

# Supporting Information

## Hands-on Kinetic Measurements and Simulation for Chemical Process Engineering Students

Stefan Herrmann<sup>1,♥</sup>, Daniel Felder<sup>1,2,♥</sup>, Maria Padliger<sup>1,♥</sup>, Sebastian Brosch<sup>1,♥</sup>, Matthias Geiger<sup>1,♥</sup>, Felix Stockmeier<sup>1,2</sup>, Kristina Baitalow<sup>1</sup>, Deniz Rall<sup>1,2</sup>, Robert Femmer<sup>1</sup>, Florian Roghmans<sup>1</sup>, Martin Hauser<sup>1</sup>, Jannik Mehlis<sup>1</sup>, John Linkhorst<sup>1</sup>, and Matthias Wessling<sup>1,2,\*</sup>

<sup>1</sup>RWTH Aachen University, AVT.CVT - Chair of Chemical Process Engineering, Forckenbeckstraße. 51, 52074 Aachen, Germany

<sup>2</sup>DWI - Leibniz Institute for Interactive Materials, Forckenbeckstr. 50, 52074 Aachen, Germany

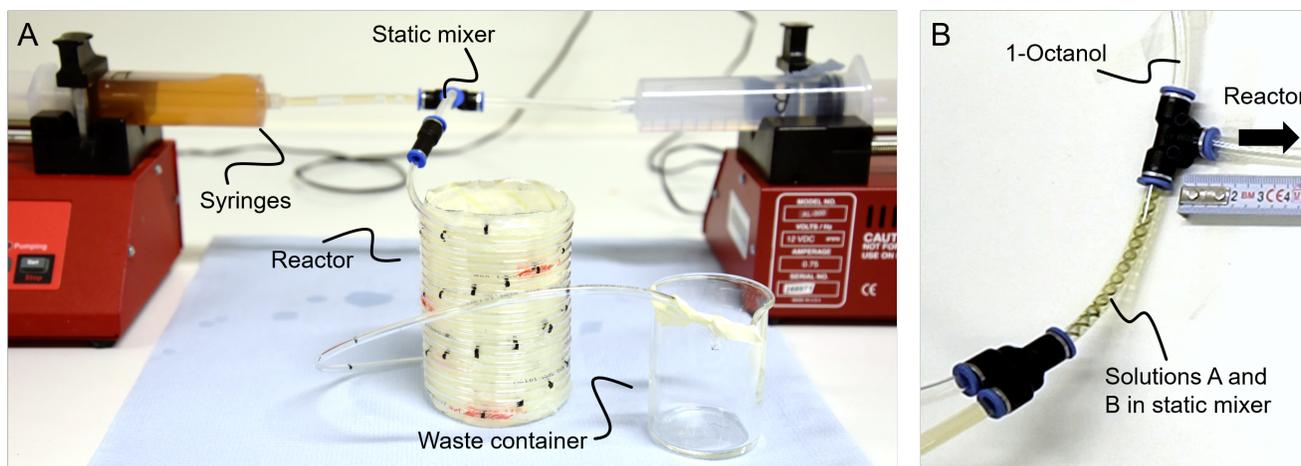
♥these authors contributed equally to this work

\*manuscripts.cvt@avt.rwth-aachen.de

### S1 Handout for Experiments

Student handouts to give an overview of the experimental setup and support experimental planning are appended to this document.

### S2 Lab Setup



**Figure S1.** A: experimental setup of the reactor in continuous flow mode. B: for the bubbletrain reactor, an additional T-junction is inserted at which the 1-octanol phase is added to the same reactor as shown in A.

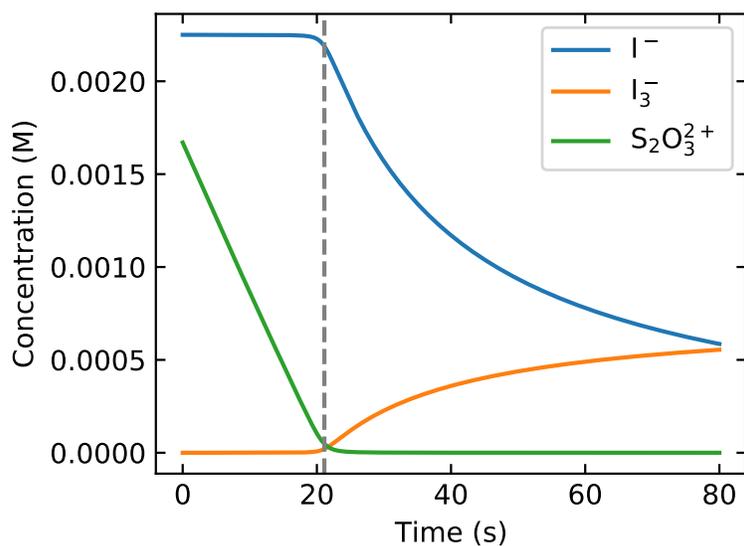
### S3 Python Code

Students are given a jupyter notebook. A printout of the notebook is appended to this document. The notebook is available at <https://git.rwth-aachen.de/avt.cvt/public/acvt>.

### S4 Color Sensor

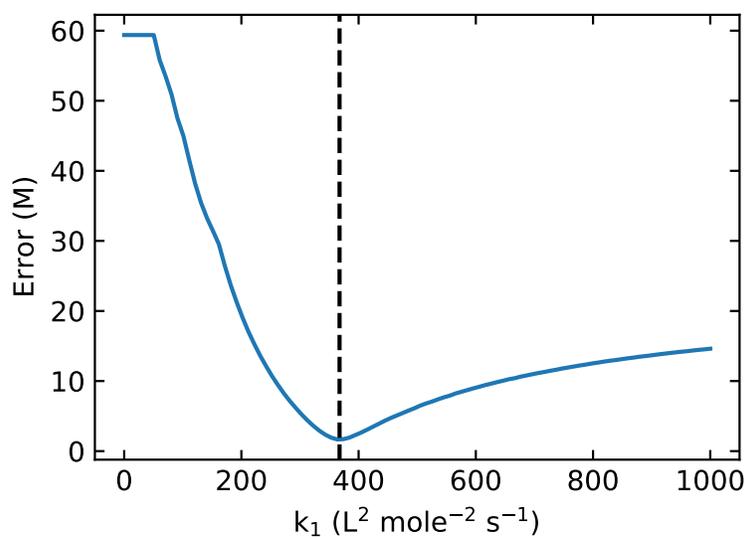
Schematics and code for the color sensor are available at <https://git.rwth-aachen.de/avt.cvt/public/acvt/-/tree/main/sensor>.

## S5 Concentration Profiles



**Figure S2.** Concentration profiles for  $k_1 = 300 \text{ L}^2 \text{ mol}^{-2} \text{ s}^{-1}$  and a  $\text{Fe}_3^+$  concentration of  $14 \times 10^{-3} \text{ mol L}^{-1}$ .

## S6 Error Function



**Figure S3.** Example of the fit error function for the batch reactor for a  $\text{Fe}_3^+$  concentration of  $14 \times 10^{-3} \text{ mol L}^{-1}$ .

# Applied Chemical Process Engineering

## Part A: Experiments

### Landolt Reaction

<b>Chemicals:</b>	<b>Material:</b>
Starch	Beakers, graduated cylinder
Sodium thiosulfate	Set of scales
Potassium iodide	Tubing, fittings
Acetic acid	Static mixer (3D-printed in house)
Iron(III) chloride	Syringe pumps + 3 syringes (60 ml)
DI water	Waste container

**Reaction solutions:**

A1:

<b>Chemicals:</b>	<b>Mass (g)</b>
Acetic acid	4,5060
Iron(III) chloride	2,0245
DI water	249,22

A2:

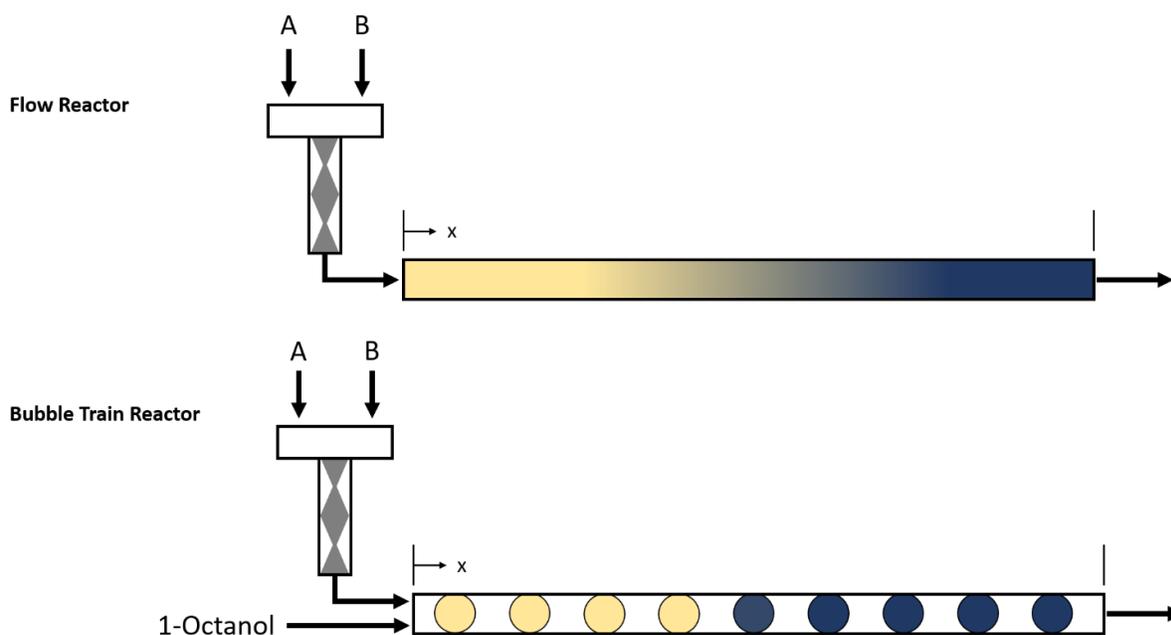
<b>Chemicals:</b>	<b>Mass (g)</b>
Acetic acid	9,0022
Iron(III) chloride	0
DI water	498,5

B:

<b>Chemicals:</b>	<b>Mass (g)</b>
Starch (1 wt% boiled in 20 ml DI water)	0,2 (0,04 wt%)
Sodium thiosulfate	0,254
Potassium iodide	3,573
DI water	478,61

## Procedure:

The reactors are set up according to the schematic shown below.



### A: Continuous Flow Reactor

1. Each group is given 200 ml of solution A1, 400 ml of solution A2 and 400 ml of solution B.
2. The flow reactor is set up, the outlet is placed in a beaker and two syringe pumps are set up as the reactor inlets.
3. Solution A is prepared by mixing solutions A1 and A2. For that, a graduated cylinder is used.
4. By changing the ratio of solutions A1 and A2, different iron(III) chloride concentrations are obtained.
5. One syringe is filled with solution A and the other syringe is filled with solution B. Both syringes are placed in the syringe pumps and marked appropriately. The syringe pumps are set up according to the diameter of the used syringes and the flow rates are set up to be the same for both pumps.
6. The experiment is performed for different concentrations of iron(III) chloride. Results are recorded in tables and the location of color change is measured. In addition, a photograph of the reactor is made. **Cameras or smart phones must not be touched wearing gloves!**

### B: Bubble Train Reactor

1. Each group is given 200 ml of solution A1, 400 ml of solution A2 and 400 ml of solution B. Additionally, students are provided with 1-octanol as the continuous phase.
2. The flow reactor is set up, the outlet is placed in a beaker and two syringe pumps are set up as the reactor inlets. A third syringe pump is set up for the 1-octanol and connected *via* a T-junction between the mixer and the reactor.
3. Solution A is prepared by mixing solutions A1 and A2. For that, a graduated cylinder is used.
4. By changing the ratio of solutions A1 and A2, different iron(III) chloride concentrations are obtained.
5. One syringe is filled with solution A and the other syringe is filled with solution B. Both syringes are placed in the syringe pumps and marked appropriately. The syringe pumps are set up according to the diameter of the used syringes and the flow rates are set up to be the same for both pumps. A third syringe containing 1-octanol is placed in the third syringe pump and a flow rate approx. four times of the individual reactant solutions is set up.
6. The experiment is performed for different concentrations of iron(III) chloride. Results are recorded in tables and the location of color change is measured. In addition, a photograph of the reactor is made. **Cameras or smart phones must not be touched wearing gloves!**

### Cleanup

1. After the experiment is finished, the reaction solution is disposed of in the canister marked „acids“. The canister must not be filled over 90%!
2. All used materials are cleaned thoroughly and the syringes are disposed of in the bin marked „plastics“.

### Task:

- a) Determine the reaction rate at different iron(III) chloride concentrations
- b) What is the minimum and maximum concentration of iron(III) chloride which still allows the reaction to take place within the bounds of the reactor? Vary the ratio of A1/A2.
- c) Plot the concentration over the reaction rate. Which reaction order can you assume for iron(III) chloride (x), potassium iodide (y) and sodium thiosulfate? Plot your results logarithmically and show the equation for determining the reaction order. Reference graphs for potassium iodide and sodium thiosulfate are given in Figure 1, 2 and 3.

$$v_0 = k[\text{Fe}^{3+}]_0^x [\text{I}^-]_0^y$$

**A: Continuous Flow Reactor**

$V_{A1}$ [mL]	$V_{A2}$ [ml]	$C_{FeCl_2}$ [mmol/L]	Flow rate*[ml/min]	Length [cm]	Time [s]

\*Total flow rate = sum of flow rates of both pumps

**B: Bubble Train Reactor**

$V_{A1}$ [mL]	$V_{A2}$ [ml]	$C_{FeCl_2}$ [mmol/L]	Flow rate*[ml/min]	Length [cm]	Time [s]

\*Total flow rate = sum of flow rates of all three pumps

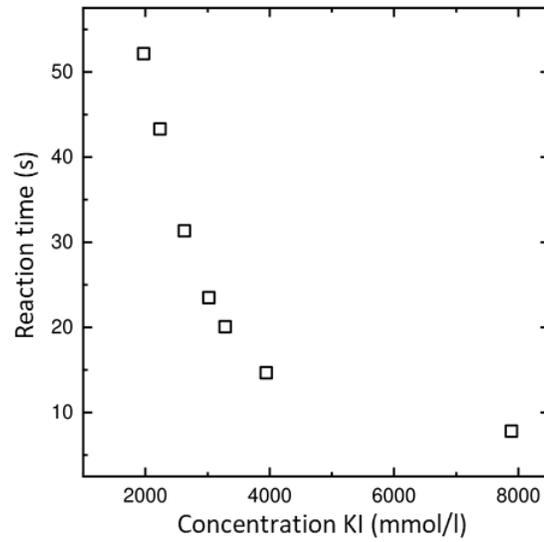


Figure 1: Reaction time over concentration of potassium iodide

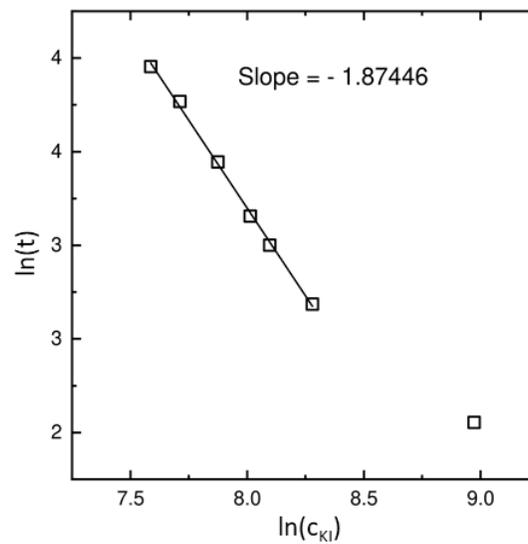


Figure 2: Logarithmic plot of reaction time over potassium iodide concentration

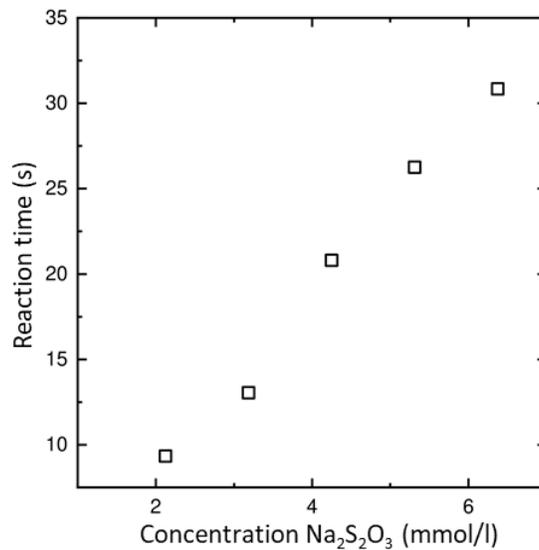


Figure 3: Reaction time over sodium thiosulfate concentration

**Physical constants:**

- Molar mass sodium thiosulfate:  $M_{\text{Na}_2\text{S}_2\text{O}_3} = 158,11 \text{ g/mol}$
- Molar mass potassium iodide:  $M_{\text{KI}} = 166,0028 \text{ g/mol}$
- Molar mass iron(III) chloride:  $M_{\text{FeCl}_3} = 162,2 \text{ g/mol}$
- Volume static mixer = 0,9455 ml
- Distance black marks 10 cm, red marks 1 m
- Tube diameter of reactor = 2,5 mm

**Chemical equations:**



# Applied chemical process engineering - Python

## Import of Packages

```
In [1]: !pip install scipy
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint as odeint
from scipy.optimize import fmin
import math
from IPython.display import display, Math, Latex
```

Collecting scipy

Using cached scipy-1.7.2-cp39-cp39-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (39.8 MB)

Requirement already satisfied: numpy<1.23.0,>=1.16.5 in /opt/conda/lib/python3.9/site-packages (from scipy) (1.21.2)

Installing collected packages: scipy

Successfully installed scipy-1.7.2

## Experimental Values:

$c_0$  := Starting Values of concentrations in mol/L  
 $t_{exp}$  := Time measured in the experiment

```
In [2]: c0 = [[14e-3 * i, 2.25e-3, 0, 1.67e-3] for i in [1, 1.5, 2, 3]]
t_exp = [33, 21, 15, 10]
```

## Chemical Reactions

- $\text{Fe}^{3+} + \text{S}_2\text{O}_3^{2-} \rightleftharpoons [\text{Fe}(\text{S}_2\text{O}_3)]^+$
- $2\text{Fe}^{3+} + 3\text{I}^- \longrightarrow 2\text{Fe}^{2+} + \text{I}_3^-$
- $\text{I}_3^- + 2\text{S}_2\text{O}_3^{2-} \longrightarrow 3\text{I}^- + \text{S}_4\text{O}_6^{2-}$
- $2\text{I}_3^- + \text{starch} \longrightarrow \text{starch-I}_5^- + \text{I}^-$

Note: Not all reactions are relevant here! The first reaction is reversible and very fast, therefore it can be neglected. The last reaction is the formation of the blue complex. This happens instantaneously when  $\text{I}_3^-$  is present in the solution. The complex-forming reaction can be neglected, when  $\text{I}_3^-$  is used as indicator for the color formation.

## Indices:

$c_0[0] = \text{Fe}^{3+}$

$c_0[1] = \text{I}^-$

$c_0[2] = \text{I}_3^-$

$c_0[3] = \text{S}_2\text{O}_3^{2+}$

## Reaction Rates:

Define a function that calculates the reaction rates. The reaction of Iron and Iodine is first and second order with respect to iron(III) ions and Iodine ions, respectively. Assume all remaining reactions to be elementary.

$$R_i = k_i(T) \prod_j c_j^{n_{ij}}$$

Calculate each component's concentration gradient for a given reaction rate. Take into account that there are multiple reactions.

$$\frac{dc_i}{dt} = \sum_j \nu_{i,j} R_j$$

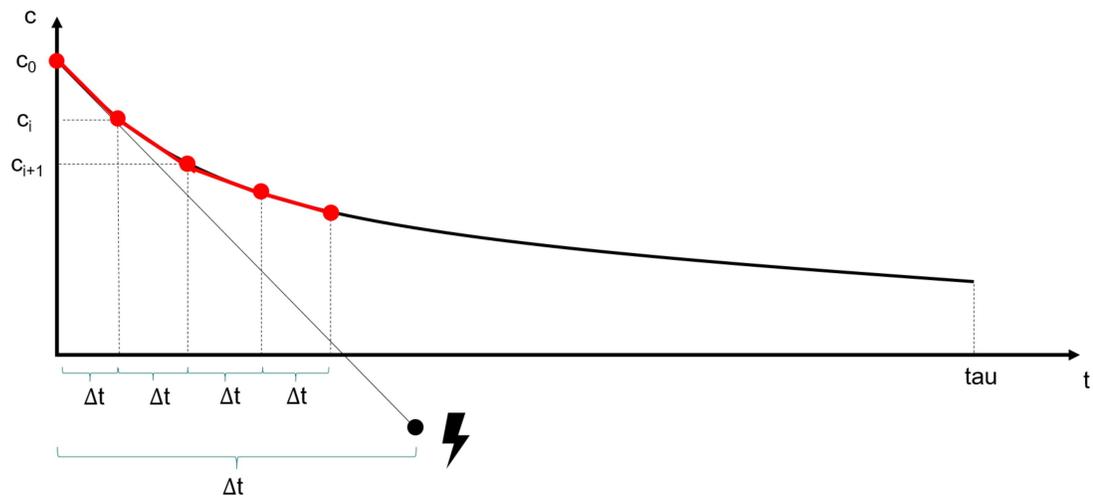
In [3]:

```
def reaction_rates(c, t, k):
    r = [
        k[0] * c[0] * c[1]**2,
        k[1] * c[2] * c[3]**2
    ]
    dc = [
        -2 * r[0],
        -3 * r[0] + 3 * r[1],
        r[0] - r[1],
        -2 * r[1]
    ]
    return(dc)
```

### Simulated Batch Experiment:

Simulate a batch experiment using the Explicit-Euler approach to solve the differential equation.

$$c_{i+1} = c_i + \frac{dc_i}{dt} \Delta t$$



In [4]:

```
def batch_experiment(c, k, t_max, t_step):
    steps = int(t_max / t_step)
    cs = np.zeros((steps+1, len(c)))
    cs[0] = np.float64(c)
    for i in range(steps):
        for j in range(len(c)):
            cs[i+1][j] = cs[i][j] + reaction_rates(cs[i], 1, k)[j] * t_step
            if cs[i+1][j] < 0:
                cs[i+1][j] = 0
    return(cs)
```

## Parameters for the Simulation

You need some parameters to simulate the batch experiment. Try out different values. You might want to plot the concentrations of your simulated experiment.

```
k      := Vector of reaction constants (starting values: [300, 6e8])
t_max := Maximum time of the simulated reaction
t_step:= Time Step used for the simulation
```

```
In [5]: k = [300, 6e8]
t_max = 80
steps = 50000
t_step = 80 / steps
c_lim = 2e-05
```

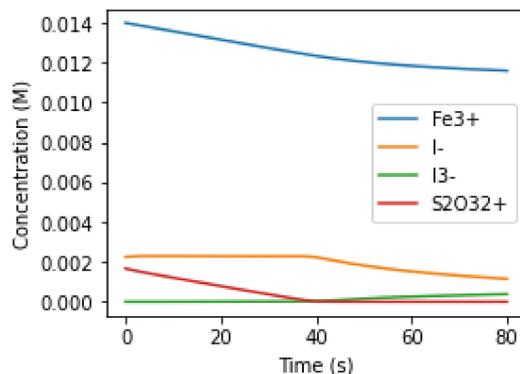
This function uses `odeint` to solve the differential equation. This is much faster than the self-implemented explicit euler, saving a lot of computational time during the fitting.

```
In [6]: def ode_experiment(cs, k, t_max, t_step):
steps = int(t_max/t_step)
t = np.linspace(0, t_max, steps)
c = odeint(reaction_rates, cs, t, args=((k,)))
return(c)
```

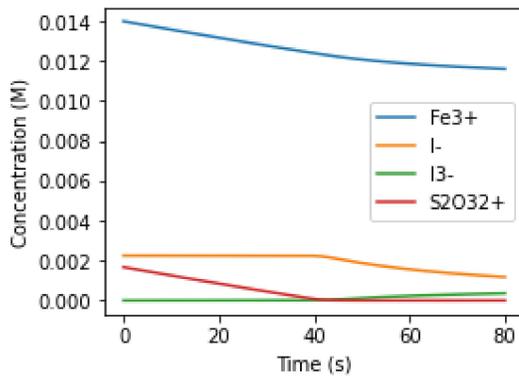
Write a function to plot the concentration profiles. Compare the results of the self-implemented explicit euler and the `odeint` function. Investigate the influence of the step-size on the simulation results.

```
In [7]: def plot(c):
t = np.linspace(0, t_max, len(c))
plt.figure(figsize=(4, 3))
plt.plot(t, c[:])
plt.xlabel('Time (s)')
plt.ylabel('Concentration (M)')
plt.legend(['Fe3+', 'I-', 'I3-', 'S2O32+'])
```

```
In [8]: plot(batch_experiment(c0[0], k, t_max, t_step))
```



```
In [9]: plot(ode_experiment(c0[0], k, t_max, t_step))
```



### Simulated Time:

Determine the time of the color change. The necessary concentration for the color change is not known, assume  $c_{I_3,lim} = 2e - 5$ .

```
In [10]: def simulate_time(c, k, t_max, t_step):
cs = ode_experiment(c, k, t_max, t_step)
for i, c in enumerate(cs):
    if c[2] > c_lim:
        break
sim_time = t_step * i
return(sim_time)
```

### Error Function:

Define the error function.

The error function sums up the squares of the difference between the simulated and experimental time for all experiments.

```
In [11]: def error_func(k, c0, times):
sim_times = []
for c in c0:
    sim_times.append(simulate_time(c, k, t_max, t_step))
deltas = [np.sqrt((sim_time - time)**2) for sim_time, time in zip(sim_times, times)]
# print('Error: {}'.format(sum(i**2 for i in deltas)))
return(sum(i**2 for i in deltas))
```

### Optimize:

Use the function `scipy.optimize.fmin` to fit the reaction constants `k` to minimize the time difference between experiment and simulation.

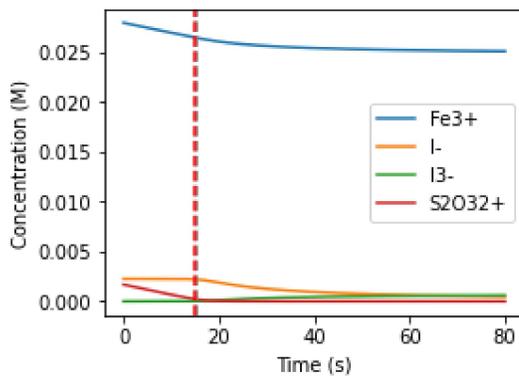
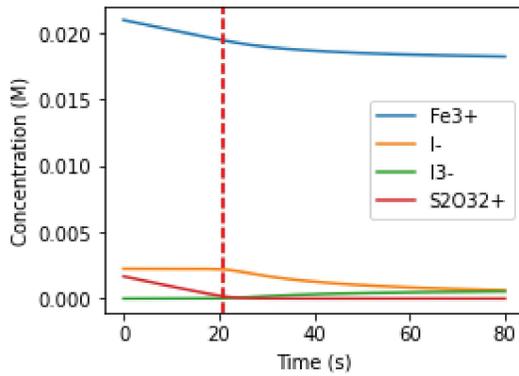
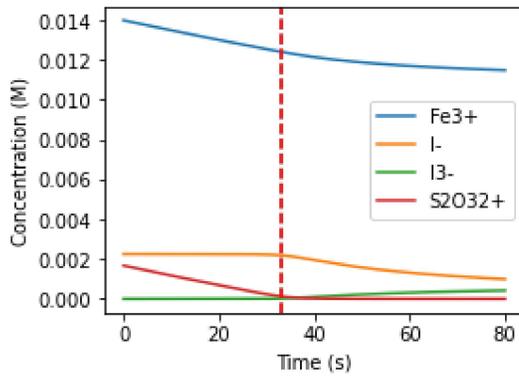
```
In [12]: kr = fmin(error_func, k, args=(c0, t_exp), xtol=0.0001, ftol=0.0001, disp=1)
print(kr)
```

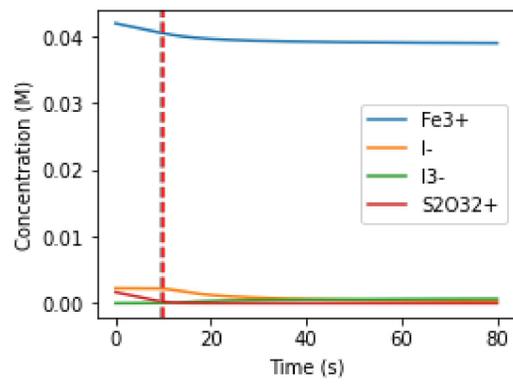
```
Optimization terminated successfully.
Current function value: 0.146266
Iterations: 92
Function evaluations: 243
[3.61126846e+02 4.41495203e+07]
```

### Plotting:

Plot the simulated concentrations. Mark the simulated and experimentally measured time of the color change.

```
In [13]: for c0i, time in zip(c0, t_exp):
          c = batch_experiment(c0i, kr, t_max, t_step)
          t = np.linspace(0, t_max, steps)
          plt.figure(figsize=(4, 3))
          plt.plot(t, c[1:])
          plt.xlabel('Time (s)')
          plt.ylabel('Concentration (M)')
          plt.legend(['Fe3+', 'I-', 'I3-', 'S2O32+'])
          plt.axvline(x=simulate_time(c0i, kr, t_max, t_step), color='gray', linestyle='--')
          if time:
              plt.axvline(x=time, color='r', linestyle='--')
```





In [ ]: