**Detailed Final Attempt**

**1.Running Simulation**

We will run a simulation in GRO. The GRO file we will use is ABFDFM(1)(4).gro

This bacteria has 3 regions of density varying with radius at t=0. These are similar to the settings used by TheresaCorr in her lab report

Parameters:

set ( "dt", 0.1 ); // timestep of simulation, original=0.1

set ( "population\_max", 2000000 );

t := 0;

i := 0;

set ( "ecoli\_growth\_rate", 0.0034); // growth rate of bacteria, original=0.0034

nocell:=50; //nocell, original=50

phi:= (2\*pi)/nocell;

Regions:

c\_ecolis(1500, 0, 0, 800,{"p3","p2"},program p());

c\_ecolis(1000, 0, 0, 500, {"p3","p2"},program p());

c\_ecolis(200, 0, 0, 200, {"p3","p2"},program p());

(Starts with 2700 cells)

Firing:

t1> 190 & t1<193:{

s\_set\_signal(k1,3,0,0)

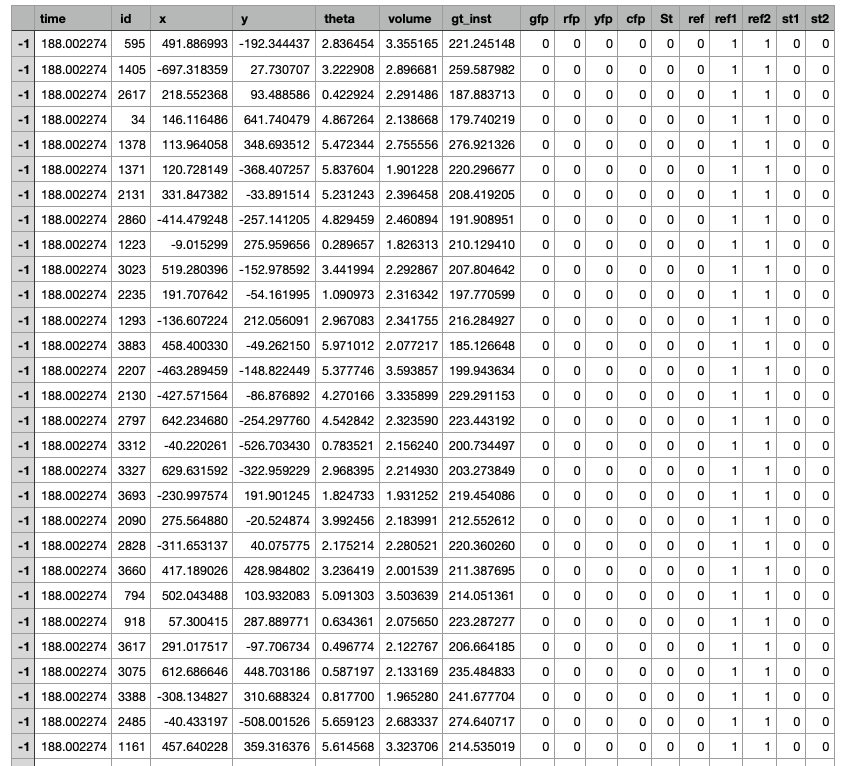
};

We are going to run this simulation, we started recording data for all of the bacteria at 188.00 mins. This is just before 190 minutes, when we expect the initial potassium firing to occur in the centre of the biofilm. The simulation was stopped at 328.41, at this point the potassium signal has degraded and the “wavefront” has reached the boundary of the biofilm. There are 6143 cells at this time.

(Insert Screenshots/GIFS)

**2.Saving Data**

All the data is saved in a csv file titled “original.csv” and treated as described in the lab book, wherein we concatenate the concentration with the time and xy data for each bacteria. Note that concentration values= -1 will be treated as =0 throughout.



**3.Plotting avg.Concentration VS Time at fixed radii**

***i) Defining radius***

When attempting to record concentration at a fixed radius, we must first define which data points fall within a selected “radial band”.

For a selected radius=r, we will collect points within the interval r +/- dr.

The next “ring” will start in such a way that we span the entire data set.

We take the average concentration of the N points that fall within r +/- dr to be the concentration at a radius r from the centre of the biofilm.

[Insert Diagram??]

Errors:

1. There is an uncertainty on this average due to the standard deviation on the mean

Concentration of the N points

Can add this to code for calculating the mean

1. There is an uncertainty on the associated values of r due to taking a range of values r+/- dr

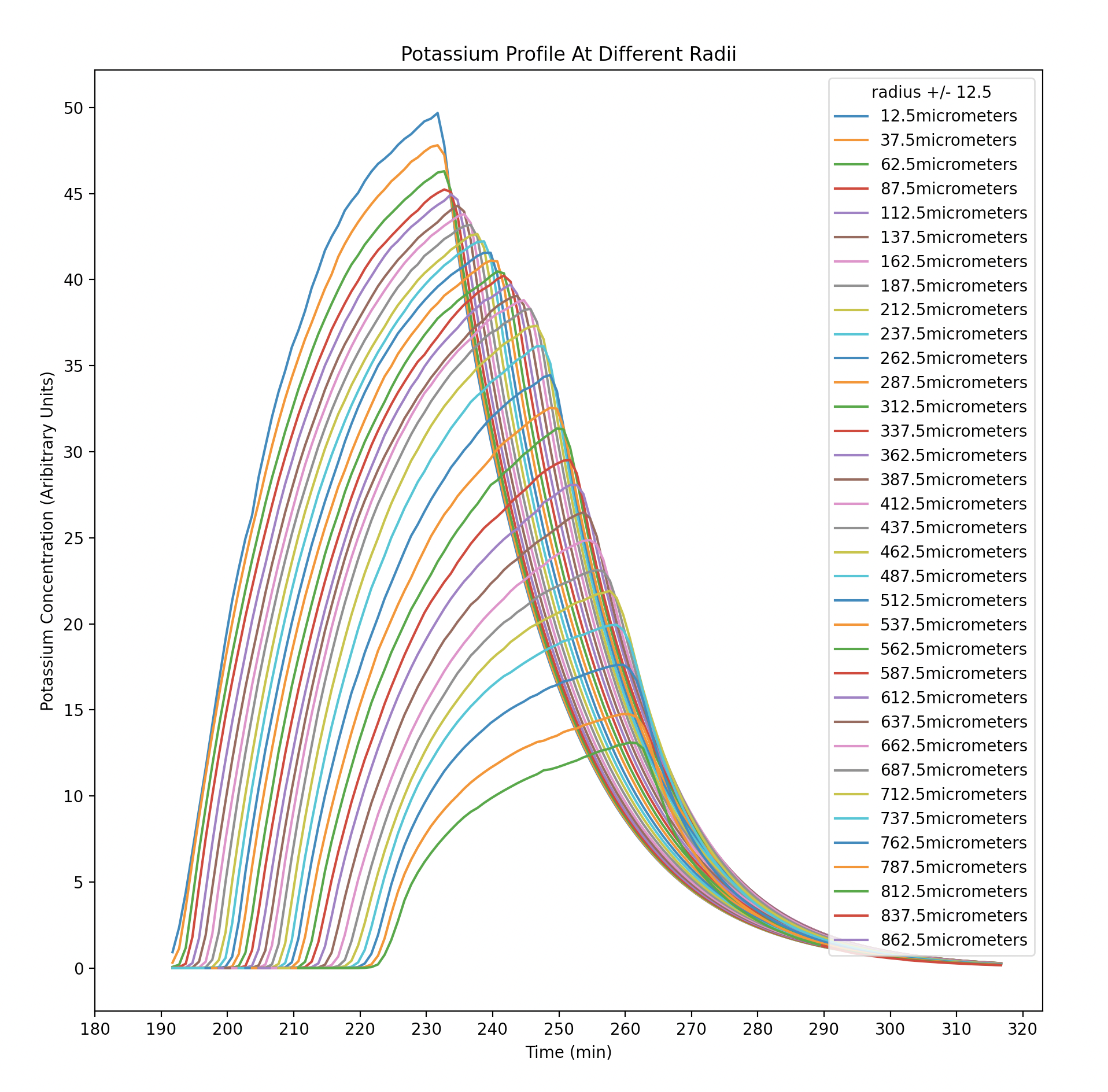
In the code: dr=0.5\*grid\_length

1. Recordings of time are taken by searching in intervals in 0.1. This means a value of 189.002335 will be recorded as 189/s. We do not combine any data points using this method as the timestep of the simulation output data is = 1/s

The grid\_length = 25micrometres. So for a given time we average the concentration of all points that fall within r+/-12.5.

When the wave reaches 800 micrometer (and maybe a little period before that), the wave has reached the area where there are no cells and subsequently the simulation breaks down. Therefore that part of the data should be removed.

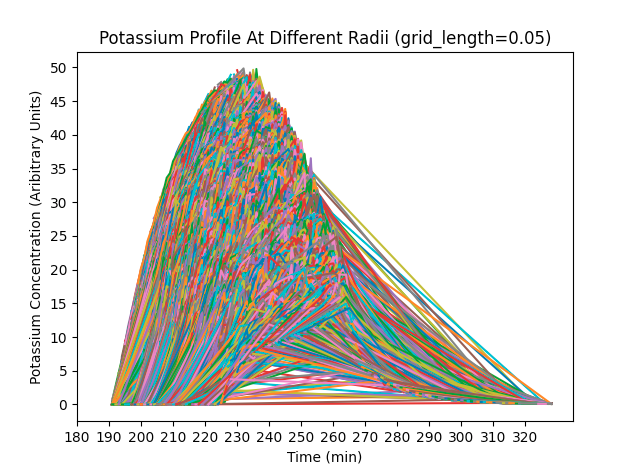
We set the value of variables r\_cut=850 and t\_cut=380 and delete the data with radial position or time position larger than them.

The following graph without associated uncertainties was produced:

In order to achieve a higher resolution and analyse peak movement at lower radii, we repeat this changing r\_grid\_length=0.05 micrometer and t\_grid\_length=0.05min. This reduced bin size will provide greater accuracy.

We set the value of variables r\_cut=800 and t\_cut=260 and delete the data with radial position or time position larger than them.

As expected this graph is incredibly messy due to the much greater resolution:



**4.Finding the Wave Peak**

We opted to use the peak concentration value at a given time as the “reference point” of the potassium wave as it travels radially outwards through the biofilm.

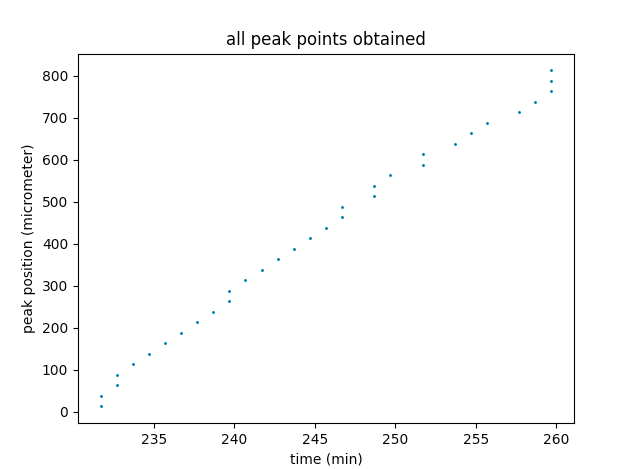
From the previous graph, for each radial distance band we find the time at which the mean concentration is the largest. This data is then stored in an array.

Errors:

1. Is there some uncertainty on the peak point?

**5.Plotting Peak position against Time**

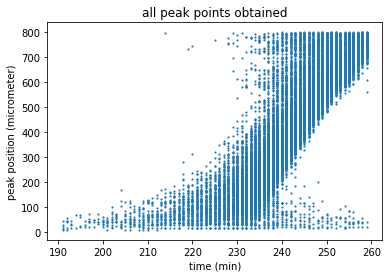
Plotting the peak positions against time we get the following graph (grid\_length=25micrometres)



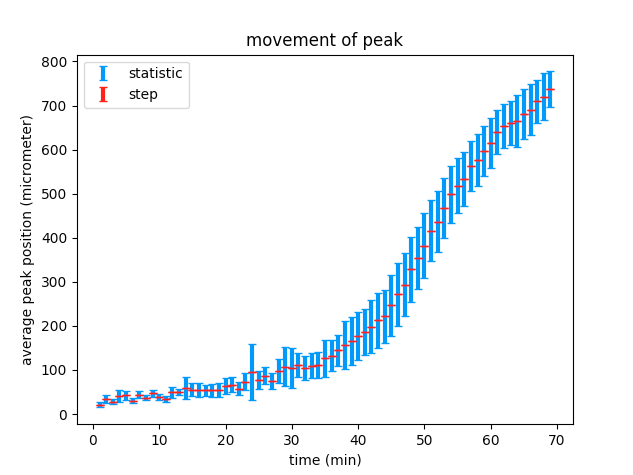
Using radial bands of width 25micrometres we see a straight line function with some vertical component at small radii and large radii. We also see multiple peaks that supposedly occur at different radii and the same time, indicating an uncertainty in the position of the peaks, (eg-they’re not the same in all directions), or too large of a bin size in the code.

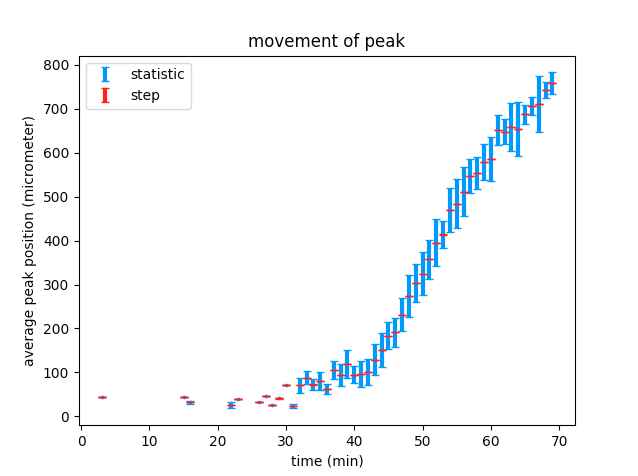
However, decreasing the radial band width may allow us to produce a graph with a higher resolution and this will make it easier to plot a function on top of the data.

With a radial band width of 0.1 we see the peaks piled on top of each other. This is likely due to a resolution smaller than that offered by the simulation.

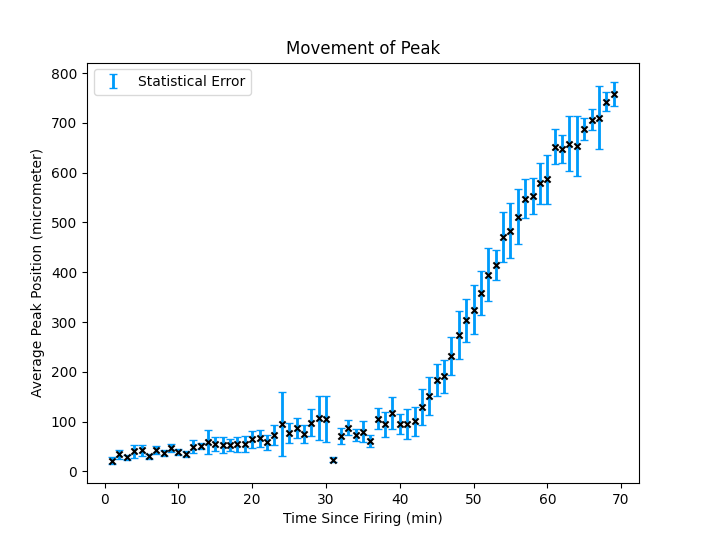
(grid\_length=0.05 micrometres,0.05 minutes)

Averaging these points and taking the statistical uncertainty to be the standard deviation/sqrt(N) results in the following graph. In the slow velocity regime near the centre, the error bars appear reasonable, however at larger radii these are too big. Using a larger radial bandwidth will counteract this. Clearly for both graphs the uncertainty on the radius “step” is negligible compared to the statistical uncertainty, so will be ignored from here.

0.05

0.5

However, at r\_grid=0.5 we cannot resolve any detail in the slow regime, so we will combine the graphs. 0.05 grid length up to 31minutes and r\_grid=0.05 for the remainder of the time since firing.

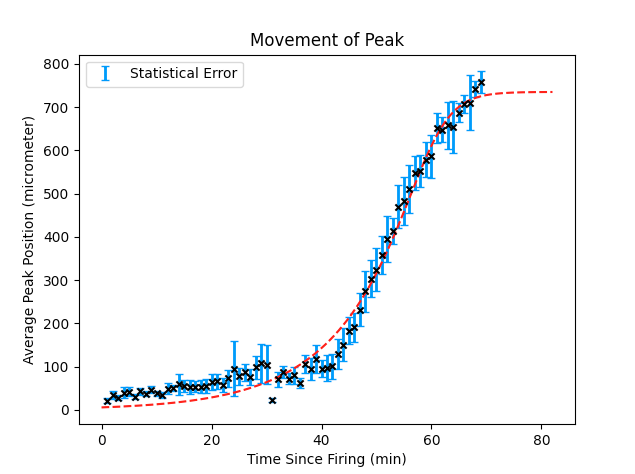


Although the statistical error appears to not match the deviation of plotted points from the overall trend, it is important to note that this statistical error is due to the poor time resolution of the dump\_single function and the fact the peak may not travel at the same speed in all directions. As a result we must average multiple peak positions at the same time. Meaning that any fit we plot to this trend will follow the trend but may be translatable up or down within the error bars, thus producing an uncertainty.

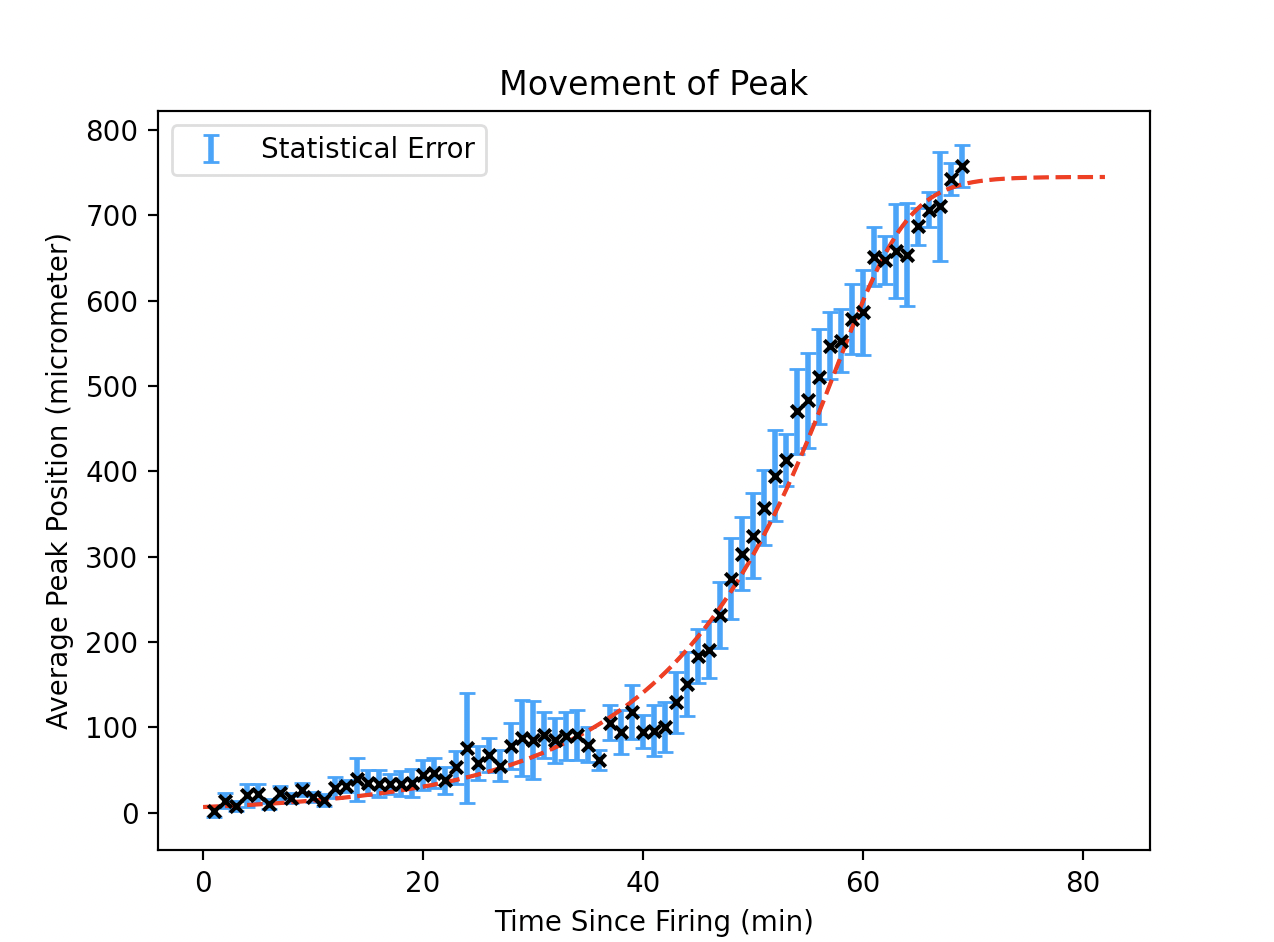
Now, we want to plot a fit to this data, Sigmoidal fits are seen commonly in biology and would well represent the three regimes that appear on this graph. We initially considered the “logistic sigmoid function” with some parameter on x. However, quickly realised this would not fit the growth phase of the process.

Instead we decided to use the following parameterised sigmoid:

We simply used the curve\_fit function of scipy.optimize to make the fit.



[a,b,c,d] = [5.80018773e+00 3.31881798e-01 1.94798396e-09 2.41444031e-01]



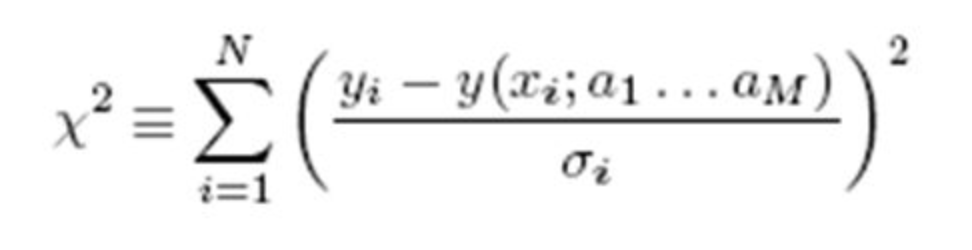
The graph above is generated with peak points extracted from original.csv.

A combination of data was introduced under the following circumstance:

1.Analysis with a spatial resolution of 0.05 micrometer gives sufficient peaks at low radial distance with acceptable uncertainty, while the uncertainty gets too large at large radii.

2.Analysis with a spatial resolution of 0.5 micrometer gives sufficient peaks at high radial distance with acceptable uncertainty, while it barely gives peak points at low radii.

Reduced chi-sqaured=



Reduced this by dividing by the DOF=n

“A mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets ("the residuals") of the points from the curve.” Mathworld

Luis Valcárcel, McGill University

October 19, 2005

HEP Graduate Student Meetings

The sigmod had a reduced chi squared=

Therefore at low radii, we use peaks from 0.05 micrometer and at high radii we use data obtained from the analysis using a resolution of 0.5 micrometer.

The sigmoid function fitted to the data was a parameterised version of logistic function, with the form

parameters [a b c d] = [6.59870967e+00 3.84989021e-01 4.67367145e-11 1.98716600e-01]

deviation [8.47911770e-01 1.41820952e-01 4.14086492e-10 7.78693292e-02]

The sigmod had a reduced chi squared=0.9854

**6.Plotting Average Velocity against Time**

To have a general description of the movement of the wave, we look into the average velocity of the reference point (peak point).

Average velocity is calculated using , where R is the radius position of the peak point and t is the time at which peak reaches R.

Uncertainty propagation suggests .

Uncertainty of v has two sources:

1.statistical uncertainty due to differences between the peak points: in the expression above we take the position uncertainty calculated before as dR, while dt is 0

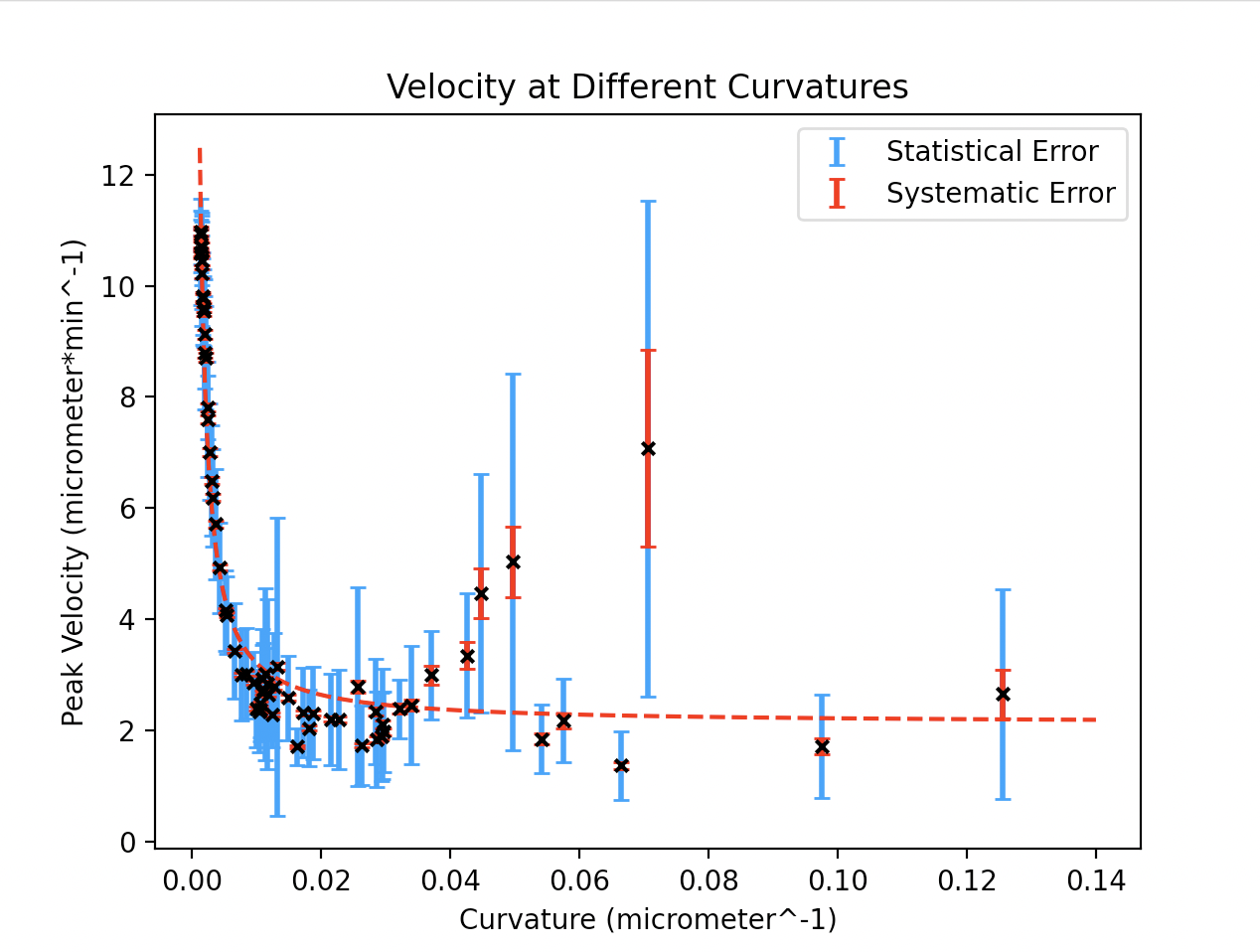
2.systematic uncertainty due to the discrete nature of the simulation and analysis: in the expression above we take half of the resolutions as dR and dt.

We will fit the data to a power law with the form

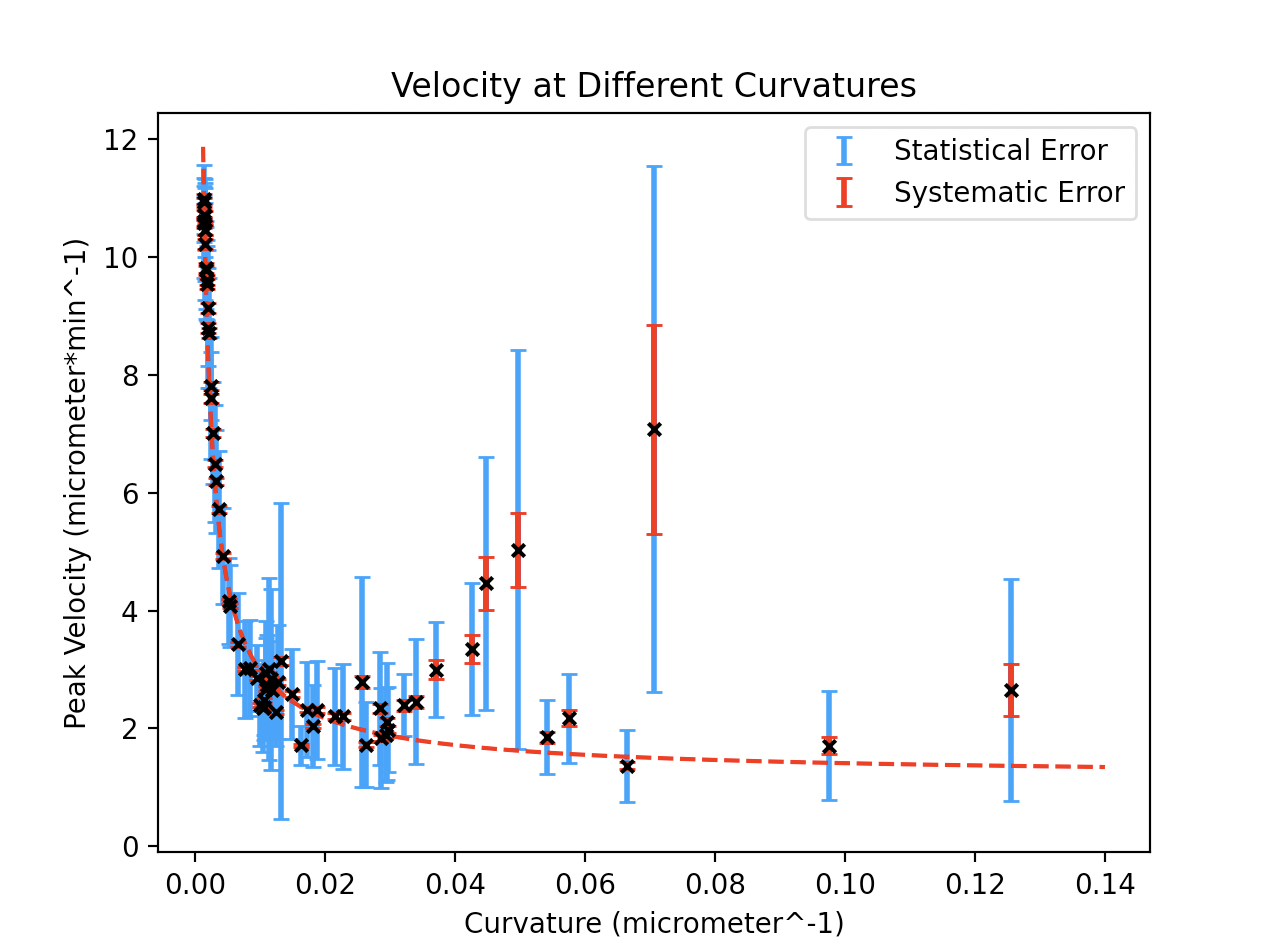
.

Graph generated with data from the last peak position-time diagram is:

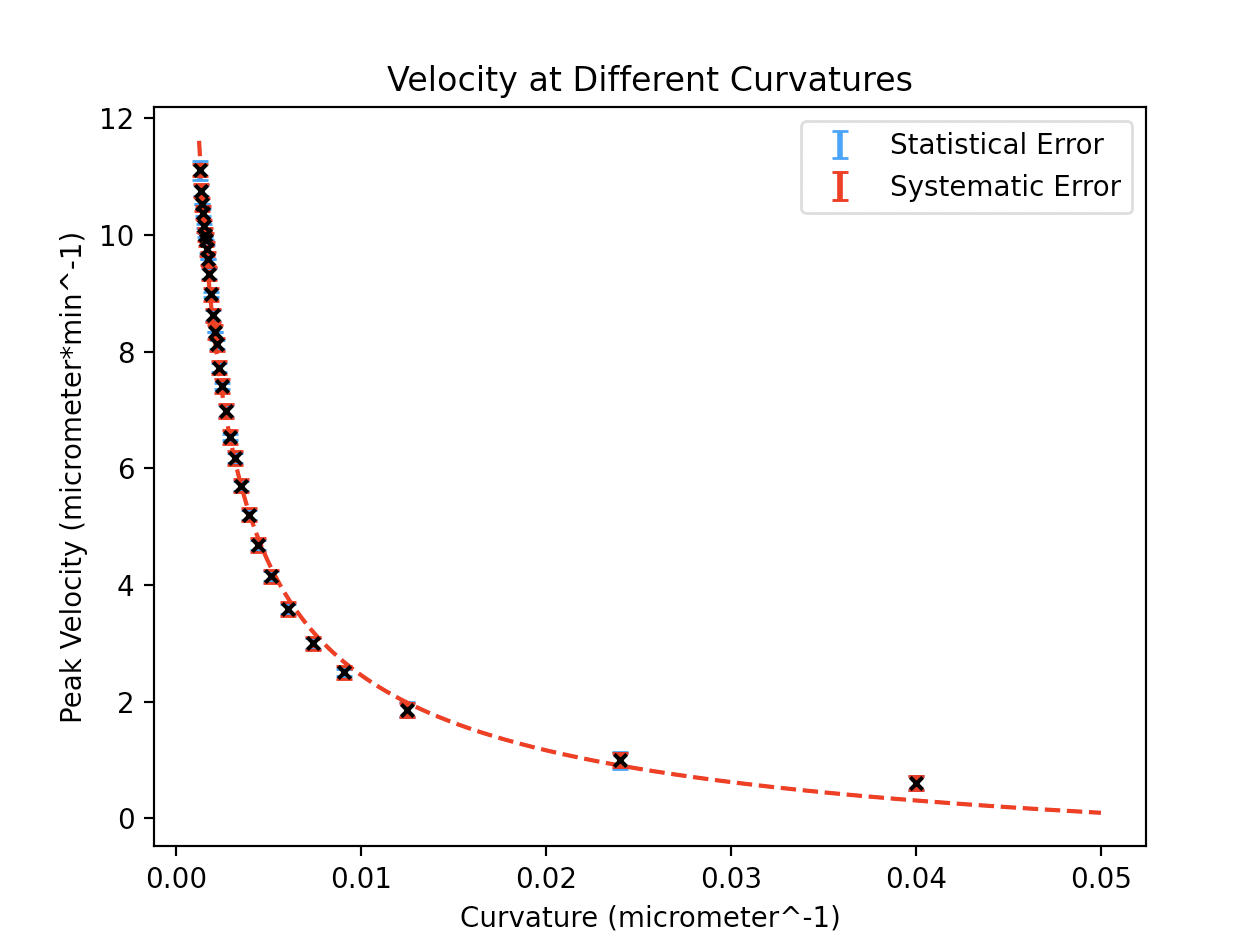
Here we do not use sigma to inform the curve\_fit function as we must work out whether to utilise the statistical or systematic error



It is unwise to combine the statistical and systematic error at this point as they are not uncorrelated. So as before, we make the decision to use the statistical error when producing a curvefit. (scipy.optimise(...sigma=statstical\_error, sigma=True)



Using the statistical errors to inform our curvefit, we can see the data fits the power law at low curvatures well with reasonable uncertainties, but gradually the uncertainty grows and fluctuates dramatically.

From our experience of part 5, we know that with larger spatial resolution we will obtain data with smaller statistical uncertainties, and we tried r\_grid\_length = 10 micrometer and obtained the graph below.

Note, at larger r\_grid we cannot record values near to the centre of the biofilm, hence 1/r does not go as high on this graph. Also, note that the statistical and systematic errors are both essentially equal at this point, and minimal, this indicates we have reached an r\_grid length that finds the balance between our competing sources of error. Overall the systematic/step error contributes the largest uncertainty here so will be used to produce our curvefit.:

parameters [a b c] = [ 0.26496686 0.58284071 -1.42491055]

deviation [0.03450324 0.01763811 0.20334351]

Reduced Chi-Squared = 1.55

(note: using stat\_error we get reduced chi square=2.95)

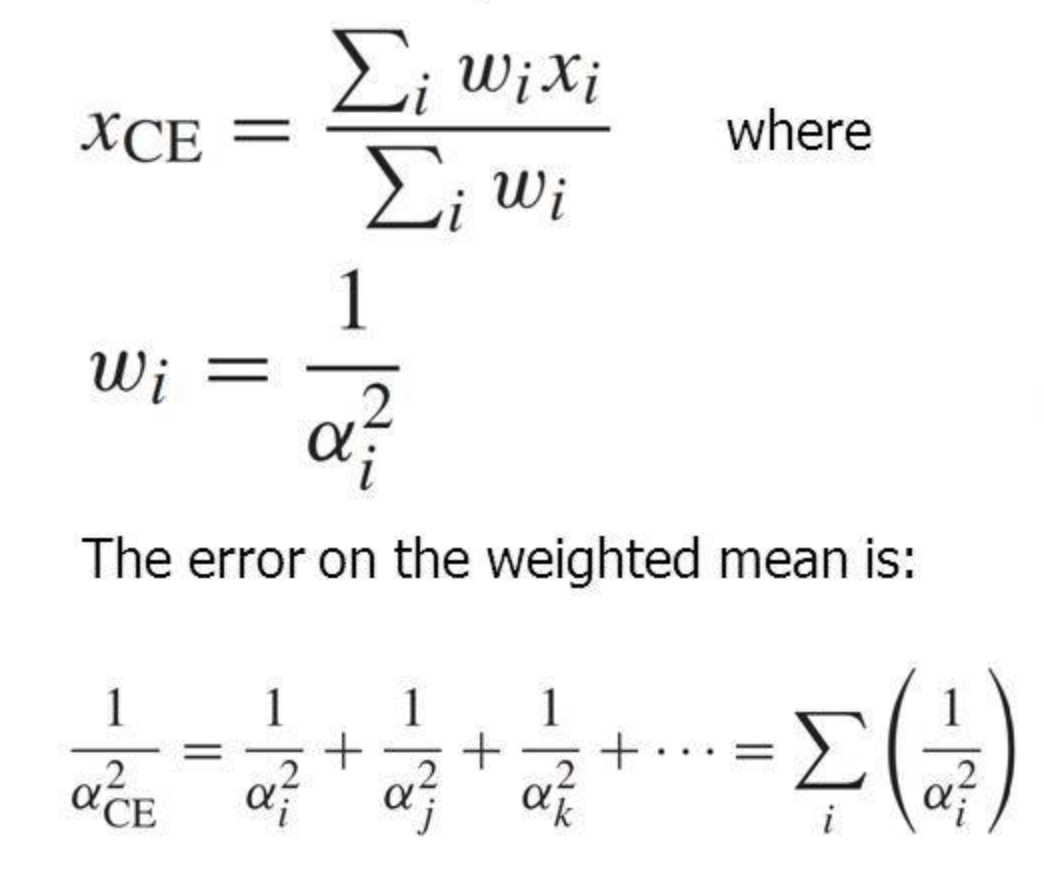
Fits quite well.

**7.Investigate the Stochastic Nature of the Simulation**

As pointed out by the documents concerning Gro, in the simulation guided by ABFDFM.gro, the initial distribution of the bacterium is randomly set.

Therefore we will run multiple simulations to look into how the stochastic nature of the simulation would affect the data we obtain and the result of our simulation.

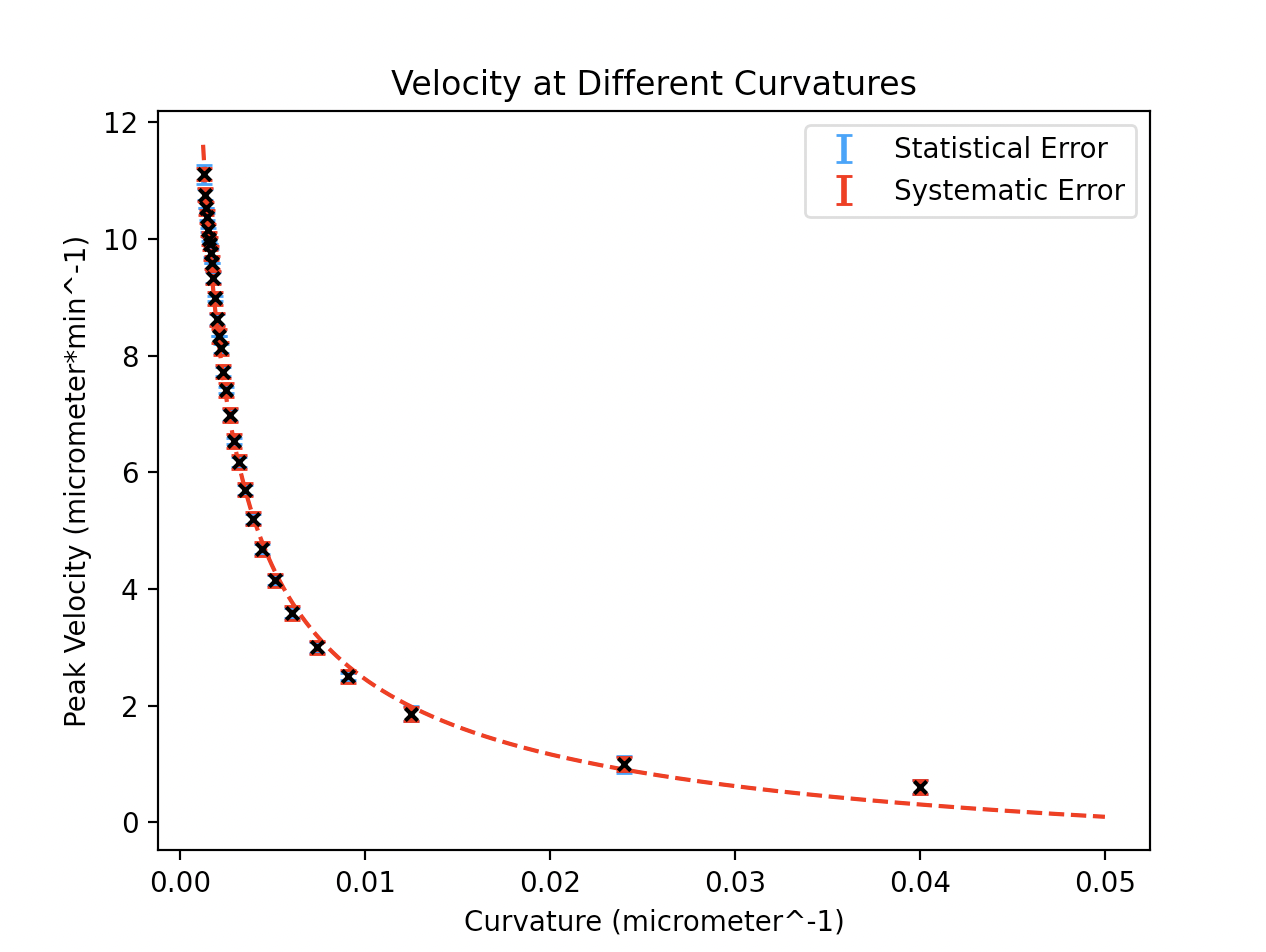
We will start with the average velocity curves by comparing the power law fits generated with data from different runs. We will produce the best curevfit for each one and record the parameter values with the associated uncertainties. From these repeats we will calculate a weighted mean with a propagated uncertainty, and these will be our final values of the parameters for our power law model of velocity against curvature.



Weighted means and there uncertainties were calculated as above.

Run1:

Simulation data stored as original.csv.



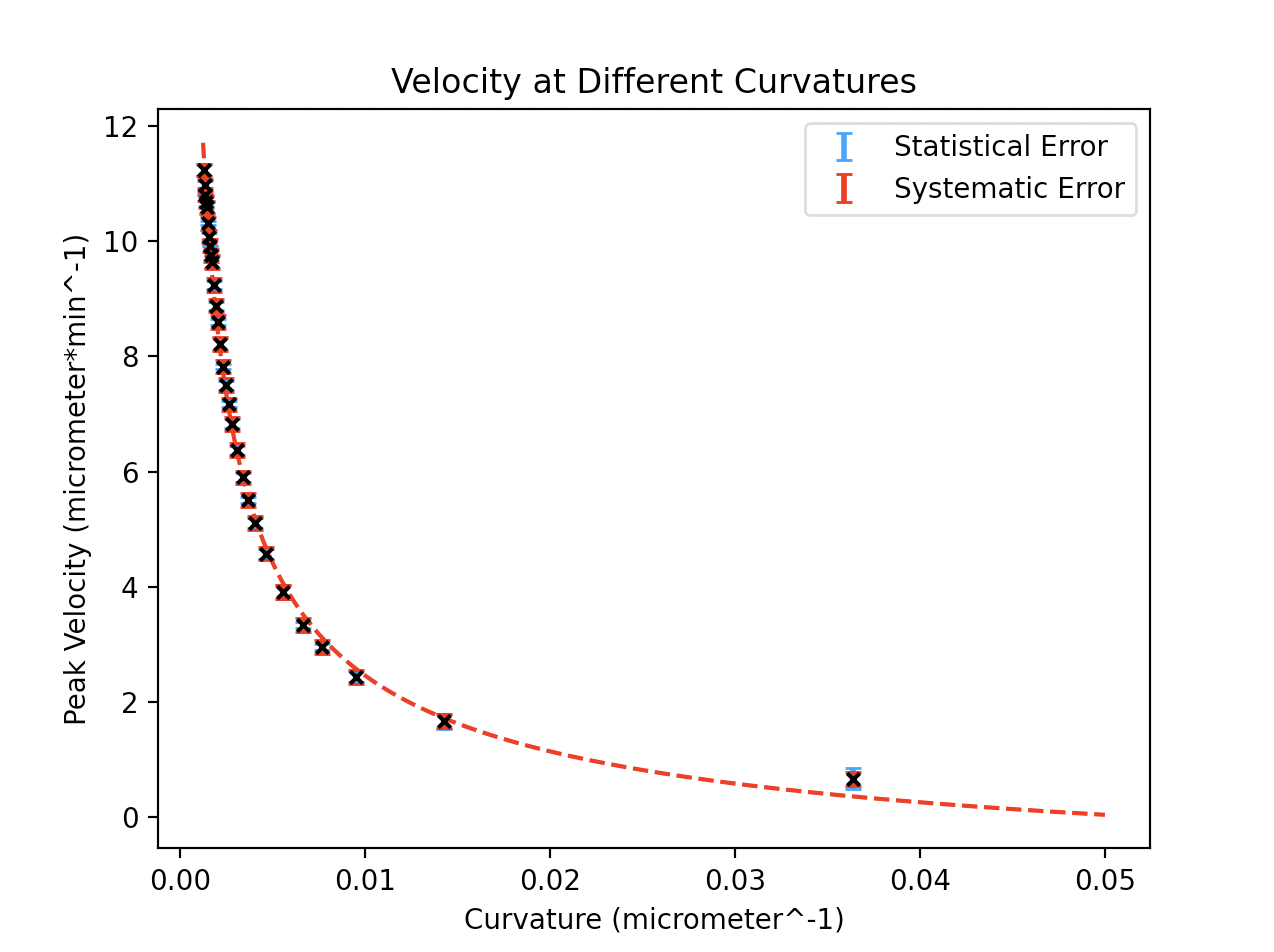
parameters [ 0.26496686 0.58284071 -1.42491055]

deviation [0.03450324 0.01763811 0.20334351]

reduced chi-squared = 1.551294004860183

Run2:

Simulation data stored as original\_processed\_1.csv.



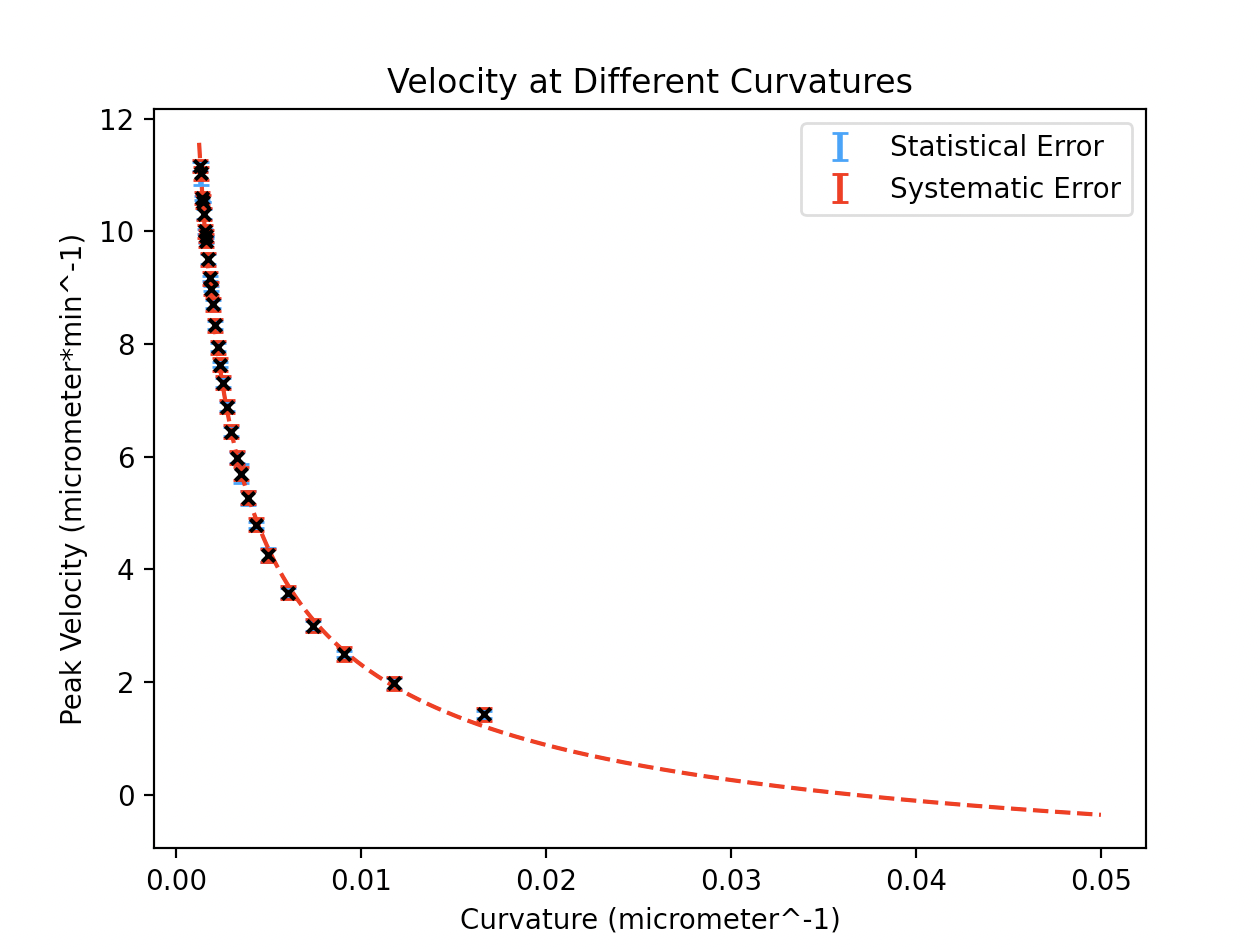
parameters [ 0.29526065 0.56961086 -1.60937 ]

deviation [0.0413555 0.0186976 0.24017637]

reduced chi-squared = 1.4160684422381613

Run3:

Simulation data stored as original\_processed\_2.csv.



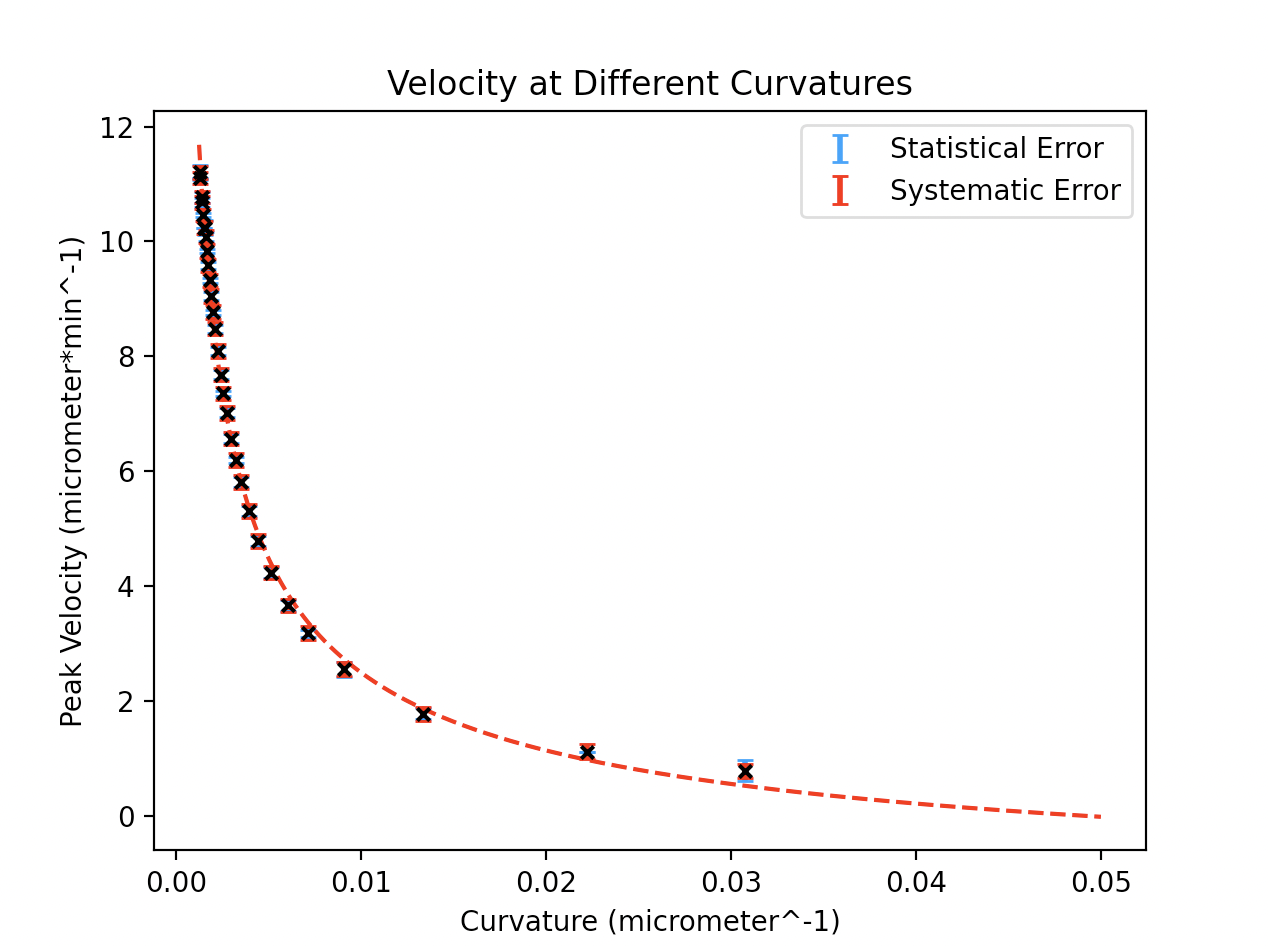
parameters [ 0.41581398 0.52535875 -2.36059388]

deviation [0.0758406 0.02372257 0.37460015]

reduced chi-squared = 0.6796828100432024

Run4:

Simulation data stored as original\_processed\_3.csv.



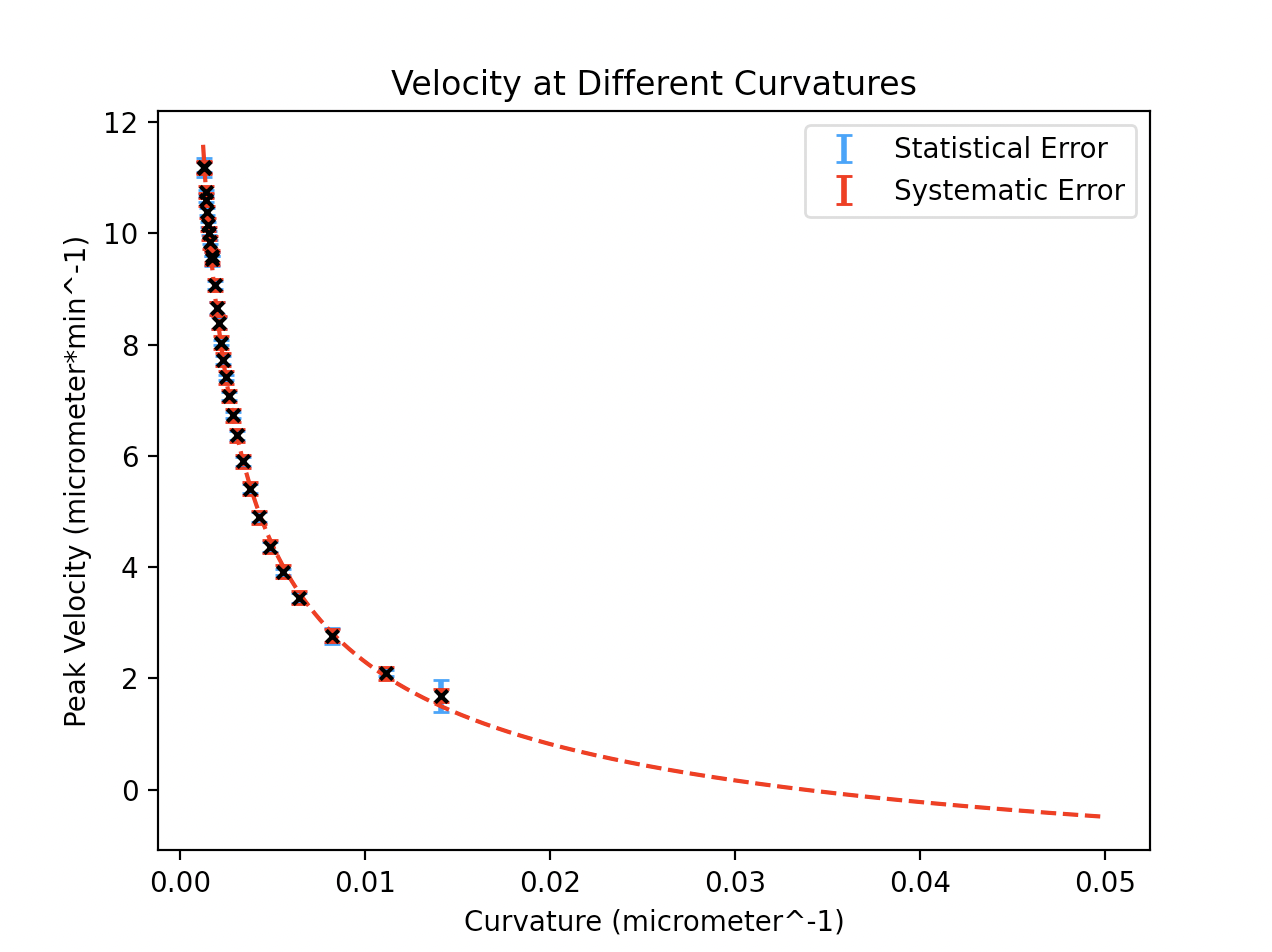
parameters [ 0.33539919 0.55218912 -1.76427786]

deviation [0.0453802 0.01803409 0.23813735]

reduced chi-squared = 1.426309249115734

Run5:

Simulation data stored as original\_processed\_4.csv.

parameters [ 0.49193393 0.50416715 -2.71694887]

deviation [0.09689476 0.0252522 0.44468046]

reduced chi-squared = 0.5720743273816464

.

a=0.31167180802388816 +/- 0.02136547092127166

b=0.5542199361408728 +/- 0.008946169347633549

c=-1.738686951567446 +/- 0.11839714964951213

**8.Investigate the Effect of the Diffusion Coefficient on the average velocity of the peak**

To investigate the influence of diffusion coefficient, we will run multiple simulations with different diffusion coefficients.

As we have seen in the previous graphs, the simulation breaks down at around 750micrometres, here the position of the peak plateaus. This is despite the actual biofilm being about 800 micrometres in radius. For the sake of argument we will define the “edge” of the biofilm as being at 750 micrometres.

In this part of the experiment we are interested in the speed at which a biofilm can “communicate” from it’s centre to the edges of itself. We wish to quantify how this “communication speed” is affected by the diffusion coefficient of the potassium ions. We expect this speed to increase with the diffusion coefficient , however is this limited??

Run1(original):

Diffusion coefficient = 0.4

[r\_grid\_length=0.5]

average\_speed = 11.363636363636363

stat\_error = 0.13167966063735198

step\_error = 0.060881134937592236

Run2:

Diffusion coefficient = 0.6

[r\_grid\_length=0.5]

average\_speed = 12.798634812286691

stat\_error = 0.23161558404612576

step\_error = 0.069661690164115

Run3:

Diffusion coefficient = 0.8

[r\_grid\_length=0.5]

average\_speed = 13.25088339222615

stat\_error = 0.25680283255845265

step\_error = 0.07244330336279463

Run4:

Diffusion coefficient = 1.0

[r\_grid\_length=0.5]

average\_speed = 13.661202185792348

stat\_error = 0.14210929411198275

step\_error = 0.0749722216385635

Run5:

Diffusion coefficient = 1.2

[r\_grid\_length=0.5]

average\_speed = 13.636363636363637

stat\_error = 0.1084986383466675

step\_error = 0.07481900005501163

Run6:

Diffusion coefficient = 1.4

[r\_grid\_length=0.5]

average\_speed = 14.478764478764475

stat\_error = 0.12786675048180282

step\_error = 0.08002453804035235

Run7:

Diffusion coefficient = 1.6

[r\_grid\_length=0.5]

average\_speed = 14.395393474088293

stat\_error = 0.08998010421302541

step\_error = 0.07950855837116738

Run8:

Diffusion coefficient = 1.8

[r\_grid\_length=0.5]

average\_speed = 15.03006012024048

stat\_error = 0.17146337923956298

step\_error = 0.08344056992550172

Run9:

Diffusion coefficient = 2.0

[r\_grid\_length=0.5]

average\_speed = 14.763779527559052

stat\_error = 0.1617505210437761

step\_error = 0.08178972755221793

Run10:

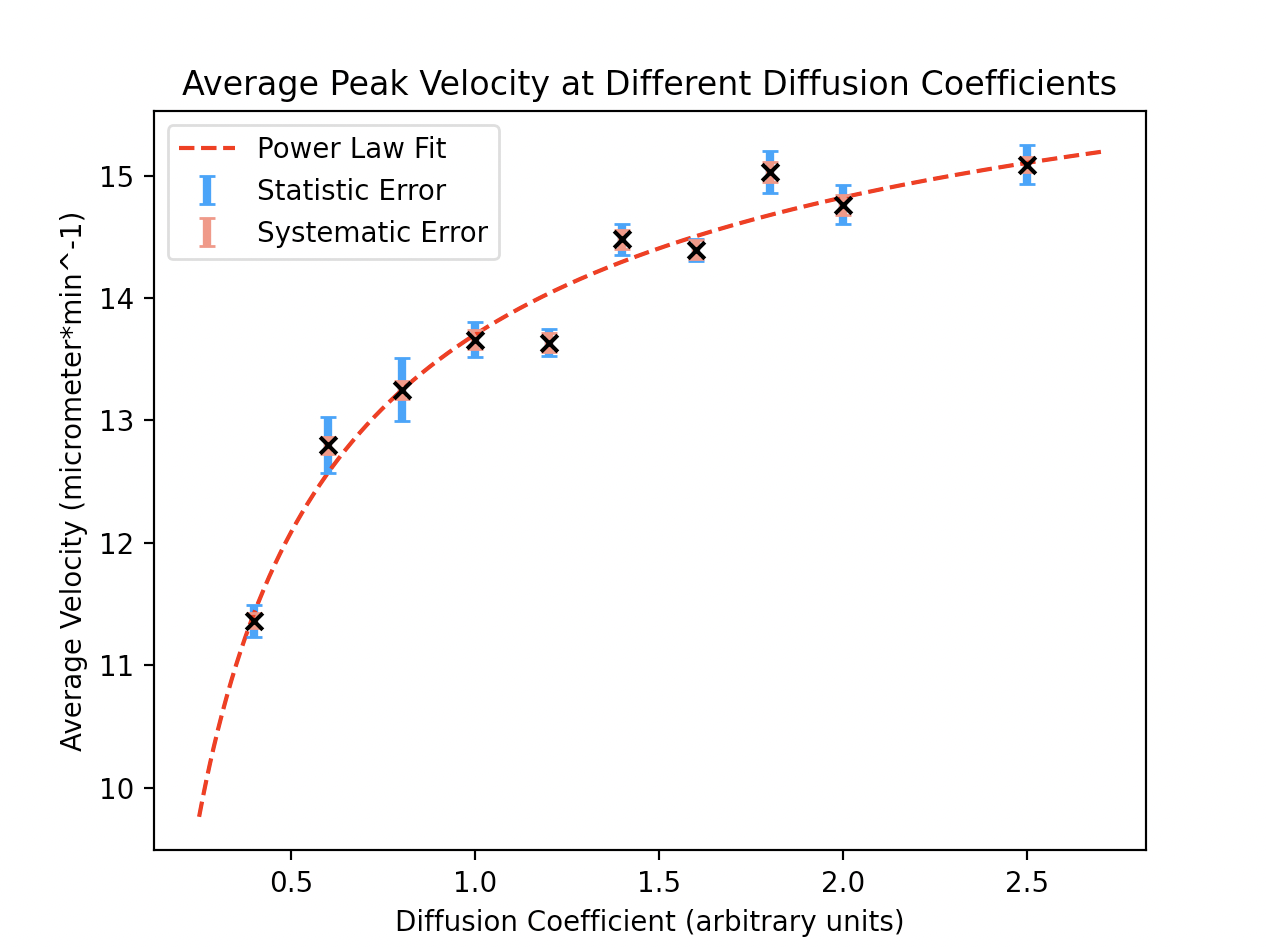
Diffusion coefficient = 2.5

[r\_grid\_length=0.5]

average\_speed = 15.090543259557348

stat\_error = 0.16185926739639897

step\_error = 0.06319864719850085



.

power\_law\_fit

parameters [a b c] = [-3.66376675 -0.52673544 17.36614818]

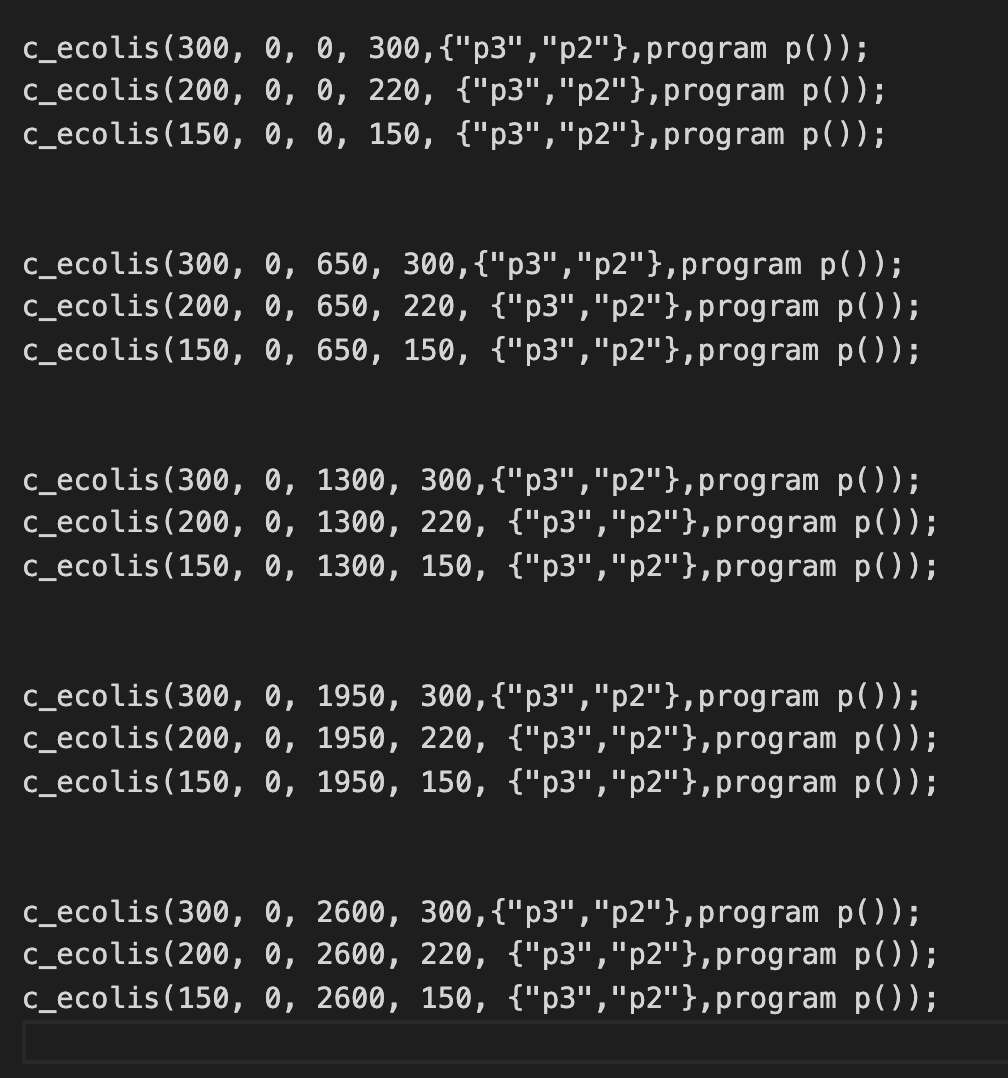
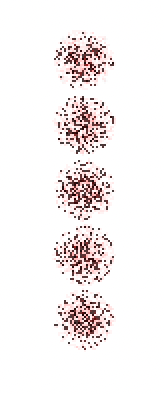
deviation [0.562909 0.07413126 0.53731769]

reduced chi-squared = 2.541275222927501

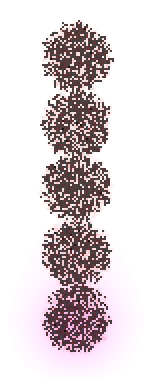
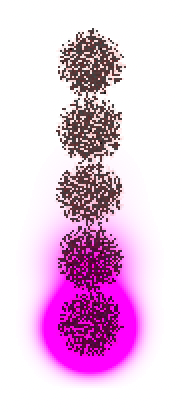
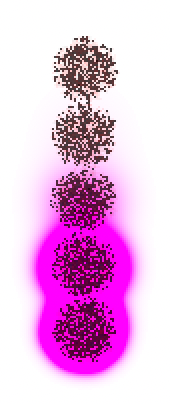
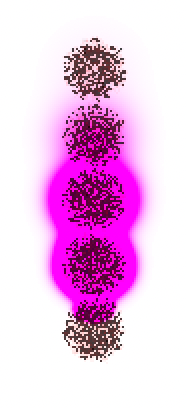
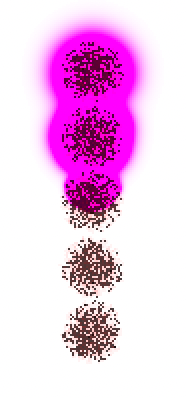
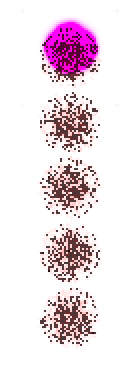
**9.Inter-colony Interaction**

After analysising the propagation of potassium signals within the biofilm, and drawing connections between the wavefront velocity and the curvature of the wave. We have decided to treat each biofilm as a point mass, with an average concentration per bacteria that essentially turns on and off during a potassium trigger. This is done by averaging over the whole biofilm at a given time and calculating the average potassium concentration per unit cell in the film. From this we can plot graphs of entire biofilm concentration levels against time.

By simulating 5 separate biofilms in a line, each with equal separation and producing a trigger at the top of the uppermost biofilm, we seek to make a measurement of how fast the “information” potassium trigger, travels across multiple biofilms. Also we seek to see if there is any degradation in this process or if it can occur ad infinitum.

****

**Screenshots of the propagation of the potassium**

****

**We take the centres of each colony to be the position of the point-mass biofilm.**

**Colony position time/mins**

Colony\_1 0 51.89978

Colony\_2 650 82.899307

Colony\_3 1300 113.898834

Colony\_4 1950 145.899704

Colony\_5 2600 172.901352

Taking times from the time of the first peak, to produce a graph that goes through the origin.

**Colony position time/mins**

Colony\_1 0 0

Colony\_2 650 30.999527

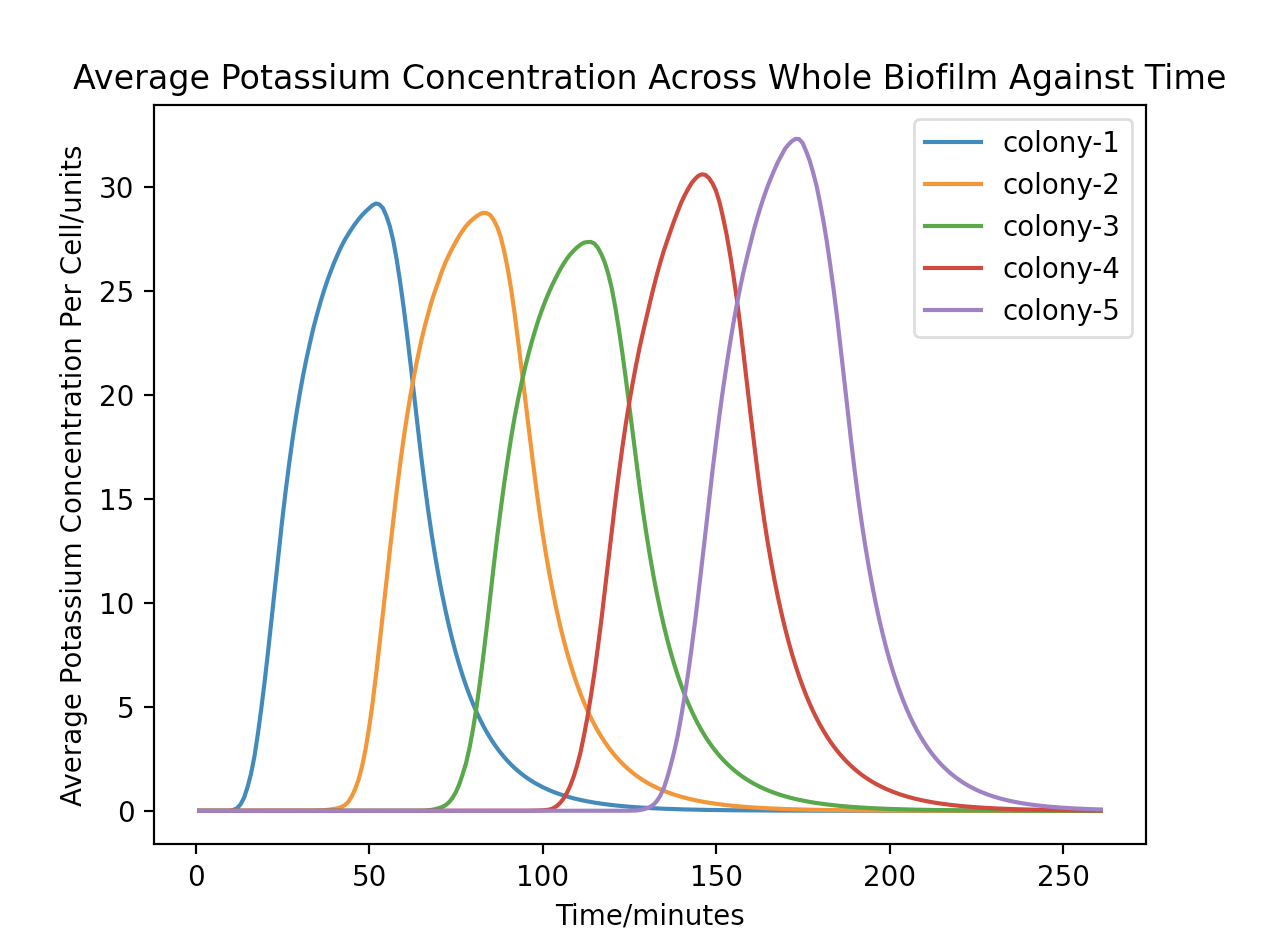
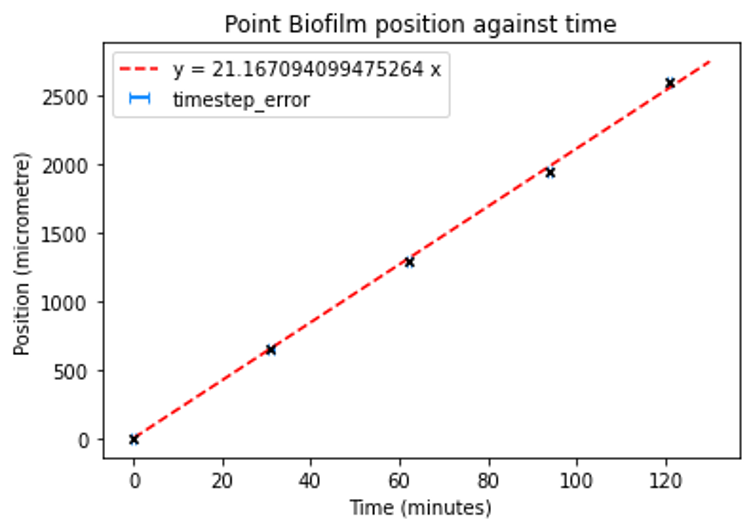
Colony\_3 1300 61.999054

Colony\_4 1950 93.999924

Colony\_5 2600 121.001572

**The time measurement have an associated uncertainty of 0.5min due to the time step of the simulation.**

**The position of each biofilm is treated as a point mass in this situation. We are not interested in measuring the position of the potassium wave here, only the peak overall concentration of the biofilm. We measure how fast this travels from our 1st colony to the 5th colony, e.g- through the 5 separate point masses.**

****

**9.Investigate the Effect of the Diffusion Coefficient on the balance-phase velocity of the peak**