NTNU
Kunnskap for en bedre verden

TMA4212 - Numerical solution of differential equations by difference methods

Project 1

Reaction-Diffusion Equations

Authors:
Henrik Grønlund
Magnus Rønneseth
Theodoros Xenakis

Date: 28.02.2025

Table of Contents

List of Figures

1 Abstract

Diffusion-reaction partial differential equations (PDEs) are widely used to model phenomena ranging from microscopic chemical reactions to large-scale epidemiological dynamics. This article examines the convergence properties of a modified Crank-Nicholson scheme for solving diffusion-reaction PDEs, demonstrating that the method achieves second-order accuracy in both time and space. Numerical experiments are conducted to confirm these theoretical results. Finally, the applicability of the scheme is illustrated through the solution of a diffusion-reaction system, specifically the spatially extended SIR model.

2 Introduction

When dealing with systems consisting of many interacting components, reaction-diffusion equations emerge. The reaction-diffusion system is a set of partial differential equations (PDEs), and the solution of this system presents spatial patterns, moving fronts, oscillations and other interesting phenomena, which is why it is used to study complex chemical or biological processes (Li A. 2020).

It is precisely this that the first part of the report covers, namely how a reaction-diffusion system, specifically the SIR-model, can simulate the spread of an infectious disease. Afterwards, the report takes a look at a modified version of the Crank-Nicolson scheme and we analyze how it behaves in terms of consistency and stability.

Finally, we test our theoretical implementations and results by doing some numerical experiments. We verify that our error analysis is correct, by finding that the numerical error increases quadratically with respect to the step sizes in the spatial and temporal directions. In this section we also show how our SIR-model can simulate the spread of a disease and how varying different parameters affects its behavior.

3 Theory

In this project we will analyze reaction-diffusion equations of the form

$$u_t = \mu u_{xx} + f(u), \quad u \in \Omega \tag{1}$$

where $\Omega : \{x \in (0,1),\ t \geq 0\}$ and μ is some positive constant. Reaction-diffusion equations are interesting because they describe the behavior of a large range of systems, where we have both time and spatial dependent variables. These equations have diverse applications, but are most notably used to describe chemical reactions, who often have both diffusion and production of a certain material (Hobbs 2025). In general, such formulas can be described as

$$\begin{aligned}\begin{bmatrix}\text{Time rate of change}\\ \text{of concentration}\end{bmatrix} &= \begin{bmatrix}\text{Change in component}\\ \text{due to diffusion}\end{bmatrix}\\ &+ \begin{bmatrix}\text{Rate of concentration} & & \text{Rate of consumption}\\ \text{of component} & - & \text{of component}\end{bmatrix}.\end{aligned}$$

For time-dependent PDE's with diffusion, we know that they should be solved by implicit methods, since explicit methods are too costly to be of interest (Fischer 2025). The method of interest in this report is a modified Crank-Nicholson (CN) scheme,

$$U_m^* = U_m^n + \frac{r}{2}(\delta_x^2 U_m^* + \delta_x^2 U_m^n) + kf(U_m^n), \quad r = \mu\frac{k}{h^2} \tag{2}$$

$$U_m^{n+1} = U_m^* + \frac{k}{2}(f(U_m^*) - f(U_m^n)). \tag{3}$$

To help visualize how a reaction-diffusion equation behaves, one can look at the SIR-epidemiological model (Britton 2012). Consider an infectious disease for which a population, N, can at any given time, t, be divided into three groups: susceptible, S, infective/infectious, I, and removed (or recovered), R. Where a part of the population is classified as removed if they for some reason never again will be infected or infect others. From Britton (2012), the relation of these three variables is described by the following system of ordinary differential equations:

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I \\ \frac{dR}{dt} &= \gamma I \end{aligned} \tag{4}$$

with $N(t) = S(t) + I(t) + R(t)$ constant. Further let's scale our model so that S, I and R represent a fraction of the population, i.e. $S(t) + I(t) + R(t) = 1$. So far, the model only describes how the disease develops over time, *not* in space. A modification to take account for space is proposed in Murray (2003):

$$\begin{aligned} S_t &= -\beta IS + \mu_S \Delta S \\ I_t &= \beta IS - \gamma I + \mu_I \Delta I \end{aligned} \tag{5}$$

where Δ is the Laplacian operator. This model is often referred to as the SIR-model with spatial spread, and is a system of reaction-diffusion equations. The β parameter measures how efficiently the disease transmits from infective to susceptible individuals. The "mortality rate" of an infected individual is γI, in other words, $\frac{1}{\gamma}$ is the life expectancy of an infective. In the originally proposed model, the life expectancy was interpreted as the expected time until an infected individual dies. However, by the introduction of the 'Recovered' state, the life expectancy becomes the expected time until an infected individual recovers from the disease. μ_I and μ_S are diffusion coefficients for infected and susceptible individuals respectively. One can interpret them as the rate at which the individuals move between locations (Murray 2003).

The first thing one often does when trying to solve a PDE such as (5), is to make a finite difference grid, $\mathbb{G}_h$, which is defined as follows:

$$\mathbb{G}_h : \begin{cases} x_m = mh, & h = 1/M \text{ (step size in x-direction)} \\ t_n = nk, & k = T/N \text{ (step size in t-direction)} \end{cases}$$

where $m = [0, 1, \cdots, M]$ and $n = [0, 1, \cdots, T]$.

It is also important to note the difference between $u(x_m, t_n)$ and $U(x_m, t_n)$:

$$\begin{aligned} u(x_m, t_n) = u_m^n, &\quad \text{(exact solution)} \\ U_m^n \approx u_m^n, &\quad \text{(numerical result)} \end{aligned}$$

With our finite difference grid defined and the numerical method established, we now seek to establish the reliability of our approach. This can be achieved by doing an error analysis, where we examine the consistency and stability of our method.

4 Error Analysis

Before implementing the modified CN-method, see 2 and 3, we will do a consistency/stability analysis. The reason for this is that consistency and stability are both measures of the reliability of the method.

We start by investigating the consistency of our method. Given that we started out with the CN method we wish to know if the modification has changed the consistency. When analyzing the consistency of a numerical method, we need to consider the local truncation error, τ_n^n. For the ordinary CN method, we have $\tau_m^n \sim O(k^2 + h^2)$. In our modified method we are using an intermediary point U_m^* and the steps will be; $U_m^n \to U_m^* \to U_m^{n+1}$. Further, it will be assumed that $f(u)$ is linear, such that $f(u) = au$ for a constant a.

To calculate the local truncation error we solve 3 and insert this into 2.

$$\begin{aligned} U_m^* &= \frac{U_m^{n+1} + \frac{ka}{2}U_m^n}{1 + \frac{ka}{2}} \\ \Rightarrow U_m^{n+1} &= \left[1 + ka + \frac{k^2a^2}{2}\right] U_m^n + \frac{r}{2}\left[\delta_x^2 U_m^{n+1} + (1+ka)\delta_x^2 U_m^n\right]. \end{aligned}$$

Applying this to an exact point u_m^n gives,

$$u_m^{n+1} = \left[1 + ka + \frac{k^2a^2}{2}\right] u_m^n + \frac{r}{2}\left[\delta_x^2 u_m^{n+1} + (1+ka)\delta_x^2 u_m^n\right] + k\tau_m^n. \tag{6}$$

Taylor-expansion of the terms gives

$$u_{m\pm1}^n = u_m^n \pm h\partial_x u_m^n + \frac{h^2}{2}\partial_{xx}u_m^n \pm \frac{h^3}{3!}\partial_{xxx}u_m^n + \frac{h^4}{4!}\partial_{xxxx}u_m^n + \cdots$$

$$\begin{aligned} u_{m\pm1}^{n+1} = u_m^n &\pm h\partial_x u_m^n + \frac{h^2}{2}\partial_{xx}u_m^n \pm \frac{h^3}{3!}\partial_{xxx}u_m^n + k\partial_t u_m^n + \frac{k^2}{2}\partial_{tt}u_m^n + \frac{k^3}{3!}\partial_{ttt}u_m^n \\ &\pm hk\partial_{xt}u_m^n + \frac{h^2k}{2}\partial_{xxt}u_m^n \pm \frac{k^2h}{2}\partial_{ttx}u_m^n \\ &+ \frac{h^2}{4!}\partial_{xxxx}u_m^n + \frac{h^2k^2}{12}\partial_{xxtt}u_m^n \pm \frac{h^3k}{3!}\partial_{xxxt}u_m^n \pm \frac{k^3h}{3!}\partial_{xttt}u_m^n + \frac{k^4}{4!}\partial_{tttt}u_m^n + \cdots \end{aligned}$$

$$\delta_x^2 u_m^n = h^2\partial_{xx}u_m^n + + \frac{h^4}{4!}\partial_{xxxx}u_m^n + \cdots$$

$$\delta_x^2 u_m^{n+1} = u_{m+1}^{n+1} - 2u_m^{n+1} + u_{m-1}^{n+1} = h^2\partial_{xx}u_m^n + h^2k\partial_{xxt}u_m^n + \frac{h^2}{12}\partial_{xxxx}u_m^n + \frac{h^2k^2}{6}\partial_{xxtt}u_m^n \cdots.$$

Inserting the terms defined above into 6, gives

$$\begin{aligned} k\tau_m^n + u_m^{n+1} &= \left[1 + ka + \frac{k^2a^2}{2}\right] u_m^n + \frac{r}{2}\left[\delta_x^2 u_m^{n+1} + (1+ka)\delta_x^2 u_m^n\right] \\ \Rightarrow k\tau_m^n &= k\left[u_t - \mu u_{xx} - au\right] + \frac{k^2}{2}\left[u_{tt} - a^2u - \mu u_{xxt} - a\mu u_{xx}\right] + O(k^3 + h^2). \end{aligned} \tag{7}$$

From our PDE, eq. 1, we can take the partial derivative in time, and we get

$$\begin{aligned} u_{tt} &= \mu u_{xxt} + au_t \\ au_t &= u_{tt} - \mu u_{xxt}. \end{aligned}$$

Using this we can cancel out the PDE, and we obtain

$$\begin{aligned} k\tau_m^n = &\frac{k^2a}{2}\left[u_t - \mu u_{xx} - au\right] \\ &+ \frac{k^3}{6}\partial_t^3 u - \mu\left[1 + \frac{ka}{2}\right]\frac{h^2k}{12}\partial_x^4 u - \frac{k^3}{12}\partial_x^2\partial_t^2 u \\ \tau_m^n \sim O(h^2 &+ k^2 + kh^2). \end{aligned}$$

We can therefore conclude that our scheme is 2^{nd} order consistent in both time and space.

Now, we will analyze the stability of our method. To do this, we disregard the error term, and solving eq. 7 for U_m^{n+1} the following formula is found

$$\left[1 - \frac{r}{2}\delta_x^2\right] U_m^{n+1} = \left[1 + ka + \frac{k^2a^2}{2} + \frac{r}{2}(1+ka)\delta_x^2\right] U_m^n.$$

From this point we can use von-Neumann analysis, as we can extend our domain further to get a periodicity of 1 in our boundary conditions, that is $f(x+1) = f(x)$, and $u(x+1,t) = u(x,t)$. Substituting U_m^n with $\xi^n e^{i\beta x_m}$ and dividing by equal terms, gives

$$\begin{gathered}\left[1 + 2r\sin(\beta h/2)\right]\xi = 1 + ka + \frac{k^2a^2}{2} - 2r(1+ka)\sin(\beta h/2) \\ \xi \leq 1 + (a + \frac{a^2h^2}{4\mu})k \leq 1 + \nu k.\end{gathered}$$

While the local truncation error measures the error this method causes after one iteration, it is also interesting to see how the errors accumulate. This can be done by implicitly defining the linear operator $\mathcal{L}_h$ from

$$\mathcal{L}_h u_m^n = \partial_t u_m^n - \mu \partial_{xx} u_m^n = f(u_m^n) + \tau_m^n.$$

By applying the operator on to the numerical approximation one obtains

$$\mathcal{L}_h U_m^n = \frac{1}{k}\nabla_t U_m^{n+1} \frac{\mu}{h^2}\delta_x^2 U_m^n = f(U_m^n)$$

which reduces to

$$\Rightarrow \mathcal{L}_h e_m^n = \mathcal{L}_h(u_m^n - U_m^n) = \tau_m^n.$$

Now define $\overline{\tau} = \max_{\overline{\mathbb{G}}}\{\tau_m^n\}$ and a *corrector* function ϕ with the property that the continuous version of $\mathcal{L}_h$, $\mathcal{L}$, renders the same result when applied to it. That is $\mathcal{L}\phi_m^n = \mathcal{L}_h\phi_m^n \quad \forall\ m, n$. Furthermore let $\mathcal{L}\phi = 1$. Next, define $v_\pm = e_m^n \pm \overline{\tau}\varphi_m^n$ such that,

$$\mathcal{L}_h v_+ = \tau_m^n + \overline{\tau} \geq 0, \quad \forall m, n$$

and

$$\mathcal{L}_h v_- = \tau_m^n - \overline{\tau} \leq 0, \quad \forall m, n.$$

From this form, we can use the discrete minimum and maximum principles, given that we have Dirichlet boundary conditions. As the errors on the bounds of the domain $\partial\mathbb{G}$, are then zero. The discrete maximum principle gives,

$$\begin{aligned} &\mathcal{L}_h v_+ \geq 0 \\ \Rightarrow\ &e_m^n + \overline{\tau}\varphi_m^n \leq \max_{R\in\partial\mathbb{G}}\{e_R + \overline{\tau}\varphi_R\} \leq \overline{\tau}\ \max_{\partial\Omega}\ \varphi \\ &e_m^n \leq \overline{\tau}\left\{\max_{\partial\Omega}\ \varphi\ -\ \min_{\Omega}\ \varphi\right\}. \end{aligned}$$

The same argumentation follows for the minimum principle, and we get

$$\begin{aligned} &\mathcal{L}_h v_- \leq 0 \\ &e_m^n \geq \overline{\tau}\left\{\min_{\partial\Omega}\ \varphi\ -\ \min_{\Omega}\ \varphi\right\} \end{aligned}$$

By using $\varphi(x,t) = -\frac{x^2}{2\mu}$ we ensure that φ satisfies the desired properties, and we have

$$\begin{aligned} \max_{\partial\Omega}\ \varphi &= 0 \\ \min_{\partial\Omega}\ \varphi = \min_{\overline{\Omega}}\ \varphi &= -\frac{1}{2\mu} \end{aligned}$$

This gives, $|e_m^n| \leq |\frac{\bar{\tau}}{2\mu}|$ and our global error has the same order as the local, i.e. is of 2^{nd} order in both space and time.

We conclude this section by formulating our results on the global error in the following theorem.

Theorem. *Consider a diffusion-reaction PDE of the form*

$$u_t = u_{xx} + f(u), \quad x \in (0,1),\ t \in (0,T)$$

with initial condition $u(x,0) = u_0(x)$, and Dirichlet boundary conditions. Let $U_m^n \approx u(mh, nk)$ be the numerical approximation given by the modified Crank-Nicholson numerical scheme defined by 2 and 3 with step lengths $h = 1/M,\ k = T/N$ in space and time respectively. If applied to a sufficiently smooth problem $u(x,t)$, the global error in the numerical approximation satisfies

$$\max_{n \in 0,..,N} ||U^n - u(x,t_n)||_2 \leq C(h^2 + k^2), \quad \text{as } h, k \to 0$$

for some constant $C \geq 0$. That is, the numerical scheme is globally convergent of order two in both time and space.

5 Numerical Experiments

5.1 Error Verification

Now that theorem 4 has been formulated, it would be interesting to see that it can be verified numerically. To do this, we look at the linear Diffusion-Reaction Partial Differential Equation (PDE) given by

$$u_t = \mu u_{xx} + au, \quad x \in (0,1).$$

By separation of variables, the analytic solution of the PDE is

$$u(x,t) = \left[A\cos(kx) + B\sin(kx)e^{-(\mu k^2 - a)t}\right].$$

If we impose the Dirichlet boundary conditions $u(0,t) = u(1,t) = 0$ and the initial condition $u(x,0) = u_0 \sin(\pi x)$ the solution to this can be shown to be

$$u(x,t) = u_0 \sin(\pi x)e^{-(\mu\pi^2 - a)t}. \tag{8}$$

When verifying our theoretical results concerning the local truncation error, τ, we will compare how our modified CN-method differs from the analytical solution, given by 8. To do this, we will solve the PDE with different steps, both in time and space as we are interested in both errors, and calculate the error with respect to the analytical solution. For each stepsize, we find the highest local truncation error, in other words, at each stepsize we take the max norm of the error, $||\tau||_\infty = \max_n |\tau^n| = \max_n ||U^n - u^n||$.

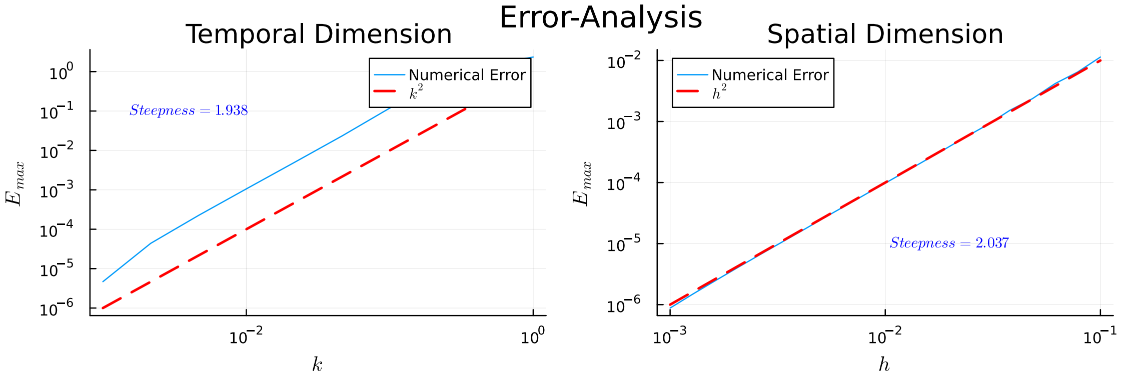

Figure 1: Showing how the numerical error increases quadratically relative to the step sizes, both for time and space.

For the temporal dimension we used $M = 1000$, $h = 1/M = 10^{-3}$, $u_0 = 5$, $a = -0.001$, $\mu = 1$ and varied h from 10^{-3} to 10^{0}. For the spatial dimension we used $N = 20$, $k = 10^{-4}$ $u_0 = 5$, $a = -1$, $\mu = 1$, then varied k from 10^{-3} to 10^{0}. As shown in figure 1, the local truncation error of the modified CN-method is of 2^{nd} order, both in time and in space, thus confirming our theoretical results.

5.2 Infectious Disease

Now, consider the infectious disease model introduced in section 3. To solve this system, we employed a finite difference method with a discretized Laplacian. By doing this we are simplifying our problem from a PDE to an ODE, which we can solve by using a standard ODE-solver, or the modified Crank-Nicholson method, defined in 3.

The model can be solved in both 1-dimension or 2-dimensions, where the only difference lies in the diffusion term. For 1D we have $\Delta S = \partial_x^2 S = S_{xx}$ and $\Delta I = \partial_x^2 I = I_{xx}$, while in 2D we have $\Delta S = S_{xx} + S_{yy}$ and $\Delta I = I_{xx} + I_{yy}$. It is computationally much easier to solve the problem in 1D, but it does not give as much information as a 2D-solution.

In particular, since we want to simulate a real life disease, a 2D solution is what we will implement in this report. Furthermore, we are imposing the Dirichlet boundary conditions with $U_{i,j}^n = 0$ for $i \vee j = 0$. The interpretation of this is that we have an isolated geographical area with no infections on the boundary, e.g. an island.

Using the approach described above to solve our system, and defining $\beta = 3$, $\gamma = 1$, $\mu_I = 0.001$ and $\mu_S = 0.001$, we obtain the following plot:

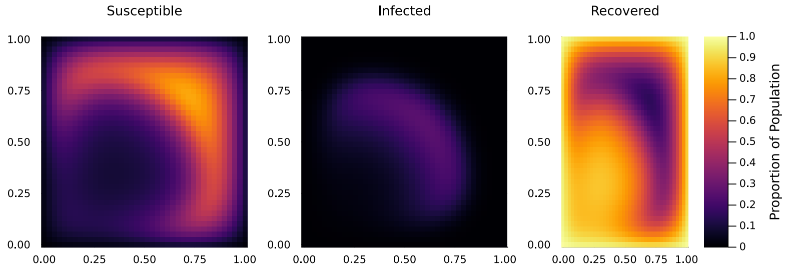

Figure 2: Plot showing the spread of a disease when $\beta, \gamma, \mu_I, \mu_S$ are kept constant throughout the grid.

From figure 2 we see that the disease spreads wavelike from patient zero in the bottom left corner. We also notice that there is a phase shift between the wave in susceptible and infected because in order to be infected, one first has to be susceptible to the disease.

Although these parameters do provide a good representation of how the disease spreads, they may not be entirely realistic when trying to simulate a real-life disease. Since humans do not move with a constant speed in all directions, we introduce spatially varying parameters. More specifically, we create a ”Manhattan-grid”, where the roads have a higher ”traffic” of people, meaning that the disease spreads faster in the roads compared to buildings (or blocks). To do this, we set β to be a matrix, where the elements that represent the road have a higher value than those that represent the buildings, in our case, 50 times larger. Figure 3 visualizes how these changes affect the spread of the disease:

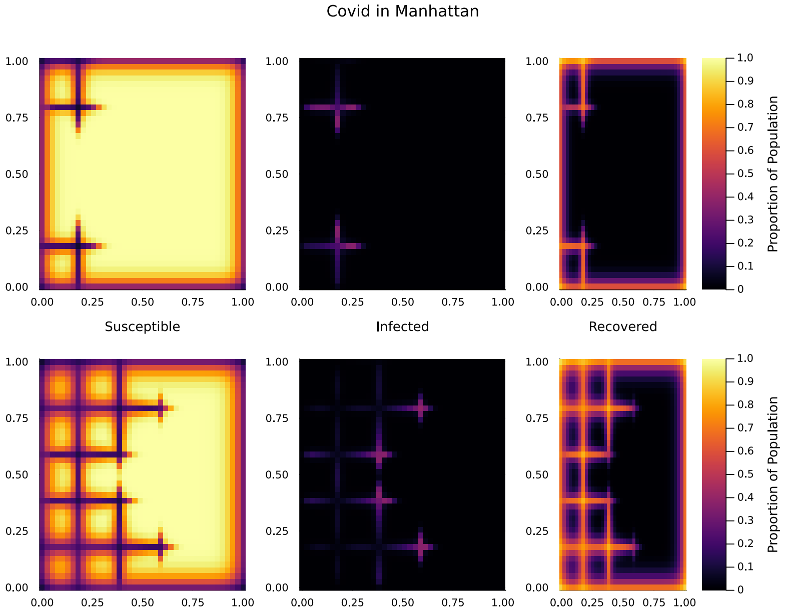

Figure 3: Plot showing how a disease will spread in a "Manhattan" grid. That is, a model where β is set to be spatially dependent, and forms a structure like Manhattan.

Figure 3 shows how a disease would spread in a "Manhattan"-like city. We clearly see that the disease spreads much faster in the roads compared to the buildings. Like figure 2, we see a phase shift between the plots for susceptible and infected.

We can make our model even more realistic by playing around with the parameters. Imagine you have a city with a Manhattan grid, same as in figure 3. We can complicate this city by adding a river that runs diagonally through it, from bottom left to top right, with two bridges that crosses this river, a large bridge and a small bridge. Naturally, the rate of how fast one becomes infected or susceptible, μ_I and μ_S, respectively, will be practically zero in the river (since there are no people there). The bridges will have approximately the same rates as the roads in the grid. If we change our interpretation of β to be that of population density (which makes sense since a higher population density means that the disease spreads more efficiently between susceptible and infected) we can introduce a β-matrix that has higher values in the city-west (taller buildings and more apartments) than in the east. Finally, we construct a hospital somewhere in the city. To model this, we set the γ parameter to be higher in a certain square of the city, meaning that patients recover (or die) faster in this area compared to in the streets and in their homes. A visualization of how the parameters are defined can be seen in figure 4.

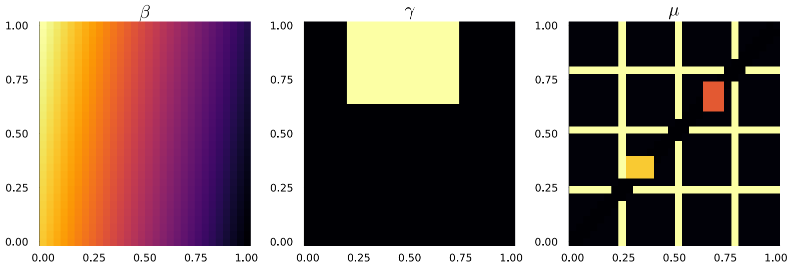

Figure 4: Image showing how β, γ and μ are defined.

Note that the plot of μ in figure 4 only represent the relative difference between the blocks, roads, river and bridges within each μ_S and μ_I matrix. This means that for both μ_I and μ_S the values in the road is K times bigger than in the houses, for some constant K. However, the value of corresponding elements in μ_I and μ_S are different. Using these matrices for β, γ, μ_I and μ_S results in the following simulation:

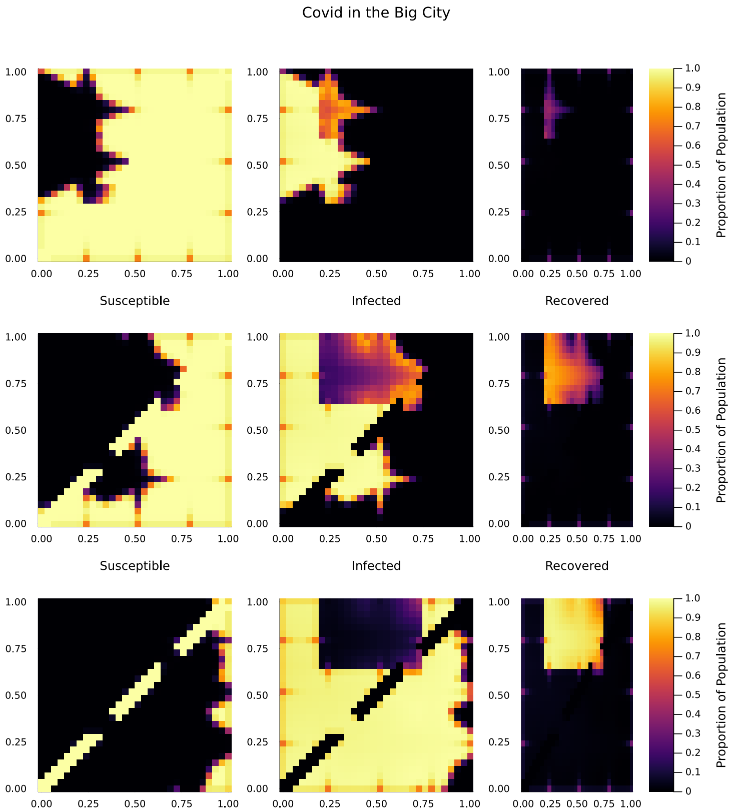

Figure 5: Image showing how the disease spreads in a Manhattan-city with a hospital, river and two bridges.

From figure 5 it is possible to see how adding a hospital, river and two bridges affected the spread of the disease in time and space. Since μ_S is zero in the river, the disease never spreads to this area, thus creating the yellow areas in ”Susceptible” plot. At the bridges however, we see that the disease spreads just like it does in the roads, as we intended. We also see that inserting a hospital affected how quickly people recovered. This highlight the effect of setting the γ value lower in the hospital relative to other areas.

6 Conclusion

In this report we have explored diffusion-reaction PDEs, and numerical methods for solving them.

Particularly, we analyzed a modified Crank-Nicholson numerical scheme, investigating its stability, consistency, and convergence properties. Our findings indicate that the method retains the second-order accuracy in both space and time, similar to the standard Crank-Nicholson scheme. These theoretical results were further validated through numerical experiments.

We then applied our approach to solve an initial/boundary value problem (I/BVP) based on the spatially extended SIR model. The simulations provided valuable insights into how different parameters influence spreading of diseases.

Future extensions of this work could involve introducing time-dependent parameters to better capture real-world dynamics, such as daily commuting patterns, seasonal variations, or periodic outbreaks like the flu. Another potential improvement could be incorporating periodic reintroductions of the disease to simulate recurring epidemics.

Despite these possible extensions, the models presented here effectively illustrate how the SIR model operates in a spatial context and demonstrate the impact of various parameters on the system's behavior.

Bibliography

Britton, N. F. (2012). *Essential Mathematical Biology*. Springer London, pp. 90–91.

Fischer, P. (2025). *notes_time1c*. URL: http://fischerp.cs.illinois.edu/tam470/refs/notes_time1c.pdf (visited on 26th Feb. 2025).

Hobbs Bruce E. & Cecília, Joan (2025). *Reaction-Diffusion Equation*. URL: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/reaction-diffusion-equation (visited on 26th Feb. 2025).

Li A. Chen R., Farimani A.B. et al. (2020). 'Reaction diffusion system prediction based on convolutional neural network'. In: *Scientific Report* 10, p. 3894.

Murray, J. D. (2003). *Mathematical Biology II*. Springer New York, pp. 661–662.

Appendix

A AI Declaration

In conducting this project we've taken use of general artificial intelligence (GAI) through the use of ChatGPT.

We have not used software such as "Github Copilot" for generating code. However, as we have been programming in a language that is relatively new to us, we have occasionally used ChatGPT for translating code from Python to Julia. As we had an interest in learning Julia, we have been very *selective* with this translation. That is, we have not asked ChatGPT to translate complete functions or large parts of code. An example of what we did ask ChatGPT to translate was the code used to create a tridiagonal matrix

```
np.diag(a*e[1:], -1) + np.diag(b*e, 0) + np.diag(c*e[1:], 1)
```

Summarising scientific knowledge in a written report is a valuable skill that we ought to hone and develop. As such, we have been careful not to use GAI for generation of *new* text for the report. However, we have at several points in the writing process used the technology for proofreading, and suggestions of alternative formulations.

In summary, the usage of ChatGPT has been restricted to:

- Proofreading of written text and suggestions of alternative formulations.
- Translation of small code chunks from Python to Julia.
- Suggestion of bug-fixes for erroneous code.
- Finding sources, much like one would use a search engine like *Google*.

This means that we *have not* used GAI for:

- Generation of new text for this report.
- Generation of new code.
- Mathematical analysis of the PDE and it's properties.
- Mathematical analysis of the numerical methods used.

B Code Structure

Attached to this report there is a compressed folder containing the code used for the numerical analysis and implementation. Within the folder there are also two files "Project.toml" and "Manifest.toml", that contains information about the packages and environment used to run the code. Full information about how to set up an equivalent environment is contained in the "README"-file.

The numerical convergence verification - i.e. the parts concerned with problem 1 - is fully implemented in the Jupyter notebook "modCN_numExp.ipynb".

The functions concerned with the applied epidemiological model (problem 2) are contained in the Julia file "SIR2D.jl". Specific I/BVP examples, their solutions, and graphical presentations are contained in the notebook "SIR2D.ipynb".