

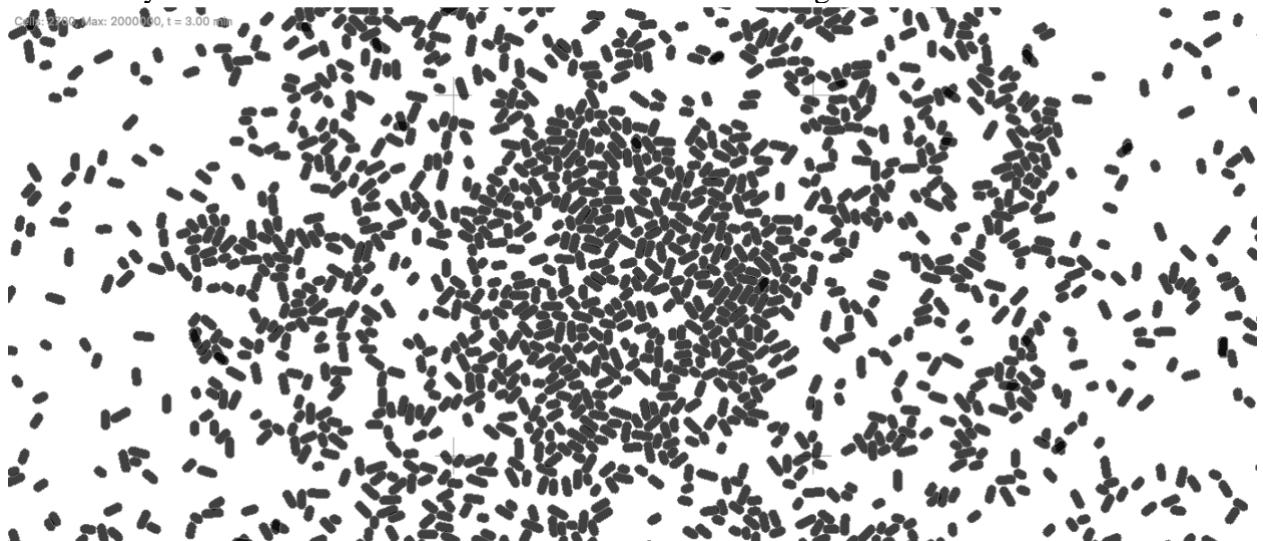
Biofilms Lab (Yr 3, Sem 2)

Luke Peters & Peng Wang

23/03/2021

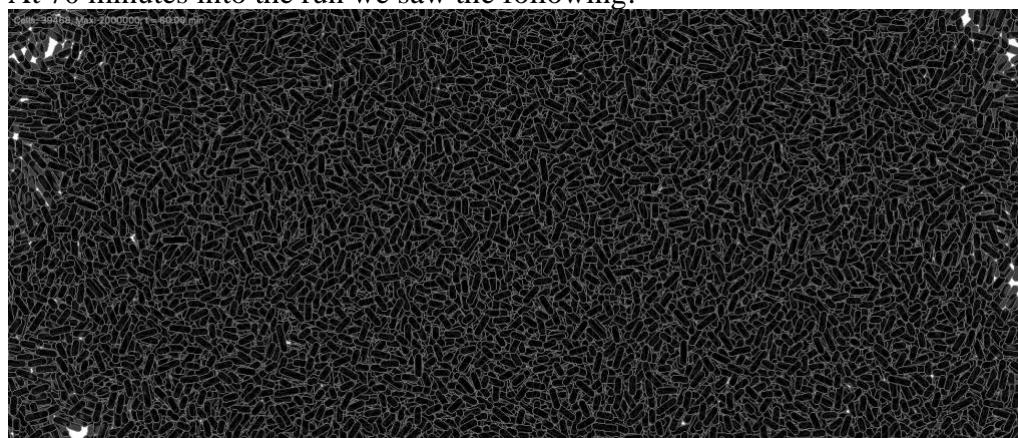
12:00

We initially downloaded GRO and ran the standard ABFDFM.gro file.



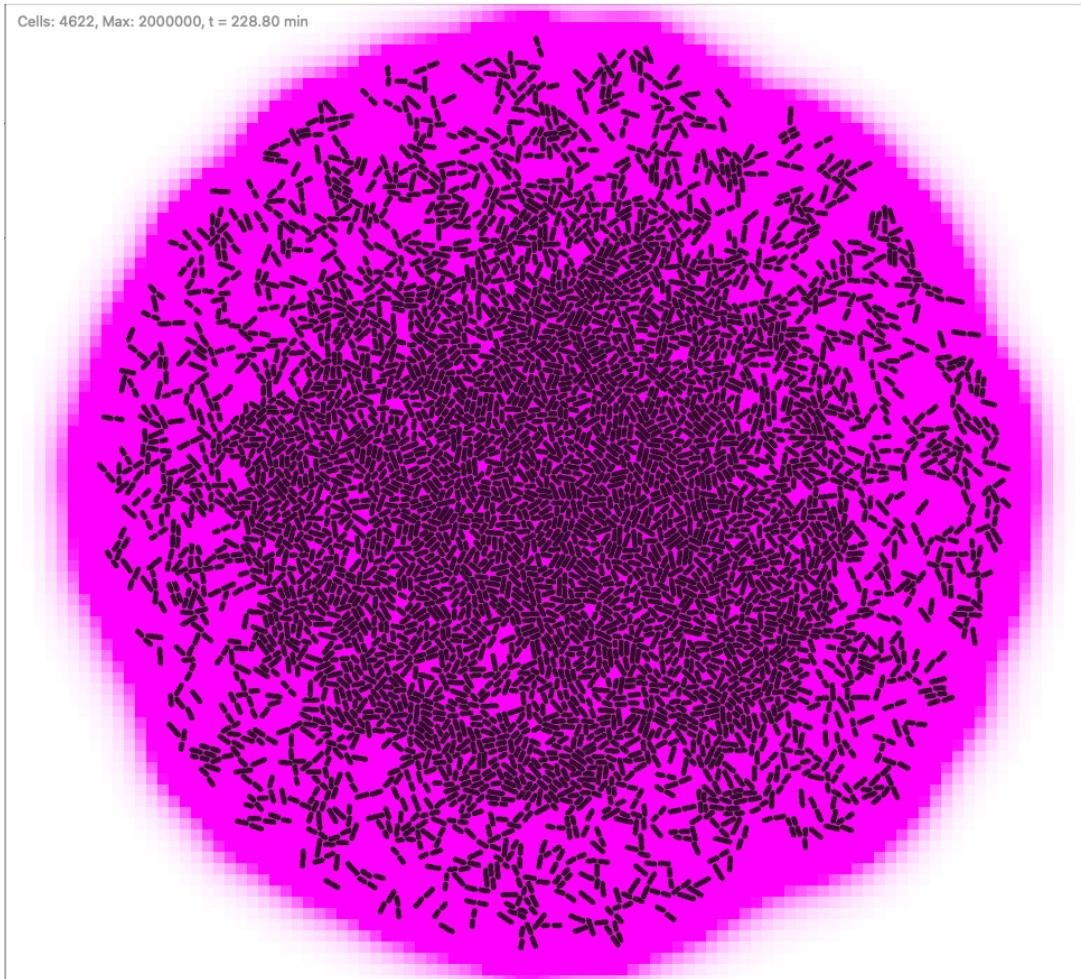
This simulation took a very long time so we increased the timestep of the simulation from 0.1 to 5. Allowing us to view the simulation at later times.

At 70 minutes into the run we saw the following:



The growth of the cell was a lot more rapid using a time step of 5 min.

In order to view potassium release by cells, we ran the simulation for a longer time. After reaching threshold potassium concentration the bacteria released potassium:
At 228.8 minutes.



By selecting a single bacteria and analysing the single_dump data and get_signal(K1) data, we clearly see that at the time St and st2 were activated (0 to 1) and ref 1 and ref 2 were deactivated (1 to 0).

At the corresponding time in the get_signal(K1) data the potassium concentration of the selected cell increases dramatically.

We decided to output the single_dump data and get_signal(K1) data to the same csv file as this will aid data analysis. This was done by editing the code as follows:

```
p.s>=1 & selected:  
{  
    fprintf(fa,s_get_signal(k1));//att  
    dump_single(fa);  
    fprintf(fb,s_get_signal(k1), "\n");  
    p.s:=0  
}
```

This csv format will be used to store data for the entirety of the lab.

concentra	time	id	x	y	theta	volume	gt_inst	gfp	rfp	yfp	cfp	St	ref	ref1	ref2	st1	st2
-1	7.599995	192	-561.112122	-183.243698	3.430980	2.245425	153.404083	0	0	0	0	0	0	1	1	0	0
-1	7.599995	512	-576.441956	-260.296082	1.585095	1.448493	279.085480	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1338	-708.770325	-302.514038	4.892122	2.079102	184.873062	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1339	-727.713440	-211.094513	-0.123156	2.337281	157.383896	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1170	-748.797058	-225.874954	2.973242	1.868771	213.466583	0	0	0	0	0	0	1	1	0	0
-1	7.599995	829	-757.410400	-196.823929	6.202288	1.800346	212.033875	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1291	-701.917419	-326.082672	1.288998	1.786301	230.769089	0	0	0	0	0	0	1	1	0	0
-1	7.599995	42	-647.754089	-402.831360	0.053328	1.472201	287.855377	0	0	0	0	0	0	1	1	0	0
-1	7.599995	200	-627.560791	-213.881165	5.099823	1.452703	287.187744	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1354	-582.460083	-446.735413	0.126767	2.436097	145.181931	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1349	-612.179932	-340.522217	4.255225	1.910643	212.623032	0	0	0	0	0	0	1	1	0	0
-1	7.599995	113	-588.498169	-391.119873	1.757727	2.517464	131.766693	0	0	0	0	0	0	1	1	0	0
-1	7.599995	954	-723.809265	-261.933868	0.303660	1.807035	226.585007	0	0	0	0	0	0	1	1	0	0
-1	7.599995	731	-587.012024	-311.515228	1.427707	2.605933	124.827995	0	0	0	0	0	0	1	1	0	0
-1	7.599995	214	-689.822876	-252.632889	4.706070	2.024315	194.448822	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1282	-655.269287	-311.214874	2.401533	1.916545	196.984955	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1263	-638.513550	-297.436676	0.713958	2.478935	135.210754	0	0	0	0	0	0	1	1	0	0
-1	7.599995	225	-616.954285	-326.090424	4.901303	1.528066	267.434662	0	0	0	0	0	0	1	1	0	0
-1	7.599995	1341	-718.086426	-201.867538	2.490907	2.228594	171.298294	0	0	0	0	0	0	1	1	0	0

We have decided to spend some time analysing the .gro file, identifying the parameters used in the code and their relation to the simulation will better allow us to manipulate the simulation and perform analysis.

MODULE PARAMETERS

dt: timestep of simulation[minutes]

population-max: limit on number of bacteria

t: initial time

i: ????

ecoli_growth_rate: rate of cell growth

nocell: no. of cells at the start

phi: $2\pi/nocell$

SET ENVIRONMENTAL SIGNALS

signals: 0.0=off, 1.0=on

signals_draw: 0.0=off, 1.0=on [if off then released potassium not drawn in pink]

grid:("continuous", "gro_original", 20,20,1);

k1 := s_signal([kdiff := 0.4, kdeg := 0.07]);

Signal identified by K1

Parameters of K1:

kdiff: diffusion coefficient

kdeg: degradation coefficient

GENETIC ENTITIES OPERONS

(Not relevant right now)

ACTIONS

s_get_QS: Senses a signal and if a comparison is successful then a genetic element protein is activated, otherwise deactivated

1. If potassium>threshold(0.8) & St is absent: Activates gene st1 "stress"
2. If potassium>threshold(0.9): Transcription factor deactivated
3. If potassium<threshold(1.1) & St is absent: Activates ref1 "refractory"
4. If potassium<threshold(1.0): Activates ref 2 "refractory"

s_emit signal: Emits a signal with a given concentration:

1. If ST is present, emit potassium at concentration(0.3) under the entire area of the bacterium

Saving data

Route 1: Location of CSV files

(Cells must be manually selected to save data.)

If a cell is selected:

fa=allcells1.csv dump_single

fb=allcells11.csv s_get_signal(k1) -Concentration present in the selected cell

We will a single .csv file which includes the collated dump_single and s_get_signal data. Eg:
titles: concentration, id, x,y etc.....

Saving images

We must also come up with some method to save images of the simulations. Screenshots can be saved automatically to a selected directory. We implement the following code to do this.

Route 2: Location of images

t2 > 3 : (every 3 minutes)

(save an image as):

snapshot (tostring(t1) <> tostring(n) <> route2 <> ".tif"),

n := n + 1,

t2 := 0 (resets to zero to maintain loop, every 3)

```
105 //snapshot ( route2 <> tostring(n) <> ".tif" ) used to save out timelapse images of simulation
106 t2 > 3 :
107 {
108 snapshot [ route2 <> tostring(t1) <> "min" <> ".tif" ],
109 n := n + 1,
110 t2 := 0
111 }
```

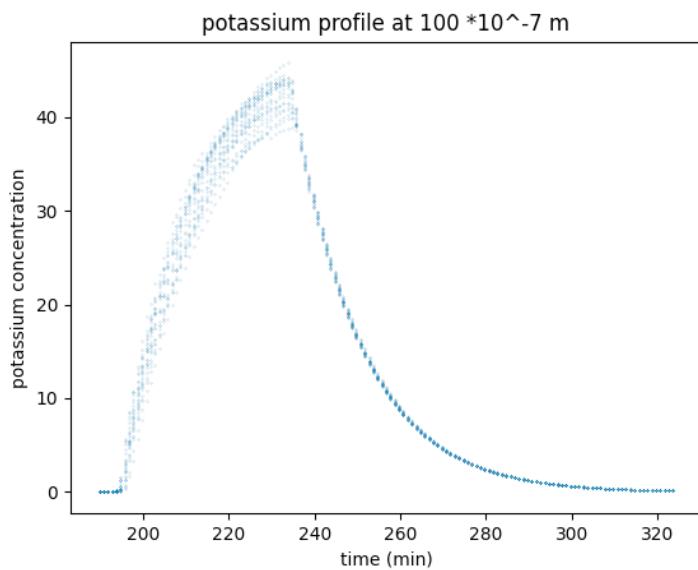
As a first test to explore the potassium profile. We explore the potassium concentration variation over time.

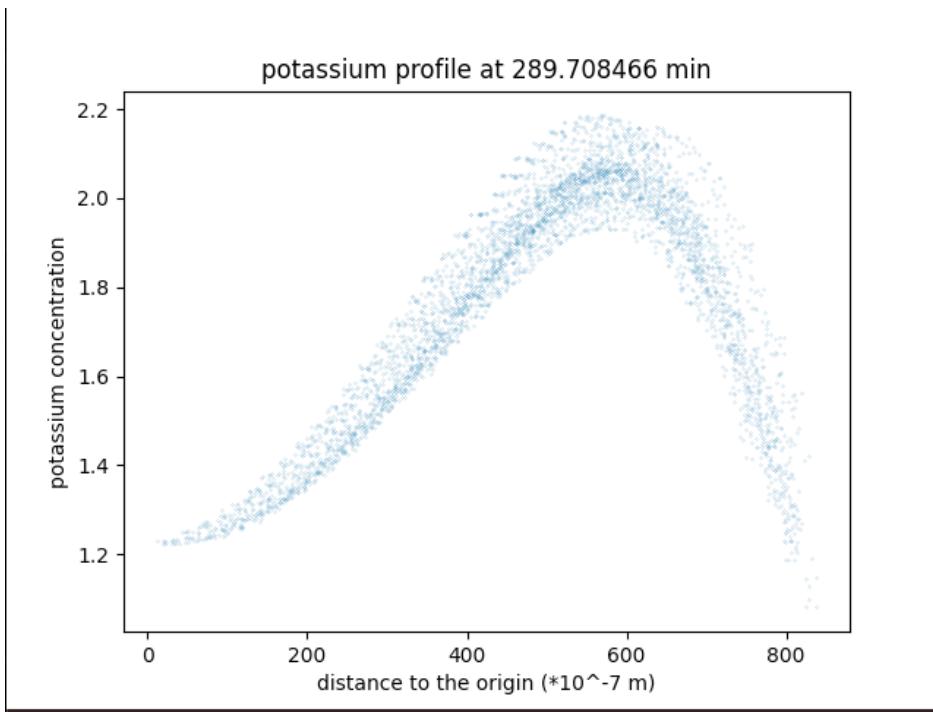
We used these original parameters in our .gro simulation:

```
include gro
set ( "dt", 0.1 ); // timestep of simulation
set ( "population_max", 2000000 );
t := 0;
i := 0;
set ( "ecoli_growth_rate", 0.0034);
nocell:=50;
phi:= (2*pi)/nocell;

//Cells signals settings
set ("signals", 1.0); //0.0: off, 1.0: on
set ("signals_draw", 1.0);
grid("continuous", "gro_original", 20,20,1);
k1 := s_signal([kdiff := 0.4, kdeg := 0.07]);
```

Coding in python to read in csv data we plot the following graphs





We continue to analyse the code for the simulation, essentially using Theresa's paper to guide us in commenting sections of the code.

```

program p() :=
{
    p := [ t := 0, s := 0 ];      initial state

    true : {
        | p.t := p.t + dt,
        | p.s := p.s + dt
    }

    p.s>=1 & selected:
    [
        | fprintf(fa,s_get_signal(k1), ","); //att
        | dump_single(fa);
        | fprintf(fb,s_get_signal(k1), "\n");
        | p.s:=0
    ]
};

program main() :=
{
    t1 := 0;
    t2 := 0;
    r:=0;
    n:=0;
    true:
    {
        t1 := t1+dt;
        t2 := t2+dt;
    };

    //snapshot ( route2 <> tostring(n) <> ".tif" ) used to save out timelapse images of simulation
    t2 > 3 :
    {
        snapshot ( route2 <> tostring(t1) <> "min" <> ".tif" ),
        n := n + 1,
        t2 := 0
    }
}

t1> 190 & t1<193:{      arbitrary signal between 191-193 min
    | s_set_signal(k1,3,0,0)
};

t1>193:{                ("fire seed")
    | t1:=0;
    |
    | 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());
};

}

```

time evolution

sense signal k1's value and output .csv file

time evolution;
t1 used for potassium trigger,
t2 used for snapshot.

output a snapshot to route2 every 3 minutes

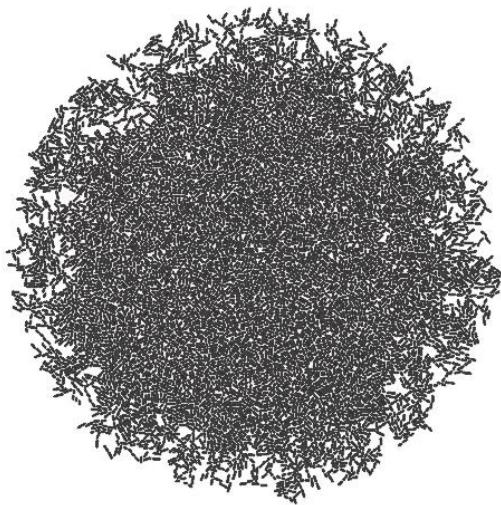
arbitrary signal between 191-193 min
("fire seed")

???

The code in question marks “???", indicate that when t1 increases above a threshold values, 193/s in this case. We set off the initial “potassium trigger”. However we see that t1 is then set to zero, implying that at a global time of 380 there will be another trigger in the centre. We ran a simulation to test this hypothesis.

Cells: 7202, Max: 2000000, t = 361.41 min

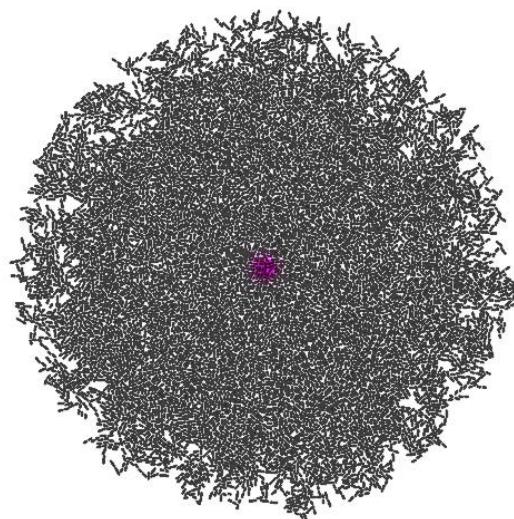
+



Cells: 8013, Max: 2000000, t = 385.51 min

+

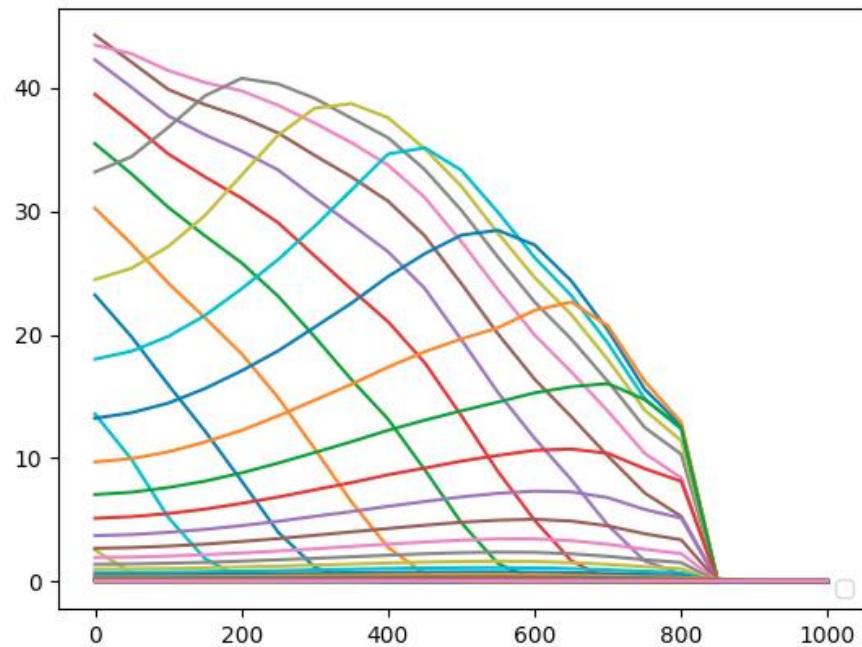
+



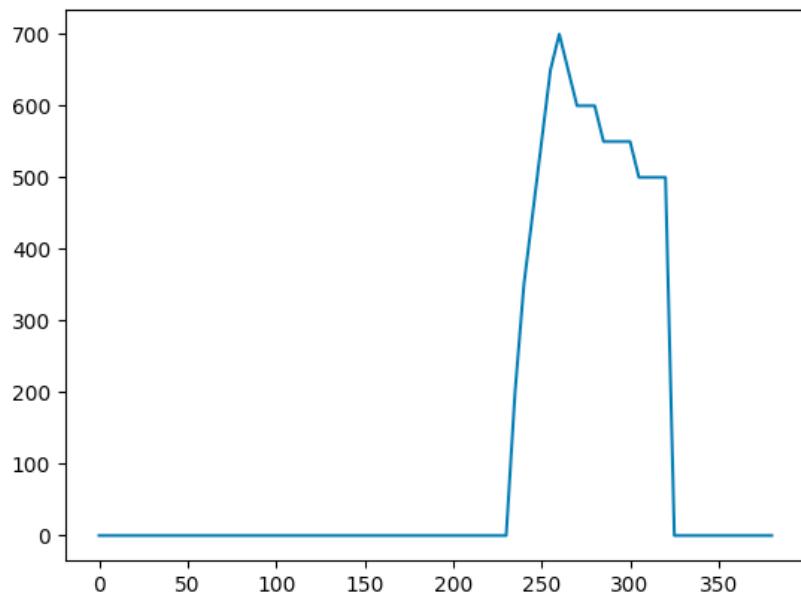
As expected there is a potassium trigger at the centre of the biofilm at 385/s.

We then decided to measure the average concentration across the entire biofilm as a

To analyse the potassium wave velocity as a function of curvature, we must first define the position of the wavefront. The assumption we make here is that the point of highest concentration can represent the wavefront. Although this is not strictly true as there may be potassium further forward than the peak, the peak is a point of constant phase that we can easily identify across multiple simulations.

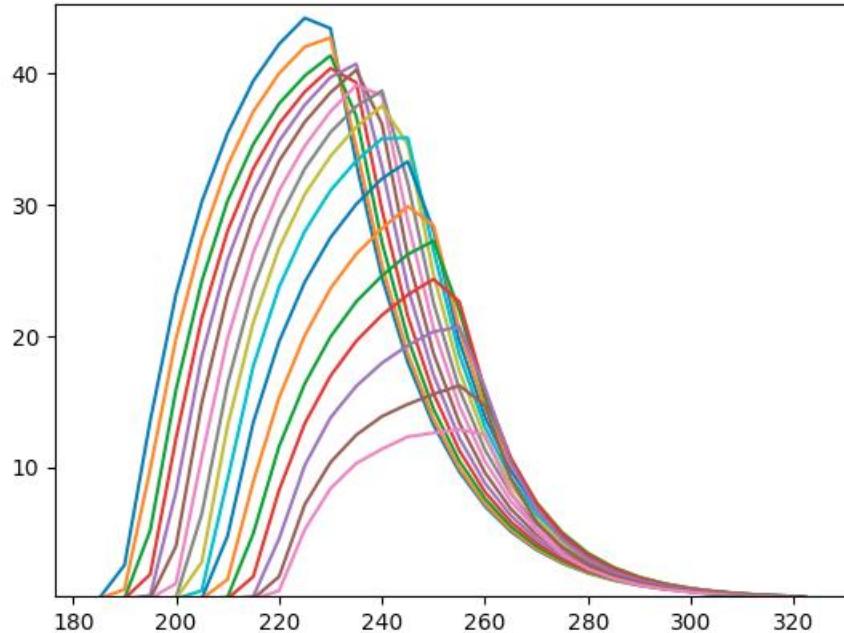


Average concentration versus radius at different fixed times.

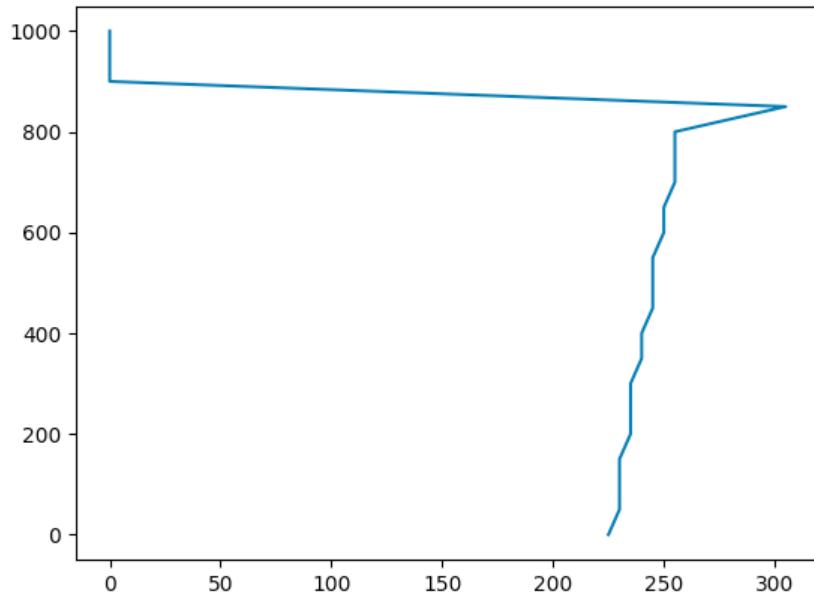


Position of peak versus time.

We produced a graph showing the mean potassium concentration at fixed radius over time. A radius of 15(micrometres) refers to an average of all bacteria that are within 10 and 20 micrometres of the origin. This was repeated for multiple radii and the results were plotted as follows. This radial “banding” is done as if we were to only take measurements when the radius $r=\sqrt{x^2+y^2}$ is an exact value we would receive no data. This banding/binning is our largest source of uncertainty throughout the experiment.



Average concentration versus time at different radii.

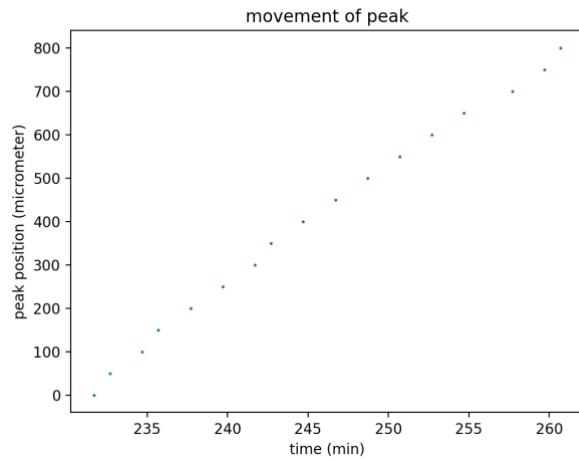


Position of peak versus time. Not a great graph, we need to investigate this.

All the graphs above are quite messy, we opted to make them smoother.

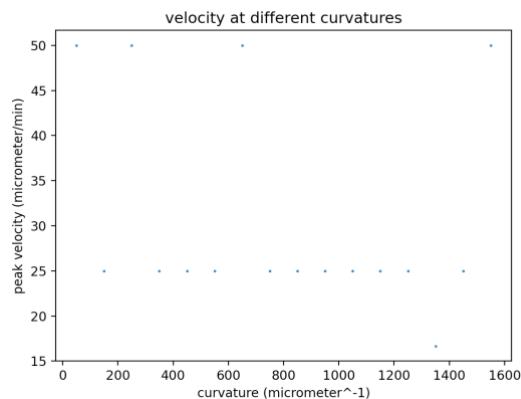
By narrowing the range of average and deleting the empty points, we obtained better graphs.

We have written code to find the position of the peak and the time. It essentially takes the max at every radial concentration.



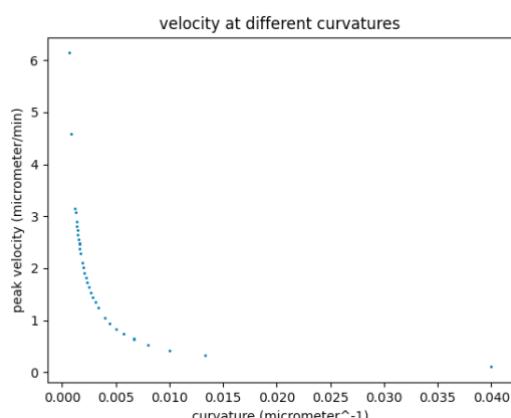
Here we get a linear relationship. However, note that the time starts a long time after the emission of the potassium at the centre of the biofilm. We must change the resolution in order to analyse the movement at low radii

15/04/2021

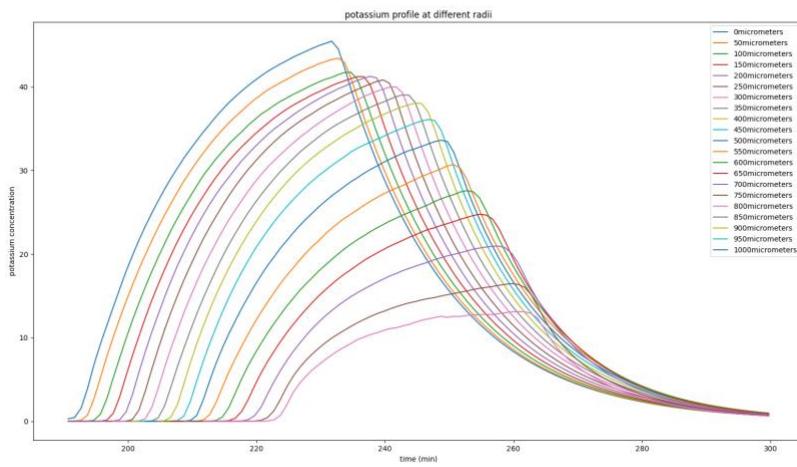


This graph is plotted using a differential velocity method. It does not agree with any power law and the data appears to be very highly uncorrelated. This is possibly due to an error in the code, more likely the uncertainty in this measurement is very large.

Repeating using the average velocity, Position



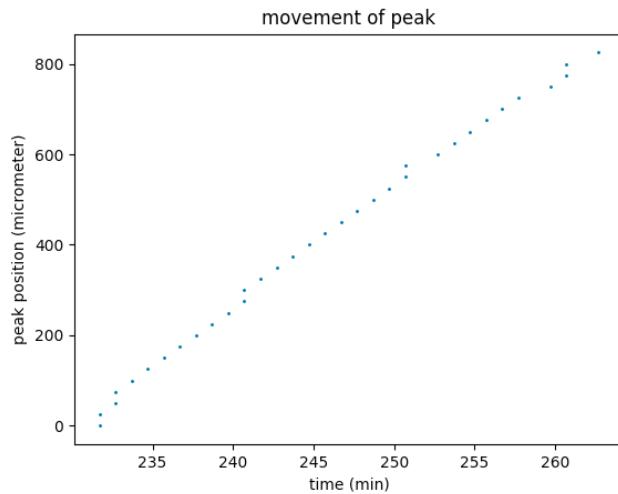
This velocity curvature graph appears to follow a power law as suggested in papers.



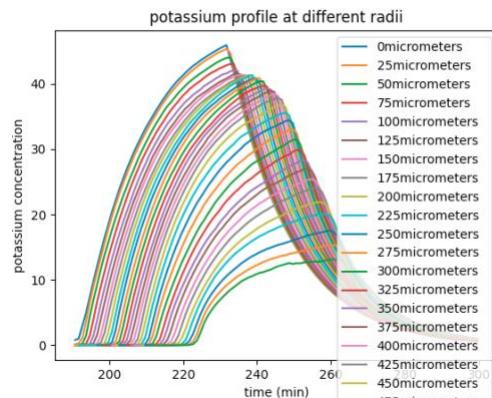
This is another graph indicating the potassium concentration at multiple radii as a function of time. Here, we note the decrease in the max concentration as the radius increases.

Below is an example run using profileAndPeak.py.

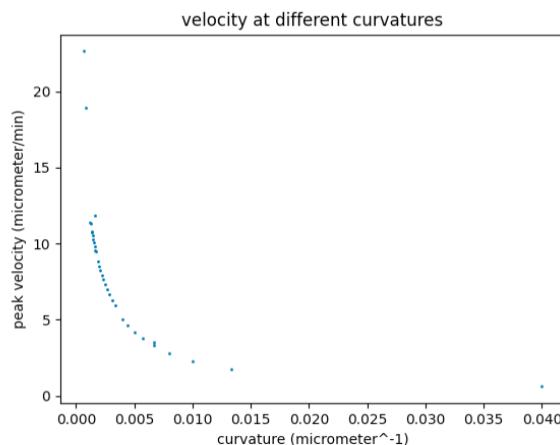
csv name=original_processed,dt=0.1min, dr=25micrometer:



In our calculation average concentration at (r,t) refers to the average concentration of all the data points that fall in the region $(r-r+dr, t-t+dt)$.



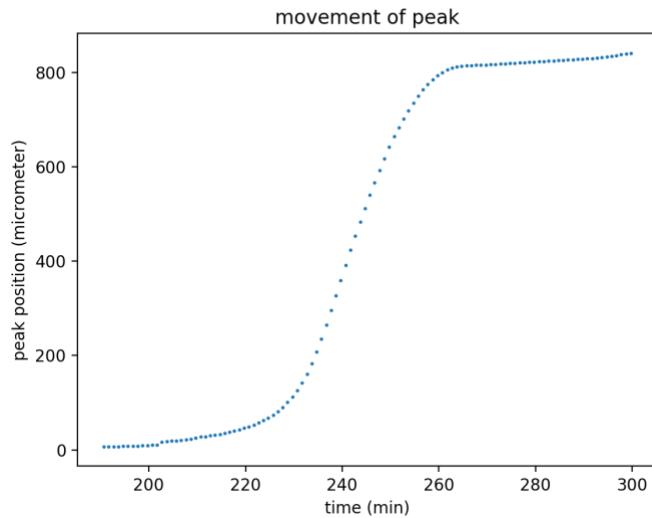
“Velocity” here is the peak’s average velocity since firing (assuming the peak starts at the origin upon firing).



At this point we have a decent baseline understanding of how to analyse the data and the skeleton of the code we will use to conduct a thorough analysis.

By using small time-step and radial-step($dt=0.05\text{min}$, $dr=0.05\text{micrometer}$), we obtain the average position-time graph of the peak.

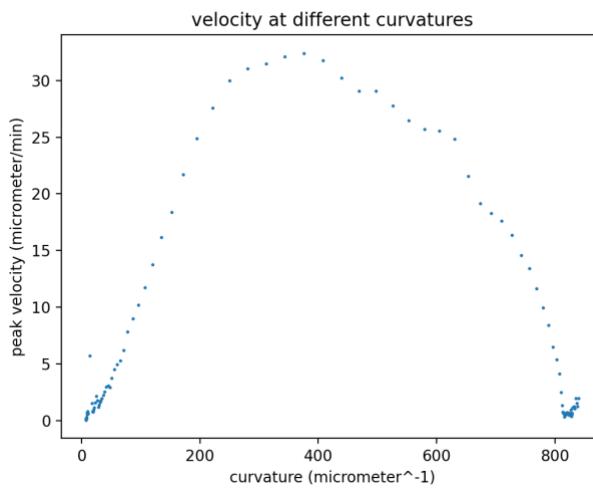
(Here we see multiple peak radii at the same time. This is due to the resolution of the simulation. Here So we take an average of these values to find the peak position)



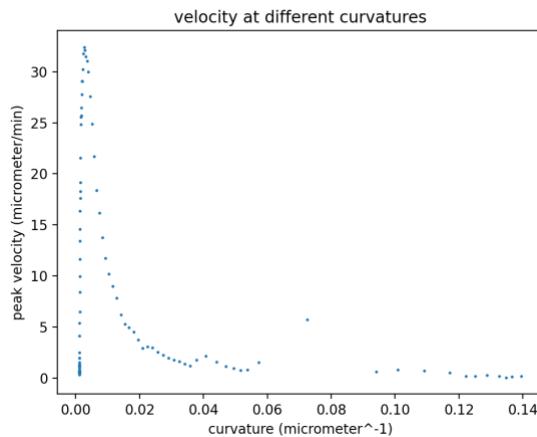
3 phase shape. Left and right similar to parts of different gaussians. What about the middle one?

Broken of symmetry may be the result of following factors:

1. Changes in cell density. (reference points at 200, 500)
2. Changes of curvature (including the corresponding difference in the phase of diffusion/cell-trigger/degradation—diffusion from a larger curvature to a smaller one; cell trigger is obviously a “one side” process and so is degradation)
3. Changes of “outside distribution” - spatial distribution of cells outside the radius



*here the x-axis is radius rather than curvature



Rather similar to the pair correlation function?

Before next week:

1. Read cardiac calcium ion paper.
2. Sort the current code.

Plan for next week:

1. Produce graphs on a more neat basis.
2. Fit the movement curve.
3. Test the similarity between velocity curve and pair correlation curve.
4. Fit concentration profiles to gaussians(2part-wise).
5. Try different parameters and see their influence on the movement of the peak.
6. Error analysis for contents obtained so far.

-How was the peak found?

With the potassium profile at a given radius we can find the peak point by simply picking the point with largest concentration.

Taking the time of the point and radius of the considered profile, we obtain a space-time position of the peak, which, if we take the peak as the reference point, is a data point of the whole space-time trajectory of the potassium wave.

-How did we get average velocity?

We have multiple radii. Each radius has a peak concentration at a given time.

First we calculate the average position of the wave at each time point of the grid.

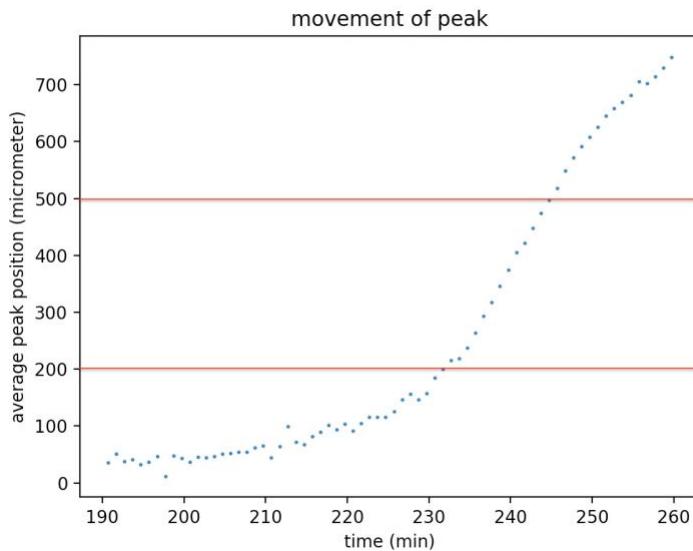
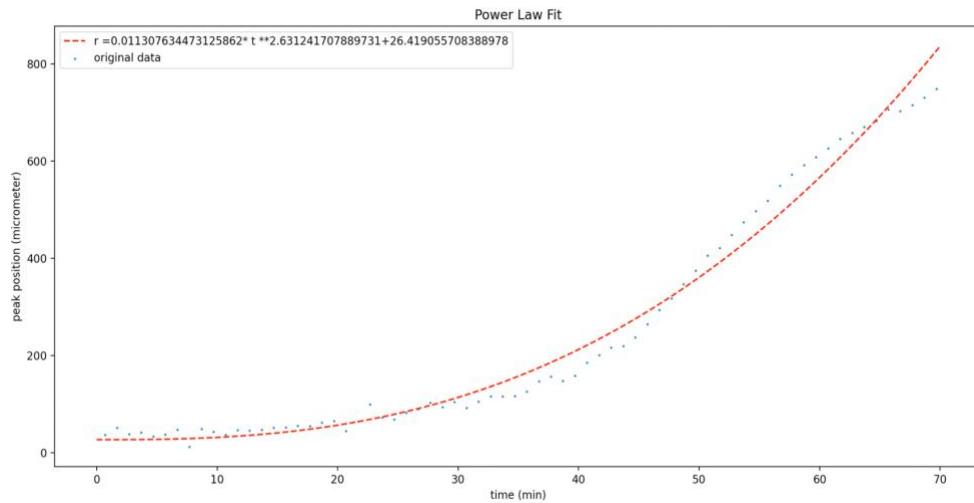
Assuming constant velocity between neighboring time points, the velocity is regarded as the change in average position over time period, the corresponding radius is regarded as the midpoint during the small movement.

20/04/2021

After running some basic simulations, we see the simulation essentially breakdown at radii of about 800 micrometres, as the wave reaches the edge of the biofilm. To avoid this corrupting the analysis we remove this data.

(We utilise curve_fit() from scipy.optimize to fit the curves. I believe this uses a least squares regression.)

After that we tried to fit the movement curve to a power law, yet the result wasn't satisfying.
(Power law used here is $r = at^b + c$.)



There is clearly three different regions. In the code the cell density varies in different radial bands(0-200, 200-500, 500-80), so we assume the movement may experience 3 different phases as a result. So we may attempt to fit them separately to different power laws.

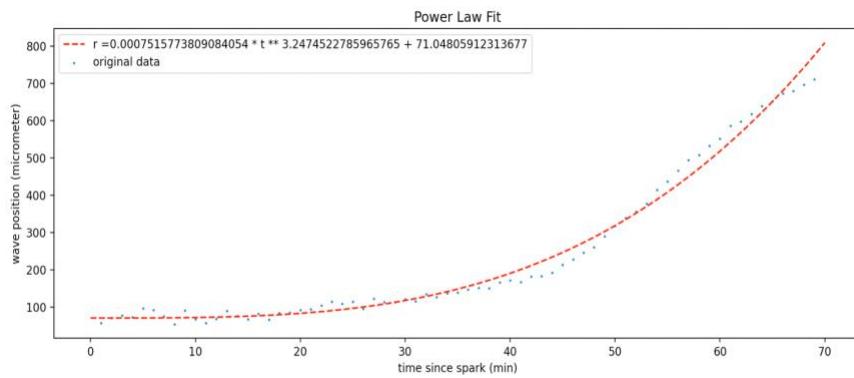
To test the speculation mentioned above, we ran a “homogenous” case, in which everything stayed the same except for now the cell density should be (on average) constant.

```

117   t1>193:{  
118     t1:=0;}  
119  
120     c_ecolis(3000, 0, 0, 800,{"p3","p2"},program p());  
121  
122   };  
123

```

Peak position curve fitted to a power law.(Power law used here is $r = at^b + c$.)



(3000 cells, size 800)

Despite this new homogenous setup the 3-phase position pattern still occurs.
Therefore a new speculation was come up:

The Lag hypothesis

The cross-membrane transportation of potassium ions takes a noticeable amount of time to happen, which suggests the activation and release of potassium as well as the accumulation of the peak concentration do not happen immediately, therefore in the fire-diffuse-fire model the following occur:

- i) The position of potassium peak is actually behind that of the most recently activated cell
- ii) The peak's formation is also affected by the “forward” cells’ potassium release.

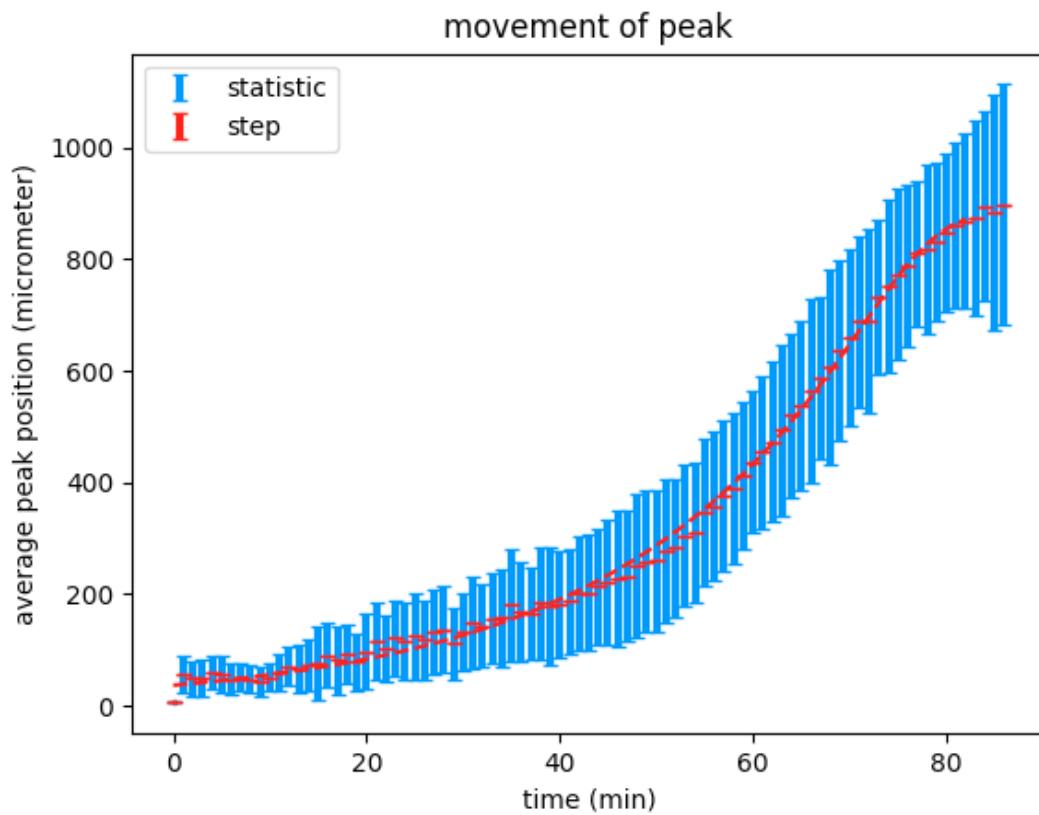
And with above statements we explain the three-phase character as below:

- 1.The starting phase – the first set of cells were arbitrarily activated, due to the lack of “backward” cells and the potassium accumulation, the wave moves slowly.
- 2.The stable phase - the system reaches a balance of potassium accumulation, backward cells and forward cells, peak position moves at a constant velocity.

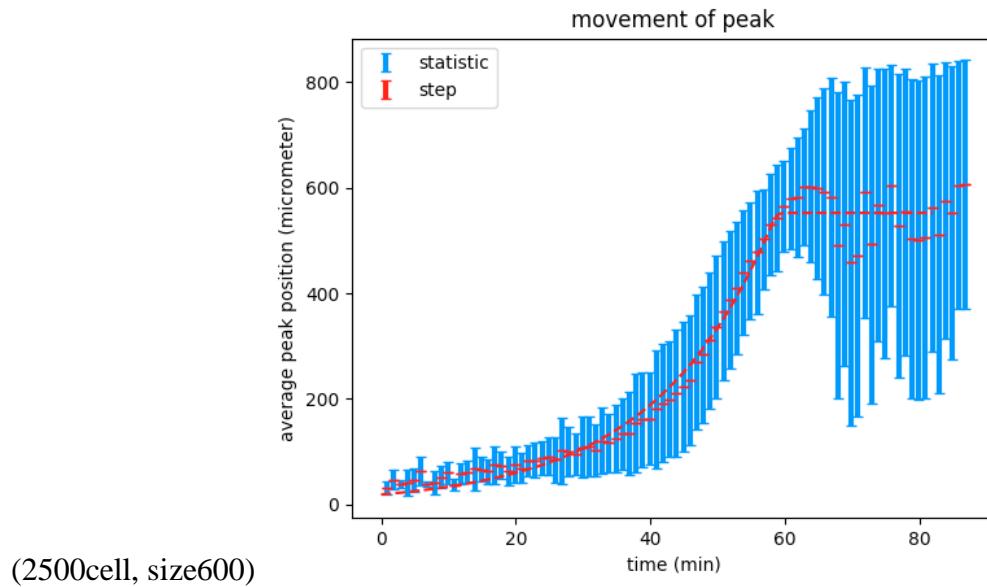
3. The decaying phase - as the peak approaches the edge of the colony, due to the lack of forward cell and loss of potassium (diffused out of the colony), the balance is broken and the velocity decreases.

Forward and backward here refer to cells in front of and behind the max potassium concentration point.

We will run simulations of homogeneous colonies of different sizes to test this speculation.
(3600cell, size1000)



(3000cell, size900)



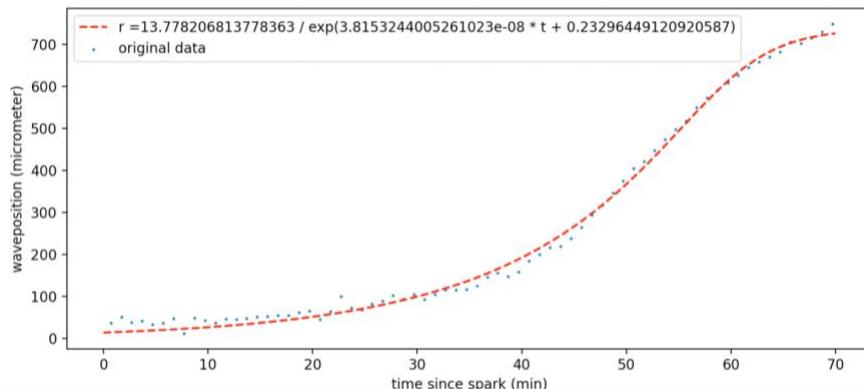
All cases show 3-phase character, which supports our hypothesis. This isn't due to the density, the three phase density is merely a coincidence/.

The phase of interest is the second one, we are interested in the relation between the velocity and the diffusion and degradation of potassium.

Now we return to original case.

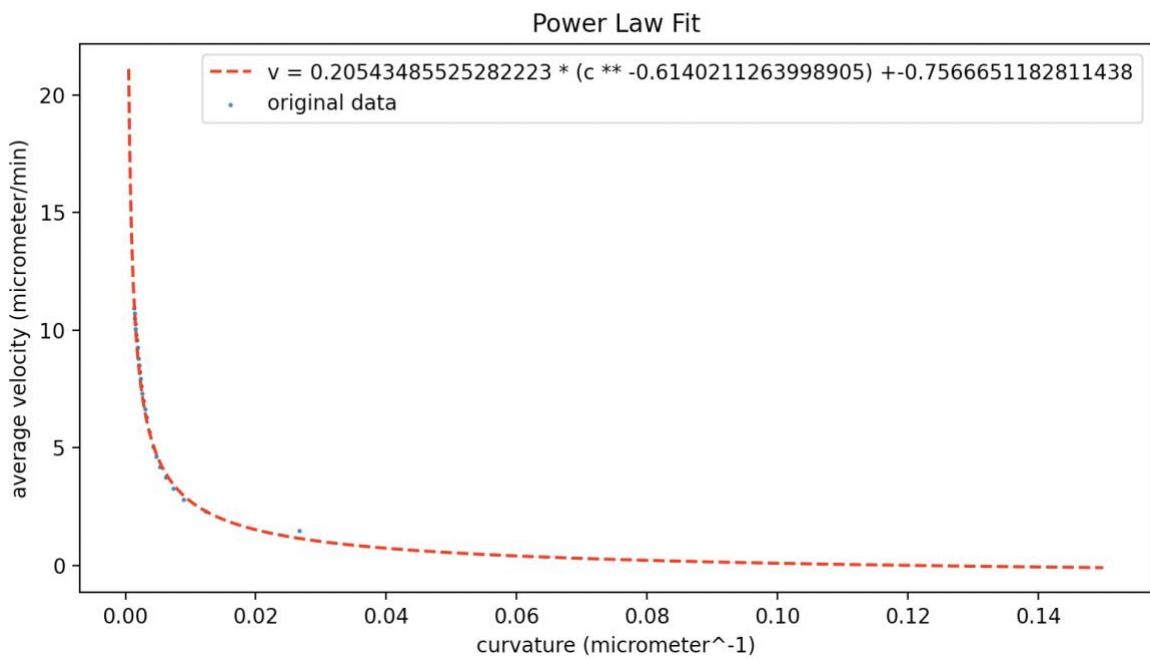
To do

- 1.fit potassium profiles to gaussians and determine the errors(?)
- 2.fit the average velocity versus curvature curve and determine the errors (systematic: randomness;discreteness; Statistic?)



Fitted to modified sigmoid function(dt=0.05min,dr=0.05micrometer).

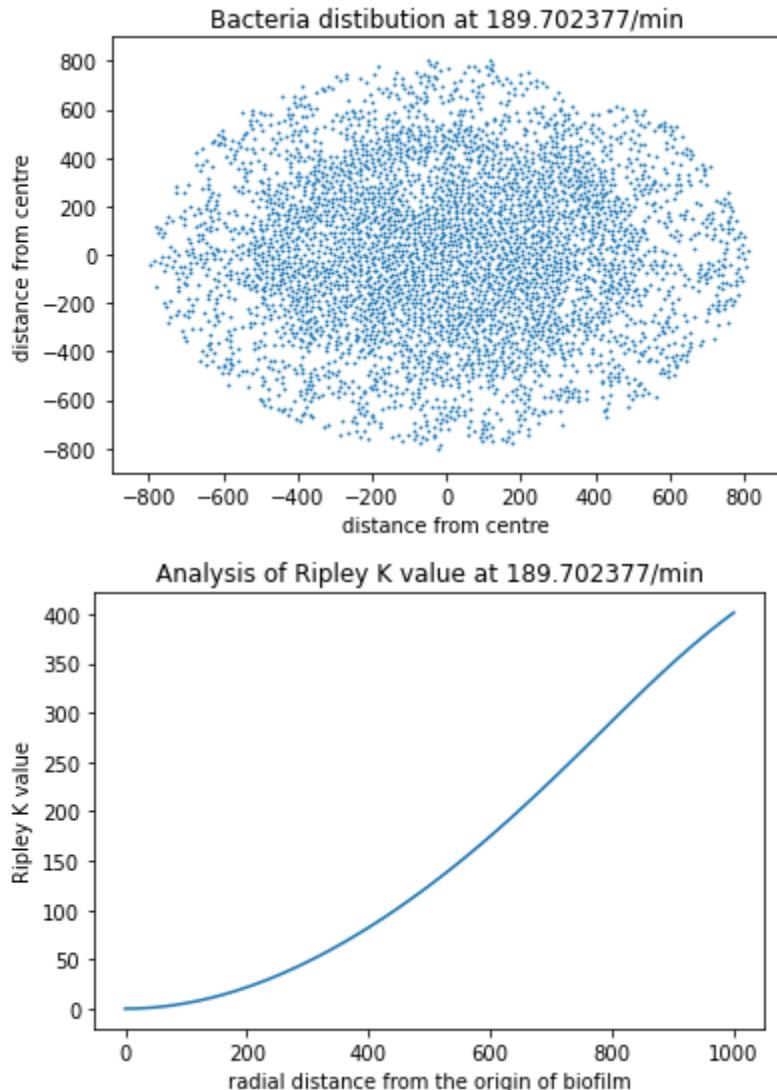
$$\text{(Sigmoid function used here is } r = \frac{a}{e^{bt} + c}.)$$



Average velocity - fits well to a power law(dt=0.1min, dr=25micrometer).
 (Power law used here is $v = lc^m + n.$)

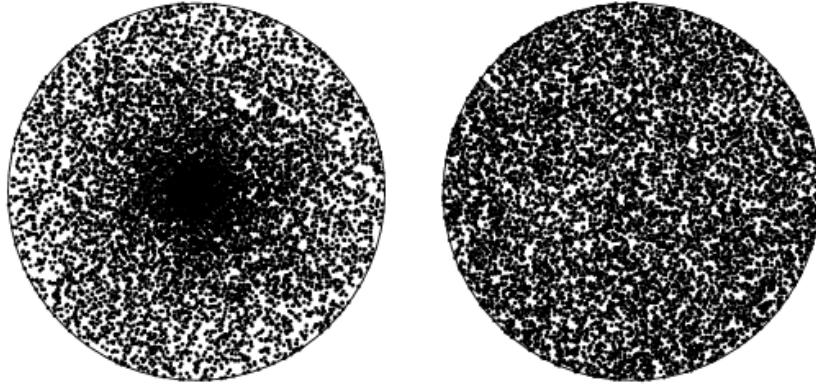
Spatial analysis of the bacterial distribution

Using astropy.stats from astropy libraries we import Ripley's K Estimator. We then wrote a python program to import the spatial coordinates of the biofilm at a selected time and output a graph of the Ripley K value against radial distance from origin. The graph below was produced:



As a check of how this Ripley K value relates to a random distribution we produced a number of “false” datasets using the np.random function. We produced a set of x and y coordinates randomly distributed over the same area of study utilised for the biofilm distribution, with an identical number of coordinates.

In order to produce a randomly distributed circular distribution the following methodology was used:



To generate random points over the [unit disk](#), it is *incorrect* to use two uniformly distributed variables $r \in [0, 1]$ and $\theta \in [0, 2\pi]$ and then take

$$\begin{aligned}x &= r \cos \theta \\y &= r \sin \theta.\end{aligned}$$

Because the area element is given by

$$dA = 2\pi r dr,$$

this gives a concentration of points in the center (left figure above).

The correct transformation is instead given by

$$\begin{aligned}x &= \sqrt{r} \cos \theta \\y &= \sqrt{r} \sin \theta\end{aligned}$$

The code reads the simulated data coordinates at a selected snapshot in time. It then makes a random uniform distribution of the same number of coordinates within the same maximum radius. It plots both distributions and then calculates the Ripley K value as a function of radial distance from each point. The search area is automatically set to the area of the biofilm and the radius of search is set to the max radius of the distribution. By setting it this way the ripley K values can be calculated and compared with a random distribution of the same sample size and radius for any time in the simulation.

Here we use a dataset called `from_one.csv`. Identical to the original `.gro` file except we start with a single bacterium and record the data from this start.

We normalise the Ripley K value against the randomised sample using the Besag function.

The K-function can be normalized as proposed by Besag (11) so that its expected value is r (linear):

$$L(r) = \sqrt{(K(r)/\pi)}. \quad (2)$$

From:

Biophysical Journal Volume 97 August 2009 1095–1103

On the Use of Ripley's K-Function and Its Derivatives to Analyze Domain Size

4.2 Edge-effect corrections

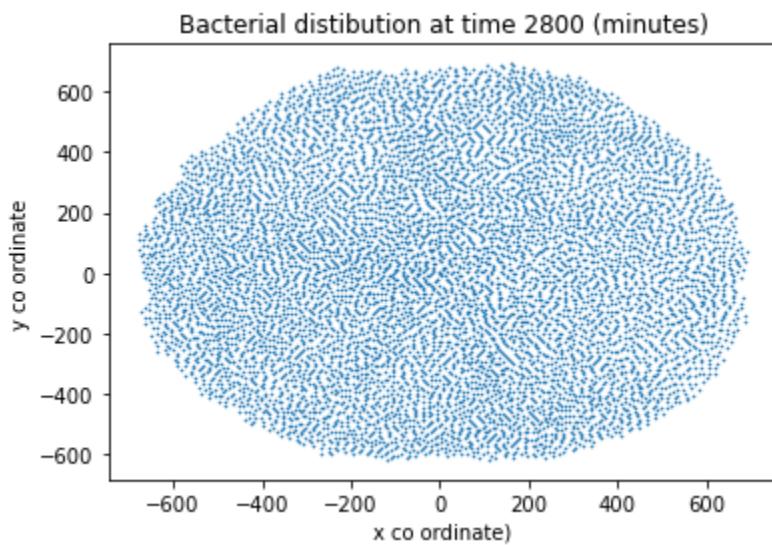
Correcting the edge effects by Ripley's method, equation (12), is impossible if a single value of L_{jr} equals 0, that is, as soon as r is big enough for a circle around a point to be completely outside the domain. If the domain is a rectangle, K 's computation is thus limited to half of its length. Diggle (1983), p.72, gave correction formulas applicable up to half of the width. Goreaud and Pélassier (1999) improved the edge-effect correction to allow computing K up to half of the rectangle's length.

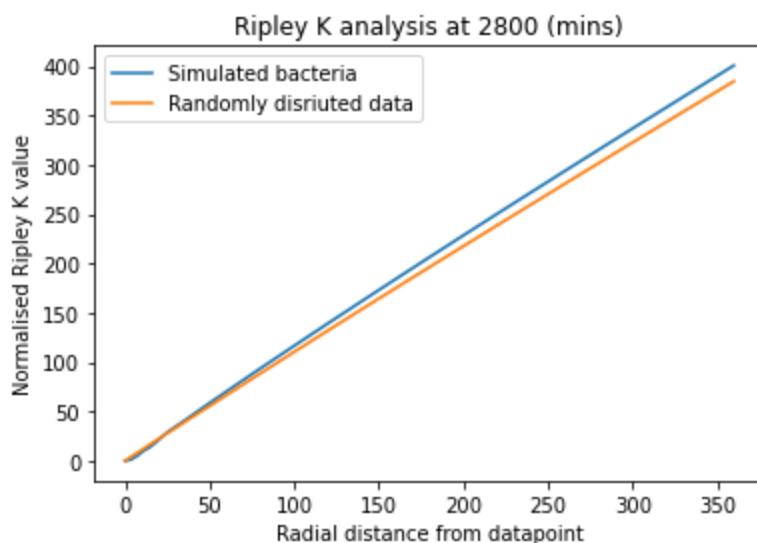
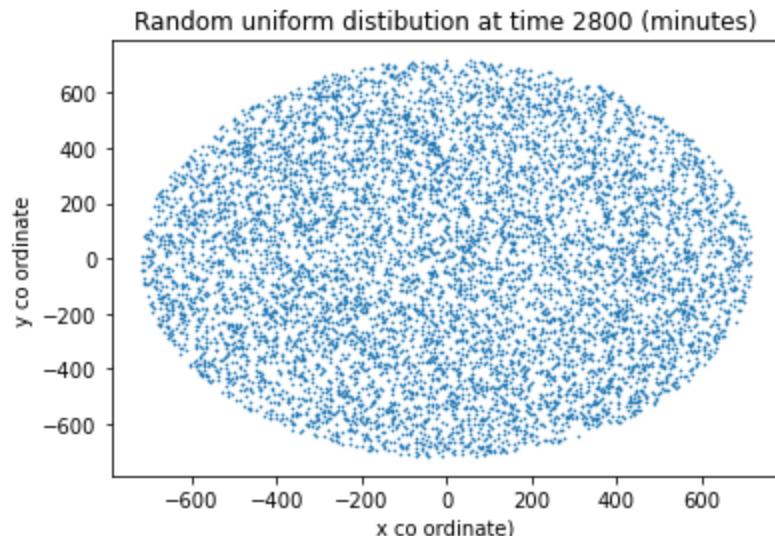
We will rather use Besag's method, equation (13), which is not limited. Detailing the density estimator, we get the expression of K , corrected from the edge effect:

$$\hat{K}(r) = \frac{A}{N^2} \sum_{i=1}^N \frac{\pi r^2}{A_{ir}} \sum_{j=1, i \neq j}^N c(i, j, r) \quad (17)$$

Eric Marcon, Florence Puech. Generalizing Ripley's K function to inhomogeneous populations. 2009. Ffhalshs-00372631f

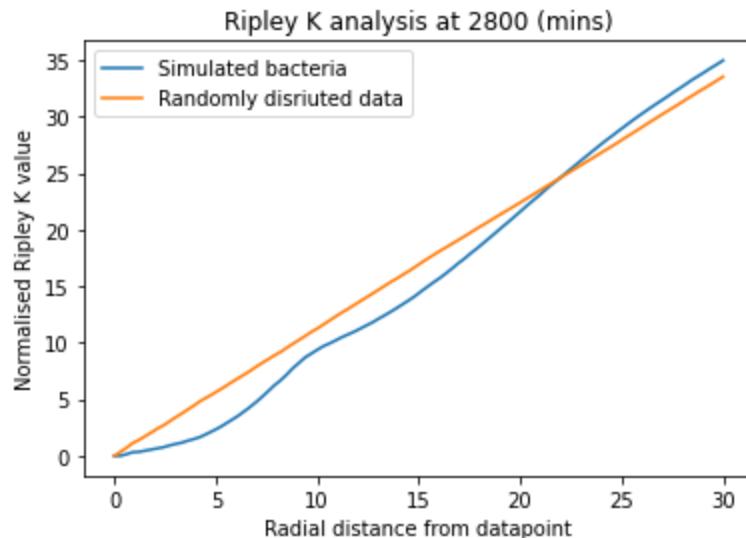
Although Astropys estimation function includes edge effects, it is clear based on Diggle and Goreaud and Pelissier that to produce meaningful results the radius used should be around $\frac{1}{2}$ of the maximum radius of the distribution, this is implemented from here onwards.





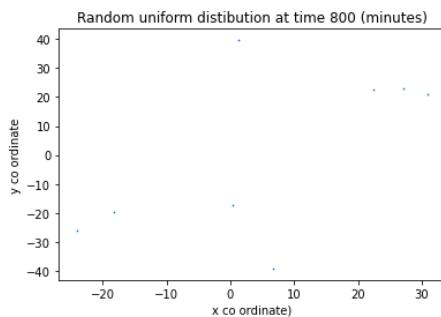
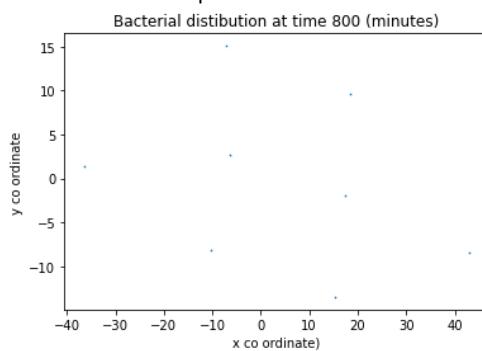
As predicted by Besags function we see a linear relation for the randomly distributed data. The simulated biofilm has a normalised ripley K value that is greater than the random. This indicated greater clustering in the simulated data compared to the null hypothesis. Interestingly at very small radii it appears that the ripley K may be smaller indicating greater dispersion at small distances from a given bacterium

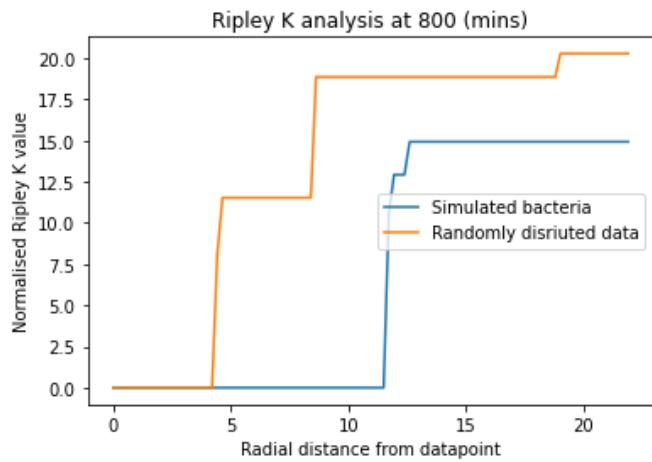
The x axis will be rescaled to view these effects at small radii from a given datapoint.



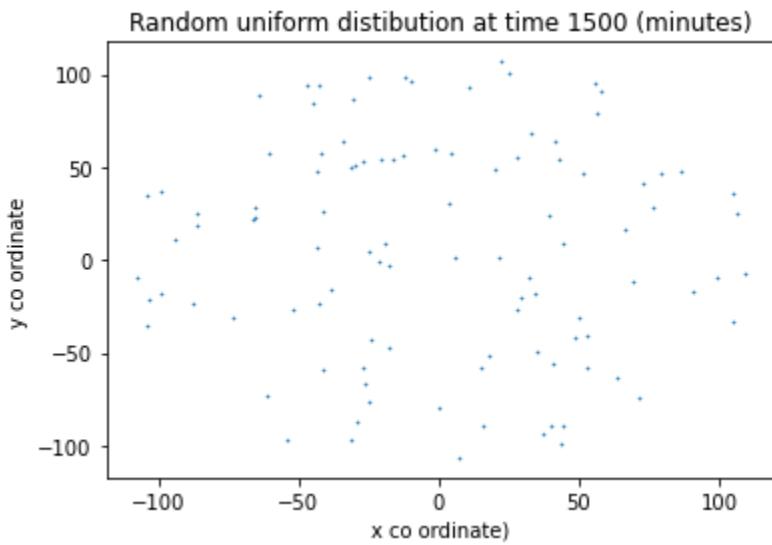
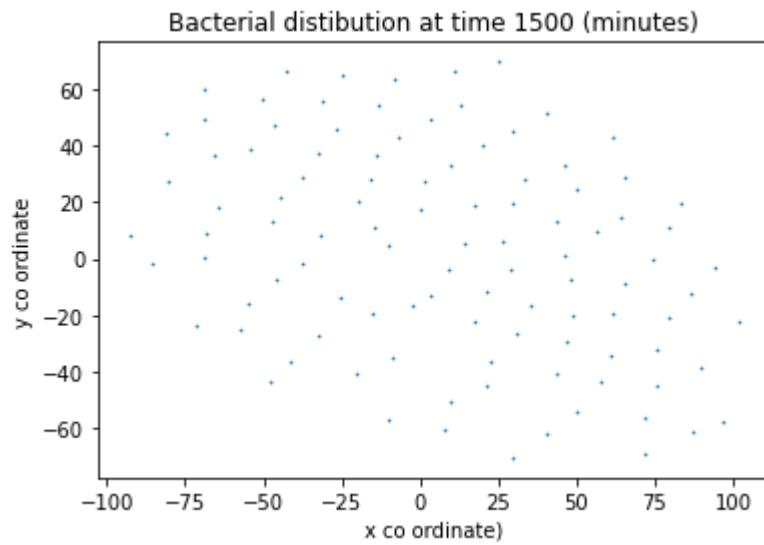
Below radial distances of about 22.5 the simulated data is more dispersed than the null hypothesis. This is likely due to the geometric limitations imposed by the dimensions of the bacteria. (eg- they cannot overlap and they have some size).

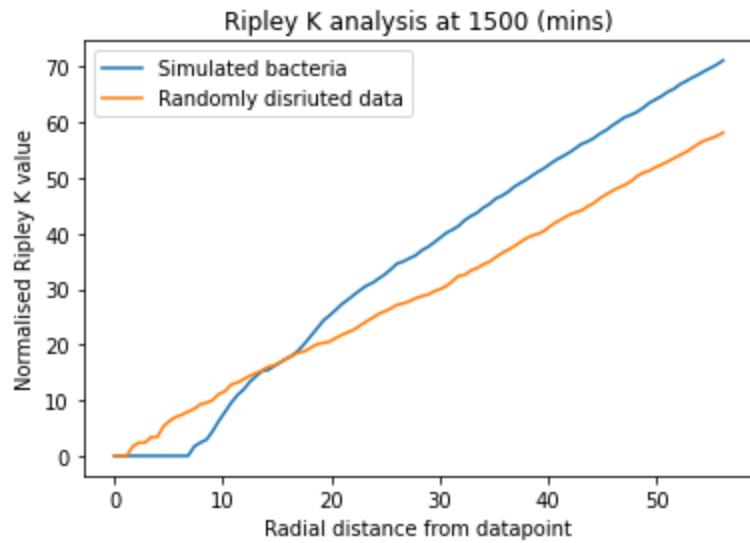
At earlier times this plot is useless



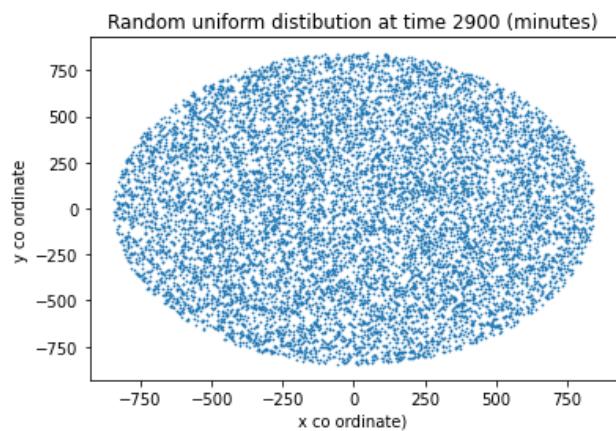
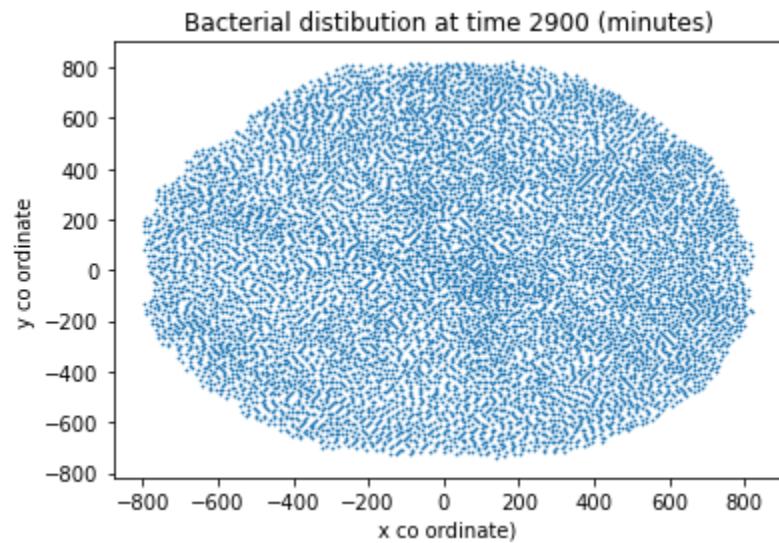


However, at slightly later times the pattern begins to appear

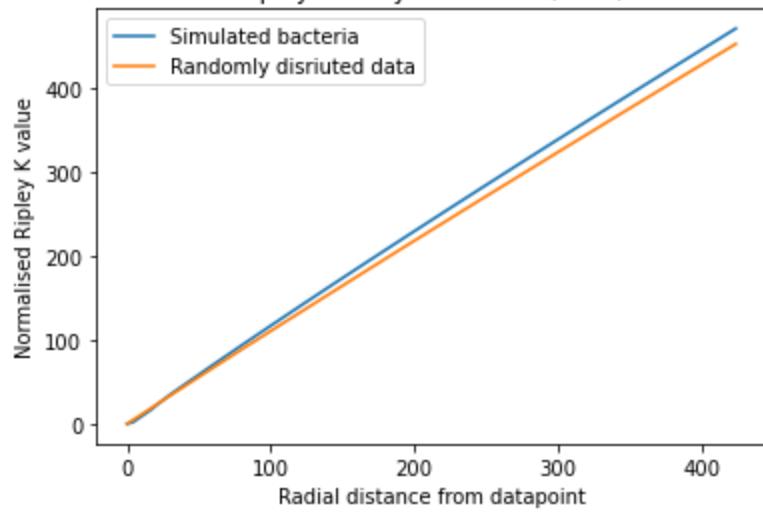




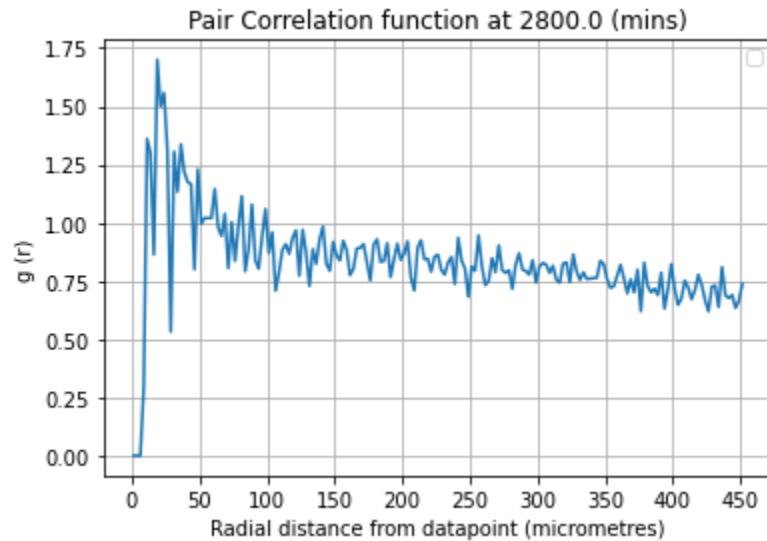
And finally once the biofilm is fully formed we get the graphs at 2900 minutes



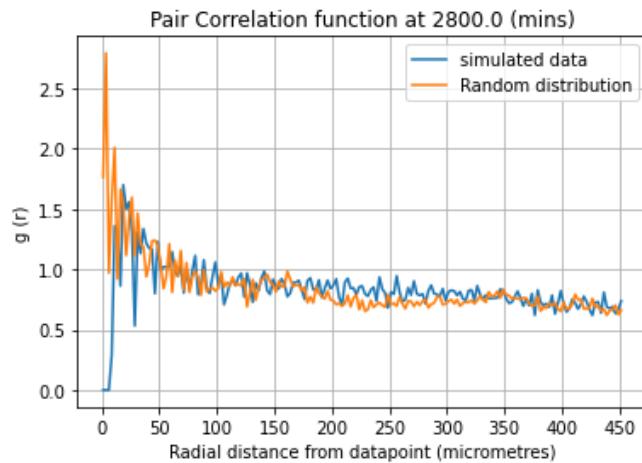
Ripley K analysis at 2900 (mins)



Another method to analyse the spatial distribution of the bacteria is the pair correlation function. In statistical mechanics, the radial distribution function, (or pair correlation function) in a system of particles (atoms, molecules, colloids, etc.), describes how density varies as a function of distance from a reference particle.



This can once again be paired with the null hypothesis



Density VS Radius

In order to analyse the density as a function of radius we create a program that at a given time records the no of cells per unit fixed area. This is done using the relationship that if we take an initial central circle dr . Then for subsequent radial rings we have boundaries given by:

$$r_n = \sqrt{n * dr^2}$$

This ensures the area of every measurement is identical.

Hence the density is n/area where n is the no_bacteria inside a radial band and A is the area $\pi * r^2$.

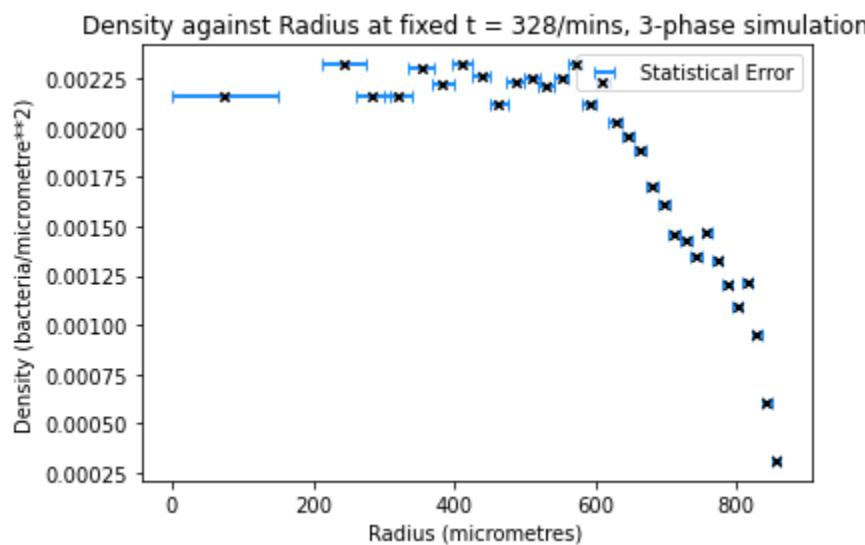
This produces an uncertainty on the radius

$$\sigma_r = \frac{r_n - r_{n-1}}{2}$$

This is used as an error bar on the plots below. We plot for our current situation with 3 phases of density, and also the case where we use a homogenous distribution.

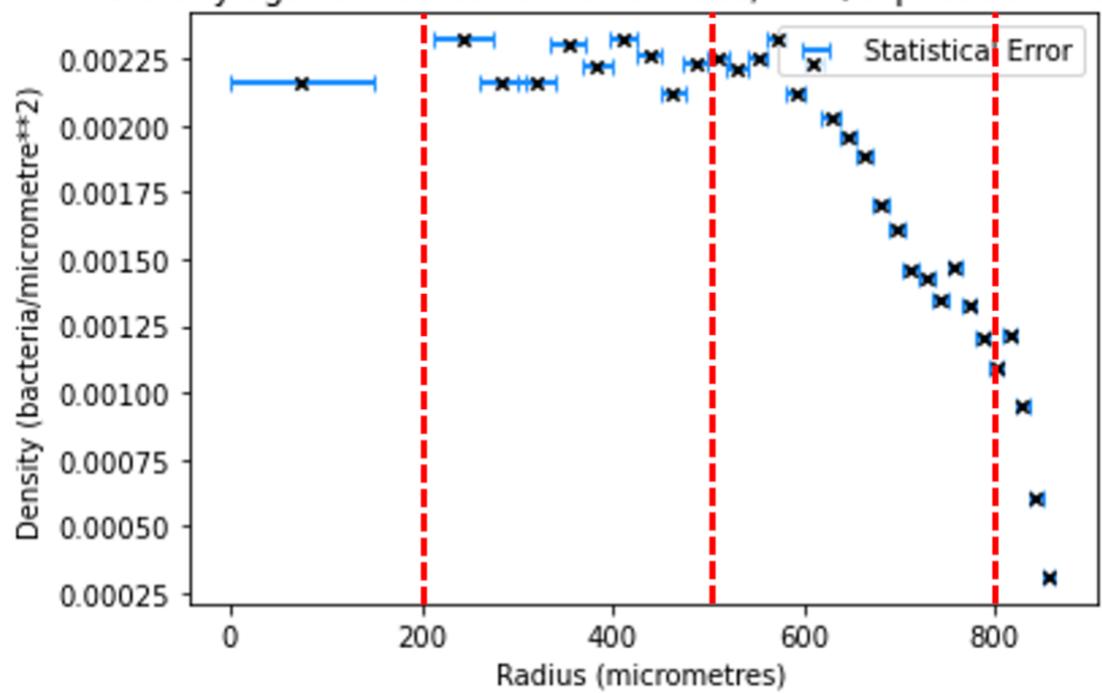
Using original.csv with dr=150

```
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());
```

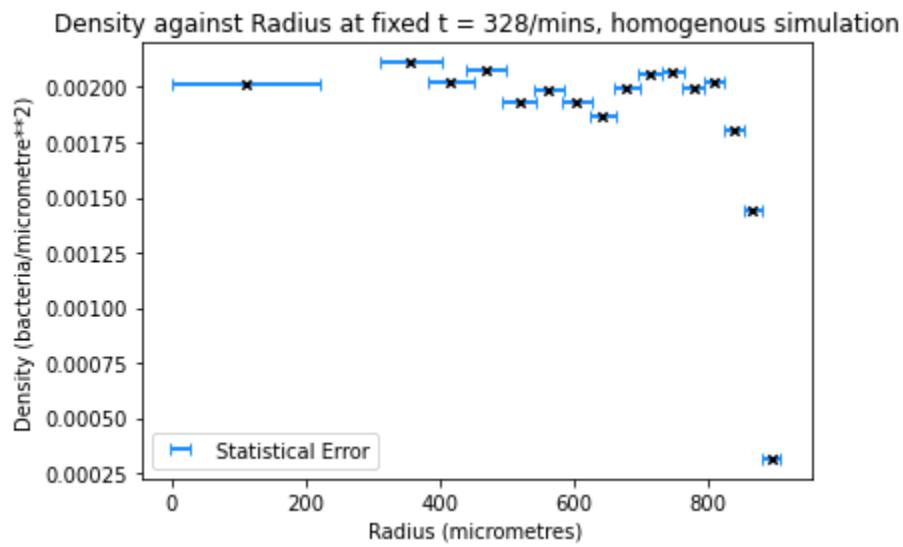


We can see these three phases most clearly in the slower drop of the density between 500 and 800 where with the homogenous case we see a more constant density with a sharp drop at the outer radii.

Density against Radius at fixed t = 328/mins, 3-phase simulation



Using Homogenous.csv with dr=220



Detailed Final Attempt

We now return to the segment of the lab analysing the peak movement, attempting to fit the proposed sigmoid and power laws using more rigorous error analysis.

1.Running Simulation

The GRO file we will use from here is ABFDFM(1)(4).gro

These settings are based on those used by Theresa Corr in her 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 start recording data for the simulation at 188mins, just before the initial potassium fire. We stop the simulation at 328.41mins, when the potassium signal has degraded and the “wavefront” has reached the boundary of the biofilm. There are 6143 cells at the end.

(Insert Screenshots/GIFS)

2.Saving Data

All the data is saved in “original.csv”. (Note: any concentration values= -1 will be treated as =0 throughout)

	time	id	x	y	theta	volume	gt_inst	gfp	rfp	yfp	cfp	St	ref	ref1	ref2	st1	st2
-1	188.002274	595	491.886993	-192.344437	2.836454	3.355165	221.245148	0	0	0	0	0	0	1	1	0	0
-1	188.002274	1405	-697.318359	27.730707	3.222908	2.896681	259.587982	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2617	218.552368	93.488586	0.422924	2.291486	187.883713	0	0	0	0	0	0	1	1	0	0
-1	188.002274	34	146.116486	641.740479	4.867264	2.138668	179.740219	0	0	0	0	0	0	1	1	0	0
-1	188.002274	1378	113.964058	348.693512	5.472344	2.755556	276.921326	0	0	0	0	0	0	1	1	0	0
-1	188.002274	1371	120.728149	-368.407257	5.837604	1.901228	220.296677	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2131	331.847382	-33.891514	5.231243	2.396458	208.419205	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2860	-414.479248	-257.141205	4.829459	2.460894	191.908951	0	0	0	0	0	0	1	1	0	0
-1	188.002274	1223	-9.015299	275.959656	0.289657	1.826313	210.129410	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3023	519.280396	-152.978592	3.441994	2.292867	207.804642	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2235	191.707642	-54.161995	1.090973	2.316342	197.770599	0	0	0	0	0	0	1	1	0	0
-1	188.002274	1293	-136.607224	212.056091	2.967083	2.341755	216.284927	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3883	458.400330	-49.262150	5.971012	2.077217	185.126648	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2207	-463.289459	-148.822449	5.377746	3.593857	199.943634	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2130	-427.571564	-86.876892	4.270166	3.335899	229.291153	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2797	642.234680	-254.297760	4.542842	2.323590	223.443192	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3312	-40.220261	-526.703430	0.783521	2.156240	200.734497	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3327	629.631592	-322.959229	2.968395	2.214930	203.273849	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3693	-230.997574	191.901245	1.824733	1.931252	219.454086	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2090	275.564880	-20.524874	3.992456	2.183991	212.552612	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2828	-311.653137	40.075775	2.175214	2.280521	220.360260	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3660	417.189026	428.984802	3.236419	2.001539	211.387695	0	0	0	0	0	0	1	1	0	0
-1	188.002274	794	502.043488	103.932083	5.091303	3.503639	214.051361	0	0	0	0	0	0	1	1	0	0
-1	188.002274	918	57.300415	287.889771	0.634361	2.075650	223.287277	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3617	291.017517	-97.706734	0.496774	2.122767	206.664185	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3075	612.686646	448.703186	0.587197	2.133169	235.484833	0	0	0	0	0	0	1	1	0	0
-1	188.002274	3388	-308.134827	310.688324	0.817700	1.965280	241.677704	0	0	0	0	0	0	1	1	0	0
-1	188.002274	2485	-40.433197	-508.001526	5.659123	2.683337	274.640717	0	0	0	0	0	0	1	1	0	0
-1	188.002274	1161	457.640228	359.316376	5.614568	3.323706	214.535019	0	0	0	0	0	0	1	1	0	0

3.Plotting avg.Concentration VS Time at fixed radii

i) Defining radius

To record concentration at fixed radius, we define the “radial band”.

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

The next “band” will start in such a way that we span the entire data set with these bands

We take the average concentration of the N points that fall within $r +/- dr$ to be the cell 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

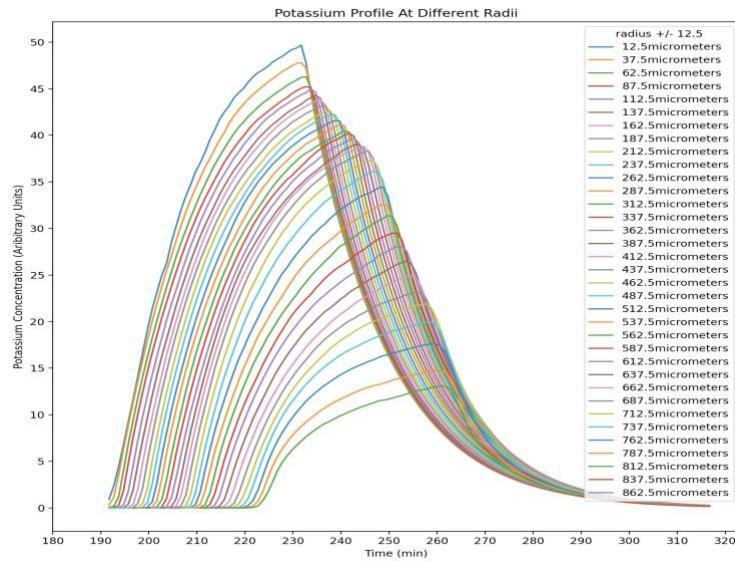
We Can add this to code for calculating the mean

- 2) There is an uncertainty on the associated values of r due to taking a range of values $r +/- dr$
In the code: $dr=0.5*grid_length$
- 3) 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 $grid_length = 25$ micrometres. So the average concentration is over all points that fall within $r +/- 12.5$

We set the value of variables $r_cut=850$ and $t_cut=380$, e.g- ignoring data when the potassium degraded or the simulation has broken down.

The following graph without associated uncertainties was produced:

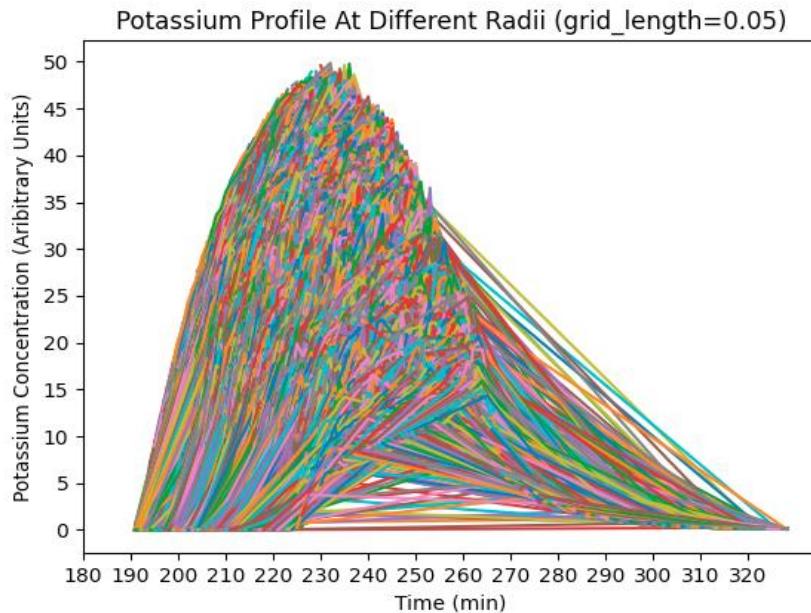


In order to achieve a higher resolution and also analyse peak at lower radii, we change `tr_grid_length=0.05 micrometer` and `t_grid_length=0.05min`.

This increased resolution 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

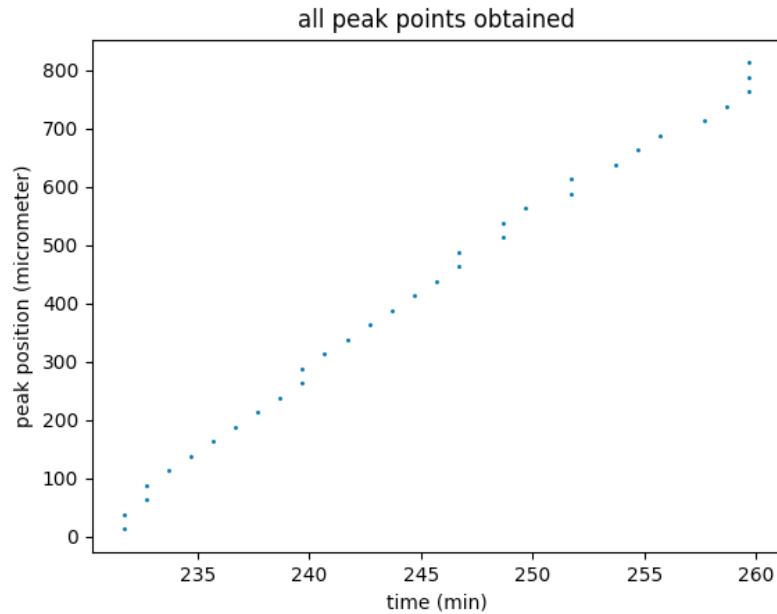
For each radial band we find the time at which the mean concentration is the largest. This data is then stored in an array.

Errors:

- 1) Is there some uncertainty on the peak point?
- 2)

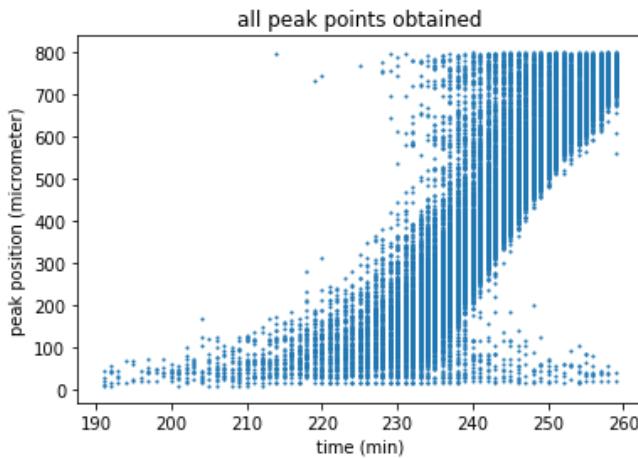
5. Plotting Peak position against Time

Plotting the peak positions against time we get the following graph
(grid_length=25micrometres)



Using radial bands of 25micrometres we see a straight-line function with a strange vertical component at small and large radii. We also see stacked peaks at the same time, indicating an uncertainty in the position of the peaks, (eg-they're not the same in all angular directions, or the resolution is too low to separate them.)

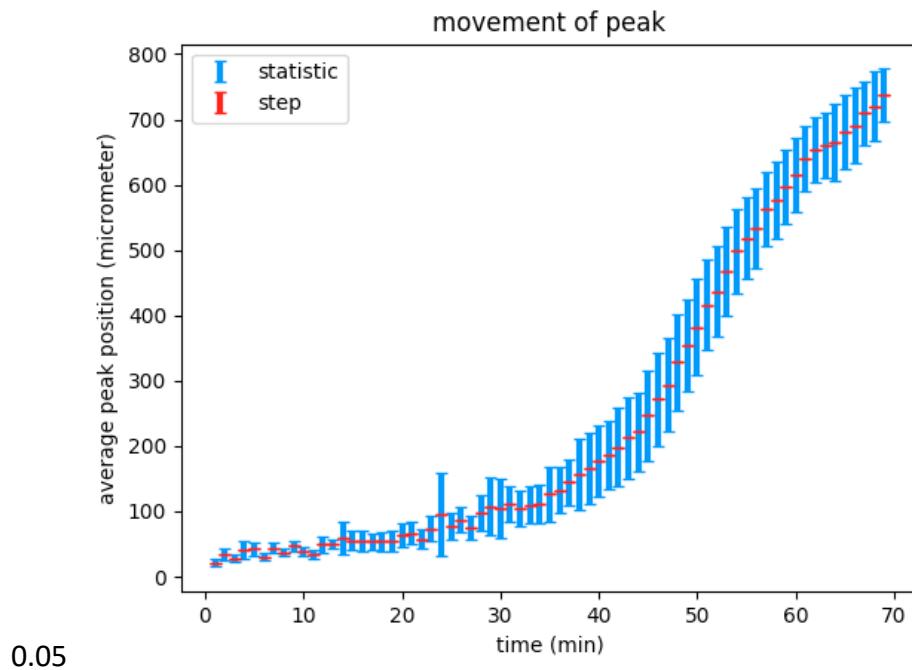
Decreasing the radial band width to 0.1, we see the peaks piled on top of each other. This is likely due to a resolution smaller than that offered by the simulation.

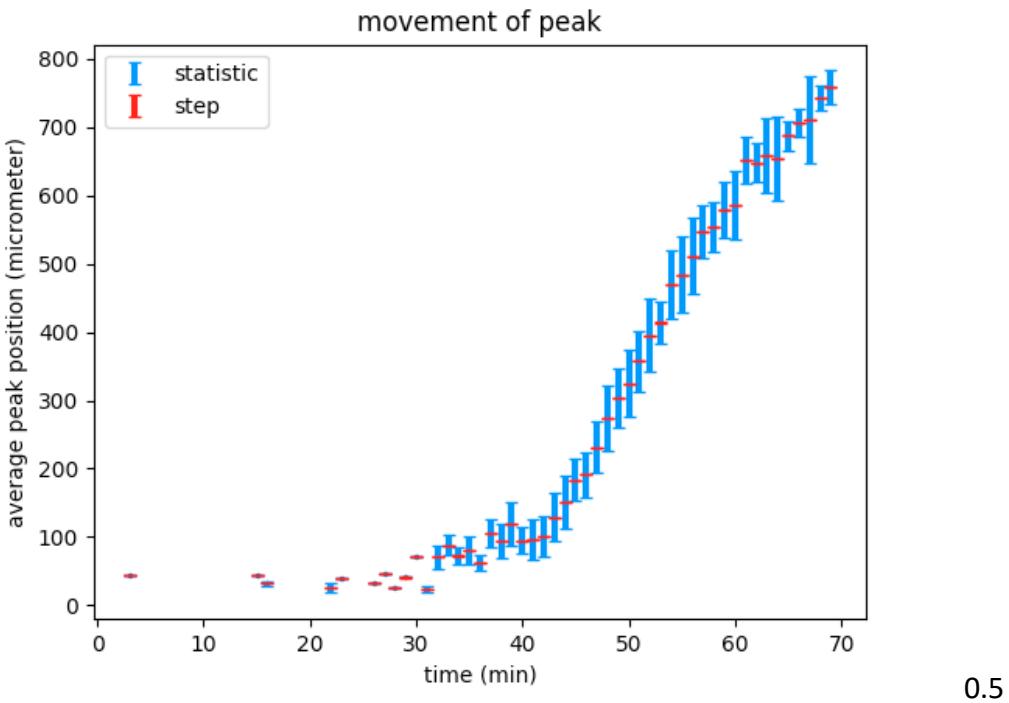


(grid_length=0.05 micrometres,0.05 minutes)

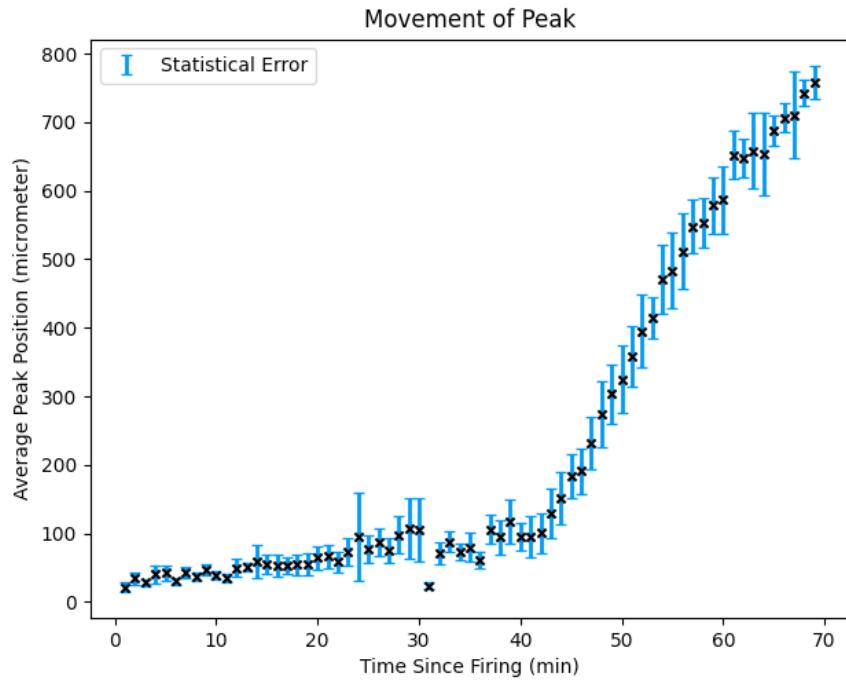
Averaging these points and taking the statistical uncertainty to be the standard deviation/sqrt(N) results in the following graph. In the “slow velocity” regime near the centre, the error bars are reasonable. However, at large radii they are too big. Using a larger radial bandwidth will counteract this.

Clearly for both graphs the uncertainty on the “radial band/step error” is negligible compared to the statistical uncertainty, so will be ignored from here.





At $r_{\text{grid}}=0.5$ we cannot resolve any detail in the slow regime, so we will combine the graphs. 0.05 grid length up to 31minutes and $r_{\text{grid}}=0.05$ for the remainder of the time since firing.



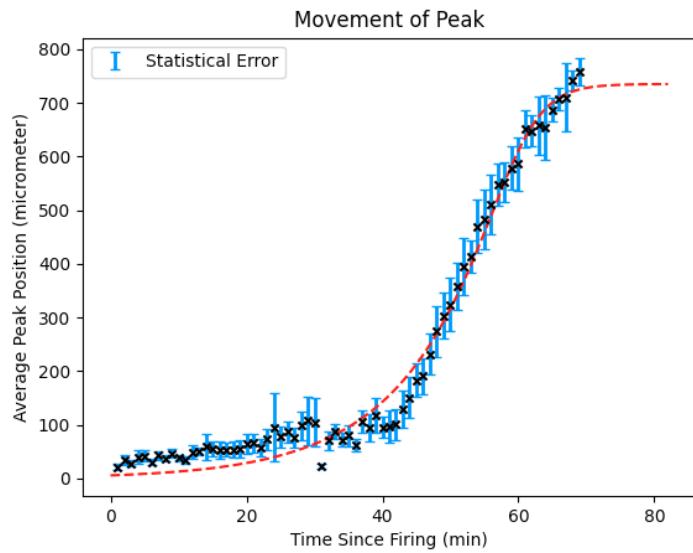
Although the statistical error appears to not match the deviation of plotted points from the overall trend, it is important to note that this statistical error is due to the poor time resolution of the dump_single function and the fact the peak may not travel at the same speed in all directions.

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.

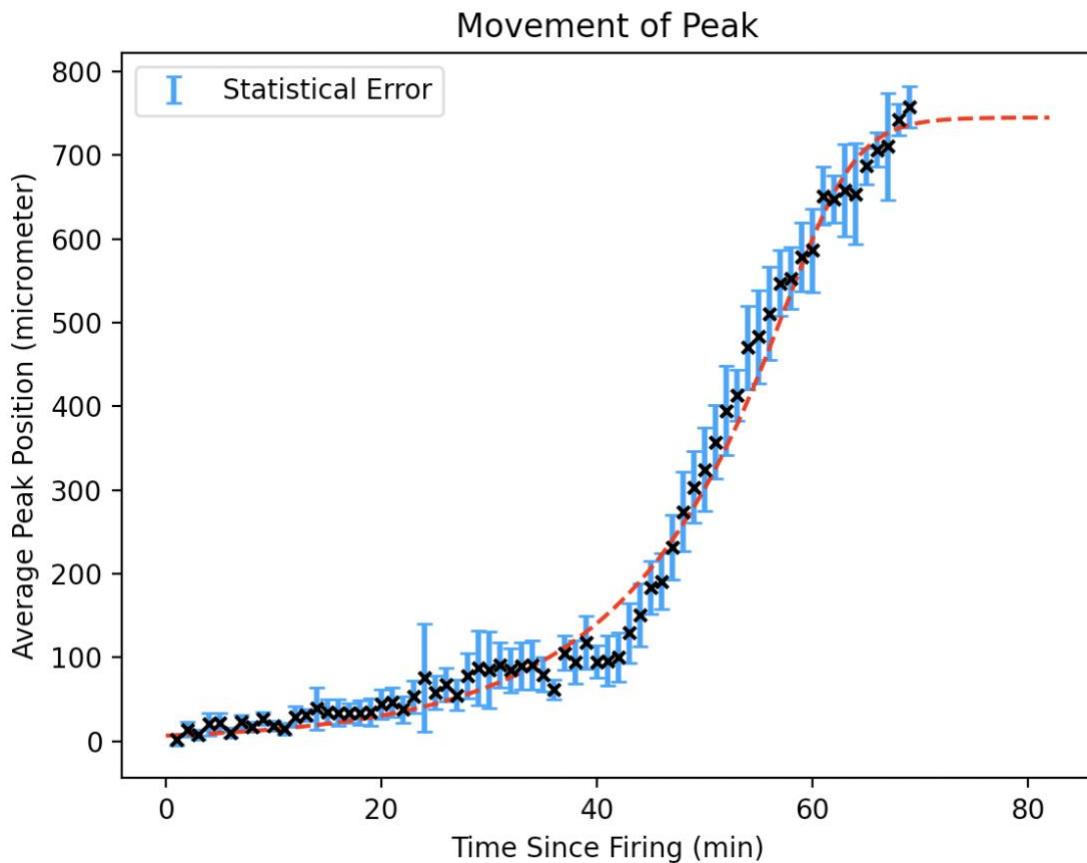
Now, we want to plot our sigmoidal fit. We initially considered the “logistical sigmoid function”, with some parameter on x. This didn’t match the curve at all.

We decided to use the following parameterised sigmoid:

$$r = \frac{a}{(e^{-bt} + c)^d}$$



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



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

2nd graph here uses the stat error and absolute_sigma=true

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-squared=

$$\chi^2 \equiv \sum_{i=1}^N \left(\frac{y_i - y(x_i; a_1 \dots a_M)}{\sigma_i} \right)^2$$

We calculate the reduced chi squared as the chi squared divided by the DOF=n-1

"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

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

$$r = \frac{a}{(e^{-bt} + c)^d}$$

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 sigmoid had a reduced chi squared=0.9854

6. Plotting Average Velocity against Time

Use the average velocity of the peak point.

$$v_{average} = \frac{R}{t}$$

R is the radial position of the peak point and t is the time of the peak

Propagation by fractional quadrature suggests $dv = |V| \sqrt{\left(\frac{dR}{R}\right)^2 + \left(\frac{dt}{T}\right)^2}$.

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

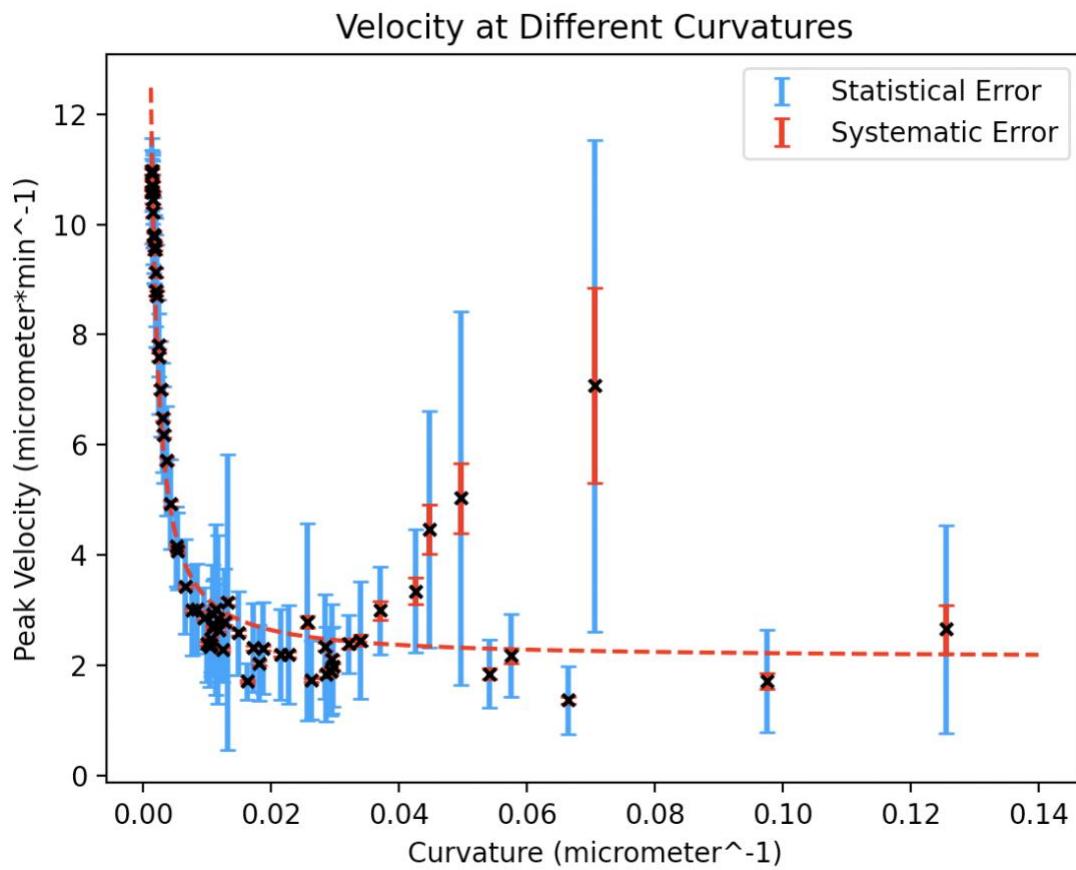
2. systematic uncertainty due to the discrete nature of the simulation and analysis: in the expression above we take half of the resolutions as dR and dt.

We fit the data to a power law

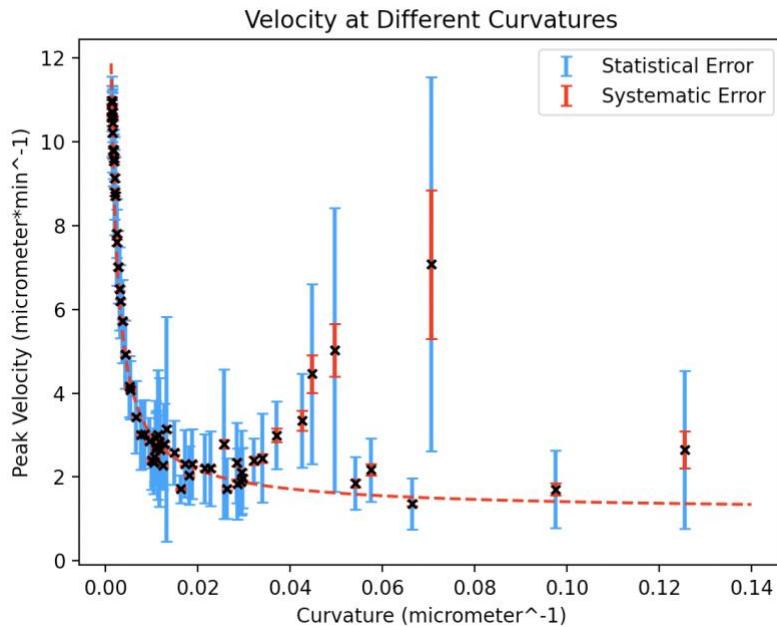
$$v_{average} = a \cdot curvature^{-b} + c.$$

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 need to work out whether to use the statistical or step error



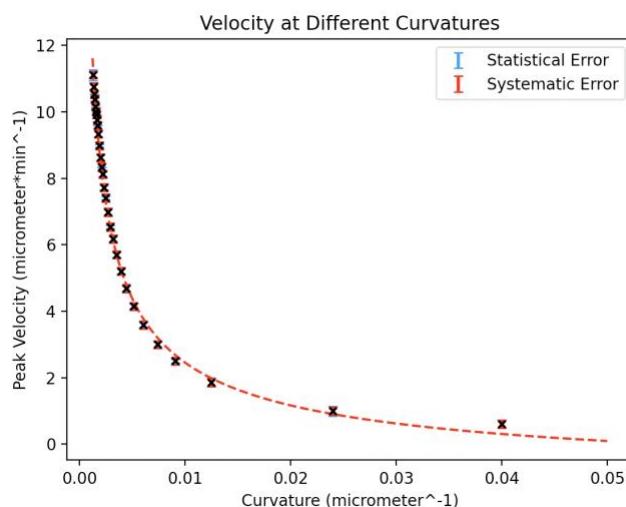
It is unwise to combine the statistical and systematic error at this point as they are not uncorrelated. So as before, we make the decision to use the statistical error when producing a curvefit. (scipy.optimize(...sigma=statstical_error, sigma=True)



Using the statistical errors to inform our curvefit, we see the data fits the power law at low curvatures. At small radii, (high curvature) the uncertainty grows and fluctuates dramatically.

From previously, we know that using larger spatial resolutions, we obtain data with smaller statistical uncertainties, we tried $r_{\text{grid_length}} = 10$ micrometres 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, the statistical and step errors are both almost equal at this point, and minimised, this indicates we have reached an r_{grid} length that balances our 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

The ABM used in gro has a stochastic nature.

We will run multiple simulations using the same initial parameter to see these stochastic effects.

We will compare the power law fits generated with data from different runs. We will produce the 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. The standard deviation of these points should give us some insight into how much each run differs.

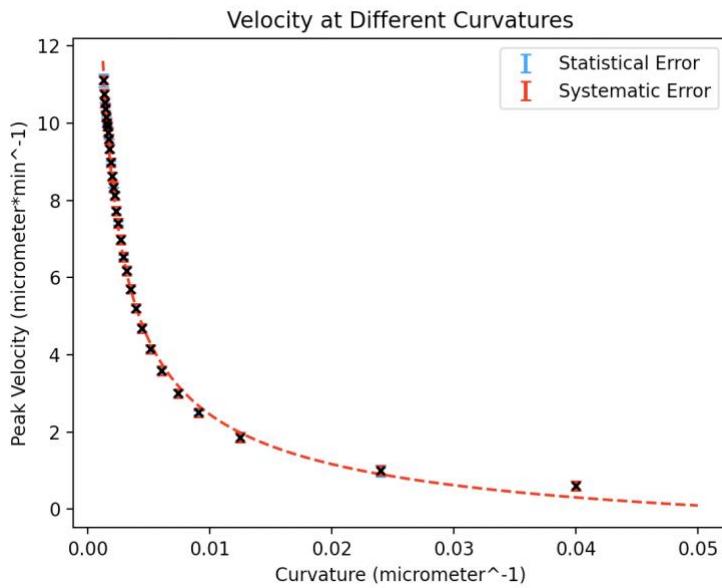
$$x_{CE} = \frac{\sum_i w_i x_i}{\sum_i w_i} \quad \text{where}$$
$$w_i = \frac{1}{\alpha_i^2}$$

The error on the weighted mean is:

$$\frac{1}{\alpha_{CE}^2} = \frac{1}{\alpha_i^2} + \frac{1}{\alpha_j^2} + \frac{1}{\alpha_k^2} + \dots = \sum_i \left(\frac{1}{\alpha_i^2} \right)$$

Weighted means uncertainties were calculated as above.

Run1:
Simulation data stored as original.csv.

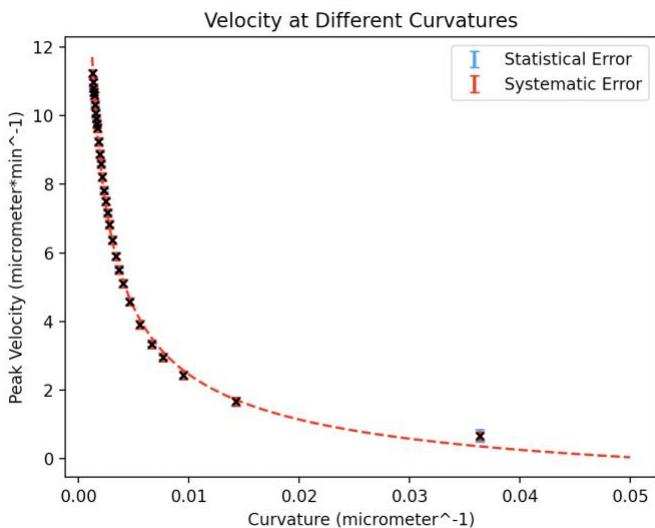


parameters [0.26496686
0.58284071 -1.42491055]

deviation [0.03450324
0.01763811 0.20334351]

reduced chi-squared =
1.551294004860183

Run2:
Simulation data stored as original_processed_1.csv.



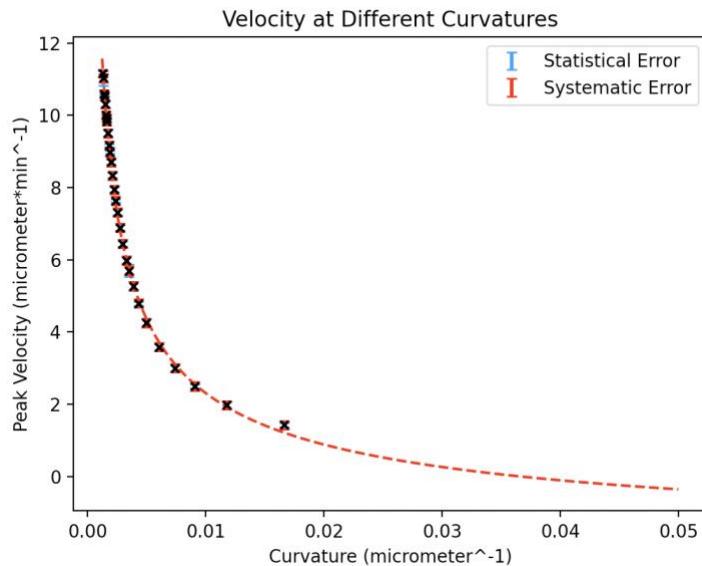
parameters [0.29526065
0.56961086 -1.60937]

deviation [0.0413555 0.0186976
0.24017637]

reduced chi-squared =
1.4160684422381613

Run3:

Simulation data stored as original_processed_2.csv.



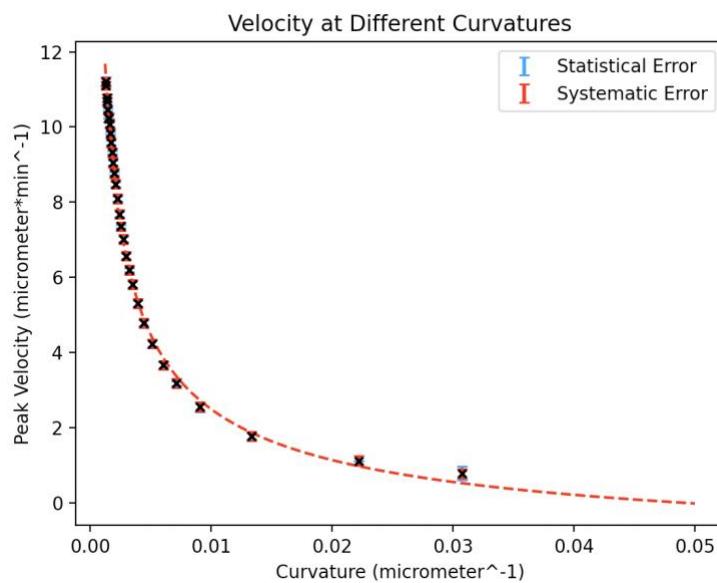
parameters [0.41581398
0.52535875 -2.36059388]

deviation [0.0758406
0.02372257 0.37460015]

reduced chi-squared =
0.6796828100432024

Run4:

Simulation data stored as original_processed_3.csv.

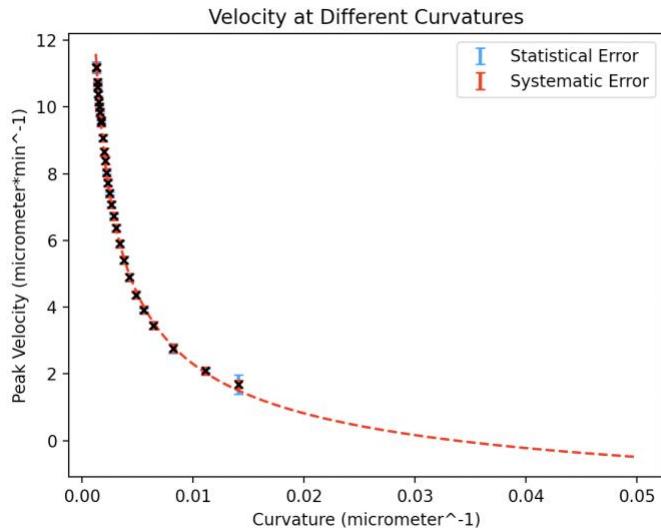


parameters [0.33539919
0.55218912 -1.76427786]

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

$$v_{average} = a \cdot curvature^{-b} + c.$$

$$a=0.31167180802388816 +/- 0.02136547092127166$$

$$b=0.5542199361408728 +/- 0.008946169347633549$$

$$c=-1.738686951567446 +/- 0.11839714964951213$$

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 750 micrometres, 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 its 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:

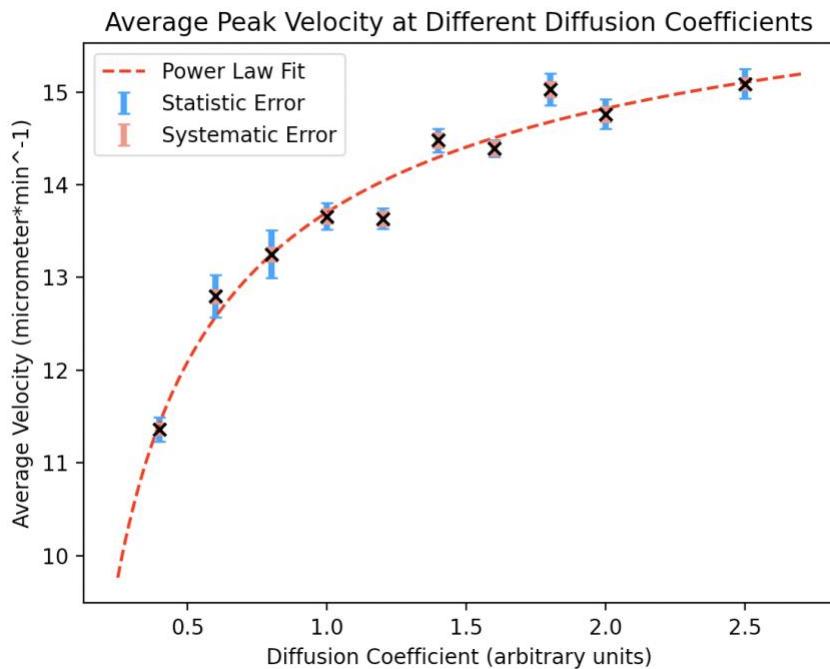
Diffusion coefficient = 2.0

```
[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
```



$$v_{average} = a \cdot D^b + c.$$

```
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

We have decided to treat a biofilm as a point mass, with an average concentration per bacteria that turns “on and off”. To measure this phenomena we average over the entire biofilm at a given time. We can then plot graphs of entire biofilm average concentration levels against time.

We simulated 5 equally separated biofilms in a line and producing a K⁺ trigger at the top of the uppermost biofilm. We want to analyse the speed that the “information”, travels across multiple biofilms. Also, we seek to see if there is any degradation of concentration during this process .

```

c_ecolis(300, 0, 0, 300,{"p3","p2"},program p());
c_ecolis(200, 0, 0, 220, {"p3","p2"},program p());
c_ecolis(150, 0, 0, 150, {"p3","p2"},program p());

c_ecolis(300, 0, 650, 300,{ "p3", "p2"},program p());
c_ecolis(200, 0, 650, 220, {"p3","p2"},program p());
c_ecolis(150, 0, 650, 150, {"p3","p2"},program p());

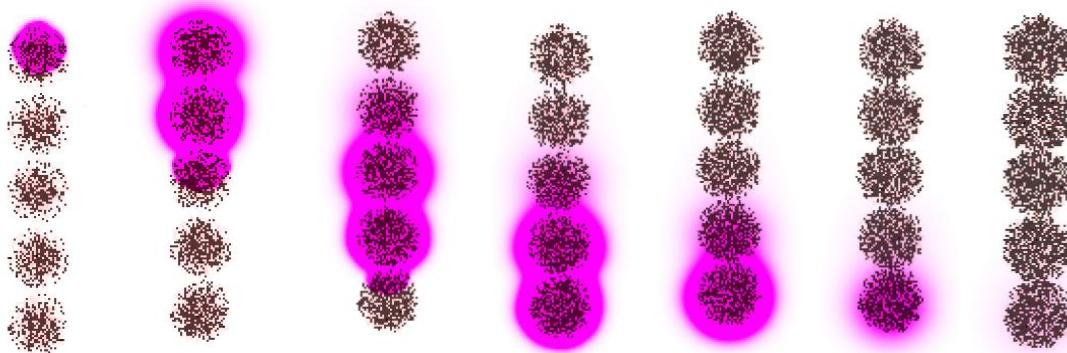
c_ecolis(300, 0, 1300, 300,{ "p3", "p2"},program p());
c_ecolis(200, 0, 1300, 220, {"p3","p2"},program p());
c_ecolis(150, 0, 1300, 150, {"p3","p2"},program p());

c_ecolis(300, 0, 1950, 300,{ "p3", "p2"},program p());
c_ecolis(200, 0, 1950, 220, {"p3","p2"},program p());
c_ecolis(150, 0, 1950, 150, {"p3","p2"},program p());

c_ecolis(300, 0, 2600, 300,{ "p3", "p2"},program p());
c_ecolis(200, 0, 2600, 220, {"p3","p2"},program p());
c_ecolis(150, 0, 2600, 150, {"p3","p2"},program p());

```

Screenshots of the propagation of the potassium



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

<u>Colony</u>	<u>position</u>	<u>time/mins</u>
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 measurements have an uncertainty of 0.5min due to the resolution of the simulation.

The position of each biofilm is a point particle in this treatment. We are not interested in measuring the position of the potassium wave here, only the speed of the peak overall concentration of the biofilm. We measure how fast this travel from the 1st colony to the 5th.

