June 5, 2023

Controlled Release Design and Tartrazine Diffusivity Analysis

Lucas Culverhouse
ECH 145B | Dr. Jiandi Wan
May 10, 2023
Section A02
Lienna Chan | Valdemar Roman | Irving Rosales Gonzalez

Abstract

In this report we investigate the effects of cross-linking and bead diameter on release kinetics for alignate particles. We discuss a manufacturing methodology for varying the desgin of an alginate release system. We conclude ACME should consider producing alginate beads with a diameter of 2.632 mm in order to achieve their design targets.

Table of Contents

Introduction

Polymer devices are increasingly important for biomedical applications in drug delivery. Polymers present many benefits such as biocompatability, ease of production and controllable release characteristics. These devices allow for less invasive drug delivery systems that can be tightly contolled based on the drug and dose needed. This presents uses for drugs where toxicity and effectiveness need to be tightly controlled.

To understand the design requirements for these polymer devices, first we have to understand the mass transport principles behind the diffusion from polymers. Diffusion models from polymers are the subject of considerable research and and modeling. Ritger and Peppas ,1987 [1] establish theoretical models for the diffusion mechanisms from such polymer systems. These rely on the classical models of Fickain diffusion, device geometry, and diffusive properties. The usefulness of alginate as a biomedial polymer for drug delivery is expolored in a review by Lee and Mooney [2], where its favorable biomedical and manufacturing properties are explored. Alginate is cheap, being derived from brown seaweed, making it ideal for use in high volume drug production [2]. To understand release kinetics, tartrazine is chosen as a common, non-toxic food dye that has similar release kinetics to the desired ACME compound [3, 4].

This report outlines a experimental process and parameters to meet the design goals of ACME's drug release system, that is a one hour sustained release at 0.010 ± 0.002 min^{-1}. Here we propose an ideal spherical alginate bead radius for meeting this design goal, along with several suggestions for general polymer based relase design.

Theory

Theoretical Fickain Diffusion. To understand the reaction release rates from a polymer we can use the following exponential relation, regardless of the limiting mechanism and device geometry [1]:

$$\frac{M_t}{M_\infty} = F = kt^n \tag{1}$$

Where M_t is the mass released at a specific time, and M_∞ is the mass released after infinite time; this means F is the mass fraction of the relevant species released. The exponent n describes the rate controlling mechanism for the device and k is a constant that indicated the characteristic diffusive length and diffusion coefficient [3]. Equation 1 can be differentiated to get the rate of release for the polymer device:

$$\frac{\mathrm{d}F}{\mathrm{d}t} = knt^{n-1} \tag{2}$$

We can find k for a spherical geometry using the following equation:

$$k = 6\sqrt{\frac{D}{\pi r^2}} \tag{3}$$

We can use the following table to provide theoretical estimations of the release rates:

Control Mechanism	Slab (n)	Sphere (n)
Fickian Diffusion	0.5	0.43
Polymer Relaxation	1.0	0.85
Both Contribute	0.5 < n < 1.0	0.43 < n < 0.85

Table 1: Summary of n-values for different geometries and rate control mechanisms [3]

Experimental Fickain Diffusion. In order to get an experimental measure of the release rate of designed alginate beads we can derive the following from Equation 1:

$$\ln\left(\frac{M_t}{M_\infty}\right) = n\ln(t) + \ln(k) \tag{4}$$

This allows us to get the rate of release from our experimental data by using a linear fit on a log-log plot. As we cannot easialy directly measure the mass released from our beads, we must use the definition of concentration to find the mass released from the beads:

$$M_t = VC \tag{5}$$

To get the concentration we will use a spectrometer to correlate the absorbance of the solution collected at a given time with a concentration using the following relation:

$$C \propto \frac{\text{Abs}}{\text{const.}} \tag{6}$$

Where C is the concentration and we relate it with a constant obtained as the slope from plotting known concentrations versus maximum measured absorbance. These various known concentrations are obtained by a serial dilution of a solution of known concentration.

Experimental Methods

Preparation of Alginate Solution. Tartazine powder was dissolved in DI water to make 10 mL of tartrazine solution, and 0.1 g of alginate power was then added. This solution was mixed using a magnetic stir bar for appoximately 45 minutes until all of the powder was dissolved. This solution was then stored for later use in a sealed container covered with aluminum foil in a dark dry area.

Preparation of Tartrazine Alginate Beads. A beaker of around 50 mL of calcium chloride was prepared. Then the previously prepared alginate solution was loaded into a syringe with the desired tip size, making note of the total amount of solutiion to be dispensed. Then over approximately 20 seconds multiple alginate drops were dispensed into the calcium chloride solution. These were then left in the solution for 80 seconds, until they were transfered to a buchner funnel to be dryed by vaccum filtration.

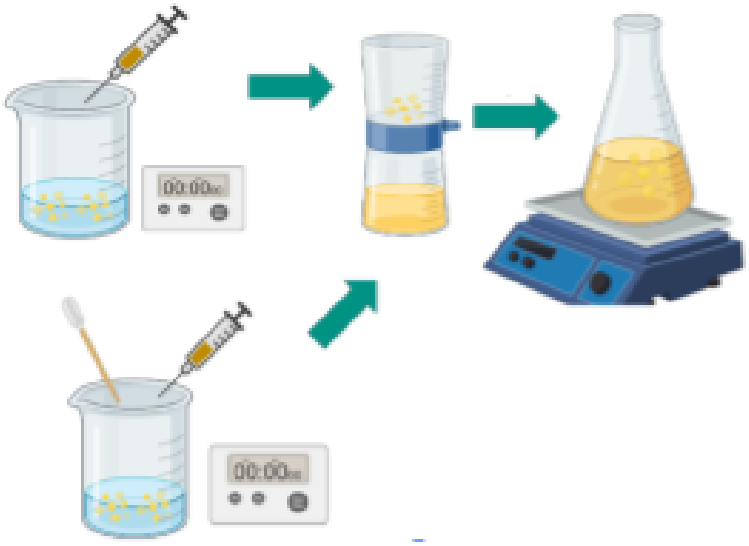


Figure 1: Experimental Set up for Alginate Bead Preparation

Beads were then quickly collected using a spatula and measured with calipers; after which they were transferred to 100 mL of DI water containting a magnetic stir bar at 200 rpm.

Spectroscopic Measurment of Tartrazine Concentrations. First, an *Ocean FX* spectrometer was initially calibrated using a sample of DI water to account for the absorbance of the water solvent in the tartrazine solution; additionally the background or 'dark' interference was calibrated for.

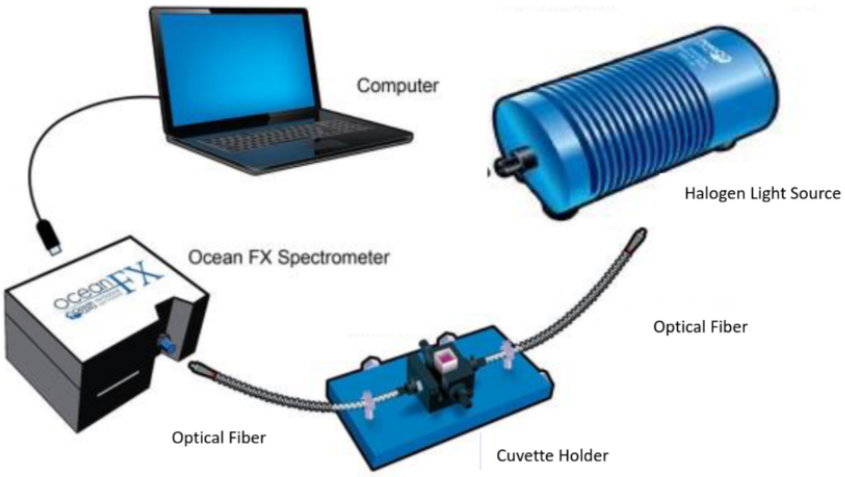


Figure 2: Experimental Set up for Spectroscopic Measurment [3]

Once the calibration was completed and the alginate beads were prepared, a beaker containing 100 mL of DI water was placed on a hot plate with magnetic stir bar, with no heating and stirring set to 200 rpm. Beads were then placed into the beaker and timing was started. Approximatley every 5 minutes, a cuvette was rinsed with the solution and then filled. This was then placed into the spectrometer to measure the maximum absorbance. Quickly after this measurment was completed, and the data recorded, the solution in the cuvette was emptied back into the container to keep the volume of liquid constant.

After approximately 45 minutes of continued measurements, the experiment was stopped. This process was completed multiple times for different bead parameters.

Results

Spectrometer Calibration. The specrometer calibration produced the following relationship between absorbance and concentration, which was used for all later experimental conversions between measured absorbance and concentration.

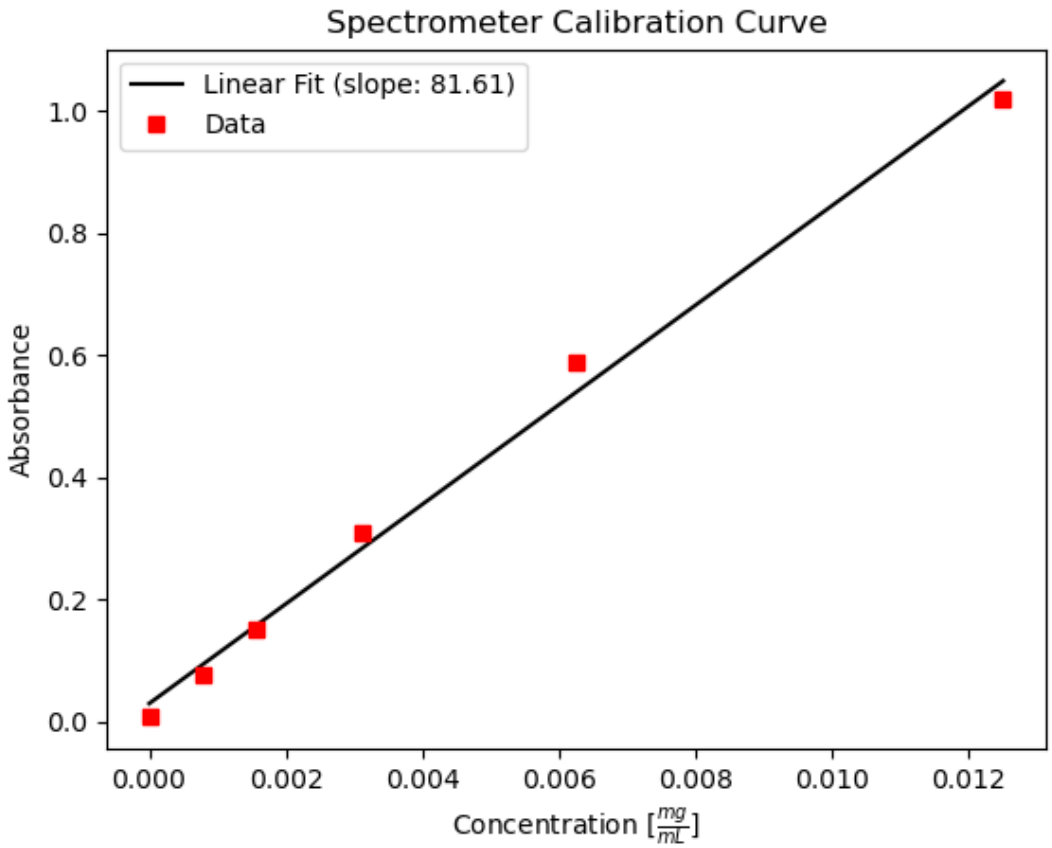


Figure 3: Proportionallity between maximum measured absorbance and tartrazine concentration. Serial dilution from a stock solution of $0.5\ \frac{\text{mg}}{\text{mL}}$ was performed to obatian all concentrations.

Tartrazine Release Rate. Below is a table showing the experimental and theoretical k and n values for the selected trial.

Method	k	n
Theotetical	0.0569	0.43
Experimental	$4.36 \cdot 10^{-5}$	0.596

Table 2: Calculated k and n values from experimental and Theoretical Methods

We can see the mass fraction $\frac{M_t}{M_\infty}$ released over time for several trials below.

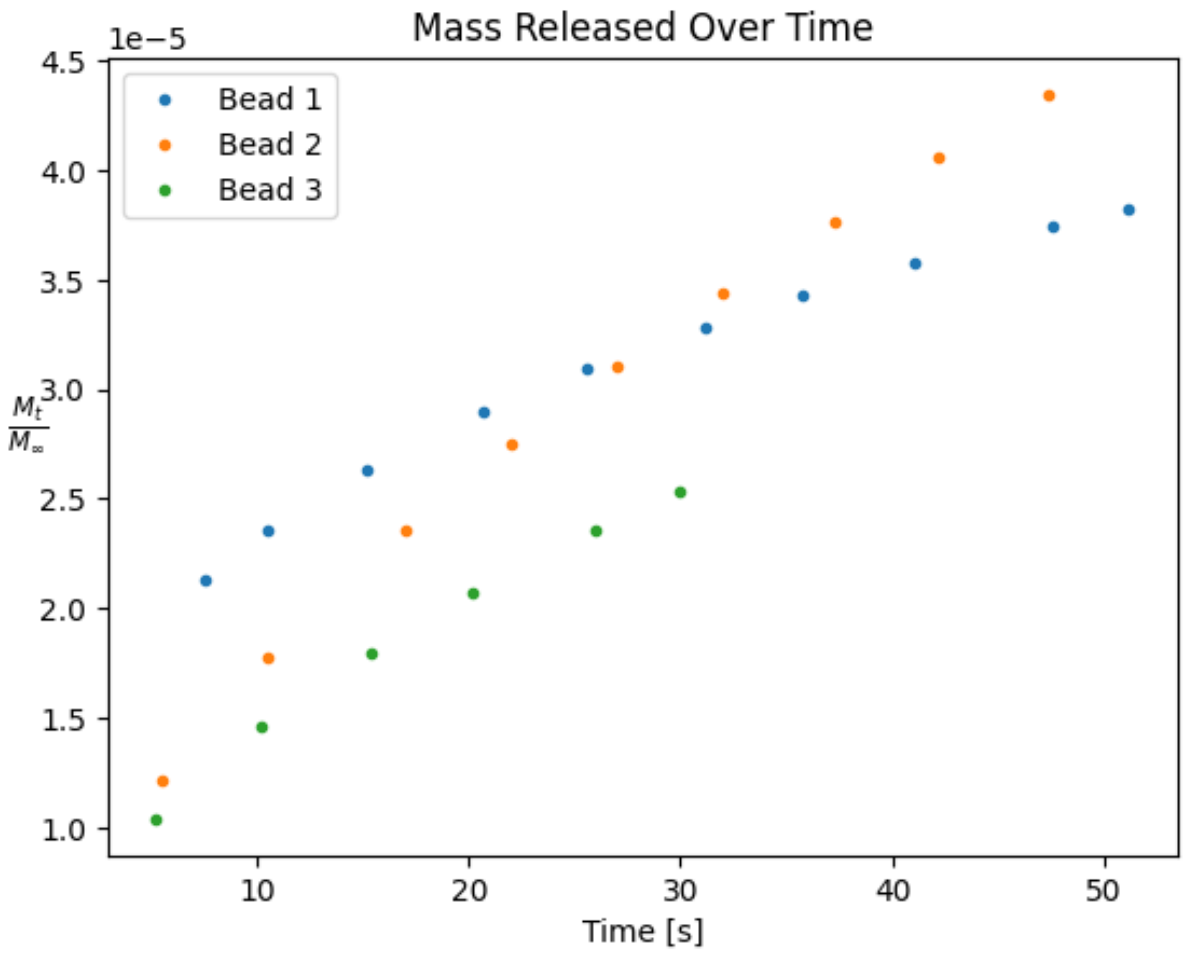


Figure 4: The mass fraction released by the alginate beads over time. Several different bead diameters were tested to derive the best overall parameters for bead production

We can also see the fitting for the experimental parameters found for k and n

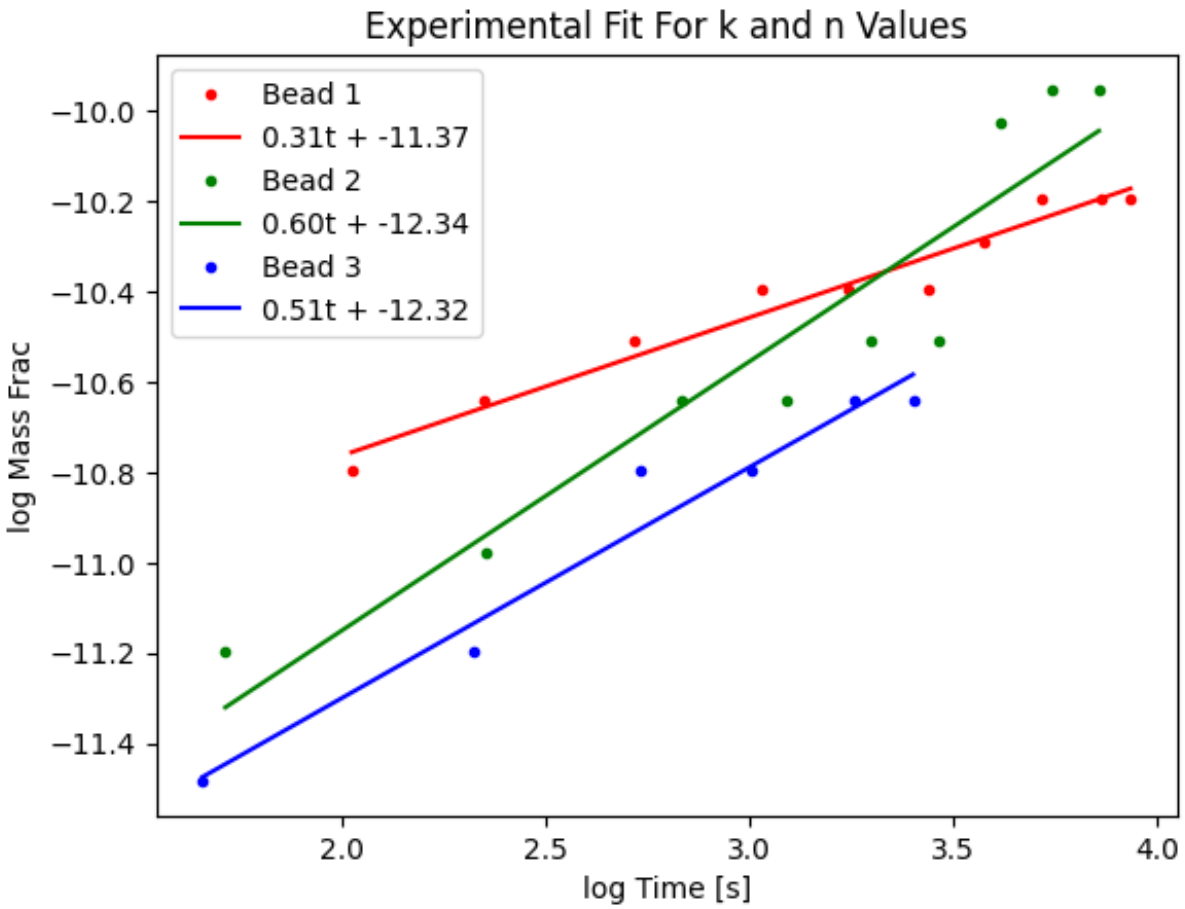


Figure 5: log-log fit for the parameters of various beads tested

The final selected trial for the design challenge is show below, generated using the theroetical k and n values derived and show above in Table 2:

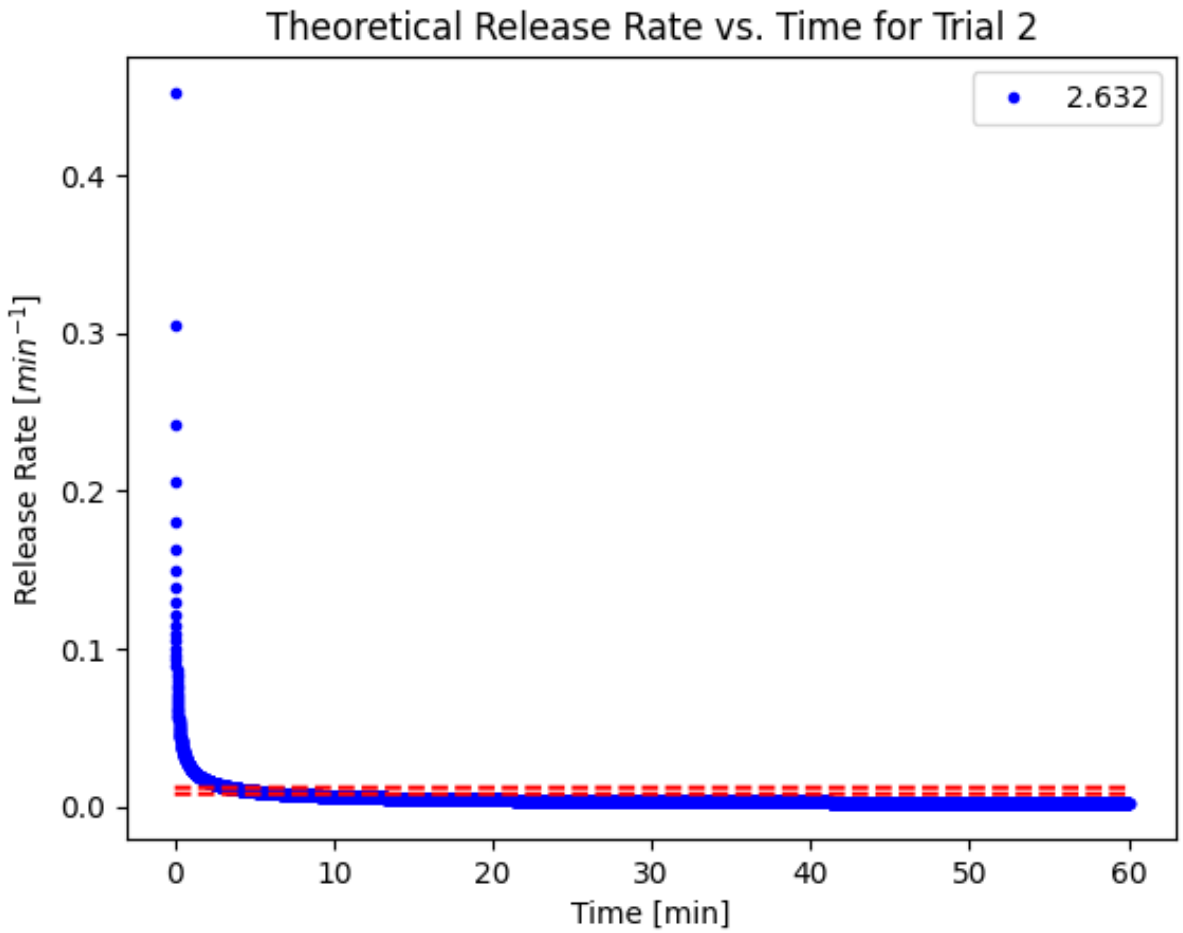


Figure 6: Release rate over 1 hour from theoretically derived k and n values.

Discussion

We can see that there is a wide variation in the obtained values for the experimentally fitted data and for the theoretically obtained values. As the theory is quite well established [1] there seems to be a source of error polluting the data. This can be seen when insepecting the raw data as a lack of precsion in the collected absorbance values. The software provided for the *Ocean FX* spectrometer exports data with some preset level of precision. This value is not immediatley apparent when looking at the view it provides, but only after the fact when analyzing the data. This leads to large, stair-stepping effects in the data and makes it unreliable for analysis.

We can see, however that with a bead diameter of 2.632 mm, assuming the release rate is controlled by Fickian diffusion that our long term release rate gets close to the ACME target over 1 hour.

The trend in the release rates is also odd, as from the theory we would expect the release rate to decrease as beads got bigger, whereas we obsever the opposite trend. This may be beacuase of two factors:

1. Imprecision in the measurement of radius. Calipers produce some systematic error from the non rigid beads
2. Not exactness of the cross-linking time effects

Conclusions

More testing would have to be done to analyze the optimal parameters for ACME's drug release system. Understanding the magnitude of the effects of both cross-linking and bead size would take more trails to be certain of the overall effects. However we can conclude that the experimental data must be well collected to obtain useful results, as the collection methods are very sensistive to this parameter.

Alginate beads are relativly easy to produce and can be synthesiszed rapidly, ensuring an easy to manufacture final product. These seem like a great candidate for the final version of this ACME product.

From this report we would conclude a bead diameter of 2.632 mm is desireable for ACME to meet its design target.

Nomenclature

C: Concentration $\left[\frac{\text{mg}}{\text{mL}}\right]$

n: Exponential constant in release rate equation

k: Diffusion constant in release rate equation

M_t: Mass released at specific time

M_∞: Mass released at infinite time

F, $\frac{M_t}{M_\infty}$: Mass fraction of species released

Bibliography

[1] P. L. Ritger, and N. A. Peppas, “A simple equation for description of solute release i. fickian and non-fickian release from non-swellable devices in the form of slabs, spheres, cylinders or discs,” *J. Controlled Release*, vol. 5, no. 1, p. 23, 1987.

[2] K. Y. Lee, and D. J. Mooney, “Alginate: properties and biomedical applications,” *Prog. Polym. Sci.*, vol. 37, no. 1, p. 106, Jan. 2012.

[3] J. Wan, “Lab 3 - design an alginate-based controlled drug release system,” 2023.

[4] U.S. National Library of Medicine. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/compound/Tartrazine

Appendix

In [46]:

```
from typing import NamedTuple

import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize as so
import pandas as pd

# Loading and storing data values

# Spectronometer calibration curve calculations
Calibration = NamedTuple("Calibration", slope=float, intercept=float)
# Converts from absorbance to concentration in mg / mL
calibration = Calibration(slope=81.607231705, intercept=0.029639142857)

# Bead diameters
bead_diameter = (  # m
    4.493 * 0.001,
    2.632 * 0.001,
    3.505 * 0.001,
)

TARTRAZINE_MOLAR_MASS = 0.5343  # kg / mol
TARTRAZINE_DENSITY = 1930.0  # kg / m^3
TARTRAZINE_VOLUME = 1.0 * 1e-6 # m^3
TARTRAZINE_DIFFUSIVITY = 4.9e-10  # m^2 / s
SOLUTION_VOLUME = 100 * 1e-6  # m^3

# Absorption at 427nm wavelength
sample_times = ( # s
    np.array(
        [
            454,
            628,
            909,
            1244,
            1534,
            1868,
            2145,
            2465,
            2854,
            3070,
        ]
    ),
    np.array(
        [
            333,
            630,
            1020,
            1320,
            1620,
```

```
            1920,
            2235,
            2533,
            2841,
        ]
    ),
    np.array(
        [
            315,
            612,
            923,
            1212,
            1560,
            1800,
        ]
    ),
)

spectro_trials = (
    pd.read_excel("./Data/Lab3/Final/Bead1.xlsx", sheet_name=None),  # Bead 1
    pd.read_excel("./Data/Lab3/Final/Bead2.xlsx", sheet_name=None),  # Bead 2
    pd.read_excel("./Data/Lab3/Final/Bead3.xlsx", sheet_name=None),  # Bead 3
)

concentrations = (  # kg / m^3
    [],
    [],
    [],
)
for trial, data in enumerate(spectro_trials):
    for i, sample in enumerate(data.values()):
        # Get the maximum value of the abs(orbance)
        max_abs = sample["abs"].max()
        conc = (1 / calibration.slope) * (max_abs - calibration.intercept)
        concentrations[trial].append(conc)
concentrations = (
    np.array(concentrations[0]),
    np.array(concentrations[1]),
    np.array(concentrations[2]),
)

print(concentrations)
```

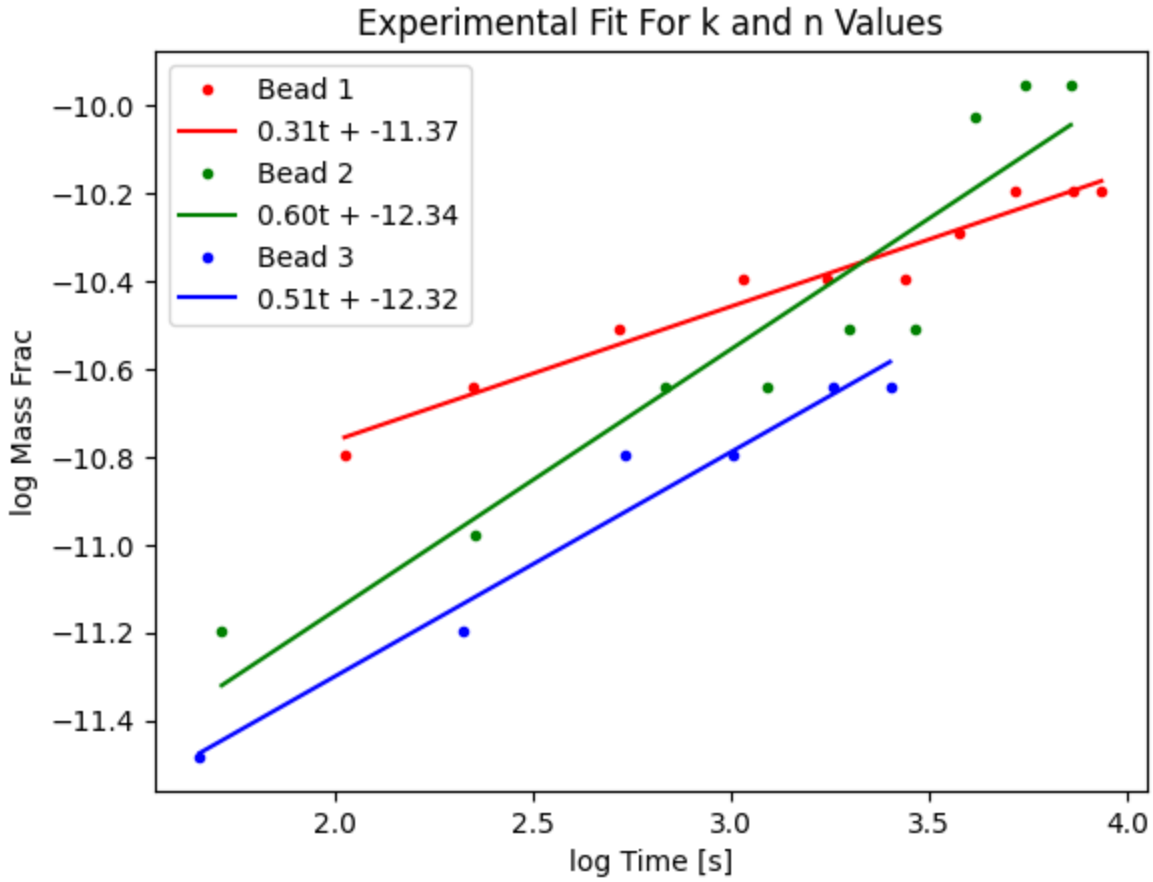
```
(array([0.00073965, 0.00086219, 0.00098473, 0.00110727, 0.00110727,
       0.00110727, 0.0012298 , 0.00135234, 0.00135234, 0.00135234]), array([0.0004
9457, 0.00061711, 0.00086219, 0.00086219, 0.00098473,
       0.00098473, 0.00159742, 0.00171996, 0.00171996]), array([0.00037204, 0.0004
9457, 0.00073965, 0.00073965, 0.00086219,
       0.00086219]))
```

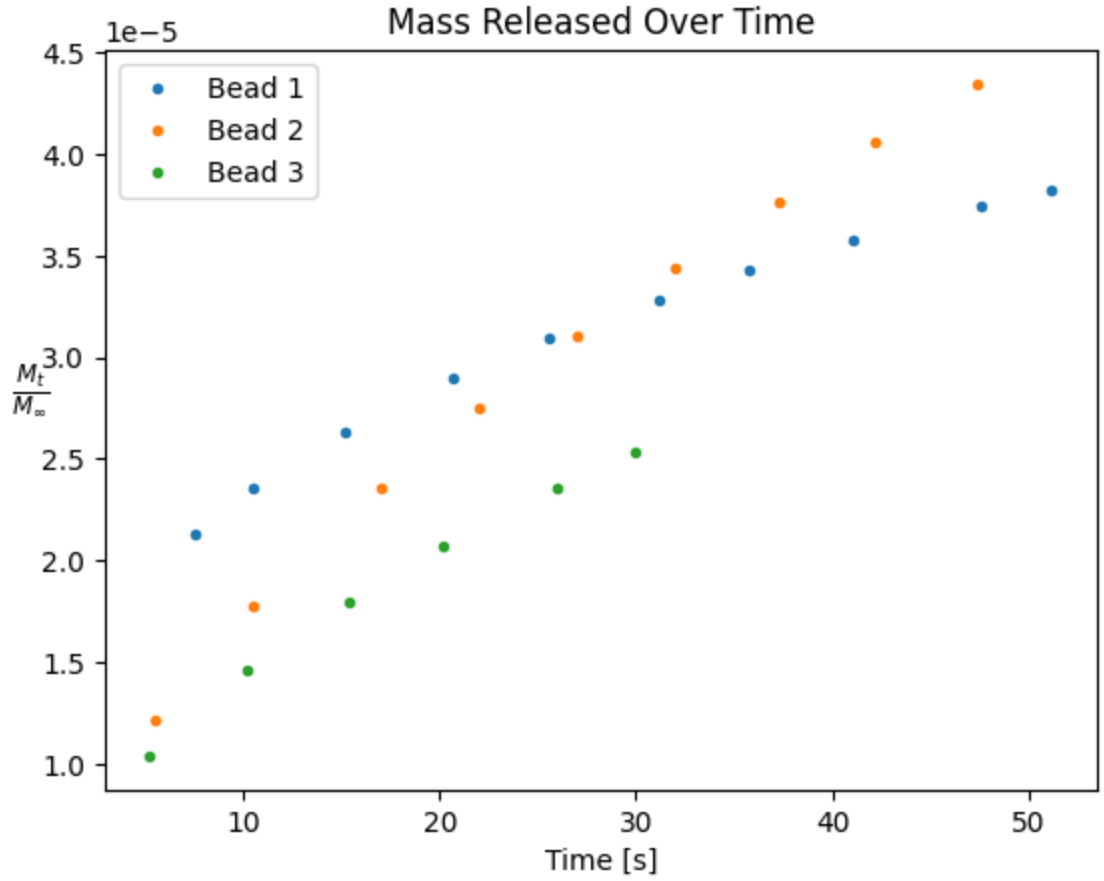
In [47]:

```
time_masses = [
    TARTRAZINE_MOLAR_MASS * SOLUTION_VOLUME * conc for conc in concentrations
]

inf_mass = TARTRAZINE_DENSITY * TARTRAZINE_VOLUME
```

```
n_vals = np.zeros(3)
k_vals = np.zeros(3)
F = [mt/inf_mass for mt in time_masses]
def Fcalc(k, n , time):
    return k * (time ** n)

colors = ['r', 'g', 'b']
for i in range(3):
    time_mins = sample_times[i] * (1 / 60)
    log_time = np.log(time_mins)
    log_mass = np.log(time_masses[i] / inf_mass)
    slope, inter = np.polyfit(log_time, log_mass, 1)
    n_vals[i] = slope
    print(slope)
    print(np.exp(inter))
    k_vals[i] = np.exp(inter)
    plt.plot(log_time, log_mass, colors[i]+".", label=f"Bead {i+1}")
    plt.plot(log_time, slope * log_time + inter, colors[i]+"-" ,label=f"{slope:.2f}
plt.title("Experimental Fit For k and n Values")
plt.xlabel("log Time [s]")
plt.ylabel("log Mass Frac")
plt.legend()
plt.show()

for i in range(3):
    time_mins = sample_times[i] * (1 / 60)
    plt.plot(time_mins, Fcalc(k_vals[i], n_vals[i], sample_times[i] * (1/60)) , "."


plt.xlabel("Time [s]")
plt.ylabel(r"$\frac{M_t}{M_\infty}$", rotation=0, fontdict={"size": 12})
plt.title("Mass Released Over Time")
plt.legend()
plt.show()
print(F)
print([np.gradient(F[i]) for i in range(3)])
dFdt = [np.gradient(F[i])/np.gradient(sample_times[i]) for i in range(3)]
print(dFdt)
```

```
0.30562000234461834
1.1490156454185313e-05
0.5957820504143857
4.36526212591157e-06
0.5113859115589585
4.449425971186136e-06
```


Experimental Fit For k and n Values
Bead 1
0.31t + -11.37
Bead 2
0.60t + -12.34
Bead 3
0.51t + -12.32
log Mass Frac
log Time [s]
−10.0
−10.2
−10.4
−10.6
−10.8
−11.0
−11.2
−11.4
2.0
2.5
3.0
3.5
4.0


Mass Released Over Time
1e−5
Bead 1
Bead 2
Bead 3
$\frac{M_t}{M_\infty}$
Time [s]
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
10
20
30
40
50

```
[array([2.04764478e-05, 2.38687865e-05, 2.72611253e-05, 3.06534641e-05,
       3.06534641e-05, 3.06534641e-05, 3.40458029e-05, 3.74381417e-05,
       3.74381417e-05, 3.74381417e-05]), array([1.36917702e-05, 1.70841090e-05, 2.38687865e-05, 2.38687865e-05,
       2.72611253e-05, 2.72611253e-05, 4.42228193e-05, 4.76151581e-05,
       4.76151581e-05]), array([1.02994314e-05, 1.36917702e-05, 2.04764478e-05, 2.04764478e-05,
       2.38687865e-05, 2.38687865e-05])]
[array([3.3923388e-06, 3.3923388e-06, 3.3923388e-06, 1.6961694e-06,
       0.0000000e+00, 1.6961694e-06, 3.3923388e-06, 1.6961694e-06,
       0.0000000e+00, 0.0000000e+00]), array([3.39233880e-06, 5.08850820e-06, 3.39233880e-06, 1.69616940e-06,
       1.69616940e-06, 8.48084699e-06, 1.01770164e-05, 1.69616940e-06,
       0.00000000e+00]), array([3.3923388e-06, 5.0885082e-06, 3.3923388e-06, 1.6961694e-06,
       1.6961694e-06, 0.0000000e+00])]
[array([1.94962000e-08, 1.49113793e-08, 1.10140870e-08, 5.42774208e-09,
       0.00000000e+00, 5.55210932e-09, 1.13646191e-08, 4.78468096e-09,
       0.00000000e+00, 0.00000000e+00]), array([1.14220162e-08, 1.48137065e-08, 9.83286608e-09, 5.65389800e-09,
       5.65389800e-09, 2.75799902e-08, 3.32039687e-08, 5.59791881e-09,
       0.00000000e+00]), array([1.14220162e-08, 1.67385138e-08, 1.13077960e-08, 5.32549262e-09,
       5.76928367e-09, 0.00000000e+00])]
```

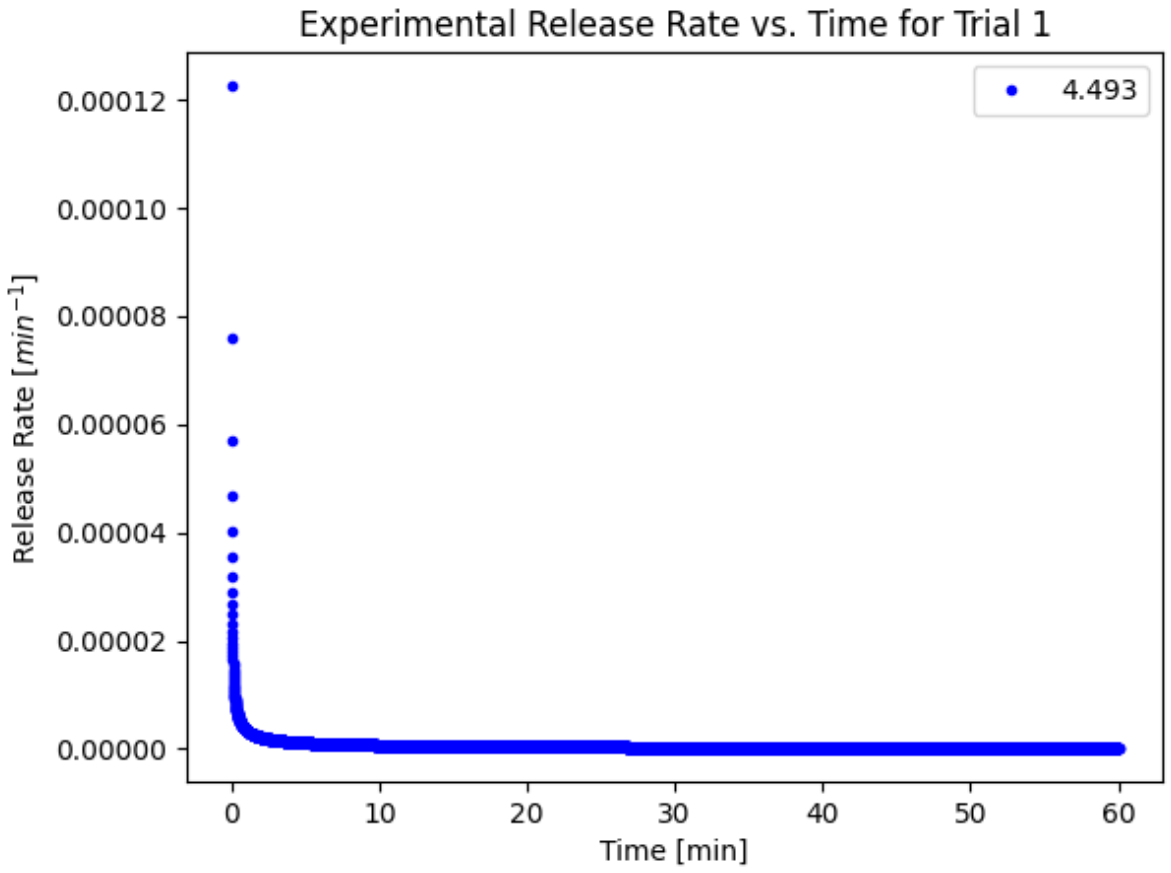
In [48]:

```python
def deriv_F(k: float, n: float, time: np.ndarray) -> np.ndarray:
    return k * n * (time ** (n - 1))

def F(k: float, n: float, time: np.ndarray) -> np.ndarray:
    return

n_theory = {"fick": 0.43, "poly": 0.85}

def k_theory(diff: float, rad: float) -> float:
    return 6 * np.sqrt(diff / (np.pi * rad ** 2))

times = np.linspace(0, 60, 10_000)
for i in range(3):
    dFdt = deriv_F(k_vals[i], n_vals[i], times)
    print(k_vals[i], "\n",n_vals[i])
    plt.plot(times, dFdt, "b.", label=f"{bead_diameter[i] * 1000}")
    plt.xlabel("Time [min]")
    plt.ylabel("Release Rate $[min^{-1}]$")
    plt.title(f"Experimental Release Rate vs. Time for Trial {i+1}")
    plt.legend()
    plt.show()

for i in range(3):
    n = n_theory["fick"]
    k = k_theory(TARTRAZINE_DIFFUSIVITY, bead_diameter[i] / 2)
    print(k,"\n",n)
    dFdt = deriv_F(k, n, times)
    plt.plot(times, dFdt, "b.", label=f"{bead_diameter[i] * 1000}")
    plt.plot(times, times*0 + 0.010-0.002, "r--")
    plt.plot(times, times*0 + 0.010+0.002, "r--")
    plt.xlabel("Time [min]")
```

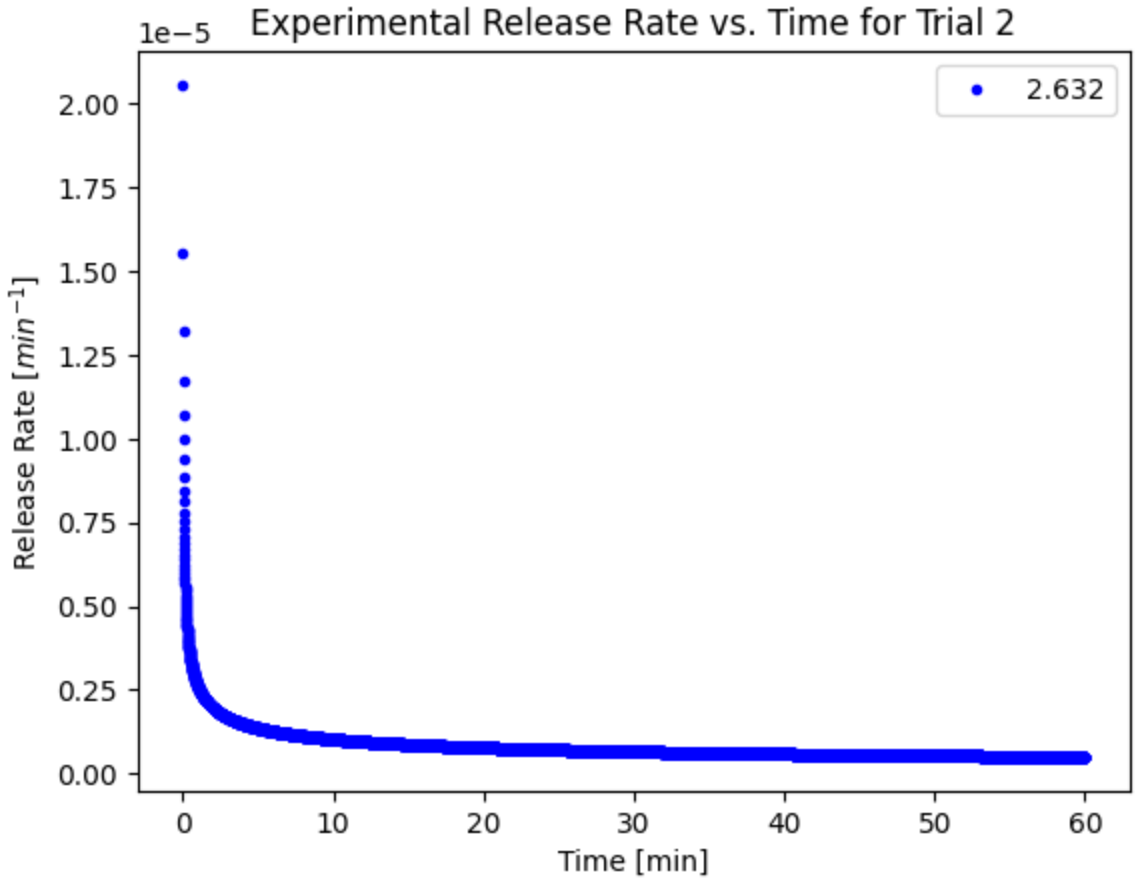
```
    plt.ylabel("Release Rate $[min^{-1}]$")
    plt.title(f"Theoretical Release Rate vs. Time for Trial {i+1}")
    plt.legend()
    plt.show()
```

```
1.1490156454185313e-05
 0.30562000234461834
```

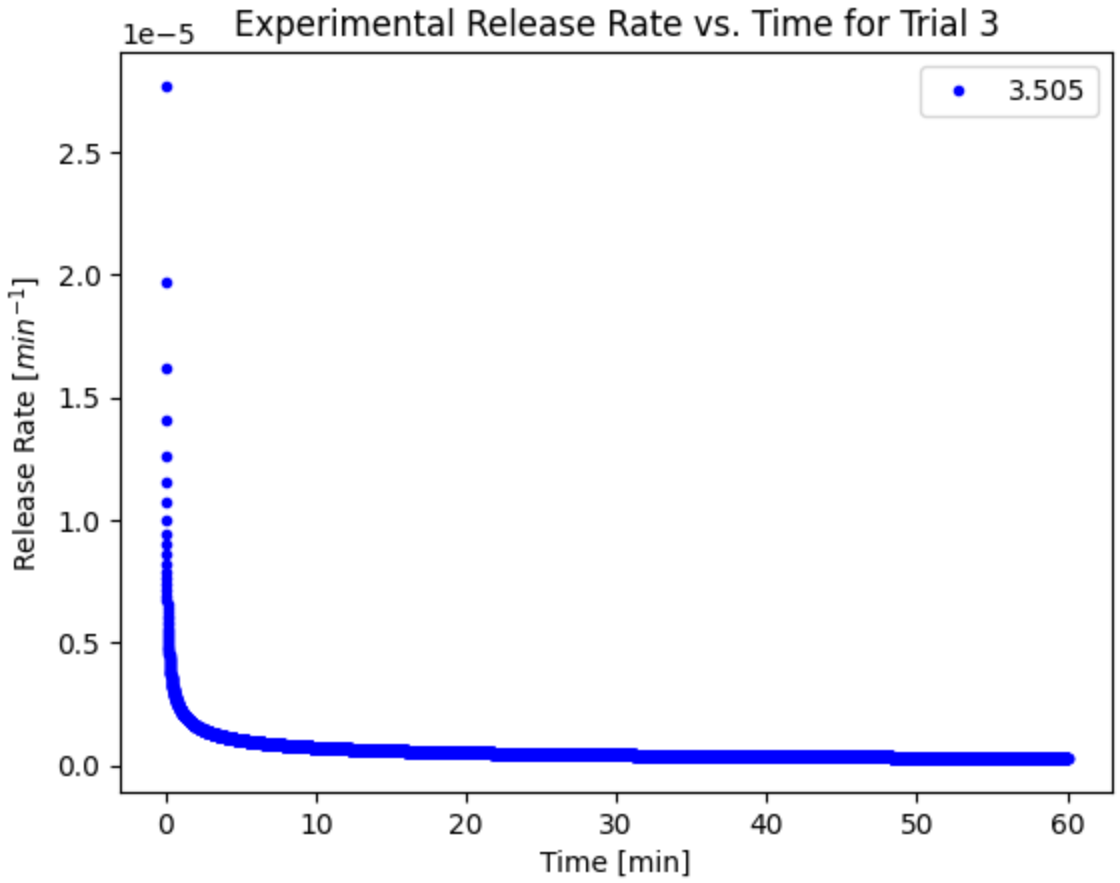
```
/tmp/ipykernel_22157/1162718564.py:2: RuntimeWarning: divide by zero encountered i
n power
  return k * n * (time ** (n - 1))
```



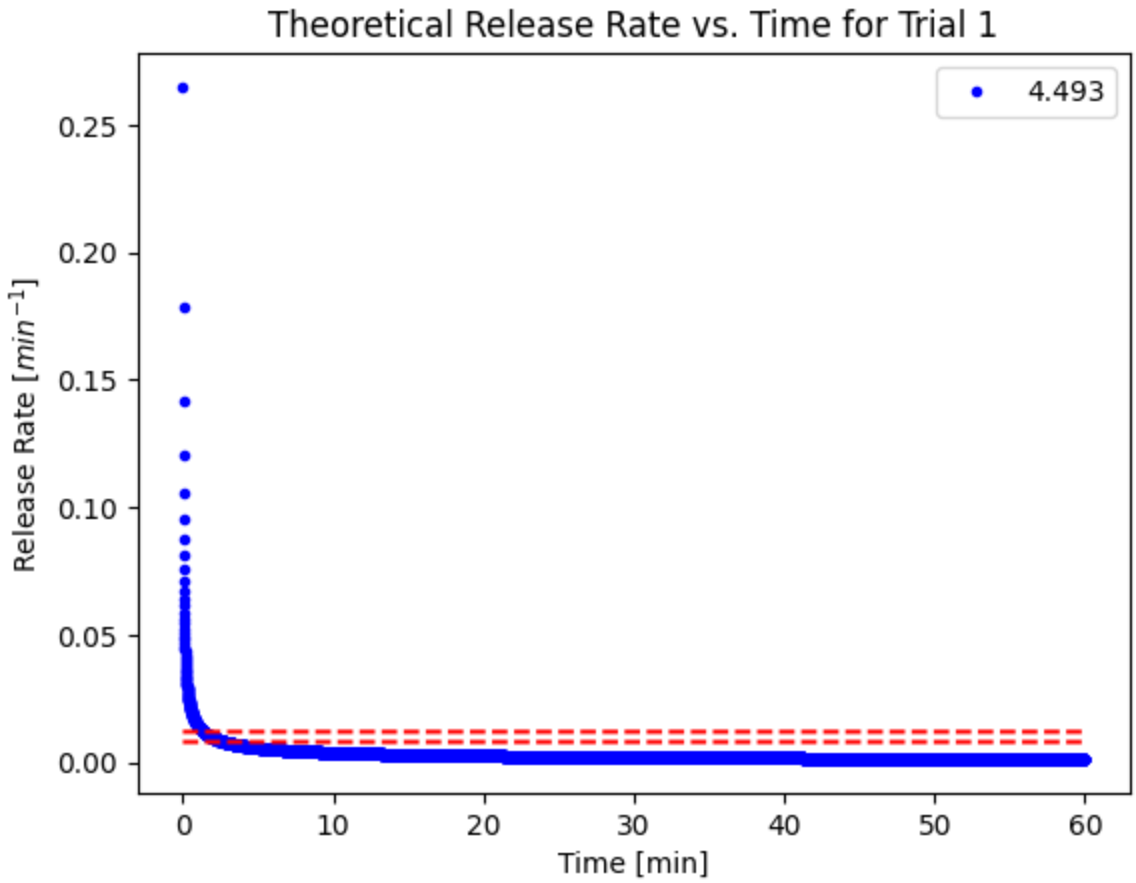

```
4.36526212591157e-06
 0.5957820504143857
```



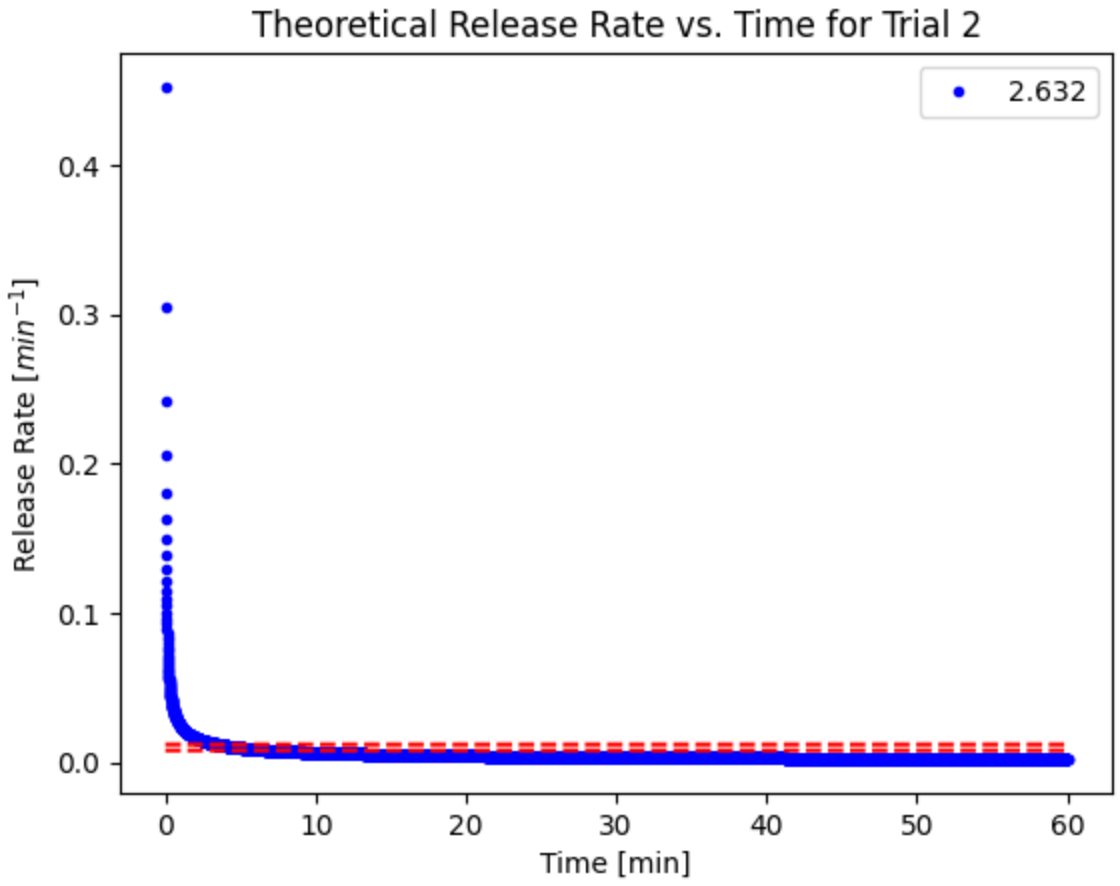

```
4.449425971186136e-06
 0.5113859115589585
```



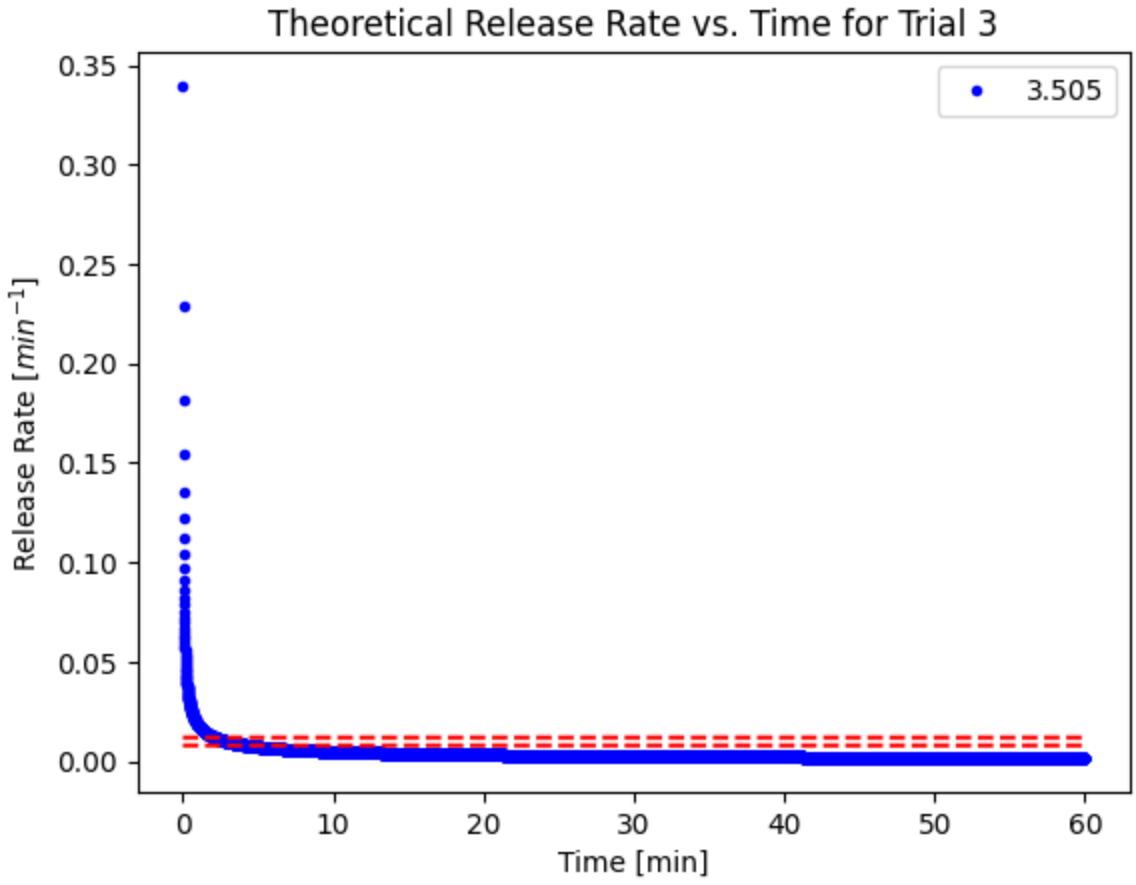
0.03335553655838699
 0.43



0.056940131366577794
 0.43




```
0.04275789607898225
 0.43
```



In [49]:
```
print(k_vals)
print(n_vals)
```

```
[1.14901565e-05 4.36526213e-06 4.44942597e-06]
[0.30562    0.59578205 0.51138591]
```

In [ ]: