Residence Time Distribution
TKP4110

Eva Tula Larsen Aunet and Anine Bodsberg

October 2021

1 Introduction and Objectives

In this experiment, the residence time in a real non-reactive system where back-mixing is allowed will be studied and compared with that of an ideal plug flow reactor with fluid assumed to only flow in one direction. This will be done by injecting methylene blue into the reactor, and using the difference in top and bottom sensor voltage to calculate the residence time distribution for three different concentrations at three different pump rates.

The learning objectives of this experiment are to know how the residence time in a real reactor differs from that of an ideal plug flow reactor, learn how to translate raw data into a residence time distribution, present the data in appropriate figures and apply regression techniques to calibrate the equipment.

2 Theory

The following section is based on the theory from *RE9: Residence Time Distribution*[1].

In chemical reaction engineering, the residence time describes the time it takes from a small amount of fluid enters a reactor until it leaves the reactor. For an ideal plug flow reactor, the fluid is assumed to flow in only one direction. Here, the velocity of the molecules is constant throughout the reactor, meaning that all molecules share the same residence time. For a real reactor, a portion of the reacted fluid might intermingle with the unreacted feed. This phenomenon is called back-mixing, and will cause different molecules to have different residence times, yielding a residence time distribution. The residence time distribution can be viewed as a probability density distribution, $E(t)$, which can be calculated from

$$E(t) = \frac{\hat{n}}{n_0}, \tag{2.1}$$

where $\hat{n}$ is the molar flow of tracer out of the reactor and n_0 is the total molar amount of injected tracer. This equation only applies to non-reactive systems, as n_0 is not conserved in reactive systems. Considering the volumetric flow-rate q and the concentration of the tracer $c(t)$, the molar flows can be substituted, and equation (2.1) can be rewritten as

$$E(t) = \frac{qc(t)}{\int_0^\infty qc(t)dt}. \tag{2.2}$$

Under the assumption of steady state flow conditions, q is constant. Taking q out of the integral, equation (2.2) can finally be rewritten as

$$E(t) = \frac{c(t)}{\int_0^\infty c(t)dt}. \tag{2.3}$$

The distribution $E(t)$ depicts the probability that a tracer molecule spends a time t inside the reactor[1]. All tracer molecules must eventually have left the reactor as t approaches infinity, meaning that the probability of a tracer molecule having left the reactor between $t = 0$ and $t = \infty$ equals 1. This is shown mathematically by

$$\int_0^\infty E(t)dt = 1. \tag{2.4}$$

The mean residence time $\bar{t}$ can be calculated from

$$\bar{t} = \int_0^\infty tE(t)dt. \tag{2.5}$$

The integrals shown in equation (2.3), (2.4) and (2.5) can be calculated using numerical methods such as the method of trapezoids or Simpson's method.

The residence time for an ideal plug flow reactor τ can be calculated from

$$\tau = \frac{V_r}{q}, \tag{2.6}$$

where V_r is the reactor volume.

The voltage of the stream measured by the top (T) and bottom (B) sensor is related to the concentration c of the tracer as

$$\Delta U_T = \beta_T \cdot c \tag{2.7}$$

and

$$\Delta U_B = K_B(1 - e^{(-\frac{c}{\tau_{p,B}})}). \tag{2.8}$$

Here, ΔU_T and ΔU_B are the differences in measured voltages at the top sensor and the bottom sensor respectively to the idle resting values for the respective sensors. This is shown mathematically in equation (2.9). Further, $\beta_T = 0.073$, $K_B = 1.636$, $\tau_{p,B} = 14.50$, $U_{T,idle} = 4.5695$ and $U_{B,idle} = 4.9012$ for this experiment.

$$\Delta U_i = U_{i,idle} - U_i \tag{2.9}$$

3 Experimental

3.1 Setup

The experimental setup consisted of a tank of deionized water which was pumped through a reactor and into a waste tank. Two light sensors were placed before and after the reactor. The optical tracer was injected after the pump, but before the first sensor. A flow sheet of the experimental setup is presented in figure 3.1.1.

3.2 Residence Time Distribution

Three methylene blue solutions with concentrations of 50.8 mg/L, 87.2 mg/L and 126.8 mg/L were prepared by weighing salt on an analytical balance, dissolving it in deionized water and diluting the solution to 250 mL. Approximately 1 mL of the solution was injected quickly into the reactor to ensure that all molecules entered the system at approximately the same time. This was done three times for each solution, while keeping the pump rate steady at 400, 800 and 1200 rpm respectively.

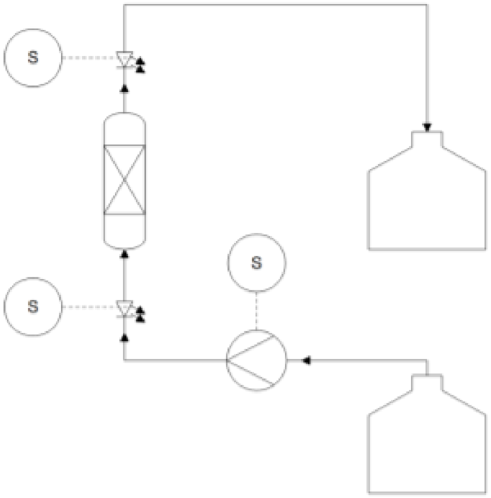


Figure 3.1.1: Flow sheet of the experimental setup, including a tank of deionized water, a pump, two light sensors, the point of injection, a reactor and a waste tank.

3.3 Pump Calibration

The pump was calibrated by registering the time it took for the pump to consume a certain volume of deionized water. This was done twice for pump rates of 400, 800 and 1200 rpm. The volume consumed and time elapsed was noted in the lab journal, shown in figure B.1.

4 Results

Figure 4.1 shows the response $c(t)$ of the top and bottom sensor for a 50.8 mg/L methylene blue solution at 400 rpm. The injected tracer reaches the bottom sensor first, before it goes into the reactor and exits through the top sensor. The tracer uses a longer time interval to pass the top sensor compared to the bottom sensor.

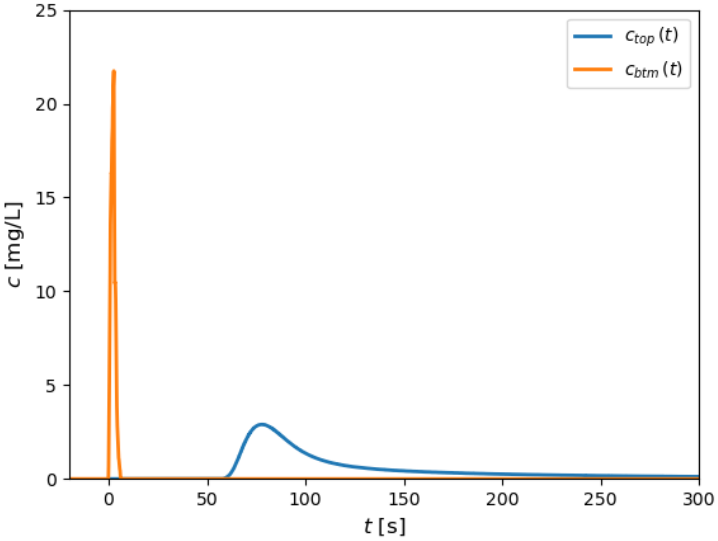


Figure 4.1: The response $c(t)$ of the top and bottom sensor for a 50.8 mg/L methylene blue solution at 400 rpm.

Figures 4.2, 4.3 and 4.4 show the residence time distribution for three methylene blue solutions with concentrations of respectively 50.8 mg/L, 87.2 mg/L and 126.8 mg/L at pump speeds of 400 rpm, 800 rpm and 1200 rpm. For each of the graphs, the solution injected at the highest pump rate spent the shortest time in the reactor, while the solution injected at the lowest pump rate spent the longest time in the reactor.

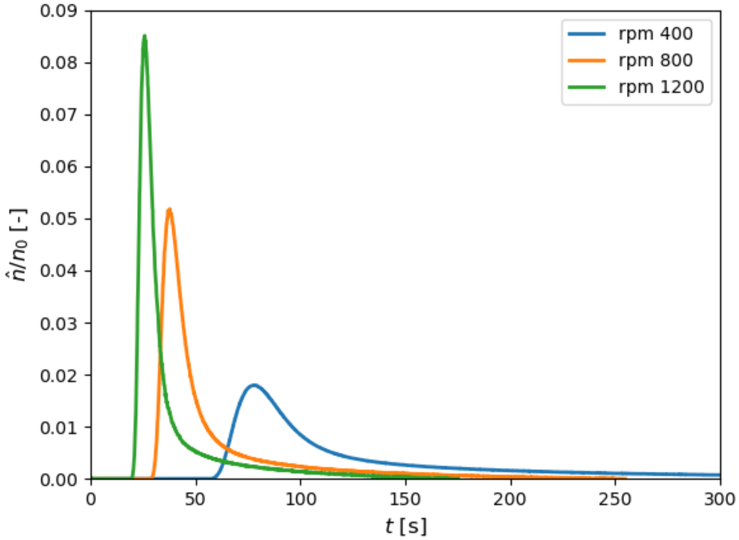


Figure 4.2: Residence time distribution from a 50.8 mg/L methylene blue solution at three different pump rates.

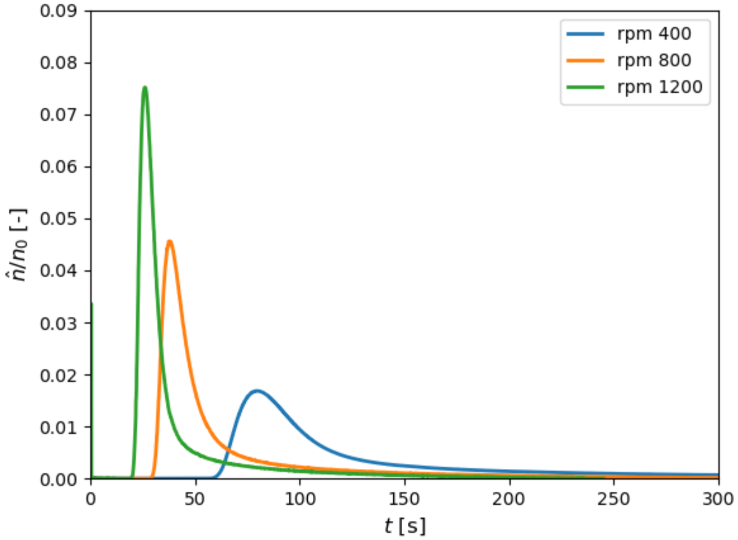


Figure 4.3: Residence time distribution from a 87.2 mg/L methylene blue solution at three different pump rates.

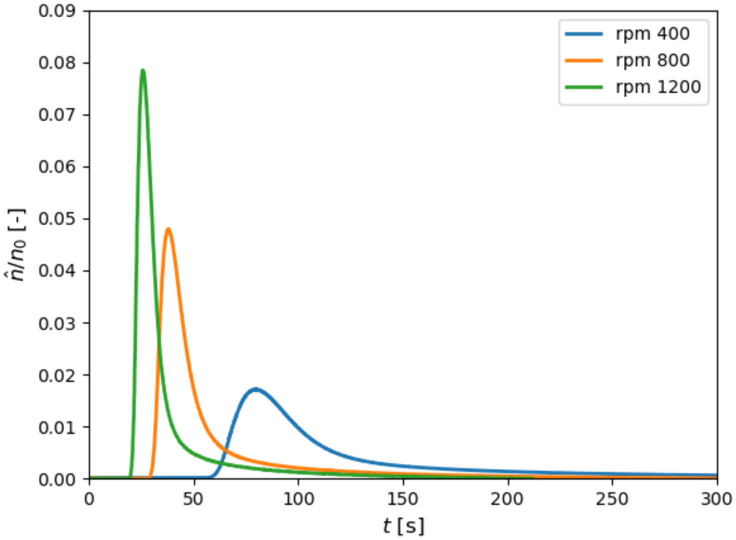


Figure 4.4: Residence time distribution from a 126.8 mg/L methylene blue solution at three different pump rates.

Figures 4.5, 4.6 and 4.7 show the residence time distribution for three methylene blue solutions with concentrations of 50.8 mg/L, 87.2 mg/L and 126.8 mg/L at pump speeds of respectively 400 rpm, 800 rpm and 1200 rpm. The residence time distributions line up almost perfectly in all three graphs.

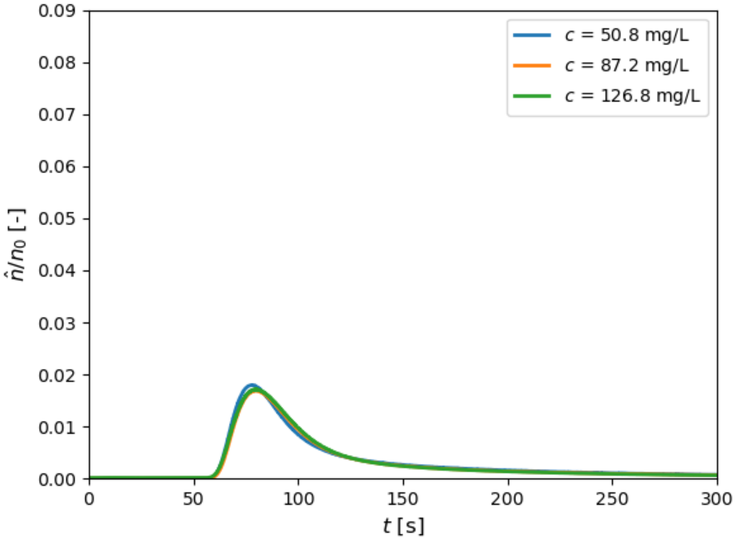


Figure 4.5: Residence time distribution from three methylene blue solutions at 400 rpm.

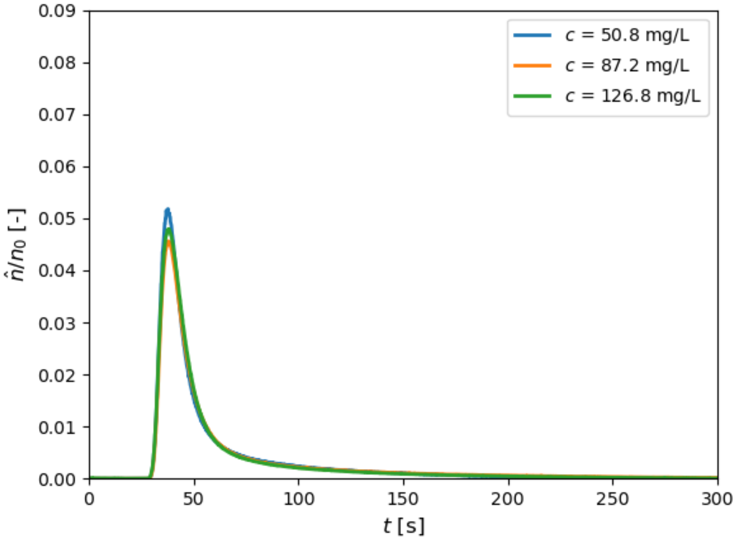


Figure 4.6: Residence time distribution from three methylene blue solutions at 800 rpm.

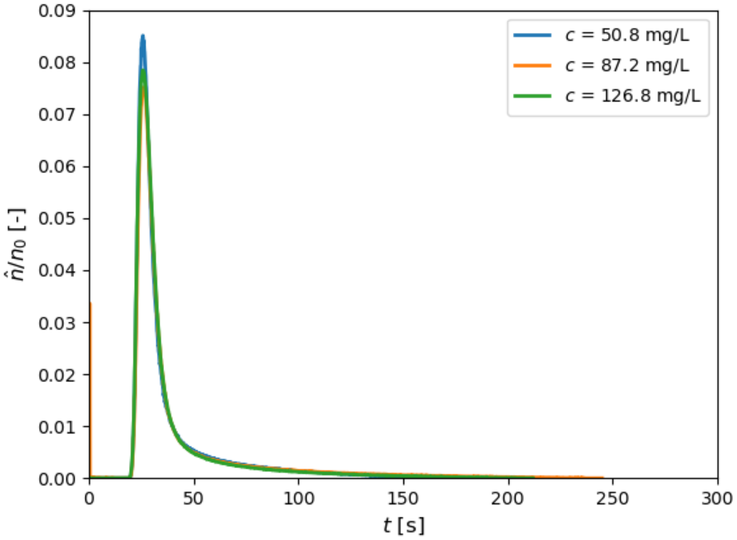


Figure 4.7: Residence time distribution from three methylene blue solutions at 1200 rpm.

Figure 4.8 shows the calibration curve with volumetric flow rate as a function of the pump rate. This graph was found by measuring the time it took to pump approximately 80 mL deionized water into the reactor at different pump rates and then using linear regression. The regression line was found to be $q = 0.0017x - 0.0405$, where x is the pump rate. The calibration curve has an R^2-value of 0.99999, which shows that the regression line fits the six data points from table A.2.1 almost perfectly.

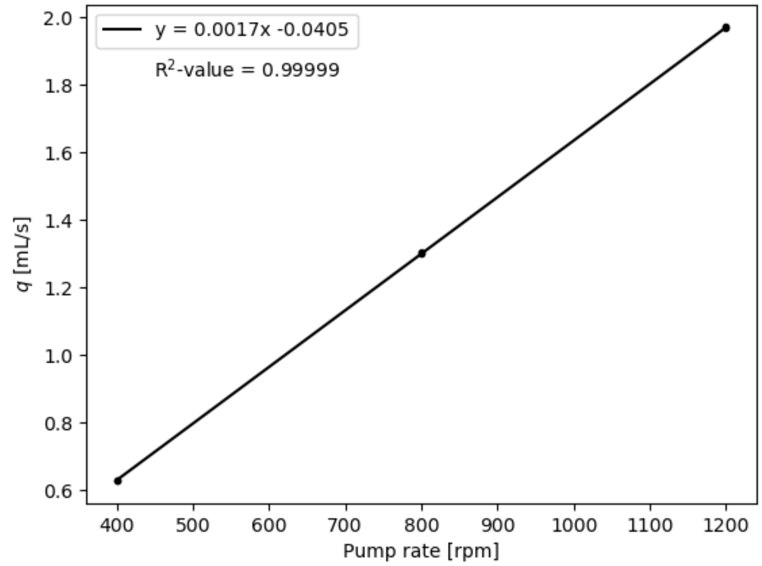


Figure 4.8: Calibration curve with volumetric flow q as a function of the pump rate.

The mean residence time, $\bar{t}$, was calculated using equation 2.5 and the python code found in appendix C. The residence time for an ideal plug flow reactor, τ, was calculated from equation (2.6), and an example calculation can be found in appendix A.3. The results for both the mean residence time and the ideal PFR residence time are given in table 4.1. From table 4.1, it is observed that τ is smaller than $\bar{t}$ for all pump rates and concentrations. The relationship between τ and $\bar{t}$ is observed to stay approximately constant as the pump rate is increased, with $\bar{t}$ being approximately twice as big as τ.

Table 4.1: Mean residence time and ideal PFR residence time for different concentrations and pump rates.

Pump rate [rpm]	Concentration [mg/L]	$\bar{t}$ [s]	τ [s]
400	50.8	141.39	69.8
400	87.2	155.00	69.8
400	126.8	144.20	69.8
800	50.8	59.49	33.8
800	87.2	66.98	33.8
800	126.8	61.40	33.8
1200	50.8	38.86	22.3
1200	87.2	42.91	22.3
1200	126.8	39.36	22.3

5 Discussion

The difference in response $c(t)$ of the top and bottom sensor, seen in figure 4.1, suggests that the tracer spreads in the solvent while going through the reactor. For an ideal plug flow reactor, the tracer would not spread in the solvent like this because the velocity of the molecules would be constant throughout the reactor, meaning that all molecules would share the same residence time. However, the reactor operates under real conditions where back-mixing is allowed. Thus, the fluid in the reactor intermingles, causing the tracer to spread out and spend more time passing the top sensor.

The solutions injected at higher pump rates spend a shorter time in the reactor because a higher pump rate leads to a higher volumetric flow rate, which again leads to a shorter residence time. This fits well with the results seen in figures 4.2, 4.3 and 4.4. Considering the similar time scales and the negligible differences between the residence time distributions for the different concentrations in figures 4.5, 4.6 and 4.7, it is reasonable to assume that the deviations are due to errors. Thus, the concentration was not found to have any effect on the residence time distribution.

The tracer molecules were observed to spend more time moving through the reactor for the real reactor model that was used in this experiment, compared to an ideal reactor model. This is because some tracer molecules undergo the process of back-mixing in a real reactor, causing the tracer to spread out in the solvent and spending more time going through the reactor. For an ideal plug flow reactor, however, the fluid is assumed to flow in only one direction, which yields a higher volumetric flow rate and thus a shorter residence time. When calculating the residence time for an ideal plug flow reactor, τ, the volumetric flow rate, q, was found using the calibration curve shown in figure 4.8. As the R^2-value of the calibration curve was very close to 1, the volumetric flow rates were not a significant error source when calculating τ.

The calculations associated with this experiment are based on a couple of assumptions. Firstly, the system was assumed to be non-reactive. This is a fair assumption, as methylene blue is a non-reactive substance. Secondly, the system was assumed to operate under steady state flow conditions. Because the pump rate was kept steady throughout the experiments, this assumption is reasonable. Finally, the density of the fluid was assumed to be constant. This assumption is also reasonable, as the system was non-reactive and the temperature was kept constant throughout the experiment.

When weighing out the salt on the analytical balance, there were some uncertainties in the masses, as it was nearly impossible to weigh out the desired amounts, and also because the scale was hard to operate. This is a major source of error in the experiment, as it would affect the concentrations of the solutions, which in turn would affect the residence times.

References

[1] B. Baumgarten. Re9: Residence time distribution. `https://folk.ntnu.no/preisig/HAP_Specials/Felles_lab/Experiments/RE9_milli-reactor_residence_time.pdf`, 2021. Accessed: 27.10.2021.

A Calculations

A.1 Concentrations of the Solutions

Equation (A.1.1) was used to calculate the concentrations, c, of the solutions.

$$c = \frac{m}{V} \tag{A.1.1}$$

Here m is the mass and V is the volume. Table A.1.1 shows the mass of methylene blue and volume water used to obtain the concentrations used in this experiment.

Table A.1.1: Concentrations of the solutions.

Sample	m [mg]	V [L]	c [mg/L]
1	12.7	0.25	50.8
2	21.8	0.25	87.2
3	31.7	0.25	126.8

A.2 Calibration Curve

The measured values for time and volume for each pump rate can be found in table A.2.1, along with the volumetric flow rates calculated from equation (A.2.1).

$$q = \frac{\Delta V}{\Delta t} \tag{A.2.1}$$

Table A.2.1: Measured data used to find q at different pump rates.

Pump rate [rpm]	ΔV [mL]	Δt [s]	q [mL/s]
400	77.5	123.40	0.6280
400	81.7	129.60	0.6204
800	83.0	63.78	1.3014
800	80.8	61.97	1.3039
1200	80.6	40.94	1.9688
1200	84.0	42.59	1.9723

A.3 Residence Time for an Ideal Plug Flow Reactor

The residence time for an ideal plug flow reactor τ was found using equation (2.6). The following calculation shows how τ was determined for the three samples at a pump speed of 400 rpm.

The volumetric flow rate q used in the calculation was determined from the calibration curve shown in figure 4.8, and found to be 0.63013 mL/s. The reactor volume V_r was 44 mL.

$$\tau_{400\,rpm} = \frac{44\,\mathrm{mL}}{0.63013\,\mathrm{mL/s}} = 69.8\,\mathrm{s}$$

A.4 RTD Integrals

By integrating the graphs in python, it was confirmed that they all followed equation (2.4), meaning that the integrals were all equal to 1. A manual calculation for the 50.8 mg/L sample at 400 rpm was done using the trapezoidal method with three trapezoids. The values used in the calulcations are found in table A.4.1.

Table A.4.1: Values for t and $\hat{n}/n_0$ used in manual calculation with the trapezoidal method.

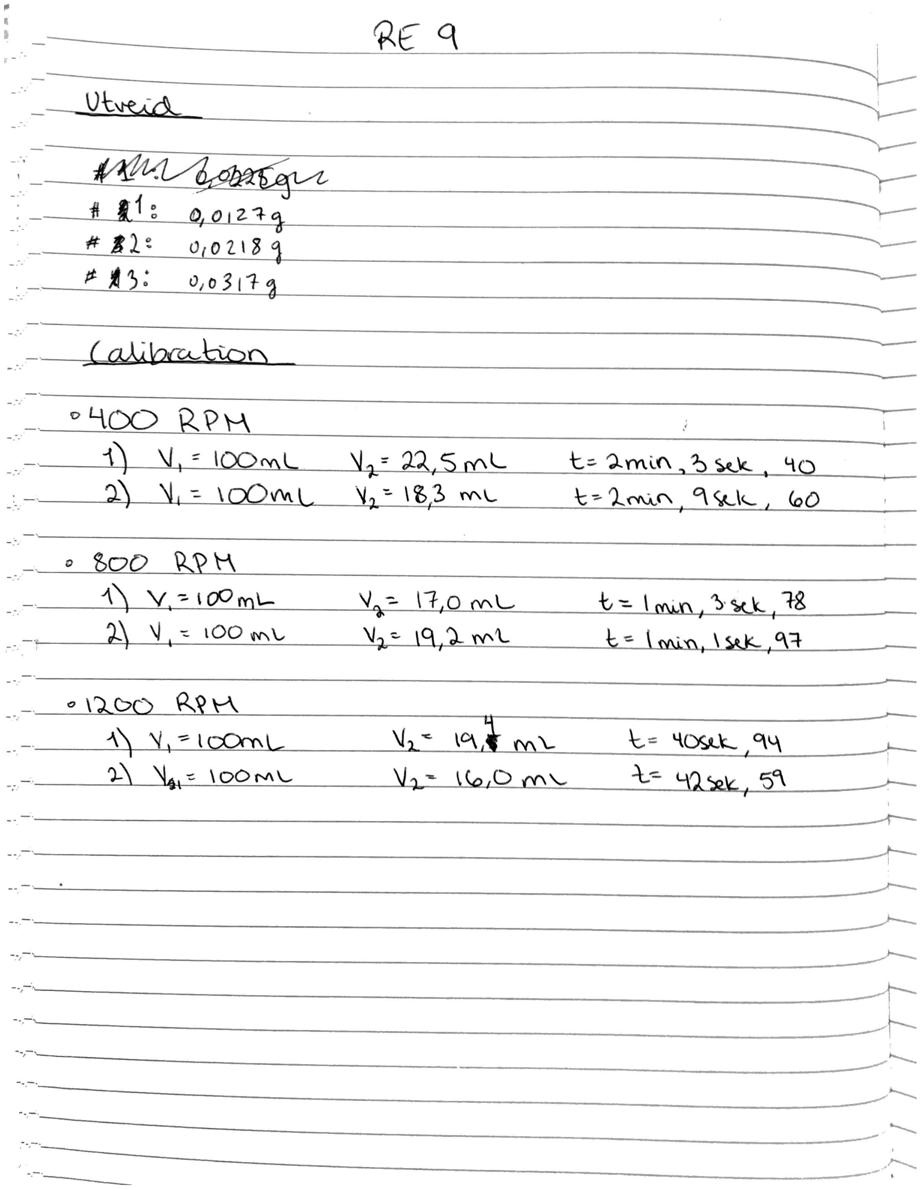
i	0	1	2	3
t [s]	0	20	62	242
$\hat{n}/n_0$ [-]	0	0.0180	0.0047	0

Intergrating $E(t)$ from $t = 0$ to $t = \infty$ with the trapezoidal method gives:

$$\int_0^\infty E(t)dt = \frac{0.0180 + 0}{2} \cdot (20 - 0) + \frac{0.0047 + 0.0180}{2} \cdot (62 - 20) + \frac{0 + 0.0047}{2} \cdot (242 - 62) = 1.0797,$$

which is close to the real value of 1.

B Lab Journal



RE 9

Utveid

~~[illegible]~~

\# 1: 0,0127 g

\# 2: 0,0218 g

\# 3: 0,0317 g

Calibration

- 400 RPM
  1) V_1 = 100mL V_2 = 22,5 mL t = 2min, 3 sek, 40
  2) V_1 = 100mL V_2 = 18,3 mL t = 2min, 9 sek, 60
- 800 RPM
  1) V_1 = 100mL V_2 = 17,0 mL t = 1 min, 3 sek, 78
  2) V_1 = 100 mL V_2 = 19,2 mL t = 1 min, 1 sek, 97
- 1200 RPM
  1) V_1 = 100mL V_2 = 19,4 mL t = 40sek, 94
  2) V_1 = 100mL V_2 = 16,0 mL t = 42sek, 59

Figure B.1: Photocopy of the lab journal used during the experiment.

C Python Code

```
"""
23.10.2021
TKP4110 RE9
Anine Bodsberg & Eva Aunet
"""

import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score

x = np.array([400, 400, 800, 800, 1200, 1200])
y = np.array([0.62804, 0.63040, 1.30135, 1.30386, 1.96873, 1.97229])

a, b = np.polyfit(x, y, 1)
y_pred = a*x + b
r2 = round(r2_score(y, y_pred), 5)
plt.plot(x, y, '.', color='black')
plt.plot(x, y_pred, label='y = {:.4f}x {:.4f}'.format(a, b), color='black')
plt.text(448, 1.82, r'R$^2$-value = '+f'{r2}')
plt.xlabel('Pump rate [rpm]')
plt.ylabel(r'$Q$ [mL/s]')
plt.legend()
plt.show()
```

```
"""
23.10.2021
TKP4110 RE9
Anine Bodsberg & Eva Aunet
"""

import numpy as np
import matplotlib.pyplot as plt
import csv
import scipy
from scipy import integrate


# funksjon som tar ut en liste fra csv-fil
def extractor(fil):
    list1 = []
    with open(fil) as f:
        reader = csv.reader(f)
        next(reader)  # skip header
        data = [r for r in reader]
    for i in data:
        list1.append(i)
    return list1


# funksjon som lager lister med tid og spenning for topp- og bunnsensor
def sorting(list2):
    ut = []  # spenning toppsensor
    tt = []  # tid toppsensor
    ub = []  # spenning bunnsensor
```

```
    tb = []  # tid bunnsensor
    for k in range(2, 6):  # tar ikke med kolonnene for rpm
        for i in range(0, len(list2)):
            if k == 2:
                if list2[i][k] != 'NA':
                    tt.append(float(list2[i][k]))
            if k == 3:
                if list2[i][k] != 'NA':
                    ut.append(float(list2[i][k]))
            if k == 4:
                if list2[i][k] != 'NA':
                    tb.append(float(list2[i][k]))
            if k == 5:
                if list2[i][k] != 'NA':
                    ub.append(float(list2[i][k]))
    return tt, ut, tb, ub


# finner konsentrasjonen i toppsensor og setter tiden den entrer bunnsensor som null
def ctop(u_top, t_top, u_btm, t_btm):
    ct = []
    t = []
    bt = 0.0730
    Ut_idle = 4.5695
    Kb = 1.636
    Ub_idle = 4.9012
    tp = 14.50
    t0 = 0
    if t0 == 0:  # finner tiden det entrer bunnsensoren
        for i in range(0, len(u_btm)):
            if t0 == 0:
                deltaV = Ub_idle - u_btm[i]
                cons = -tp * np.log(1 - (deltaV / Kb))
                if cons > 0.01:
                    t0 = t_btm[i]
    for i in range(0, len(u_top)):
        if t_top[i] >= t0:
            deltaV = Ut_idle - u_top[i]
            c = deltaV / bt
            if c < 0:
                ct.append(0)
            else:
                ct.append(c)
            t.append(t_top[i] - t0)
    return ct, t


# finner konsentrasjonen i bunnsensor og setter tiden den entrer bunnsensor som null
def cbtm(u_btm, t_btm):
    cb = []
    tb = []
    t0 = 0
    Kb = 1.636
    Ub_idle = 4.9012
    tp = 14.50
```

```
    if t0 == 0:  # finner tiden det entrer bunnsensoren
        for i in range(0, len(u_btm)):
            if t0 == 0:
                deltaV = Ub_idle - u_btm[i]
                cons = -tp * np.log(1 - (deltaV / Kb))
                if cons > 0.01:
                    t0 = t_btm[i]
    for i in range(0, len(u_btm)):
        deltaV = Ub_idle - u_btm[i]
        c = -tp * np.log(1 - (deltaV / Kb))
        if c < 0:
            cb.append(0)
        else:
            cb.append(c)
        tb.append(t_btm[i] - t0)
    return cb, tb


# lager funksjon for E
def E(integral, concentration):
    E_list = []
    for i in range(0, len(concentration)):
        Ee = concentration[i] / integral[-1]
        E_list.append(Ee)
    return E_list


# liste med alle filnavnene
n = ['sample1_400rpm.csv', 'sample1_800rpm.csv', 'sample1_1200rpm.csv',
     'sample2_400rpm.csv', 'sample2_800rpm.csv', 'sample2_1200rpm.csv',
     'sample3_400rpm.csv', 'sample3_800rpm.csv', 'sample3_1200rpm.csv']


# funksjon som fikser riktig label på grafene. k = det som holdes konstant (kons eller rpm)
def name(k, filnavn):
    if k == 1:  # konsentrasjon er kontant
        if '400' in filnavn:
            na = 'rpm 400'
        elif '800' in filnavn:
            na = 'rpm 800'
        elif '1200' in filnavn:
            na = 'rpm 1200'
    elif k == 2:  # pump rate er konstant
        if 'sample1' in filnavn:
            na = r'$c$ = 50.8 mg/L'
        elif 'sample2' in filnavn:
            na = r'$c$ = 87.2 mg/L'
        elif 'sample3' in filnavn:
            na = r'$c$ = 126.8 mg/L'
    return na


# funksjon som setter alt sammen og gir ønsket output
# (mean residence time, integral av alle E(t) og printer grafene)
def main(filene):
```

```
list1 = extractor(filene[0])
tt, ut, tb, ub = sorting(list1)
ytc, t_top = ctop(ut, tt, ub, tb)  # konsentrasjon i toppsensor med tider
ybc, t_btm = cbtm(ub, tb)  # samme for bunnkonsentrasjonen
xcb = np.array(t_btm)
ybc = np.array(ybc)
xct = np.array(t_top)
ytc = np.array(ytc)
plt.plot(xct, ytc, linewidth=2.0, label=r"$c_{top}\,(t)$")
plt.plot(xcb, ybc, linewidth=2.0, label=r"$c_{btm}\,(t)$")
plt.xlabel(r"$t$ [s]", fontsize=12)
plt.ylabel(r"$c$ [mg/L]", fontsize=12)
plt.xlim(-20, 300)
plt.ylim(0, 25)
plt.legend()
plt.show()
counter = 0
for fil in filene:
    list2 = extractor(fil)  # liste fra csv-fil
    tt, ut, tb, ub = sorting(list2)  # tid og spenning for topp- og bunnsensor
    ytc, t_top = ctop(ut, tt, ub, tb)  # konsentrasjon i toppsensor med tider
    integral = scipy.integrate.cumtrapz(ytc, t_top)  # tot integral av konsentrasjoner
    e = E(integral, ytc)  # E(t)
    xct = np.array(t_top)
    yct = np.array(e)
    if counter != 3:
        plt.plot(xct, yct, linewidth=2.0, label=name(1, fil))
        t_mean = np.trapz(e * xct, xct)  # mean residence time
        print(fil, 'har t på', t_mean)
        inte = np.trapz(e, xct)  # integralet av hele E(t)
        print('Integralet av hele E er', inte)
        counter += 1
    if counter == 3:
        counter = 0
        plt.xlabel(r'$t$ [s]', fontsize=12)
        plt.ylabel(r'$\^n/n_0$ [-]', fontsize=12)
        plt.xlim(0, 300)
        plt.ylim(0, 0.09)
        plt.legend()
        plt.show()
for i in range(0, 3):
    for k in range(0, 3):
        list2 = extractor(filene[i + k * 3])  # liste fra csv-fil
        tt, ut, tb, ub = sorting(list2)  # tid og spenning for topp- og bunnsensor
        ytc, t_top = ctop(ut, tt, ub, tb)  # konsentrasjon i toppsensor med tider
        integral = scipy.integrate.cumtrapz(ytc, t_top)  # tot integral av konsentrasjoner
        e = E(integral, ytc)  # E(t)
        xct = np.array(t_top)
        yct = np.array(e)
        if counter != 3:
            plt.plot(xct, yct, linewidth=2.0, label=name(2, filene[i + k * 3]))
            counter += 1
        if counter == 3:
            counter = 0
            plt.xlabel(r'$t$ [s]', fontsize=12)
```

```
                plt.ylabel(r'$\^n/n_0$ [-]', fontsize=12)
                plt.xlim(0, 300)
                plt.ylim(0, 0.09)
                plt.legend()
                plt.show()


main(n)
```