

Imperfections in Reaction Diffusion Simulations

Dennis Barzanoff

10th March 2024

Supervisor: James Stovold

B.Sc. (Hons) Computer Science

Number of words = 8542

This includes the body of the report only

Declaration of Originality

I certify that the material contained in this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work.

I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Name: Dennis Barzanoff

Date: 10th March 2024

Abstract

The project aims to explore how changes in light imperfections impact the realism of a BZ simulation as it's scaled in size. The code is in Metal/GPU and runs experiments varying different simulation parameters that correspond to real-world changes, and then measures results. Changes in light were found to significantly impact the correct behaviour of BZ computer simulations and more advanced light management tools such as LED panels are required to scale large BZ circuits.

Contents

1	Introduction	1
2	Methodologies	2
3	Results	4
3.1	Size Limitations of the Coincidence Detector	4
3.2	Coincidence Detector Size Limitations	9
3.3	Temperature	12
4	Experiments	14
4.1	Measuring ϕ_{min} , ϕ_{max} and propagation time for the diode at different light conditions	14
4.1.1	Results	15
A	Creating a Coincidence Detector In Simulation	19
A.1	Unsuccessful Detector	19
A.2	Successful Detector	20
B	Original Project Proposal	22
B.1	Aims & Objectives	22
B.1.1	Aims	22
B.1.2	Challenges	22
B.2	Related work	23
B.3	Methodologies	23
B.4	Programme of Work	24
C	Another Appendix Chapter	26

List of Figures

2.1	The Metal Dependency Pipeline consists of a Compute Command Encoder (represented by red circles) and a Render Command Encoder (depicted by yellow circles). The Compute Command Encoder is run multiple times, performing as many calculations as possible. Simultaneously, the Render Command Encoder operates periodically to render the computed results. These two encoders work in parallel, with the Render Command Encoder producing output every time it gets a chance, while the Compute Command Encoder continuously performs computations most of the time.	3
3.1	First working AND gate	5
3.2	Light shined over a petri dish (Edwards, 1970)	6
3.3	Light flux at different parts of a Petri dish (Edwards, 1970)	6
3.4	Radius Illumination Graph	8
3.5	Illustration of finding the angle θ	10
4.1	Diode Experiment	15
4.2	Propagation times for various $\phi_{passive}$ values within the limits of ϕ_{min} and ϕ_{max} , illustrating the experimental outcomes of testing a diode in a Belousov–Zhabotinsky reaction. Each point represents a propagation time measurement for a given $\phi_{passive}$, with the average propagation time across measurements shown by a thicker black line. The shaded area between ϕ_{min} (red dashed line) and ϕ_{max} (blue dashed line) highlights the range where the diode works correctly, with the optimal ϕ marked by a solid green line.	16
4.3	Propagation times at ϕ_{max} —	17
4.4	Comparison of two waves passing through a diode. Above $\phi_{passive} = \phi_{literature}$, below $\phi_{passive} = \phi_{max}$. Both have different propagation speeds, so they have been equalised for the purpose of comparing their interaction with the diode side by side. The second wave takes substantially more time to form and is about 700 time steps slower as seen in Figure 4.3, which is quite a lot in such simulations.	17
A.1	Initial Coincidence Detector Configuration	19
A.2	First Successful Coincidence Detector	21
B.1	Expanding wave pattern from this project using the Oregonator model . .	23
B.2	Reaction Diffusion Game by Chaotic Edge	24
B.3	Gantt Chart	25

List of Tables

A.1	Results of the experiment showing the relationship between conductor width and gap size.	19
-----	--	----

1 Introduction

Reaction diffusion is a field of science that has seen rapid innovations in the areas of image processing (Kuhnert et al., 1989) and chemical computing . Among the numerous computing approaches, it has emerged as an alternative that takes advantage of the intrinsic ability of chemically reactive circuits to process information similar to that of computers. Although there is much potential there, most research focusses on building circuits in a simulation, such as Stovold (2019), versus in real life, such as Gorecki et al. (2003). It is unclear whether exactly how the simulation maps to real life in terms of size and environmental factors that are likely to impact it.

ref

2 Methodologies

The project experimented with Flutter for the initial simulation due to the benefits of pixel-level control and fast prototyping (see Figure B.1 for a preview of the first successful waves). However, the rendering was too slow for large chemical circuits, so the GPU was needed, for which, as of March 2024 Flutter still has no official support. A custom version of the newest engine was tested to expose experimental GPU APIs as detailed in flutter.dev/go/impeller-dart. That did not work, so the Metal library was used to compute the simulation on the GPU.

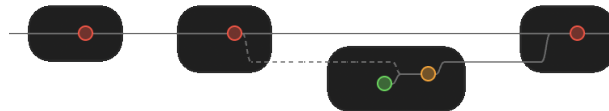


Figure 2.1: The Metal Dependency Pipeline consists of a Compute Command Encoder (represented by red circles) and a Render Command Encoder (depicted by yellow circles). The Compute Command Encoder is run multiple times, performing as many calculations as possible. Simultaneously, the Render Command Encoder operates periodically to render the computed results. These two encoders work in parallel, with the Render Command Encoder producing output every time it gets a chance, while the Compute Command Encoder continuously performs computations most of the time.

3 Results

This chapter aims to explore the operational boundaries of the chemical components in a Petri dish environment, focusing on the size limitations of the coincidence detector and the light sensitivity of a chemical diode.

The main objectives of the experiments are to find out the bounds and limitations of different chemical components in order to determine the feasibility of using them in a real-world environment.

3.1 Size Limitations of the Coincidence Detector

The goal here is to find a way to measure the size of the coincidence detector in real life, for which there is currently no data. The size of the coincidence detector is crucial for the design of the Petri dish, as it determines the minimum size of the Petri dish required to accommodate the detector.

Creating the coincidence detector is covered in Appendix A, the final component is illustrated in Figure 3.1.

A Petri dish, how likely is it to vary the resolution within the petry dish. how big is it how big is this pixel

figure out the resolution. you can figure out the math quite easily.

For this AND gate to be deployed in a Petri dish, how big does the Petri dish need to be.

NB!

The size of the Petri dish is **assumed** to be unsubstantially larger than the size of the chemical circuit inside of it. Continuing from both sizes are referred to interchangeably.

Ok, so here is the plan: 1. for the AND gate, figure out the constraints: - min size of everything - assumption is that there is no max size for the length as the waves just travel along. There is probably also no max width as the waves just expand, but there is probably a min width

2. Figure out the conditions for the AND gate to work in its smallest form, what are the conditions under which the AND gate would still work in the real world. For example, how close do they need to be in the real world before the light is not strong enough to keep them apart, maybe that's two-three pixels in the case of the simulation because 3 pixels are impossible to get across and two pixels are possible if the wave is focused towards the gap.

In my case there is either max light or no light, so I need to be careful when I see how much light they are projecting in different spots.

intensity, angle, size of the light beam can influence the BZ reaction

some can be influenced by temperature (Yamada et al., 2022). The model they are

is it the same reaction

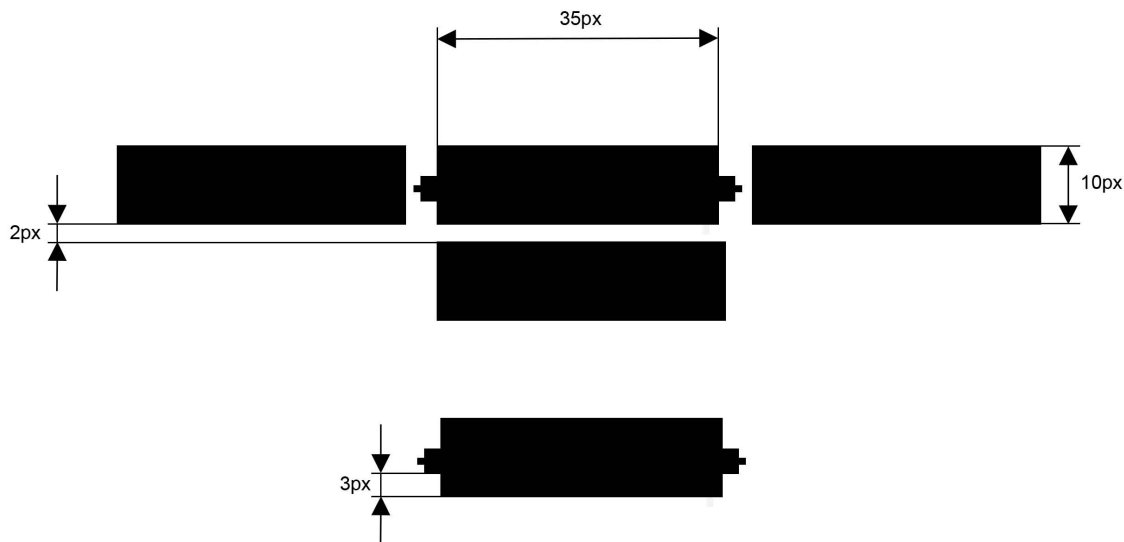


Figure 3.1: First working AND gate

using is a modified version of the Oregonator, that would be too much work for us.

alright, first I need to find what the light-sensitive reaction is then I need to inspect how they are shining the light, is it from above, from the side, is there multiple light sources. then I need to inspect a data sheet for the diode they are using to see how the light shined everywhere differs, assuming the direct light below it is the maximum shined light, and I need to find a mathematical model that describes how the light around the diode becomes less. what is the focus of that light?

Shining light on the BZ surface: (Barry et al., 1979) The catalyst in these reactions is light, using light we can modulate the speed using Φ . A light bulb is shined over the Petri dish (fig. 3.2)

Given multiple light sources, the total illumination I_{total} at a point on a surface is the sum of the illuminations from each individual light source (fig. 3.3). The illumination I_i from a single source at a given point is given by the inverse square law, adjusted for the angle of incidence θ_i :

$$I_{\text{total}} = \sum_{i=1}^n \frac{P_i}{4\pi r_i^2} \cdot \cos(\theta_i)$$

where P_i is the power of the i -th light source, r_i is the distance from the i -th light source to the point, and θ_i is the angle between the direction of the i -th light ray and the normal to the surface. The cosine term $\cos(\theta_i)$ accounts for the angle of incidence, with the intensity contribution from the light source decreasing as the angle increases.

(Edwards, 1970)

This means that now we can use that in the simulation to simulate the imperfections of light sources.

Now I have to figure out how exactly they shine the light to form an inhibitor path. Is it multiple bulbs? Is it just one bright bulb that is far away, so the θ is insignificant?

In real experiments, the chemical reaction that occurs on the gel surface is sensitive to light due to the specific catalyst used ($\text{Ru}(\text{bpy})_3\text{SO}_4$). However, to observe and record the process, they need to use light to illuminate the gel, so that the camera can capture images.

figure out how many light sources are used to illuminate a BZ reaction in a petri dish

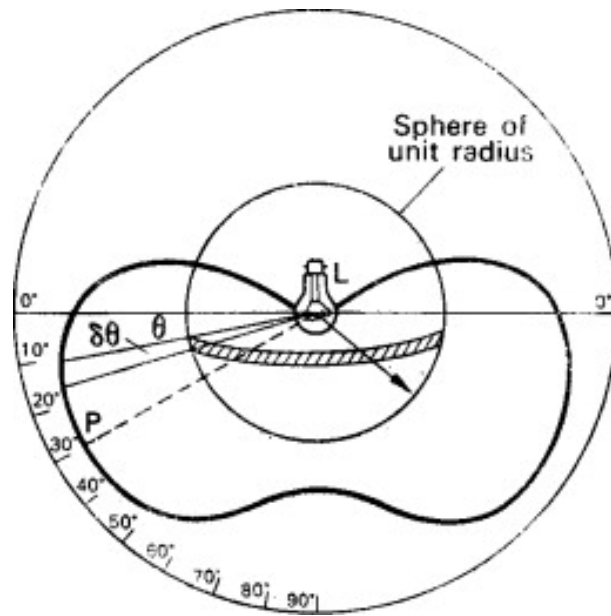


Figure 3.2: Light shined over a petri dish (Edwards, 1970)

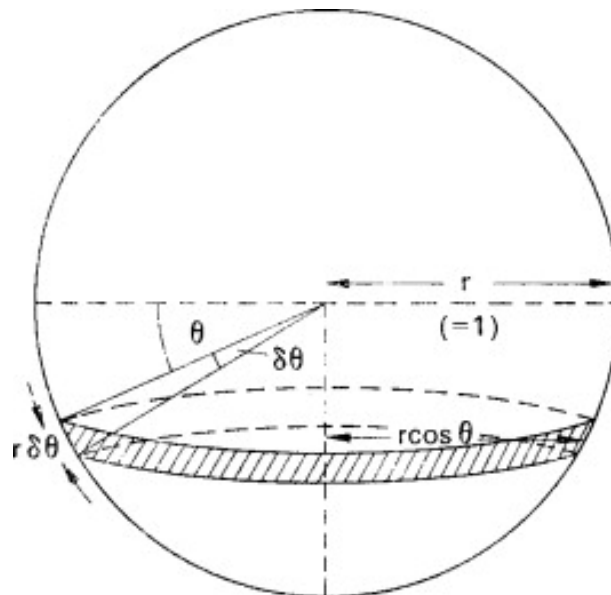


Figure 3.3: Light flux at different parts of a Petri dish (Edwards, 1970)

The experiment design, including time-varying intervals of different light intensities, allows them to balance between controlling the reaction and capturing the activity on the gel. Toth et al. (2009)

From what I see, they all project a single light bulb and then use a mask to have the shape they want projected, such that light does not kill off the reaction and there is light everywhere else. This is important to show how imperfections can affect the simulation. Gorecki et al. (2003)

This book (Cui, 2004), referenced by Gorecki et al. (2003), contains a similar setup and goes into more detail how they project the pattern onto the gel.

Also the observation angle seems to change where we see the wave, idk if that is relevant. (Cui, 2004).

Unfortunately, they do not specify how far away the camera is, but we can assume the came is far away, such that the angle of light projection is very close to 90° , so that would make the effect of the angle insignificant. However, we can still simulate that in our simulation to see what effect it would have if it were close to the medium; for larger circuits, this effect would be visible. Given a normal setup, what would be the maximum size of the Petri dish in a simulation before the effects of the light angle start to impact the simulation.

The next step is to assume a size; in the papers above they have the size of the AND gate coincidence detectors used, so that can help me assume a relatively accurate size for the mapping from pixels in my simulation to a Petri dish.

It is really challenging to find information on the exact distance the light source is from the Petri dish, but we are going to assume that the distance is 1m because that is how much it looks like from the diagrams they show where there are other objects like projectors and mirrors that are estimable. (Gorecki et al., 2003)

Now I need to prove that for a small Petri dish, this would have an insignificant effect even in my simulation, but when we scale it, the effect would be very noticeable.

Now, this is just a simulation we are doing, so to get the mapping for a real Petri dish, we can use the dimentions of the gate in a Petri dish. In Gorecki et al. (2003) they specify the exact dimentions in mm of the collision detector inside of the Petri dish, which can be used to map it to the one in our simulation. The distance between the active channel and the signal bar is 0.4 mm, which corresponds to exactly 2px in the simulation.

from paper: Both stripes are 2 mm wide, the signal channel is 10 mm long, and the detector part is 6 mm long. The gap between the signal and the detector channels is 0.4 mm. The light intensity was selected so that the non-illuminated parts of the membrane were excitable, whereas the excitations died in the illuminated areas. In the experiment, the light intensity was set at $I = 24 \text{ kLx}$ as determined by a light metre (ASONE LM-332), and the temperature was $295 \pm 1 \text{ K}$.

From the provided information in the paper and the datasheet for the JCD100V-300W halogen bulb, we have the following data:

- Luminous flux (Φ) as specified in the datasheet = 6600 lm (Fujilamp, 2024)
- Light intensity (I) at the Petri dish = 24,000 lux

Originally, we calculated the luminous flux using the estimated luminous efficacy of $17 \frac{\text{lm}}{\text{W}}$ and the power of the bulb (P) as 300 W, which gave us:

$$\Phi_{\text{estimated}} = \text{Power of bulb} \times \text{Luminous efficacy} = 300 \text{ W} \times 17 \frac{\text{lm}}{\text{W}} = 5100 \text{ lm}$$

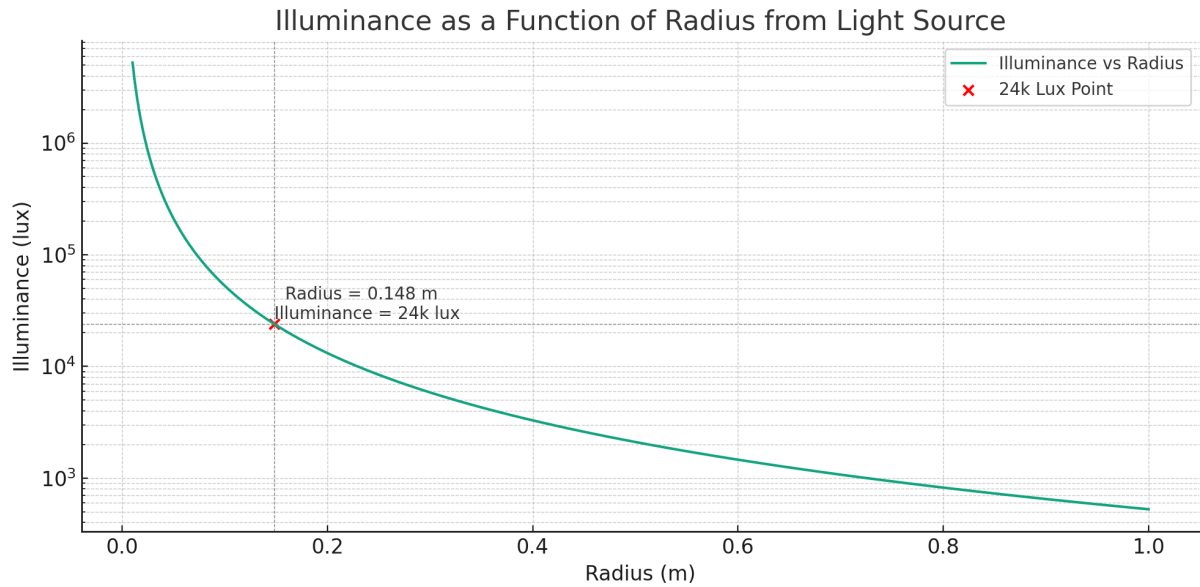


Figure 3.4: Radius Illumination Graph

However, this was an approximation. The datasheet for the bulb specifies a luminous flux of 6600 lm, which suggests that our assumption for the luminous efficacy was incorrect.

Using the inverse square law, which relates the light intensity (I) to the distance (r) from the light source, we have:

$$I = \frac{\Phi}{4\pi r^2}$$

Solving for the distance (r) with the correct luminous flux, we get:

$$r = \sqrt{\frac{\Phi}{4\pi I}}$$

Substituting the values from the datasheet and the light intensity measurement, we find:

$$r = \sqrt{\frac{6600 \text{ lm}}{4\pi \times 24,000 \text{ lux}}}$$

$$r \approx 0.148 \text{ metres}$$

Thus, the corrected distance from the light source to the Petri dish, using the accurate luminous flux from the datasheet, is approximately 0.148 metres or 14.8 centimetres. This distance is crucial, as it suggests that the light intensity measurement of 24 kLux is likely taken close to the Petri dish where the biological samples are studied, rather than at an arbitrary point close to the light source.

In figure 3.4 it is shown how the radius impacts the intensity at the light. This is necessary because it is not specified how they

Given the luminous flux Φ of the JCD100V-300W bulb as 6600 lumens, we can calculate the illuminance I at various distances from the light source. The illuminance is given by the formula:

$$I = \frac{\Phi}{4\pi d^2}$$

where d is the distance from the light source in meters.

For distances of 10 cm, 12 cm, 18 cm, and 20 cm, the illuminance values calculated are:

- At 10 cm: $I \approx 52521$ lux
- At 12 cm: $I \approx 36473$ lux
- At 18 cm: $I \approx 16210$ lux
- At 20 cm: $I \approx 13130$ lux

These values illustrate a significant change in illumination as the radius increases, indicating that the radius is a crucial factor in determining the intensity of light received at a point.

Concluding from these calculations, it is likely that the measurement of 24 kLux was made relatively close to the Petri dish. This is because the light intensity of 24 kLux is a practical measure for the conditions under which biological samples are typically studied. Measuring illuminance right next to the light source would yield an impractically high value, which is not as useful for experimental purposes. Hence, the measurement taken is more likely to be representative of the actual working conditions near the Petri dish.

Now I know how far away the Petri dish is from the light source, but I still do not know the size of the Petri dish they used.

3.2 Coincidence Detector Size Limitations

In the setup (Gorecki et al., 2003) they specify a signal channel of 10cm and a membrane filter of 2.5cm. The filter applied is usually smaller than the Petri dish itself, but for now we can just look at the dimensions we specified at .

Given the following mapping from real life measurements to simulation pixels:

- Stripe width: 2 mm = 10 px
- Channel gap: 0.4 mm = 2 px

We can establish a scaling factor for the conversion from millimeters to pixels. For the stripe width:

$$\text{Scaling factor} = \frac{10 \text{ px}}{2 \text{ mm}} = 5 \text{ px/mm}$$

For the channel gap:

$$\text{Scaling factor} = \frac{2 \text{ px}}{0.4 \text{ mm}} = \frac{2 \text{ px}}{\frac{2}{5} \text{ mm}} = 5 \text{ px/mm}$$

Thus, both measurements confirm the same scaling factor. Using this consistent scaling factor, we can convert any measurement from millimeters to pixels.

Using this scaling factor, we can convert any measurement from millimeters to pixels. For a dimension of 10 mm × 4.4 mm, the conversion would be:

$$\text{Width in pixels} = 10 \text{ mm} \times 5 \text{ px/mm} = 50 \text{ px}$$

$$\text{Height in pixels} = 4.4 \text{ mm} \times 5 \text{ px/mm} = 22 \text{ px}$$

Thus, a real-life size of 10 mm × 4.4 mm would map to a size of 50 px × 22 px in the simulation.

reference
the place
where you
said all
the cm
used in
the dish

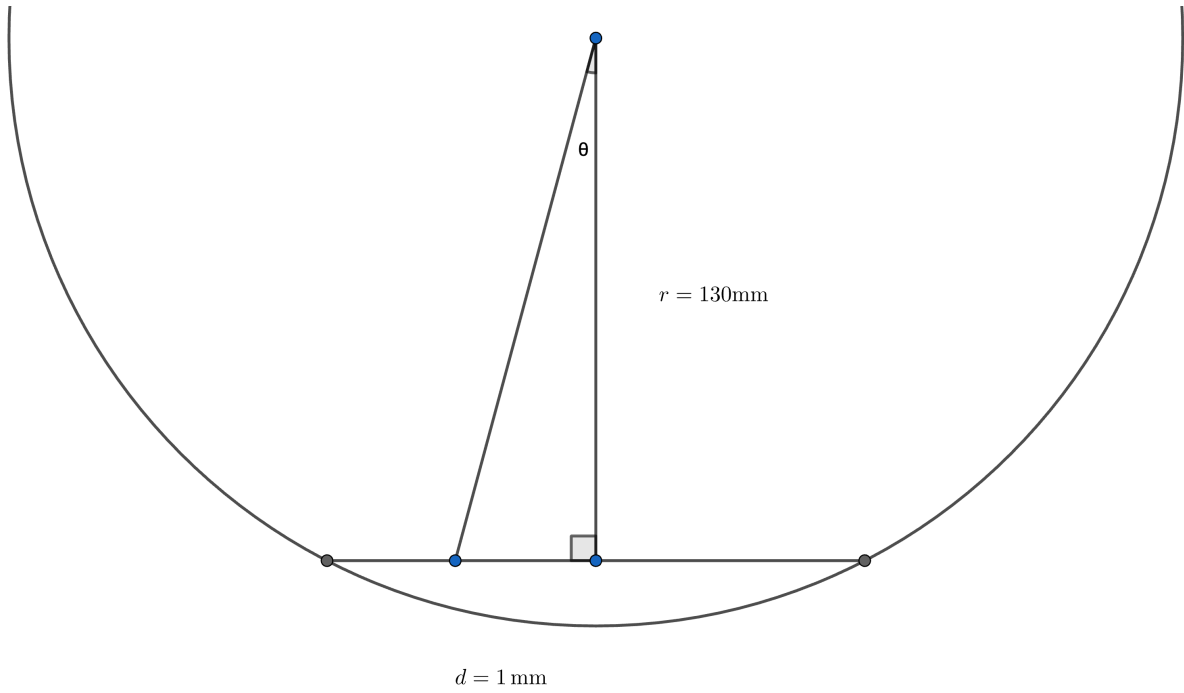


Figure 3.5: Illustration of finding the angle θ

Realistically, we only care about the size of the circuit inside of the Petri dish, so we don't really need to know the actual size of the Petri dish for this imperfection.

Next thing is to put the imperfection in the simulation with jsut an and gate. To do that, I need to come up with a formula about plugging in a pixel and getting out the illumination percentage. The percentage is basically the $\cos \theta$ as it is 0 when the angle is 90° . let's assume a parallel light source is max intensity and an angle of 0 is no intensity at all, so in the formula we would plug in $\phi = 0.054$ for when it's active and $\phi = 0.0975$ when it's passive. Active means no light, passive means a lot of light, so when the light source becomes weak at a point, that would mean thath ϕ becomes more active. let's come up with a formula in mm that tells us when we move 1mm from the center of the petri dish where the light is most intense, what angle is that. That should be pretty simple, we have a right triangle,

i made a bad assumption, i thought when we get the 0.44 angle for a 1mm distance, we can just get 0.88 for 2mm, which is proolly not true, we have to find the angle every time

To calculate the angle θ for any given radial distance r_c from the center of the Petri dish, we convert the pixel distance p to millimeters using the scaling factor s (in px/mm):

$$r_c = p \times s$$

The angle θ is then found using the arctangent function with respect to the height h of the light source above the center of the Petri dish:

$$\theta = \arctan\left(\frac{r_c}{h}\right)$$

The percentage of illumination I_p at this radial distance is given by the cosine of θ :

$$I_p = \cos(\theta)$$

Since $\cos(\arctan(x))$ simplifies, we can express I_p directly as:

$$I_p = \frac{1}{\sqrt{1 + \left(\frac{r_c}{h}\right)^2}}$$

This formula calculates the percentage of illumination at a point r_c millimeters from the center, assuming the light intensity directly beneath the source is 100

From manual tests with illumination, the circuits start behaving not as expected even with illumination percentages less than 0.99 because the simulation is very sensitive to changes in light intensity as it relies on a manifold of light-affected microinteraction. Using this value of 0.99, we can see at what size of the Petri dish do we start to see values below 0.99. We can calculate that and then confirm in the simulation.

I debugged the nonworking thing with a debug variable inside the metal function controlled by my mouse. I just verified that the simulation does break once the illumination reaches 0.99. I did not get much value from putting this inside the simulation, more like, oh yeah, I am correct. I could have verified that 0.99 is the threshold by just running it.

I verify using a simulation that if the circuit is too big, a single light source might not be enough to illuminate the whole area relatively evenly and even light perturbations in the light are going to disrupt larger circuits. What can be done is to make the circuit different the further away it is from the source, which would involve different dynamics that work under different lighting conditions. This is infeasible to do practically as it so highly depends on the light source.

The 0.99 is kind of sus, I still need to figure out a lot of things like a more optimal mapping from the real-life illumination intensity and the mathematical formula in the Oregonator model. I think my mapping is okay because ϕ_{active} is what it is and ϕ_{passive} can be moved even further, I just interpolate between these values, maybe there is a better way to simulate it. The problem with my simulation is that it does not matter how strong the light source is, you can put the whole sun over the Petri dish, the illumination at the angle is still going to be 0.99 of the original, I need a better way to simulate it that can also input the parameters of the light source, so it's not just about the strength. I need something like a threshold for when the light intensity is so strong that nothing can pass through. That is also a problem, however, because the actual circuit relies on the gaps being not too illuminated, so that the activator chemical can diffuse towards another part of the surface. It's again difficult to come up with a model. So that means a more powerful light bulb might create different things for different illuminations, but it might not work if the Petri dish is directly under it as it might create too much illumination and prevent the waves from crossing boundaries. *I don't know*. So the 0.99 is only true when the assumption that ϕ_{active} is equivalent to the Hallogen light bulb of 300W used directly above it.

Maybe, just maybe, the relationship is not linear as I assume, but logarithmic, so as the illumination in real life decreases, that creates a logarithmic decrease in ϕ from ϕ_{active} to ϕ_{passive} , so it might not be as sensitive to the small changes in light as I assume.

In the simulation very small changes in light intensity were found to very significantly affect the propagation delay of the activator signal. This is crucial for the operation of the circuit. These circuits do not have any way of synchronisation other than propagation delay, which means even the slightest change in light intensity is going to affect the speed of propagation and the whole circuit would break. This is more important for bigger circuits like the ones built by Stovold (2019), that rely heavily on exact propagation. In my simulation it was very sensitive to the illumination changes, maybe in real life as long

To resolve this, how did the var values come about

as it's more illuminated than some *threshold*, then it doesn't matter for the simulation. I need to do more digging to figure out what to do from here.

3.3 Temperature

Let's examine how temperature affects reaction diffusion.

say im assuming the light bulb is not using soft light, it has not piece of paper so inverse square law is fine.

we're looking at a paper working with old technologies. let's use an LED panel, uniform light, scale better.

Bell curve for light distribution of light.

how does the phi relate to the real world.

dig into the oregonator model, what do they define as phi? excitability? what is the mapping? it's nonlinear.

temperature is not a great route, what effect would dust have on the wave propagation.

exploring how does different light source magnitudes get affected by a sheet of paper ? less useful....

focus on the large petri dish.

why are you measuring the pixels?

dont talk about papers, talk about research

general introduction - what is this, background & literature review, what have other people done , why is this important. james has done this, how possible is this in real life. aims and objectives are next - i wanna look. specify them

then we look at the results, what is the meat of what ive found.

then there is a discussion section , opposite of backtounrd and lit review, now we go back out. place our findings in the context of our literature, how does thi simpact, tha'ts where my views come in, scientifically... how realistic is it? now that we know this new piece of information

conclusion & evaluation

what you've done, what you've larned . reflection manner shows you what you've learned.

give it some reasonable name - related to what im doing.

Ok, I need to understand how illumination maps to ϕ . For that I need to understand the Oregonator model

that's it, we need to find the minimum and maximum reaction rates in real life reactions. we already have a minimum and maximum in the simulation ,we can check if they behave the same way. if they do, that's our thresholds, then we need to find an appropriate function that describes a similar relationship and we found it

The main idea is that when there is no activator in the solution, (Reddy et al., 1995)

Beer-Lambert Law for Absorption

The Beer-Lambert Law describes how the intensity of light decreases as it passes through an absorbing medium:

$$I = I_0 \cdot e^{-\alpha \cdot l}$$

where:

- I_0 is the initial light intensity before hitting the OHP sheet.

Move to future work, looking at phi act and phi pass are related, atm we are varying passive, but active

- I is the light intensity after passing through the material.
- α is the absorption coefficient of the OHP sheet material.
- l is the thickness of the OHP sheet.

For our calculations, we assume an absorption coefficient $\alpha = 1 \text{ m}^{-1}$ and a thickness $l = 0.004$ meters (4 mm).

Fresnel Equations for Transmission

The Fresnel equations determine how much light is transmitted and reflected at an interface, depending on the angle of incidence. For non-polarized light and considering both s-polarized and p-polarized components, the average transmission T can be approximated by:

$$T = \frac{T_s + T_p}{2}$$

where T_s and T_p are the transmission coefficients for s-polarized and p-polarized light, respectively. These coefficients are calculated using the refractive indices of the air (n_1) and the OHP sheet material (n_2), and the angle of incidence θ_i .

Calculation at 45 Degrees Angle of Incidence

At a 45-degree angle of incidence, we calculated the transmission coefficients for s-polarized and p-polarized light, and found the average transmission T to be approximately 0.950. This indicates that about 95% of the light is transmitted through the OHP sheet at this angle.

Combining the effects of absorption and transmission, the final intensity I_{final} of light after passing through the OHP sheet and considering the angle of incidence is given by:

$$I_{\text{final}} = I_0 \cdot e^{-\alpha \cdot l} \cdot T$$

Substituting the given values and assumptions, we find:

$$I_{\text{final}} \approx 0.946$$

This result indicates that the combined effect of slight absorption by the OHP sheet and the reduction in transmission due to the 45-degree angle of incidence leads to a final light intensity of approximately 94.6% of the initial intensity.

These calculations demonstrate the importance of considering both material properties and geometric factors, such as the angle of incidence, when modelling the transmission of light through materials in experimental setups.

the diffusion assumes 100% goes through at the centre of the light source, it's more like 99% or something

4 Experiments

Here $\phi_{passive}$ is measured at different values because that is the value controlling the illuminated area, which makes the circuit there “passive” . This simulates less illumination of light at the diode, altering its behaviour.

4.1 Measuring ϕ_{min} , ϕ_{max} and propagation time for the diode at different light conditions

During the evaluation of ϕ_{min} , ϕ_{max} , the diode had to be tested from numerous directions to determine the minimum and maximum values of $\phi_{passive}$ where it still functions correctly as a diode, which means that it passes waves from right to left, but never from left to right. As we approach the limits, new cases start appearing where the wave becomes able to pass through from a very specific angle that changes as we change the phi value, making it impossible to automatically test, so manual testing was needed to verify if the diode works in every single use case for the limit phi values. Also, the minimum and maximum values start to highly depend on the testing conditions, and whether the diode works or not starts to depend on how you are using it; for example, do you start a wave right at the diode? If not, what is the minimum distance the diode has to work for? This greatly changes the minimum and maximum phi values.

Results of measurement:

- $\phi_{max} = 0.106127$
- $\phi_{min} = 0.09555$

That is, as long as $\phi_{min} < \phi_{passive} < \phi_{max}$, then the diode will function as expected, even though the propagation time would still be affected (as seen in Figure 4.2), causing synchronisation problems in larger circuits.

The way the experiment was set up was that a diode was set up in Figure 4.1 where a wave was launched 25 px from the diode to allow it to pass through the diode and reach a small white flag where the wave is expected and the experiments are evaluated. A step in the diagram means one simulation pass. The parameters of the experiment were the following:

- Number of $\phi_{passive}$ values: 30, of which only 19 succeeded, the remaining 11 could not pass through the diode due to the parameter being too out of range.
- Interval size: 0.0013448276
- Number of runs per $\phi_{passive}$ values: 7
- ϕ_{start} : 0.08066608

figure out
if that's
how
quotes
should be
or rever-
esed

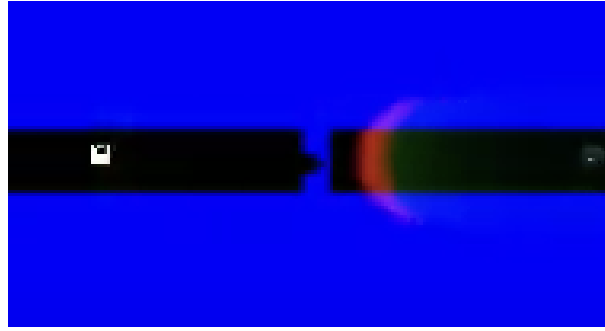


Figure 4.1: Diode Experiment

- ϕ_{end} : 0.121010914, never reached because once a wave passes through the diode, the concentration no longer reaches zero fully and that leaves a very small amount of inhibitor at the diode and due to the borderline value of $\phi_{passive}$ the next waves never manage to penetrate to get measured, so the highest measured value for $\phi_{passive}$ is 0.10487298
- Wave start: 39px before diode, arbitrary distance to allow the wave to form.
- Wave measure: 31px after diode, arbitrary distance.
- dt : 0.001, also serves as a reference point for the amount of steps.
- Measurement interval: about every 50 steps, depending on the GPU frame rendering schedule

It is possible to run the experiment much more accurately, eliminating the need for multiple runs for the same $\phi_{passive}$ value we measured at every step and did not simulate another step in the simulation until the measurement was complete. However, that would drastically slow down the experiment, making it run for hours instead of minutes. For the purposes of merely showing that the propagation changes along with the slope, it was enough accuracy, so the faster and more inaccurate solution was chosen.

So, I found the following limitation while working with the simulation. Due to the fact that in metal sampling is a rendering operation, it cannot be done during the computation of the simulation. That means that I would have to synchronise it to run with rendering speed, but that would make the simulation too slow to run many experiments. It uses the step speed of the simulation, 1 frame = about 150 steps in the simulation. This is still accurate enough. I ran at $dt = 0.001$.

4.1.1 Results

The results of the experiment can be seen in Figure 4.2. The propagation time decreases as there is less light, indicated by the decrease in the value of $\phi_{passive}$ and the reverse. This is to be expected, as a lower value brings it closer to ϕ_{active} (darkness), and illumination is an inhibitor. The reason the circuit has a higher tolerance to less light, compared to more light, is because the diode is much more resistant in the reverse direction as the wave has to go through a reverse triangle. The diode also has a high tolerance to $\phi_{passive}$ increasing due to an additional spike added to the diode, which helps waves to cross the gap easier, even when there is more light. The reason for $\phi_{max} < \phi_{end}$, meaning the diode was still letting the wave through left to right and not right to left at ϕ_{end} , but it was not at other

figure out how the point thing is called

measure without the spike, how bad

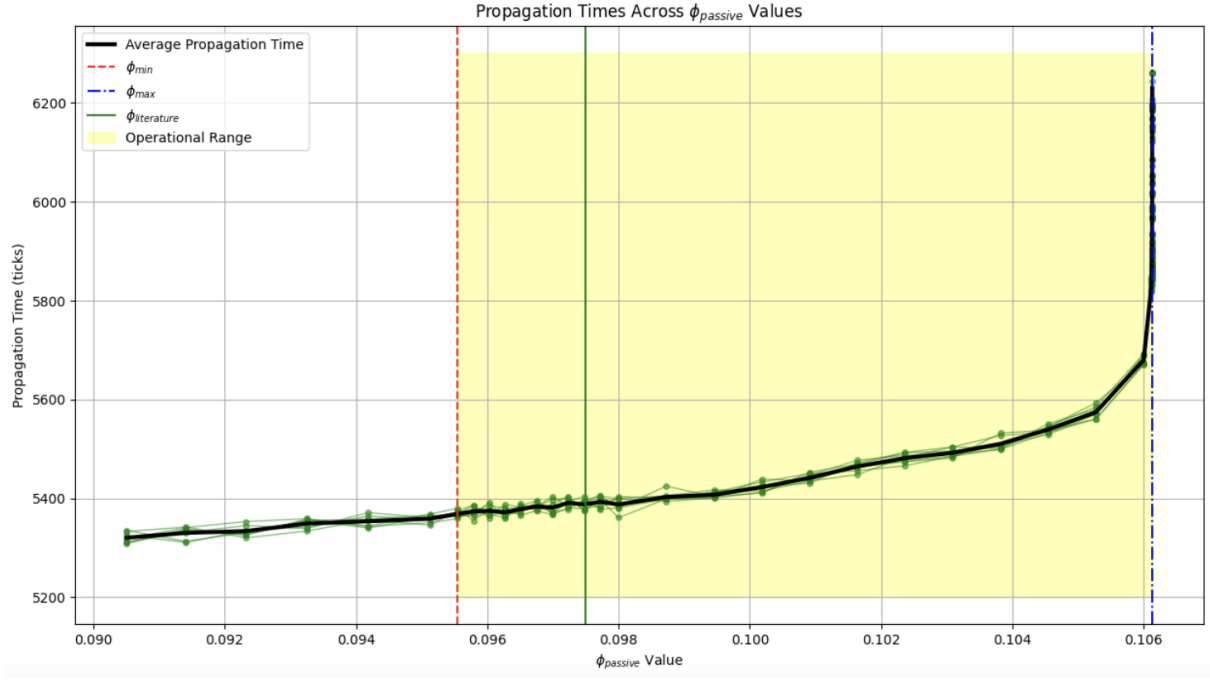


Figure 4.2: Propagation times for various $\phi_{passive}$ values within the limits of ϕ_{min} and ϕ_{max} , illustrating the experimental outcomes of testing a diode in a Belousov–Zhabotinsky reaction. Each point represents a propagation time measurement for a given $\phi_{passive}$, with the average propagation time across measurements shown by a thicker black line. The shaded area between ϕ_{min} (red dashed line) and ϕ_{max} (blue dashed line) highlights the range where the diode works correctly, with the optimal ϕ marked by a solid green line.

angles when tested, so the experiment measured propagation time, but the diode was only partially functional at these values

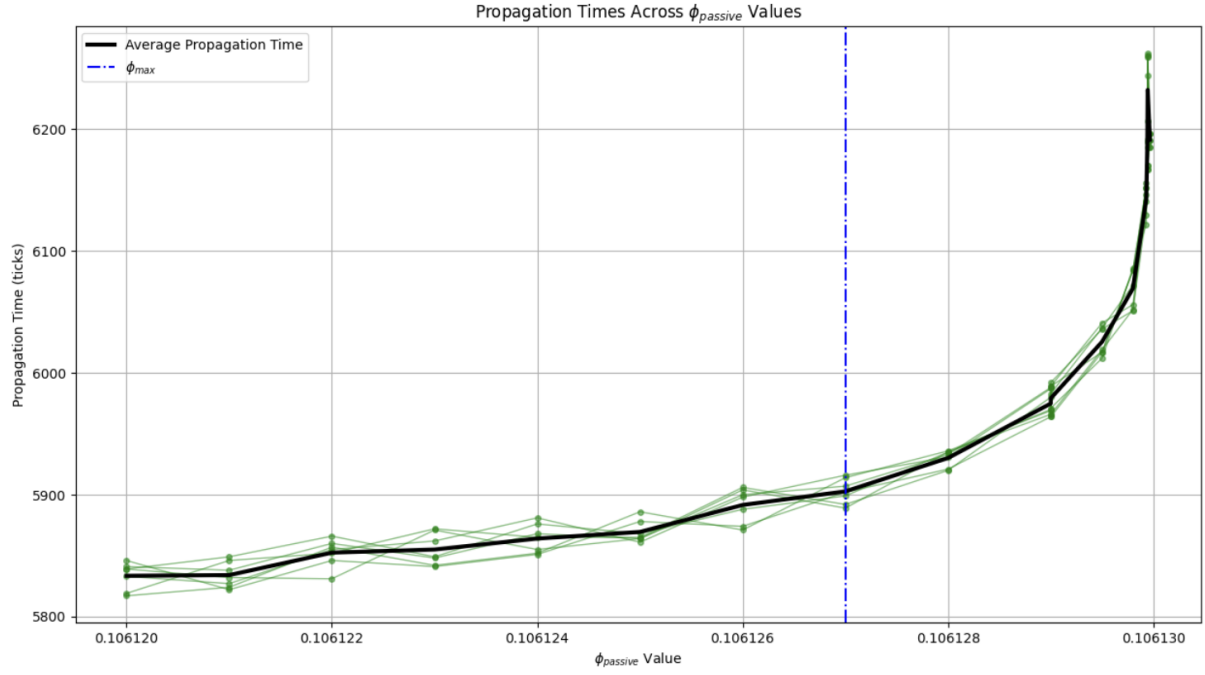


Figure 4.3: Propagation times at ϕ_{max} —

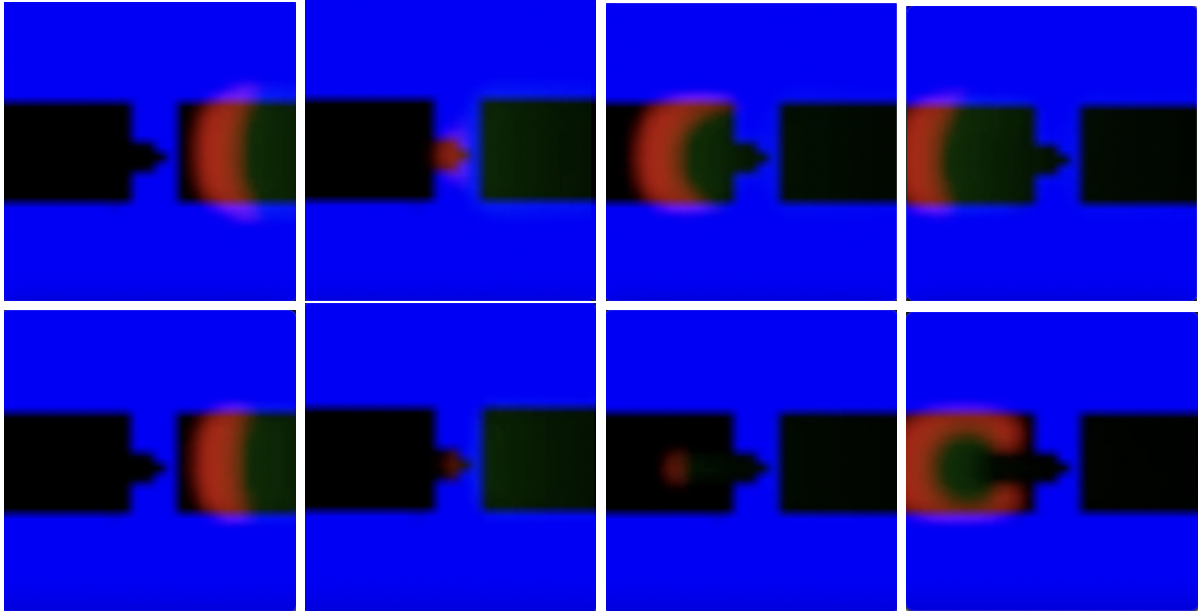


Figure 4.4: Comparison of two waves passing through a diode. Above $\phi_{passive} = \phi_{literature}$, below $\phi_{passive} = \phi_{max}$. Both have different propagation speeds, so they have been equalised for the purpose of comparing their interaction with the diode side by side. The second wave takes substantially more time to form and is about 700 time steps slower as seen in Figure 4.3, which is quite a lot in such simulations.

Bibliography

- Barry, A. L., Coyle, M. B., Thornsberry, C., Gerlach, E. and Hawkinson, R. (1979), ‘Methods of measuring zones of inhibition with the bauer-kirby disk susceptibility test’, *Journal of clinical microbiology* **10**(6), 885–889.
- Cui, J. (2004), *Synchronization of spatiotemporal patterns and modeling disease spreading using excitable media*, West Virginia University.
- Edge, C. (2020), ‘Messing around in parameter space’, YouTube. Available: <https://www.youtube.com/watch?v=4m6mdFfeQMI> [Accessed: (Oct 26, 2023)].
- Edwards (1970), *Physics for O.N.C. Courses*, Elsevier. Chapter 25 - Illumination and Photometry (Pages 473-495).
- Fujilamp (2024), ‘Datasheet for jdc 100v 300w’, http://www.fujilamp.com/html/catalogue_e/jcd_e.html.
- Gorecki, J., Yoshikawa, K. and Igarashi, Y. (2003), ‘On chemical reactors that can count’, *The Journal of Physical Chemistry A* **107**(10), 1664–1669.
URL: <https://doi.org/10.1021/jp021041f>
- Kuhnert, L., Agladze, K. and Krinsky, V. (1989), ‘Image processing using light-sensitive chemical waves’, *Nature* **337**(6204), 244–247.
- Reddy, M. R., Dahlem, M., Zykov, V. and Müller, S. (1995), ‘The effect of an illumination jump on wave propagation in the ru-catalyzed belousov-zhabotinsky reaction’, *Chemical physics letters* **236**(1-2), 111–116.
- Stovold, J. (2019), ‘Rdcsim: a gpu-accelerated, interactive simulator for reaction–diffusion chemistry’.
- Toth, R., Stone, C., Adamatzky, A., de Lacy Costello, B. and Bull, L. (2009), ‘Experimental validation of binary collisions between wave fragments in the photosensitive belousov–zhabotinsky reaction’, *Chaos, Solitons & Fractals* **41**(4), 1605–1615.
URL: <https://www.sciencedirect.com/science/article/pii/S0960077908003032>
- Yamada, Y., Ito, H. and Maeda, S. (2022), ‘Artificial temperature-compensated biological clock using temperature-sensitive belousov–zhabotinsky gels’, *Scientific reports* **12**(1), 22436.
- Zambrano, A., Zadorin, A. S., Rondelez, Y., Estévez-Torres, A. and Galas, J.-C. (2015), ‘Pursuit-and-evasion reaction-diffusion waves in microreactors with tailored geometry’, *The Journal of Physical Chemistry B* **119**(17), 5349–5355. PMID: 25839240.
URL: <https://doi.org/10.1021/jp509474w>

A Creating a Coincidence Detector In Simulation

Developing the AND gate posed a significant challenge as it is highly dependent on specific geometric configurations for proper operation. While it is acknowledged that such a coincidence detector is feasible, the focus of our investigation is on the stability of such simulations in response to environmental changes and the practicality of implementing them in a real environment. Although we acknowledge that building this in a simulation is possible, the focus of this paper is to determine the stability of the simulation in response to environmental changes and the practicality of implementing them in a real environment.

A.1 Unsuccessful Detector

Width	Gap=1px	Gap=2px	Gap=3px	Gap=4px
w=1px	NP	NP	NP	NP
w=2px	NP	NP	NP	NP
w=3px	AP	NP	NP	NP
w=4px	AP	AP	NP	NP
w=5px	AP	AP	NP	NP
w=6px	AP	AP	NP	NP
w=7px	AP	AP	NP	NP
w=8px	AP	AP	NP	NP
w=9px	AP	AP	NP	NP
w=11px	AP	AP	NP	NP
w=12px	AP	AP	NP	NP
w=13px	AP	AP	NP	NP

Table A.1: Results of the experiment showing the relationship between conductor width and gap size.

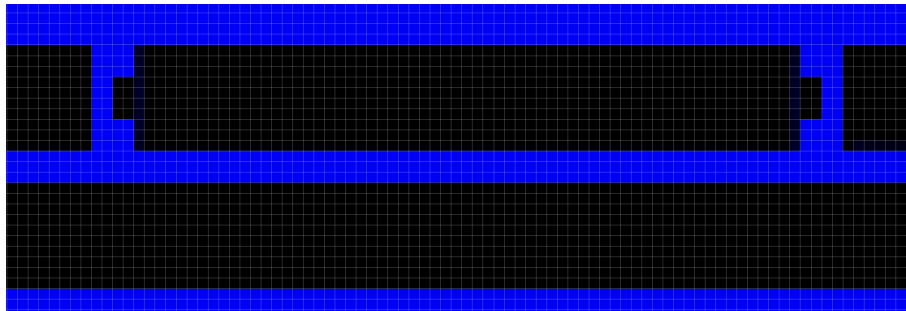


Figure A.1: Initial Coincidence Detector Configuration

In our investigation, we focus on three controllable variables:

- the width of the conductor
- its length before the gap.

The detector configuration is shown in Figure A.1. The primary question we seek to answer is how the width of the conductor influences the functionality of the gates. Specifically, our experiment aims to identify an optimal conductor width that enables the "if" gate to function correctly. This gate's operation relies on two conductors positioned closely, requiring a precise amount of force to facilitate signal transmission from one to the other. This force is generated by the collision of two waves. However, current configurations fail due to the conductors being excessively proximate.

For the purposes of this experiment, we define three outcomes based on the interaction between the waves and the conductors: - Always pass (AP): The signal always passes through the gap. - Never pass (NP): The signal never passes through the gap. - Collision pass (CP): The signal passes through the gap only upon collision of waves.

The CP outcome is of particular interest as it represents the desired state for computational functionality.

Initial tests did not yield CP outcomes, suggesting a potential discrepancy in the values of ϕ_{active} versus $\phi_{passive}$. Adjusting ϕ_{active} to a more aggressive value of 0.035f resulted in CP outcomes for every gap of 3px, marking a significant deviation from the default value of 0.054 proposed in prior literature. This finding prompts further investigation into the range of values between 0.035 and 0.054 to identify a flexible operational window.

A successful configuration identified involves a long charge, a gap of 3px, and a ϕ_{active} value of 0.035. Further exploration is required to determine a viable configuration for a gap of 2px.

Additionally, our observations reveal notable differences in wave behavior depending on orientation; specifically, vertical wave propagation exhibits distinct properties compared to horizontal propagation. A particularly intriguing observation is that waves originating off-center tend to accelerate asymmetrically, favoring one direction over the other, and resulting in a higher likelihood of collision.

A.2 Successful Detector

Section A.1 led to the conclusion that there must be more parameters in play that prevent the detector from functioning, and it was later found that the height of the diode pin on the active medium of the coincidence detector. Following this, it was discovered that by setting the pins higher or lower, we change from where the wave approaches the detector medium, allowing for a finer adjustment of the operation. This created the first successful detector, seen in Figure A.2.

Normally the wave also depends on where it is coming from, but we can ignore that in our case because we are using diodes at entry of the detector medium. Diodes add a new entry point for the incoming wave, so the incoming direction does not really matter.

There is still a small problem with this design where if two waves meet around the centre, none of them gets enough momentum to develop fully and they don't have enough concentration to diffuse into the result substrate, this can be solved by increasing the length of the detector medium to allow for both waves to form, so later designs have a longer detector medium, like in Figure 3.1

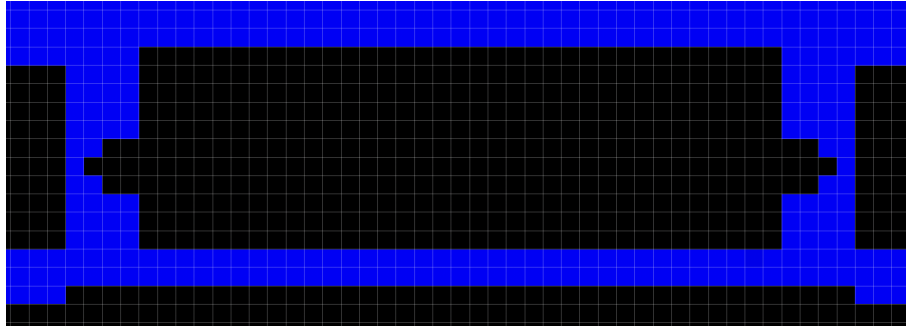


Figure A.2: First Successful Coincidence Detector

The length of the coincidence detector depends on two factors:

1. The minimum distance apart they need to be from each other while still allowing for the waves to form inside the detector.
2. The minimum distance apart they have to be, so that the centre is still usable for collusion.

B Original Project Proposal

B.1 Aims & Objectives

B.1.1 Aims

The aim of the project is to create one of these BZ simulations and add a number of imperfections, measuring how they impact the simulation. These imperfections would typically be found in the real world in the form of dust particles, terrain obstacles or other environmental factors.

The main idea is to create an application using Flutter and to simulate the reaction.

1. Interactive Simulation App Development with Flutter Flutter is a mobile application framework that allows for easy addition of elements of interactivity and allows for lower level control over pixels, which is necessary for creating one of these simulations. The reason for choosing Flutter as opposed to another framework or solution is because of the already present familiarity with the framework.

The simulation shall run on the GPU, so addressing the GPU usage is crucial. The current solutions are to use the recently added `flutter_gpu` package or integrating Dart Foreign Function Interface (FFI) to run native shaders.

2. Creating a game from the simulation. Since the project is focused around imperfections, it could be possible to let the user play with or against the waves (see figure ??) using the `flame` package (minimal game engine).

The simulation has illumination that creates a path, the aim is to create a game where the user has to run away from the waves, but is also constrained to the illuminated area, so for example a single wave could split and then catch the player from both directions.

3. Adding imperfections: One of the main themes of the project is finding imperfections. The goal is to let the player defend himself from the waves using imperfections in the simulation that impact the reaction-diffusion model. This could mean throwing dust particles at the wave or adding temperature to the model that speeds up or slows down the reaction at a particular zone, so that a part of the waves travels more slowly, disrupting the reaction. The aim is to allow the user to only use imperfections as a means of defence.

B.1.2 Challenges

Among the challenges I might face I see myself running into a computational problem where `dt`, being as small as it is, could force me to run the simulation at a very slow rate, which then would make me increase the speed at which it is run, using more CPU, but since the project is *real time*, I cannot afford to go beyond the maximum allowed CPU per frame or I would cause jitter, running code on the GPU should help, but there also needs

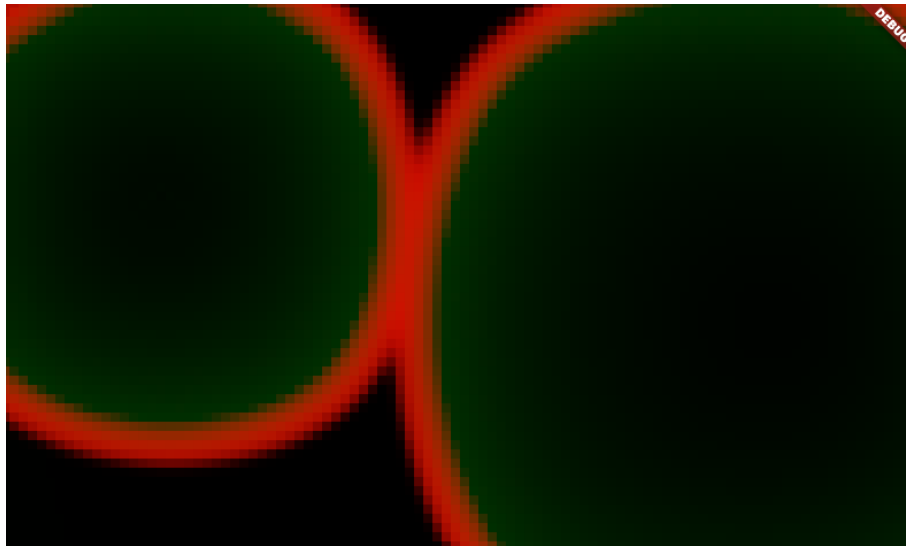


Figure B.1: Expanding wave pattern from this project using the Oregonator model

to be a way to only perform these calculations on a subset of the whole map. Also if this were a game, there could be beacons generating these waves that the user has to destroy. Flutter also has a 2D game engine that I've wanted to explore and that could be a great opportunity. The other reaction-diffusion systems are very scientific or are presented as a video. Zambrano et al. (2015) have used reaction and diffusion to find the shortest path in a maze.

B.2 Related work

The area of reaction-diffusion simulations has been explored well. The simulations of waves are used to produce computing units like counters and logic gates (Gorecki et al., 2003) as well as neurons (Stovold, 2019).

Edge (2020) has created a game (figure B.2) out of the Gray-Scott model where ships battle inside of a goo-like substance and the chemical acts as a dynamic obstacle. The game looks of very high quality despite the lack of interest from the public. Other than this project, there isn't a lot of interest in the joint field of reaction-diffusion simulations and gaming

B.3 Methodologies

1. The project is going to use an agile iterative approach where the phases of the Software Development Life Cycle are performed every sprint.
2. It's important is that the work is broken down into chunks according to the SMART task framework, which stands for Specific, Manageable, Achievable, Relevant, Time-bound. It's important for the tasks to be small and understandable, without any dependencies to other tasks
3. If one task requires the parallel completion of another task, then these tasks should be the same task with two things to do in the task.

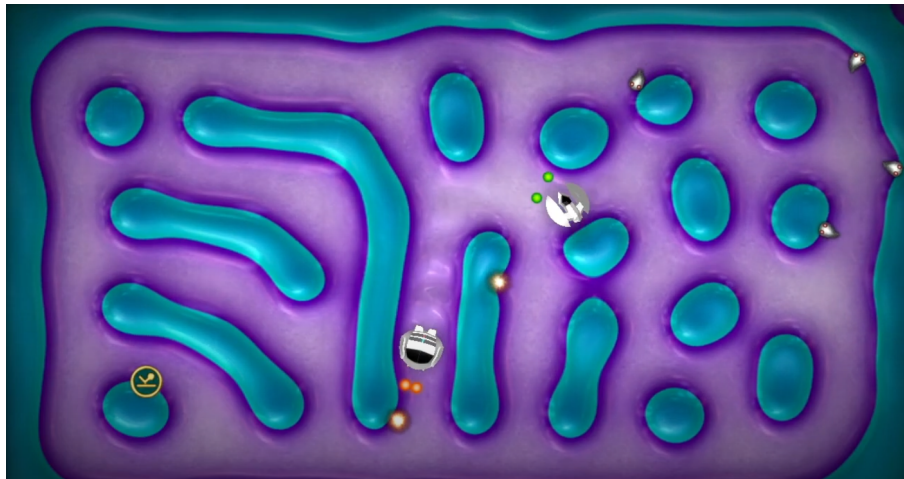


Figure B.2: Reaction Diffusion Game by Chaotic Edge

4. The project is going to use ClickUp as a Project Management System (PMS) due to its flexibility and integration with calendar and task management apps.
5. The most important task management tool to be used in this project is Reclaim.AI, which uses priorities and deadlines to auto schedule your calendar based on your preferences and hours, it has an integration with the most popular PMSs like Jira, ClickUp, Asana, etc.
6. The Project Management System tasks are going to use custom statuses:
 - a) TODO - feature not started
 - b) RESEARCHING - gathering information and reading about feature
 - c) SUBTASKING - feature clear. creating subtasks and checklists to make task execution easier
 - d) IMPLEMENTING - implementing feature
 - e) TO PRESENT - feature implemented. Waiting to be presented to supervisor
 - f) FOR REVISION - supervisor left feedback. Return task to RESEARCHING or IMPLEMENTING
7. The project is going to use the 1 week sprints to divide work into even blocks of work if the backlog ends up being too big. If the backlog remains small, the project is going to use the backlog and not use sprints.

B.4 Programme of Work

The project, being agile, doesn't have strong formal phases, it follows an iterative approach. There are however distinct phases in the planned development of this project (see figure B.3):

1. Planning & Analysis - this phase is going to specify goals of the sprint
2. Design - The project is going to be designed in more detail, that includes library choices, design choices, etc

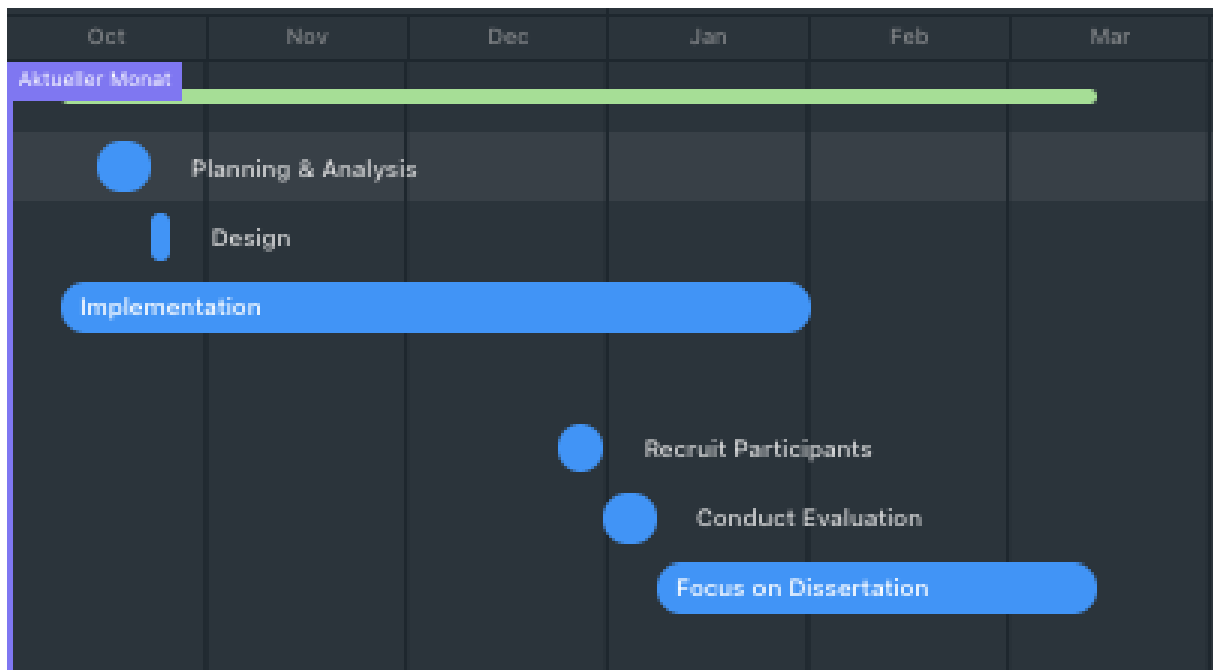


Figure B.3: Gantt Chart

3. Implementation - This phase would take up the majority of the sprint time
4. Testing - last phase before the project transitions into a finalising state
5. Searching for Participants - a stage where about 5 people need to be recruited to perform evaluation
6. Conduct evaluation - the evaluation is going to be performed using heuristics, for example how intuitive the game is, etc.
7. Focus on dissertation - time to shift focus on using the resources and documentation to create the dissertation.

This project is prototypical, so it's going to have reduced planning and design phases.

C Another Appendix Chapter

This could be about your experiments