Scientific Computing Exercise Set

Karlijn Limpens (10615342) & Jop Keuning (11014407)

I. Introduction

In this report we will have a look at the discretization of both the wave equation and diffusion equation. For the diffusion equation we will evaluate both time dependent and independent solutions. It is not always possible to calculate a analytical solution, this makes it difficult to use computers to calculate certain phenomena. To circumvent this problem we can use discretization of space. In this report we will use both spacial and temporal discretization, for both a 1D problem and a 2D problem.

II. Vibrating string

A wave through a 1D string can be described using equation 1. Where $\Psi(t,x)$ is the vibration amplitude, c is a constant, t represents time and x represents position. In this problem we use the grid points consist of discrete nodal points at which Ψ is evaluated.

$$\frac{\delta^2\Psi(t,x)}{\delta t^2} = c^2\frac{\delta^2\Psi(t,x)}{\delta x^2} \tag{1}$$

A. Discretization String

To simulate a wave through a string we want to discretize the wave equation 1. Which results in cutting the length L of the string into N similar pieces which gives us $\Delta x = L/N$. We have two boundary conditions $\Psi(x=0,t)=0$ and $\Psi(x=L,t)=0$. To implement the wave equation in a computer we use the stepping method to generate the next step in our computation which the Ψ at the two most recent time points, Y and Y_{prev}, to calculate the next Y_{next} one.

$$Y_{next} = c^2\frac{\Delta t^2}{\Delta x^2}(Y[i+1]+Y[i-1]-2Y[i])-Y_{prev}[i]+2Y[i] \tag{2}$$

To implement the boundary conditions we make sure we reset $\Psi(x=0)=0$ and $\Psi(x=L)=0$ for each time step.

B. Initialising wave

We will initialise three different waves, $\Psi(x,t=0) = sin(2\pi x)$, $\Psi(x,t=0) = sin(5\pi x)$ and $\Psi(x,t=0) = sin(5\pi x)$ if $1/5 < x < 2/5$, else $\Psi = 0$. With the use of equation 2 we will evaluation the development of the waves. We will use $c=1$, $L=1$, $N=100$ and $\Delta t = 0.001$.

C. wave development

In figures 1-3 we see the development of the different waves at different time steps. For an animation of the wave function see the file named wave_anamition.mp4

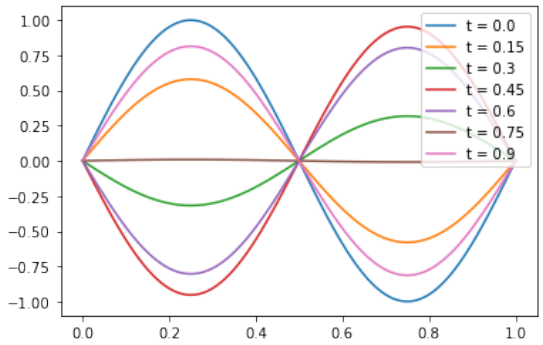

Fig. 1: The development of the initial wave: $\Psi(x,t=0) = sin(2\pi x)$ over time. Using the discretized wave equation

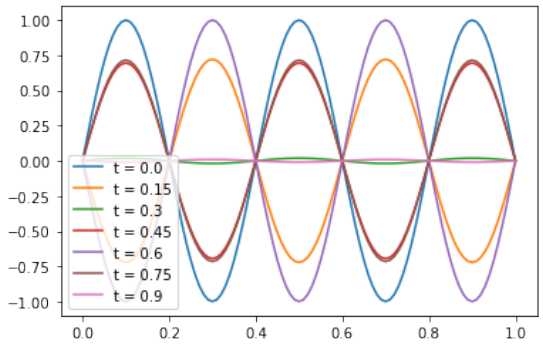

Fig. 2: The development of the initial wave: $\Psi(x,t=0) = sin(5\pi x)$ over time. Using the discretized wave equation

III. Time dependent Diffusion Equation

To simulate a diffusion of a substance through a field we need to discretize the 2D time dependent diffusion

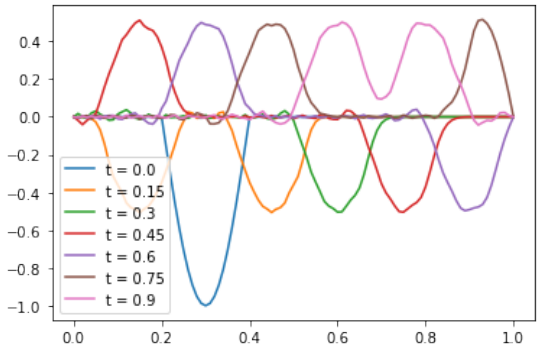

Fig. 3: The development of the initial wave: $\Psi(x, t = 0) = sin(2\pi x)$ if $1/5 < x < 2/5$, else $\Psi = 0$, over time. Using the discretized wave equation

equation 3 where c is the concentration at time t and D is the diffusion constant set at a value of 1.

$$\frac{\delta c}{\delta t} = D\nabla^2 c \tag{3}$$

The first step in doing this is to set the boundary conditions for the field. $c(x, y = 1, t) = 1$ and $c(x, y = 0, t) = 0$ constitute the vertical boundaries of the field. This means that at the top of our field, the concentration will start and stay at a value of 1 and at the bottom our field keep a value of 0. This will result in a diffusion that will go from top to bottom. For the horizontal boundaries we take $c(x = 0, y, t) = c(x = 1, y, t)$. Which means that the value $c((x = 1) + \Delta x, y, t)$ is the same as the $c(x = 0, y, t)$.

In our initial field we set all values equal to $c = 0$, except for the top row where the value is set at $c = 1$. What we want to calculate is the concentration c at each combination (x, y) at the next time step, c. We will set $\delta x = \delta y = \frac{1}{N}$ where $x = i\delta x$ and $y = j\delta y$, where $i, j \in (0, 1, 2 \dots N)$ and the time interval δt where $t = k\delta t$ with $k \in \mathbb{N}$ which can be represented in discretized form $c(i\delta x, j\delta x; k\delta t) \equiv c^k_{i,j}$ for our field. After this, we can discretize the diffusion equation 3 to the form, 4. This equation is stable if the condition $\frac{4\delta t D}{\delta x^2} \leqslant 1$ holds.

$$c^{k+1}_{i,j} = c^k_{i,j} + \frac{\delta t D}{\delta x^2}\left(c^k_{i+1,j} + c^k_{i-1,j} + c^k_{i,j+1} + c^k_{i,j-1} - 4c^k_{i,j}\right) \tag{4}$$

D. Time dependent Diffusion equation

This discrete equation is not yet complete we still need to adjust for the boundary conditions. The function needs only to be adjusted for the horizontal boundaries as we can simply not apply the function to the vertical boundaries, leaving them at a value of 1 for $y = 1$ and 0 for $y = 0$ as the boundary conditions stipulate. For the horizontal boundary conditions we need to reformulate how we calculate the values at $x = 0$ and $x = 1$.

$$c^{k+1}_{i,j} = c^k_{i,j} + \frac{\delta t D}{\delta x^2}\left(c^k_{i+1,j} + c^k_{N,j} + c^k_{i,j+1} + c^k_{i,j-1} - 4c^k_{i,j}\right) \tag{5}$$

$$c^{k+1}_{i,j} = c^k_{i,j} + \frac{\delta t D}{\delta x^2}\left(c^k_{0,j} + c^k_{i-1,j} + c^k_{i,j+1} + c^k_{i,j-1} - 4c^k_{i,j}\right) \tag{6}$$

For the boundary at $x = 0$ the adjustment to the discrete function, instead of taking the concentration value at $x = 0 - \delta x$ we take the value at $x = N$. In equation 5 we show how we calculated this x value. For the boundary at $x = 1$ the adjustment to the discrete function, instead of taking the concentration value at $x = +1\delta x$ we take the value at $x = 0$, as can be seen in equation 6.

E. Model Verification

To check the correctness of our time dependent diffusion simulation we also calculate the analytic solution for this model using equation 7.

$$c(y, t) = \sum_{i=o}^{\infty} erfc\left(\frac{1 - y + 2i}{2\sqrt{Dt}}\right) - erfc\left(\frac{1 + y + 2i}{2\sqrt{Dt}}\right) \tag{7}$$

In Table I we show the difference between out numerical results and the analytical results. We see small difference between the two computations. This shows us that our iterative model is a good estimation.

TABLE I: Concentration values for diffusion calculated both numerical and analytical and the difference between those calculations. For different value of y at $t = 0.5$

y	0.8	0.6	0.4	0.2
Analytical	$4.20e^{-7}$	0.000148	0.01141	0.20590
Numerical	$4.13e^{-7}$	0.000149	0.01150	0.20656
Δc	$-0.07e^{-7}$	$1e^{-6}$	$0.9e^{-6}$	$6.6e^{-4}$

F.

In figure 4 we show the development of the diffusion within our model for different time steps. At $t = 1$ we see a stabilised model which can be interpreted as a steady flow from the top of our grid to the bottom.

G.

For an animation of the wave function see the file named diffusion_anamition.mp4

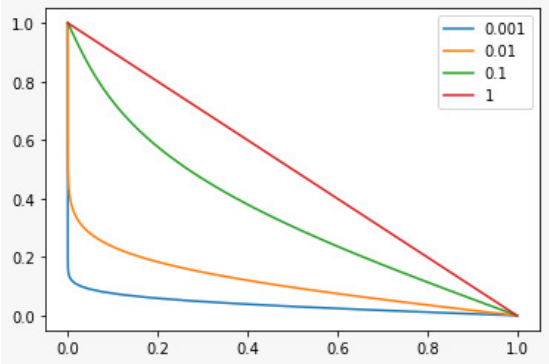

Fig. 4: The numerical solution of the concentration as a function of the y-coordinates, for t equal to 0.001, 0.01, 0.1 and 1. with D = 1

IV. Diffusion equations

In this section we will evaluate different diffusion iteration methods. Beginning with the Jacobian iteration. Equation 8 is used to generate the concentration c for the next iteration $k+1$ depending on the neighbouring values of c from the previous iteration k.

$$c_{i,j}^{k+1} = \frac{1}{4}(c_{i+1,j}^{k} + c_{i-1,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k}) \quad (8)$$

Second we have the Gauss-Seidel (GS) iteration which uses equation 9. Which also calculates the concentration of the neighbouring values but used the already updated values of c from iteration $k+1$ when they are available.

$$c_{i,j}^{k+1} = \frac{1}{4}(c_{i+1,j}^{k+1} + c_{i-1,j}^{k} + c_{i,j+1}^{k+1} + c_{i,j-1}^{k}) \quad (9)$$

Lastly we have the successive over relaxation (SOR) iteration method. For which we use equation 10, this model introduces a values of ω, when ω would be set to 1 we will again find the GS iteration model. The SOR iteration model will only converge for $0 < \omega < 2$ with an optimisation in the convergence can be found with a value somewhere between $1.7 < \omega < 2$.

$$c_{i,j}^{k+1} = \frac{\omega}{4}(c_{i+1,j}^{k+1} + c_{i-1,j}^{k} + c_{i,j+1}^{k+1} + c_{i,j-1}^{k}) - (1-\omega)c_{i,j}^{k} \quad (10)$$

For all three iteration model we need a stopping condition. In equation 11 we formulated our stopping condition. When our δ becomes smaller than ϵ we will stop our calculations. For ϵ we used a values of 10^{-5}

$$\delta \equiv max_{i,j}|c_{i,j}^{k+!} - c_{i,j}^{k}| < \epsilon \quad (11)$$

H.

After implementing all three interation method and using $N = 50$ we evaluated the final diffusion of the concentrations. In figure 5 we see the final concentration values found by the three different iteration methods and the analytical method. In figure 6 a close up is shows where we see that the Jacobian and SOR iteration models are slightly of from the analytical results. The GS iteration method seems to almost overlap with the analytical values.

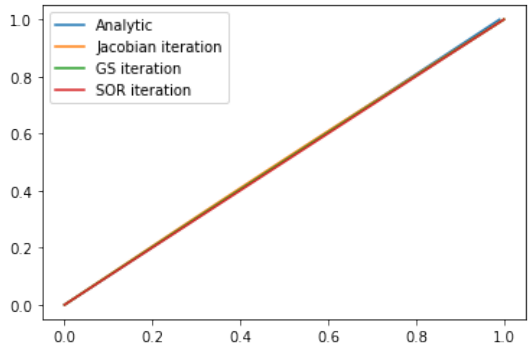

Fig. 5: the final concentration values calculated with different iteration method and the analytic values. For the SOR model we used $\omega = 1.8$

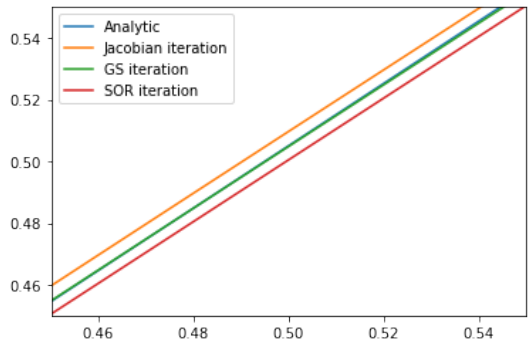

Fig. 6: a close up of the final concentration values calculated with different iteration method and the analytic values. For the SOR model we used $\omega = 1.8$

I. Convergence

The three iteration method all converge faster than the time dependant function as can be seen in figure 7. All three are logarithmic dependent on the number of iterations for their convergence but not to the same scale. The SOR method converges about 5 times quicker than the Gaus-Seidel method. The Guas-Seidel method in it's turn converges twice as fast as the Jacobian method.

This allows for the conclusion that δ is logarithmicly dependent on the amount of iterations.

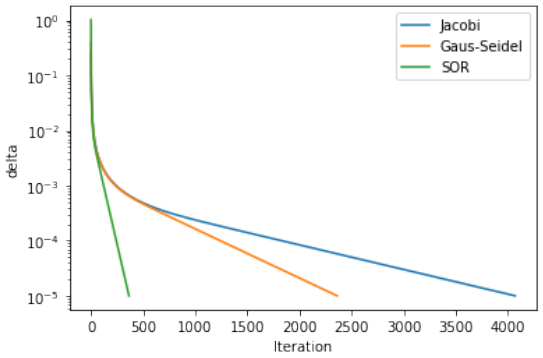

Fig. 7: Convergence of the methods

J.ω optimisation

To find the optimal value for ω in the SOR method, and see how it depends on N, the SOR method was performed for different values of ω. We tried values in the range $1.7 < \omega < 2$ and for each value of ω the SOR method was also performed with different values for N. The results can be seen in figures 8-11. In these figures we find that an ω value of 1.9 has the best performance with an N of 50 but for the higher values of N the best performing SOR method have an ω of 1.95, which performs only slightly worse with $N = 50$. In all four figures it also becomes clear that the convergence is linearly dependent on the size of N.

K. Objects within diffusion models

Lastly we will introduce objects to our SOR-iterative diffusion model. These objects will function as sinks, meaning that their concentration values will always stay zero. In figures 12 to 15 different obstacles are introduced. We see that the number of iterations differ for the different shapes that where introduced. It also clear that the higher up the obstacles are in the grid the lower the concentration is below the obstacles. Also a smaller sink area results into a higher overall concentration, as can be seen in figure 12 compared to the other figures.

We also see that a big and high placed obstacle (figure 15) results less iterations for convergence. When this is compared to multiple obstacles in Figures 14 abd 13 for which the highest number of iterations we see that a extra obstacle complicates the model and results in more iterations before convergence.

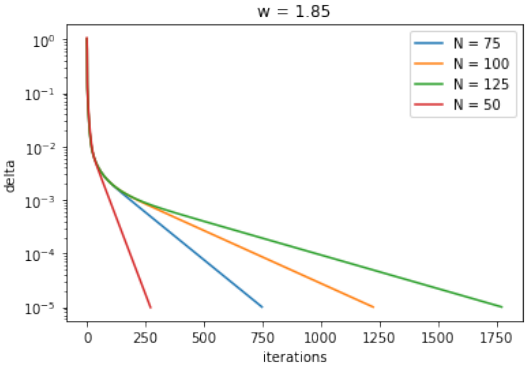

Fig. 8: $\omega = 1.85$

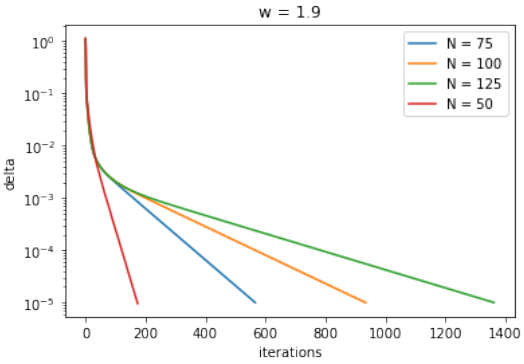

Fig. 9: $\omega = 1.9$

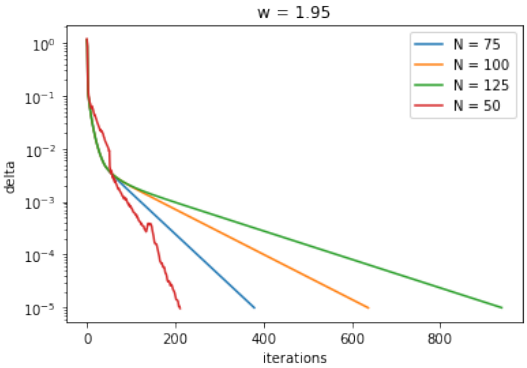

Fig. 10: $\omega = 1.95$

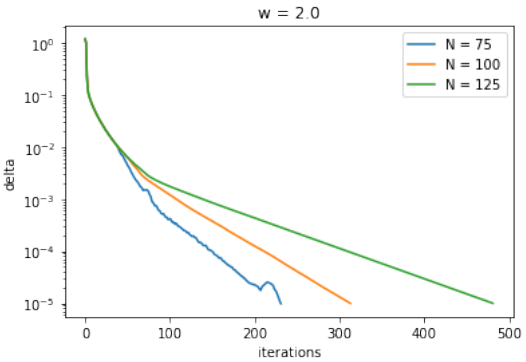

Fig. 11: $\omega = 2$

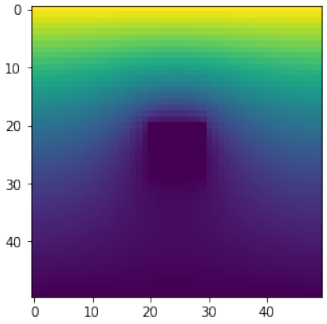

Fig. 12: k = 243

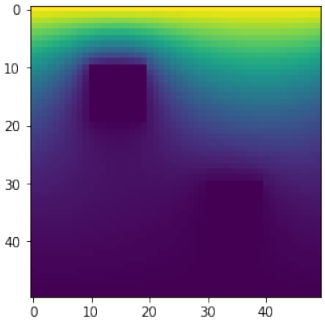

Fig. 13: k = 258

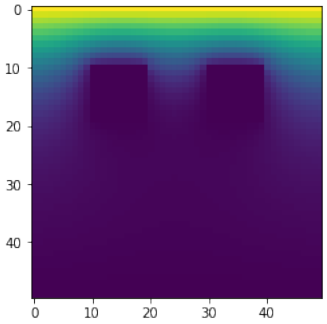

Fig. 14: k = 255

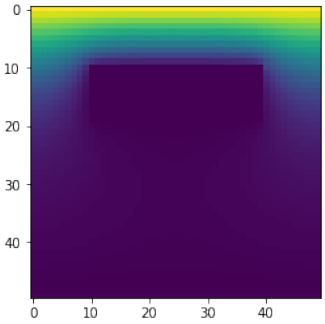

Fig. 15: k = 236