carried out using a MARINA mini 25W Submersible Aquarium heater that was connected to a relay, and then to ProCoDa to allow a constant temperature to be achieved. This was done by using ProCoDa to control the state of the heater (on/off). Code was imported to ProCoDa that turns the heater on or off based on a minimum and maximum temperature in order to keep it as steady as possible. For each level, we set the maximum temperature at the desired temperature and the minimum temperature half a degree below. This range was shifted down and not centered around the desired temperature as residual heating meant that the temperature had a tendency to overshoot and took a long time to cool down. We waited until the water hit the desired temperature at each level, and then followed the procedure outlined above.

## Results and Discussion

Performing this experiment to determine the effects of temperature and salinity variations on the gas transfer coefficient of oxygen into water yielded interesting results that both supported and refuted our expectations.

While our experiment is focused on evaluating the affect varying properties of the water had on gas transfer coefficient, or reaeration rate, it is important to consider the saturation concentration in our analysis. The saturation concentration of oxygen, $C^*$ , represents the maximum oxygen level that water can maintain at a particular temperature, pressure, and salinity. Table 1 below details this parameter, labeled O2 sat.

Table 1: Oxygen saturation for varying temperatures and salinities

| Temp (C) | Salinity (g/L) | O2 sat (mg/L) |
|---|---|---|
| 22 | 0 | 8.74 |
| 22 | 10 | 8.25 |
| 22 | 20 | 7.78 |
| 22 | 30 | 7.34 |
| 22 | 35 | 7.13 |
| 24 | 0 | 8.35 |
| 28 | 0 | 7.73 |
| 32 | 0 | 7.17 |
| 36 | 0 | 6.66 |
| 40 | 0 | 6.20 |

## Salinity

During our first round of experimentation, we varied the salt concentration of the water, in the exact manner detailed in the procedure. As salinity increases, the saturation concentration of oxygen decreases because water is a polar molecule and as salt molecules dissolve, they are dissociated by the water molecules into sodium and chlorine. The attraction of water molecules by the salt molecules results in fewer hydrogen and oxygen ions available to disassociate gas molecules, thus producing this relationship of lower saturation concentrations at higher salinities. As a result, we would expect reaeration rates to decrease as salinity decreases. Figure 2 below shows the relationship between dissolved oxygen concentrations over time for various salinity levels, depicting the relationship that dissolved oxygen levels are inversely related to salinity levels.

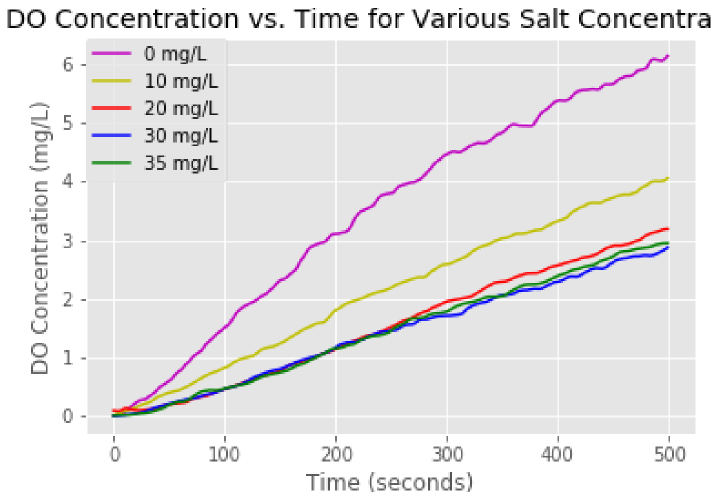

Figure 2: Dissolved oxygen concentrations as a function of time along various salinity levels

While the above figure does not show dissolved oxygen at saturation, it should be noted that this figure expresses the levels over a long time period, approximately eight minutes, thus it is reasonable to assert that overall trend in dissolved oxygen levels follows the maximum saturation trend. The impact of salinity on can be further evaluated by looking at the reaeration rate, or gas transfer coefficient, representative of the slope of the line. This can be computed from the equation $k_{v,l} = -\frac{ln\frac{C^*-C}{C^*-C_0}}{(t-t_0)}$.

The exact values of the slope for all of the different salinity levels (the curves depicted in Figure 3) are detailed in Table 2, below.

Table 2: k Values for Salinity Varied Curves

| Salinity (g/L) | k (1/s) |
|---|---|
| 0 | 0.00245 |
| 10 | 0.00137 |
| 20 | 0.00110 |
| 30 | 0.00100 |

| Salinity (g/L) | k (1/s) |
|---|---|
| 35 | 0.00101 |

The gas transfer coefficients tend to decrease with salinity, supporting our initial expectations that salinity and gas transfer would be inversely proportional. These results make sense as, like previously stated, when salinity increases, the hydrogen and oxygen ion availability decreases, resulting in decreasing rates of reaeration. This could be attributed to the increase in viscosity and decrease in diffusion that accompanies adding salt.

An interesting characteristic to highlight in these results is the effect of salinity on dissolved oxygen and reaeration rates decreasing as salinity decreases. Initially, the slopes drop significantly between 0 g/L to 10 g/L, and then from 10 g/L to 20 g/L. However, from 20 g/L to 30 g/L and then 35 g/L the change in reaeration rate is increasingly less dramatic. It is likely that the magnitude of impact the salinity drops because it is approaching salinity saturation level.

## Temperature

After completing our salinity experiments, we went on to manipulate the water temperature in accordance with the levels outlined in the procedure. As temperature increases, the saturation concentration of oxygen decreases because as temperature rises, the density decreases and the water is not able to hold as much oxygen. Additionally, higher temperatures lead to more kinematic energy in the molecules, and the resulting higher collision rates decrease retention of oxygen. The relationship between dissolved oxygen over time at various temperatures during the initial reaeration period is displayed below in Figure 3.

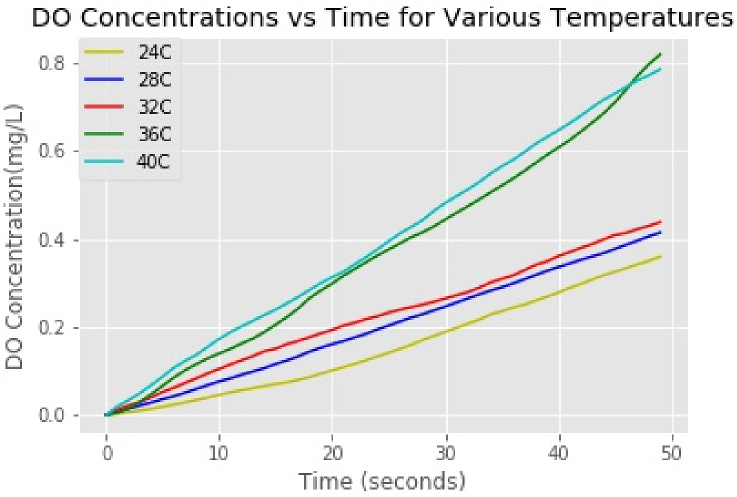

Figure 3: Dissolved oxygen concentrations as a function of time along various temperature levels

While the saturation concentration decreases with temperature, as explained above, the reaeration rate actually increases. Initially, this was not what we expected and found these results to be surprising. However, after considering the trend, we realized that these results make sense because as the temperature increases, diffusion also increases from the elevated kinetic energy in the molecules. Figure 3 (above) portrays the very small initial time step of approximately one minute, allowing this direct relationship between temperature and diffusion to be observed. A fascinating result of the temperature portion of this experiment was that although saturation concentration is inversely related to temperature, reaeration rate is actually directly related to temperature. The associated reaeration rates can be clearly observed from the gas transfer coefficient values in Table 3 below. These values were found in the same manner described in the previous Salinity subsection.

Table 3: k Values for Temperature Varied Curves

| Temperature (C) | k (1/s) |
|---|---|
| 24 | 0.00093 |
| 28 | 0.00116 |
| 32 | 0.00123 |
| 36 | 0.00257 |
| 40 | 0.00277 |

Observing the trend of the reaeration rates in Table 3, we can see that rates increase with temperatures as expected, but in varying magnitudes. Rates generally increase by small amounts, on the order of 0.0002 /s, with the exception the change between 32 C and 36 C, which increases by a much larger magnitude of about 0.0012/s. This discontinuity in trend is likely due to human or equipment error.

The linear trend that should exist between temperature and the gas transfer coefficient is supported by Figure 4 below that comes from a research paper that studied this same relationship: Reaeration Model for a Still-Water Body by Manh Tuan Nguyen Danielle S. Tan and Soon Keat Tan [1].

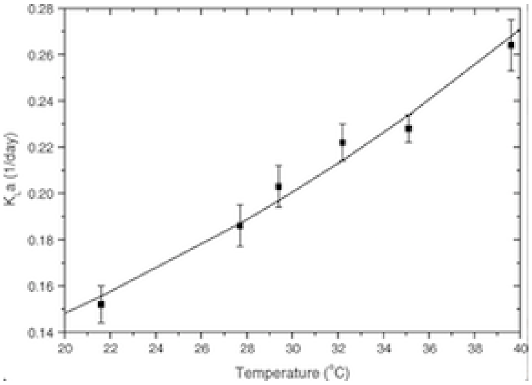

Figure 4: Gas Transfer Coefficient (kLa) plotted against their associated temperatures

It shows the linear trend in between the gas transfer coefficient and temperature for a water configuration that embodies a small lake. In order to directly compare the trend of our results with the results of this similar experiment, we plotted the our data against the same variables in Figure 5 below.

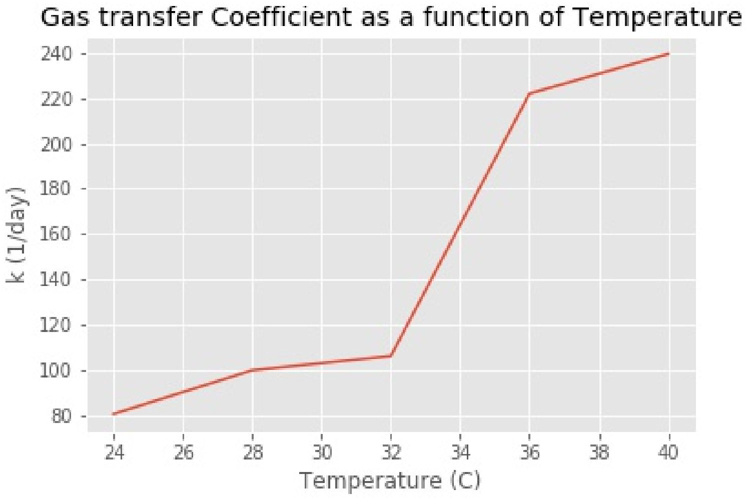

Figure 5: Gas Transfer Coefficient (k) values plotted against their associated temperatures to determine trend and compare to paper results

It is clear to see that our results do not exactly mirror the results of comparative experiment. This supports our previous assertion that the 32 C to 36 C data is slightly skewed due to some form of human or equipment error, resulting in the failure to mirror the trend. The large difference in actual k values is a result of very different experimental conditions. Our values are much higher because we had a much smaller volume of water with a much shallower depth. Additionally, we had an aeration stone in our apparatus to facilitate aeration, rather than just leaving the water open to the air. These factors were the major aspects of the experiment that caused significantly higher k values.

## Salinity vs. Temperature Comparison

The major goals of this experiment was to establish the relationship between gas transfer coefficient with varying salinity and temperature and determine which factor had a higher impact of this value. Figure 6 plots the gas transfer coefficient on a dual axis graph to best display the salinity and temperature trends against one another.

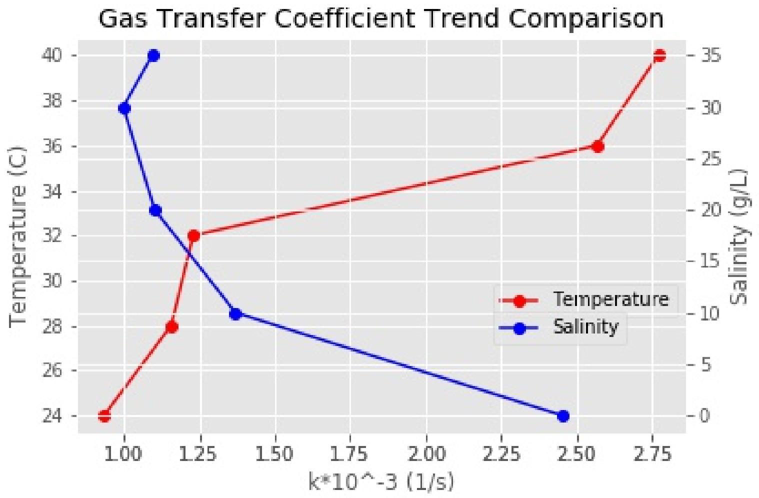

Figure 6: Plot comparing gas transfer trends of salinity versus temperature to highlight the inverse relationship

It is evident from this figure that the reaeration trends are opposite in nature. While it was discussed earlier that salinity is inversely and temperature is directly related to reaeration, this figure shows this in the clearest and most concise manner. It is was an interesting result of our experiment and an independent figure to highlight this outcome was warranted.

The Oxygen Transfer Efficiency (OTE) was calculated using the equation described in the introduction. It is the ratio of the molar rate of dissolution into the aqueous phase to the molar transfer rate of oxygen through the diffuse. The gas transfer coefficient versus the OTE for both experiments are expressed in Figure 7.

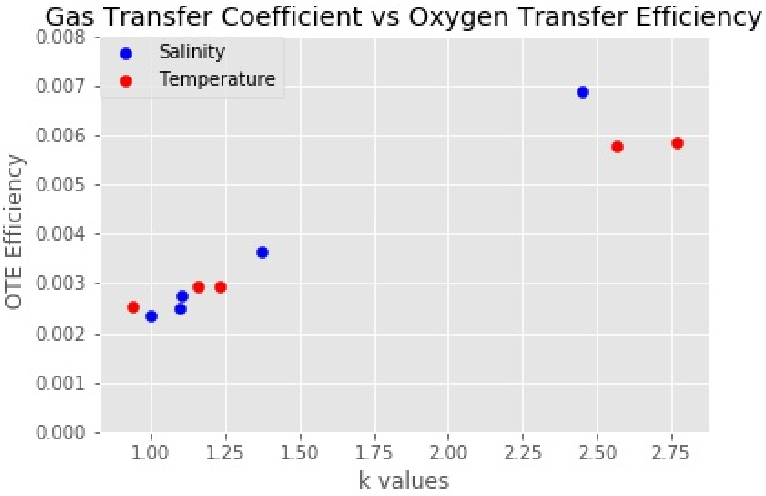

Figure 7: Oxygen Transfer Efficiency (OTE) for temperature and salinity gas plotted against gas transfer coefficient (k) to show clustering of values

Unsurprisingly, OTE trends mirrored the gas transfer coefficient trends. This is expected because the gas transfer coefficient is the most impactful parameter in the equation because most other parameters only varied slight or not at all (constant) for all rounds of the experiment. The associated code for these computations can be found in the Appendix. An interesting observation from this figure is the clustering of the values. It was slightly evident in the previous Figure 6, but is much more obvious in Figure 7. This clustering is likely due to the reaeration trends displayed in Tables 2 and 3. For the temperature, we saw that the 36 C and 40 C has significantly higher gas transfer coefficients, and this extreme difference is shown in Figure 8 because these two points are located in the upper right hand quadrant of the graph, coinciding which higher k and OTE values. The 0 g/L is clustered in the same area because it too was an outlier compared to the rest of the salinity data, having a significantly higher reaeration rate. The rest of the OTE values are clustered in the lower left hand quadrant of the figure, coinciding with all of the lower k values for both temperature and salinity. In this instance, we can observe the correlation between k and OTE values. The apparent clustering was an interesting outcome that was not expected, but looking at the k values, makes logical sense. If anything, it highlights the extreme nature of our data, in that we experience outliers in both experimental manipulations.

The associated code for all of the figures and values can be found in Appendix A.

Overall, the impact of manipulating temperature and salinity on dissolved oxygen and reaeration rates were clearly observable in our experimental results.

## Hurdles and Challenges

We experienced our fair share of problems throughout our experiment. Originally, we intended to complete the entirety of the salinity experiemnt during the first lab period and the temperature done in the second lab period, however this did not transpire until the following lab periods due to many issues related to the DO Probe and other equipment related issues. The first DO probe issue we ran into was figuring out how to calibrate it at each salinity level. However, this ended up being much easier than we expected, as we only had to completely mix the salt into the saturated water set the $C^*$ value in ProCoDa to the calculated value based on the salinity, temperature and pressure. Once this hurdle was overcome, we began testing! Unfortunately, the initial outcomes were not what we were expecting. The shape of the curve was incorrect; it dropped too rapidly, and then increased too rapidly, to a magnitude higher then what should be observed. After consulting with Monroe, it was brought to our attention that the main issue was that we did not zero the software. The next time we tried to run this experiment (still at 0 mg/L of salt), issues persisted. It was then suggested by our brilliant TA to replace the DO probe membrane and add another probe to ensure the entire probe was not broken. The fourth time we tried to run the experiment, the graph had a better shape but was not a smooth curve like we observed in our data for the Gas Transfer Lab. At this point, we were stumped and once again turned to the ever wise Monroe for help. It turned out the only issue was that the water was being over-aerated and the excessive water bubbles were interacting directly with the DO readings. The results of this run can be seen in figure 8 below. Since air bubbles were directly hitting the DO probe sensor, resulting in this spiking of data. After decreasing the air flow, turning of the stirrer, and being sure to place the DO probes out of the direct path of the bubbles, we were finally able to obtain usable data.

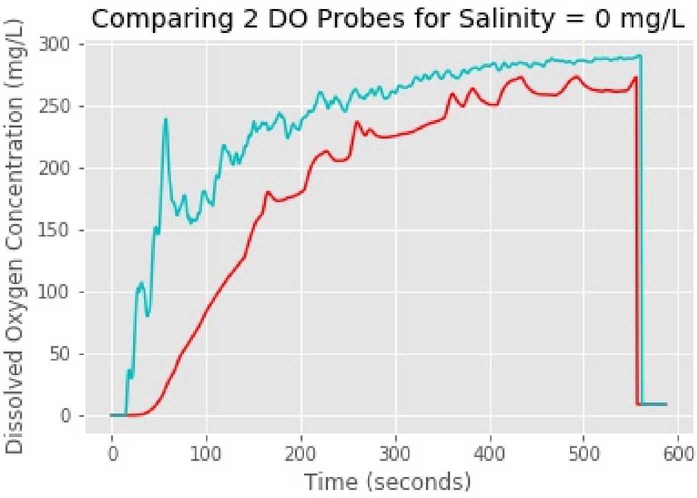

Figure 8: Graph of over aeration leading to bubbles on the DO probes, resulting in inconsistent and non-uniform data readings

Moving on to the temperature experiments, our issues were not nearly as extensive. The main problem we ran into right away was the heating plate not being able to hold a steady temperature. Monroe assisted us with by acquiring a heating wand. With his help, we were able to set up ProCoDa such that the heat turns off when the maximum temperature is hit, and turns on until the minimum temperature is hit. Our first issue with this code was that the heater was overshooting the maximum temperature when it initially heated the water at each level. The water had a tendency to heat for an approximate 0.4 degrees Celsius after before dropping back down to the lower portion of the range. To minimize this effect, we set the maximum temperature to 0.5 degrees Celsius lower than the desired temperature at the start of each heating round, then changed this once the water hit the maximum temperature and the heater turned off. However, there was still inaccuracies in this part of the experiment due to the temperature not being perfectly constant, since it varied with in a range that we set.

## Conclusion

Our results provide insight into the future impact the climate could have on delicate water properties.

Initially salinity has a larger effect on aeration rates than changes in temperature, but once a certain salinity is reached adding more salt has almost no impact. This shows that contaminating a freshwater body with a salt source results in a larger impact on aeration rates than small salinity changes in an already salty environment. This result is more relevant to the salting of roads in the winter as most of the interior streams and lake are freshwater. On the other hand, the almost constant decrease of aeration rates with temperature increases implies that changes in temperature will probably play a larger role in ocean biomes. However, there exists a complex relationship between water and air dissolved oxygen levels because while temperatures rise and salinity decreases, the air is able to hold more water vapor, possibly offsetting some of the climatic impact.

Although the objective of this experiment was to establish the relationship between salinity and temperature variations on reaeration rates, in order to determine which factor had the greater impact, the results are not black and white. As discussed in detail in the previous section, temperature and salinity had varying impact as values increased. Temperature has greater impacts at higher levels and salinity has greater impacts at lower levels. Consequently, neither salinity, nor temperature has the absolute advantage in controlling reaeration rates; it is completely dependent on the exact conditions being evaluated.

## Suggestions and comments

Upon reflection, there are some improvements that could be made to enhance the overall outcomes of our experiment. Properly linking the thermistor to the DO saturation calibration may have allowed for improved trends through improved data acquisition. This did not work when we attempted to link these parameters in ProCoDa.

Alternatively, we could modify the apparatus. As mentioned in the analysis above, using an aeration stone may not have been the most accurate way to test aeration. As in real bodies of water, it is generally done by waves or movement. Additionally, it might be useful to try using a wider container for the aeration. While we moved our DO probes out of the direct path of the Oxygen bubbles, there may have still been some interference due to the narrow width of the container.

Looking forward, we could apply a dual approach, varying both temperature and salinity parameters simultaneously. It would also be useful to investigate a specific application and for that application determine an empirical equation by choosing a specific value of one of variables (holding it constant) and varying the other. This would help to clearly define a tolerable range for those conditions.

An alternative way to analyze the data could have been looking more into the trends in changing magnitudes of impact on the reaeration rate. If an exact trend could be determined, such as the change in k value per degree or g/L of salinity, we could potentially establish an empirical formula to clearly illustrate these trends.

# Appendices

## Appendix A: Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
import aide_design
import aide_design.pipedatabase as pipe
from aide_design.units import unit_registry as u
from aide_design import physchem as pc
import aide_design.expert_inputs as exp
import aide_design.materials_database as mat
import aide_design.utility as ut
import aide_design.k_value_of_reductions_utility as k
import aide_design.pipeline_utility as pipeline
import warnings
from scipy import stats
##IMPORTANT TO IMPORT SCIPY FOR LIN REG

from aide_design.play import *
import Environmental_Processes_Analysis as EPA
import importlib
importlib.reload(EPA)

# Temperature portion
n_rows = 50
data_file_path= 'heat_24.txt'
firstrow = 1050
time_data_24 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_24 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,4,'mg/L')
concentration_data_24 = concentration_data_24 -concentration_data_24 [0]

data_file_path= 'heat_28.txt'
firstrow = 790
time_data_28 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_28 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,4,'mg/L')
concentration_data_28 = concentration_data_28 - concentration_data_28[0]

data_file_path= 'heat_32.txt'
firstrow = 288
time_data_32 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_32 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,4,'mg/L')
concentration_data_32 = concentration_data_32 - concentration_data_32[0]

data_file_path= 'heat_36.txt'
firstrow = 487
time_data_36 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_36 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,4,'mg/L')
concentration_data_36 = concentration_data_36-concentration_data_36[0]
concentration_data_36[0]

data_file_path= 'heat_40.txt'
firstrow = 645
time_data_40 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_40 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,4,'mg/L')
concentration_data_40 = concentration_data_40 - concentration_data_40[0]

plt.plot(time_data_24[0:50], concentration_data_24[0:50],'y', label='24C')
plt.plot(time_data_32[0:50], concentration_data_32[0:50],'b', label='28C')
plt.plot(time_data_28[0:50], concentration_data_28[0:50],'r', label='32C')

plt.plot(time_data_36[0:50], concentration_data_36[0:50],'g', label='36C')
plt.plot(time_data_40[0:50], concentration_data_40[0:50],'c', label='40C')
plt.xlabel(r'Time (seconds)')
plt.ylabel(r'DO Concentration(mg/L)')
plt.title(r'DO Concentrations vs Time for Various Temperatures')
plt.legend(bbox_to_anchor=(0, 1), loc=2, borderaxespad=0.)
plt.savefig('DOvT.jpeg')
plt.show()



#OXYGEN SATURATIONS
#TAKEN FROM ProCoDa or https://water.usgs.gov/software/DOTABLES/

O2_sat_24 = 8.35*u.mg/u.L
O2_sat_28 = 7.73*u.mg/u.L
O2_sat_32 = 7.17*u.mg/u.L
O2_sat_36 = 6.66*u.mg/u.L
O2_sat_40 = 6.20*u.mg/u.L

O2_sat_0 = 8.74*u.mg/u.L
O2_sat_10 = 8.25*u.mg/u.L
O2_sat_20 = 7.78*u.mg/u.L
O2_sat_30 = 7.34*u.mg/u.L
O2_sat_35 = 7.13*u.mg/u.L


#Use linear regression function - remove units

xh_24 = -(time_data_24)
DO_data_24 = concentration_data_24
yh_24= np.log((np.array(O2_sat_24.magnitude)-DO_data_24.magnitude)/(np.array(O2_sat_24.magnitude)-DO_data_24
slope24, intercept24, r_value24, p_value24, std_err24 = stats.linregress(xh_24,yh_24)

xh_28 = -(time_data_28)
DO_data_28 = concentration_data_28
yh_28= np.log((np.array(O2_sat_28.magnitude)-DO_data_28.magnitude)/(np.array(O2_sat_28.magnitude)-DO_data_28
slope28, intercept28, r_value28, p_value28, std_err28 = stats.linregress(xh_28,yh_28)
print(r_value28)
print(slope28)

xh_32 = -(time_data_32)

yh_32= np.log((np.array(O2_sat_32.magnitude)-concentration_data_32.magnitude)/(np.array(O2_sat_32.magnitude)
slope32, intercept32, r_value32, p_value32, std_err32 = stats.linregress(xh_32,yh_32)
print(r_value32)


xh_36 = -(time_data_36)
yh_36= np.log((np.array(O2_sat_36.magnitude)-concentration_data_36.magnitude)/(np.array(O2_sat_36.magnitude)
slope36, intercept36, r_value36, p_value36, std_err36 = stats.linregress(xh_36,yh_36)
r_value36
```

```
xh_40 = -(time_data_40)

DO_data_40 = concentration_data_40
yh_40= np.log((np.array(O2_sat_40.magnitude)-DO_data_40.magnitude)/(np.array(O2_sat_40.magnitude)-DO_data_4
slope40, intercept40, r_value40, p_value40, std_err40 = stats.linregress(xh_40,yh_40)
print(r_value40)

k_temp = np.array([slope24, slope28, slope32, slope36, slope40])

#the slope of each linear regression line is equal to k



#Salinity portion

n_rows = 500
data_file_path= 'salt_0.txt'
firstrow = 509
time_data_0 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_0 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,2,'mg/L')

data_file_path= 'salt_10.txt'
firstrow = 655
time_data_10 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_10 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,2,'mg/L')

data_file_path= 'salt_20.txt'
firstrow = 660
time_data_20 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_20 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,2,'mg/L')

data_file_path= 'salt_30.txt'
firstrow = 991
time_data_30 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_30 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,2,'mg/L')

data_file_path= 'salt_35.txt'
firstrow = 1038
time_data_35 = EPA.ftime(data_file_path, firstrow, firstrow+n_rows).to(u.s)
concentration_data_35 = EPA.Column_of_data(data_file_path,firstrow,firstrow+n_rows,2,'mg/L')

plt.plot(time_data_0, concentration_data_0,'m', label='0 mg/L')
plt.plot(time_data_10, concentration_data_10,'y', label='10 mg/L')
plt.plot(time_data_20, concentration_data_20,'r', label='20 mg/L')
plt.plot(time_data_30, concentration_data_30,'b', label='30 mg/L')
plt.plot(time_data_35, concentration_data_35,'g', label='35 mg/L')
#plt.plot(time_data_40, concentration_data_40,'c', label='40 mg/L')
plt.xlabel(r'Time (seconds)')
plt.ylabel(r'DO Concentration (mg/L)')
plt.title(r'DO Concentration vs. Time for Various Salt Concentrations')
plt.legend(bbox_to_anchor=(0, 1.0), loc=2, borderaxespad=0.)
plt.savefig('salinity.png')
plt.show()

#k_values - looped to chose most linear region for k value bc we are evaluting larger time step
```

```
i = 0
r_value0_max = 0
k_0 = 0
for x in range(i, (len(concentration_data_0)-100)):
  for j in range (i+100, (len(concentration_data_0))):
    xs_0 = -(time_data_0[i:j])
    DO_data_0 = concentration_data_0[i:j]
    ys_0=np.log((np.array(O2_sat_0.magnitude)-DO_data_0.magnitude)/(np.array(O2_sat_0.magnitude)-DO_data_0[i
    slope0, intercept0, r_value0, p_value0, std_err0 = stats.linregress(xs_0,ys_0)
    if ((r_value0) > (r_value0_max)):
      r_value0_max = r_value0
      linear0_start = i
      linear0_end = j
      k_0 = slope0

i = 0
r_value10_max = 0
k_10 = 0
for x in range(i, (len(concentration_data_10)-100)):
  for j in range (i+100, (len(concentration_data_10))):
    xs_10 = -(time_data_10[i:j])
    DO_data_10 = concentration_data_10[i:j]
    ys_10=np.log((np.array(O2_sat_10.magnitude)-DO_data_10.magnitude)/(np.array(O2_sat_10.magnitude)-DO_data
    slope10, intercept10, r_value10, p_value10, std_err10 = stats.linregress(xs_10,ys_10)
    if ((r_value10) > (r_value10_max)):
      r_value10_max = r_value10
      linear10_start = i
      linear10_end = j
      k_10 = slope10

i = 0
r_value20_max = 0
k_20 = 0
for x in range(i, (len(concentration_data_20)-100)):
  for j in range (i+100, (len(concentration_data_20))):
    xs_20 = -(time_data_20[i:j])
    DO_data_20 = concentration_data_20[i:j]
    ys_20=np.log((np.array(O2_sat_20.magnitude)-DO_data_20.magnitude)/(np.array(O2_sat_20.magnitude)-DO_data
    slope20, intercept20, r_value20, p_value20, std_err20 = stats.linregress(xs_20,ys_20)
    if ((r_value20) > (r_value20_max)):
      r_value20_max = r_value20
      linear20_start = i
      linear20_end = j
      k_20 = slope20

i = 0
r_value30_max = 0
k_30 = 0
for x in range(i, (len(concentration_data_30)-100)):
  for j in range (i+100, (len(concentration_data_30))):
    xs_30 = -(time_data_30[i:j])
    DO_data_30 = concentration_data_30[i:j]
    ys_30=np.log((np.array(O2_sat_30.magnitude)-DO_data_30.magnitude)/(np.array(O2_sat_30.magnitude)-DO_data
    slope30, intercept30, r_value30, p_value30, std_err30 = stats.linregress(xs_30,ys_30)
    if ((r_value30) > (r_value30_max)):
      r_value30_max = r_value30
      linear30_start = i
      linear30_end = j
      k_30 = slope30

i = 0
r_value35_max = 0
k_35 = 0
for x in range(i, (len(concentration_data_35)-100)):
  for j in range (i+100, (len(concentration_data_35))):
    xs_35 = -(time_data_35[i:j])
    DO_data_35 = concentration_data_35[i:j]
    ys_35=np.log((np.array(O2_sat_35.magnitude)-DO_data_35.magnitude)/(np.array(O2_sat_35.magnitude)-DO_data
    slope35, intercept35, r_value35, p_value35, std_err35 = stats.linregress(xs_35,ys_35)
    if ((r_value35) > (r_value35_max)):
      r_value35_max = r_value35
      linear35_start = i
      linear35_end = j
      k_35 = slope35


k_salinity = np.array([k_0, k_10, k_20, k_30, k_35])

k_temp_day = [x * 60 * 60 * 24 for x in k_heat]
T_h= np.array([24+273, 28+273, 32+273, 36+273, 40+273])

#plot k in days against T to compare to article findings

plt.plot(T_h-273, k_temp_day)
plt.xlabel('Temperature (C)')
plt.ylabel('k (1/day)')
plt.title('Gas transfer Coefficient as a function of Temperature')
plt.savefig('kVST.jpeg')
plt.show()


Salinity = np.array([0, 10, 20, 30, 35])

#represent k * 10^-3 so values fit on x axis better
k_s = [x * 1000 for x in k_salinity]
k_t = [x*1000 for x in k_heat]

#create dual y axis plot to compare salinity and temperature trends against k
from pylab import figure, show, legend, ylabel

# create the general figure
fig1 = figure()
ax1 = fig1.add_subplot(111)
line1 = ax1.plot(k_t, T_h-273, 'or-', label = 'Temperature')
ylabel("Temperature (C)")
legend(bbox_to_anchor=(0.67, 0.4), loc=2, borderaxespad=0.5)

ax2 = fig1.add_subplot(111, sharex=ax1, frameon=False)
line2 = ax2.plot(k_s, Salinity, 'ob-', label='Salinity')
ax2.yaxis.tick_right()
ax2.yaxis.set_label_position("right")
ylabel("Salinity (g/L)")
```

```
legend(bbox_to_anchor=(0.67, 0.33), loc=2, borderaxespad=0.5)
plt.xlabel("k*10^-3 (1/s)")
plt.title('Gas Transfer Coefficient Trend Comparison')
plt.savefig('dual_k.jpeg')
show()

#Oxygen Transfer Efficiency
C_def_s= np.array([O2_sat_0.magnitude,O2_sat_10.magnitude, O2_sat_20.magnitude, O2_sat_30.magnitude, O2_sat_
  C_def_h = np.array([O2_sat_24.magnitude,O2_sat_28.magnitude, O2_sat_32.magnitude, O2_sat_36.magnitude, O2_
R= 0.08206*u.L*u.atm/(u.K*u.mol)
T= 295*u.K
T_h= np.array([24+273, 28+273, 32+273, 36+273, 40+273])
V= 0.5*u.L
MW_O2= 32000*u.mg/u.mol
Q = 250
P= 1*u.atm
f_O2= 0.21


#Calculate an array of oxygen transfer efficiencies using equation 1.9 in the lab manual
#Note:flow rate is in micromols/s, but is converted to L/second in the equation using the factor 22.4/(10^6)
OTE_salinity = (k_salinity*(C_def_s)*V*R*T)/(MW_O2*Q*P*f_O2*(22.4/(10.0**6)))

OTE_heat = (k_temp*(C_def_h)*V*R*T_h)/(MW_O2*Q*P*f_O2*(22.4/(10.0**6)))

plt.scatter(k_s, OTE_salinity, color ='b', label = 'Salinity')
plt.scatter(k_t, OTE_heat, color ='r', label ='Temperature')
#plt.xlim(0.00075,0.003)
plt.ylim(0,0.008)
plt.xlabel('k values')
plt.ylabel(' OTE Efficiency')
plt.title('Gas Transfer Coefficient vs Oxygen Transfer Efficiency')
plt.legend(bbox_to_anchor=(0, 1.0), loc=2, borderaxespad=0.)
plt.savefig('OTE_both.jpeg')
plt.show()

##Code to display DO probe issues for presentation

data_file_path= '2probes_bad.txt'
firstrow = 500
time_data_probes = EPA.ftime(data_file_path, firstrow, -1).to(u.s)
 probe1 = EPA.Column_of_data(data_file_path,firstrow,-1,2,'mg/L')
probe2 = EPA.Column_of_data(data_file_path,firstrow,-1,3,'mg/L')

plt.plot(time_data_probes, probe1,'r')
plt.plot(time_data_probes, probe2,'c')
plt.xlabel(r'Time (seconds)')
plt.ylabel(r'Dissolved Oxygen Concentration (mg/L)')
plt.title('Comparing 2 DO Probes for Salinity = 0 mg/L')
plt.savefig('2badprobes.jpeg')
plt.show()
```

## Appendix B: References

[1] Nguyen, M.; Tan, D.; Tan, S.; Reaeration Model for a Still-Water Body, 2014. https://ascelibrary.org/doi/full/10.1061/(ASCE)EE.1943-7870.0000882?src=recsys&

[2] The Influence of Water Temperature & Salinity on Dissolved Oxygen Saturation. http://www.irrec.ifas.ufl.edu/teachaquaculture/curriculum/_files/modules/5_waterquality/Activity/How_DO_Changes_with_Salinity.pdf

[3] Hill, G., Oxygen Mass Transfer Correlations for Pure and Salt Water in a Well-Mixed Vessel, 2009. https://pubs.acs.org/doi/pdf/10.1021/ie8019906

[4] RAMP, Water Quality Indicators: Temperature and Dissolved Oxygen. http://www.ramp-alberta.org/river/water+sediment+quality/chemical/temperature+and+dissolved+oxygen.aspx