Scientific Computing Exercise Set 1

Romy Meester (11140046), and Natasja Wezel (11027649)

I. Introduction

THE wave and diffusion equations are very important formulas in physics to calculate and analyze various fundamental observations. In this research, we derive numerical solutions to these partial differential equations by introducing a uniform discretization of the spatial and temporal domains, and approximating the partial differential equation by finite difference expressions. With this discretization, we are able to describe the vibration of a uniform string and the diffusion of some substance through a grid.

The aim of this research is to study the wave equation, time dependent diffusion equation, and time independent diffusion equations including the Jacobi iteration, Gauss-Seidel iteration, and Successive Over Relaxation (SOR). This study therefore pursues the question: *What is the behaviour of the wave equation and the diffusion equation?*

The one-dimensional wave equation was used to describe the vibrations of a uniform string through time. The string starts at rest from different starting positions. The time dependent diffusion equation was used to describe the spreading of some resource from top (source) to bottom (sink) over a certain domain. Three other methods, the Jacobi method, the Gauss-Seidel method and the Successive Over Relaxation (SOR) method were also used to describe diffusion. The methods are compared to each other, and to the analytic solution of the diffusion equation.

II. Theory and Methods

In this assignment, we practice with discretizing partial differential equations. In the first part (see section III-A) we solve the one-dimensional wave equation, given in equation 1.

$$\frac{\partial^2 \psi}{\partial t^2} = c^2 \frac{\partial^2 \psi}{\partial x^2} \tag{1}$$

We attempt a numerical solution method by introducing a uniform discretization of the spatial and temporal domains, and approximating the partial differential equation by finite difference expressions. The grid points of this problem consist of discrete nodal points at which this function needs to be evaluated.

In the second part (section III-B), we solve the two-dimensional time dependent diffusion equation, given in equation 2.

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

Here, $c(x, y; t)$ is the concentration as a function of the coordinates x and y and time t. D is the diffusion constant. We will always consider a square domain with $0 \leq$ x, y ≤ 1 with the boundary conditions given in equation 3 and periodic boundaries given in equation 4.

$$c(x, y = 1; t) = 1 \text{ and } c(x, y = 0; t) = 0 \tag{3}$$

$$c(x = 0, y; t) = c(x = 1, y; t) \tag{4}$$

Moreover, the initial conditions are:

$$c(x, y; t = 0) = 0 \text{ for } 0 \leq x \leq 1, 0 \leq y \leq 1. \tag{5}$$

With this set of initial and boundary conditions, the solution only depends on the y-coordinate (because of symmetry) and on time. Also, for this set of conditions,

the diffusion equation can be solved analytically with the following equation:

$$c(x,t) = \sum_{i=0}^{\infty} \text{erfc}\left(\frac{1-x+2i}{2\sqrt{Dt}}\right) - \text{erfc}\left(\frac{1+x+2i}{2\sqrt{Dt}}\right) \quad (6)$$

Note that for $t \to \infty$ the concentration profile is simply a straight line,

$$\lim_{x\to\infty} c(y,t) = y. \quad (7)$$

The spatial and temporal domains are discretized. The x- and y-axes are divided in intervals with length δx. We take N intervals in the x- and y- directions, meaning that we have $\delta x = 1/N$, and $x = i\delta x, y = j\delta x$ where $i, j \in (0, 1, 2, ...N)$. Moreover, we assume a small time interval δt, so that $t = k\delta t$, where $k \in \mathbb{N}$. We want to find the solution to the concentration at these discretized space and time points. With the definition

$$c(i\delta x, j\delta x; k\delta t) \equiv c_{i,j}^k \quad (8)$$

and discretizing the diffusion equation using standard finite difference methods the following scheme is derived:

$$c_{i,j}^{k+1} = c_{i,j}^k + \frac{\partial t D}{\partial x^2} (c_{i+1,j}^k + c_{i-1,j}^k + c_{i,j+1}^k + c_{i,j-1}^k - 4c_{i,j}^k) \quad (9)$$

This scheme has a stable solution when

$$\frac{4\delta t D}{\delta x^2} \leq 1 \quad (10)$$

We will implement a finite difference simulation to solve the time dependent diffusion equation, using the finite difference scheme (eq. 9).

Moreover, in the third part (section III-C), we directly solve the time independent diffusion equation,

$$\nabla^2 c = 0, \quad (11)$$

which is the Laplace equation. We assume the same boundary conditions as earlier stated (equation 3), the same spatial discretization, and the same 5-points stencil for the second order derivatives. Now, we consider an iterative method to derive the set of finite difference equation,

$$c_{i,j} = \frac{1}{4}(c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1}) \quad (12)$$

From this equation, the Jacobian iteration is suggested using the k to denote the k-th iteration,

$$c_{i,j}^{k+1} = \frac{1}{4}(c_{i+1,j}^k + c_{i-1,j}^k + c_{i,j+1}^k + c_{i,j-1}^k) \quad (13)$$

Moreover, a stopping condition is assumed such that the solution is assumed to be converged if for all values of (i,j)

$$\delta \equiv \max_{i,j} \left| c_{i,j}^{k+1} - c_{i,j}^k \right| < \epsilon, \quad (14)$$

where ϵ is some small number, say 10^-5.

An improvement over the Jacobi iteration is the Gauss-Seidel iteration, where during the iteration a new value is used as soon as it has been calculated. Assuming that the iteration proceeds along the rows (i.e. incrementing i for fixed j), the iteration becomes:

$$c_{i,j}^{k+1} = \frac{1}{4}(c_{i+1,j}^k + c_{i-1,j}^{k+1} + c_{i,j+1}^k + c_{i,j-1}^{k+1}) \quad (15)$$

Last but not least, the Successive Over Relaxation (SOR) is derived from the Gauss-Seidel iteration by an over-correction of the new iterate:

$$c_{i,j}^{k+1} = \frac{1}{\omega}(c_{i+1,j}^k + c_{i-1,j}^{k+1} + c_{i,j+1}^k + c_{i,j-1}^{k+1}) + (1-\omega)c_{i,j}^k \quad (16)$$

This method converges only for $0 < \omega < 2$. Moreover, for $\omega = 1$, we recover the Gauss-Seidel iteration, and for $\omega < 1$ the method is called under-relaxation. The optimal ω (that minimizes the number of iterations) for our diffusion problems turns out to be somewhere between $1.7 < \omega < 2$. The exact value depends on the grid size N.

III. Results

A. Vibrating string

To define the discretized wave equation, we derived the vibration amplitude $\psi(x,t)$ from equation 1, and made a discrete approximation of the continuous space

and time. In this way, $\psi(x,t) \equiv U(i,j)$. We assumed fixed boundaries with a string of length $L = 1$, and an interval length of $\Delta x = L/N$ with $N = 100$ intervals. By using the first order forward (i.e. including $u(i+1,j)$), backward (i.e. including $u(i-1,j)$) partial equation, and second order centered partial equation of the string (e.g. $\frac{\partial^2\psi}{\partial t^2}$), we were able to equalize $\frac{\partial^2\psi}{\partial t^2} = \frac{\partial^2\psi}{\partial x^2}$:

$$\frac{u(i,j+1) + u(i,j-1) - 2u(i,j)}{\triangle t^2} = c^2 \frac{u(i+1,j) + u(i-1,j) - 2u(i,j)}{\triangle x^2} \quad (17)$$

As a result, we encountered the discretized wave equation:

$$u(i,j+1) = \left(\frac{\triangle t c}{\triangle x}\right)^2 (u(i+1,j) + u(i-1,j) - 2u(i,j)) - u(i,j-1) + 2u(i,j) \quad (18)$$

At the boundary cases (both ends of the string), the wave is always 0, so we do not update these points.

The time development ($\delta t = 0.001$) of the string with various initial conditions as the $sin(2\pi x)$, $sin(5\pi x)$, and $sin(5\pi x)$ with a constrained domain, is shown in figure 1, 2, and 3, respectively. In each of the graphs, we illustrated from $t = 0$ to $t = 1.9$ with a time difference of 1/1000 step each time.

The bold blue line stands out, because it shows the vibrating string at rest when $t = 0$. At $t = 0.5$, $sin(2\pi x)$ is perfectly reflected at the x-axis, whereas $sin(5\pi x)$ is way faster. Due to the constrained domain, the sinusoidal of figure 3 is partly shown at rest when t is also 0 and 1. However, out of that domain, the amplitude has been halved in comparison to the other ones.

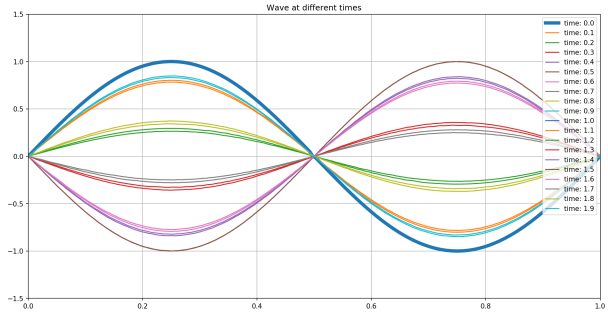

Fig. 1: The vibrating string of $sin(2\pi x)$. The string is at rest at $t = 0$. $c = 1$, and $\Delta t = 0.001$.

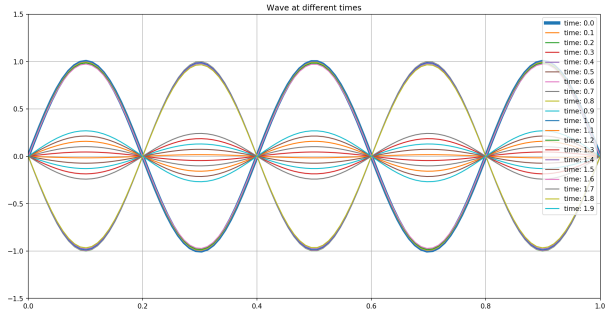

Fig. 2: The vibrating string of $sin(5\pi x)$. The string is at rest at $t = 0$. $c = 1$, and $\Delta t = 0.001$.

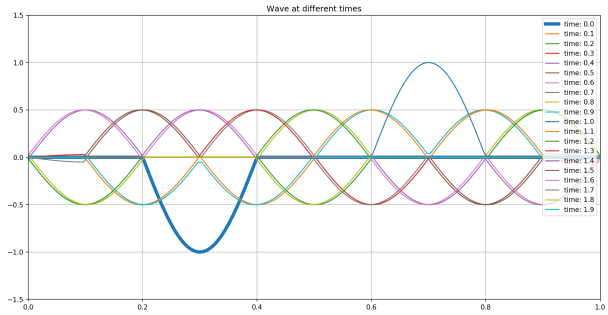

Fig. 3: The vibrating string of $sin(5\pi x)$ with the constrained domain if $1/5 < x < 2/5$, else $\Psi = 0$. The string is at rest at $t = 0$. $c = 1$, and $\Delta t = 0.001$.

In view of the plots, the animation of the time development can be seen in the files folder.

B. The time dependent diffusion equation

For the source (top) of the domain, we used the $c(x, y = 1; t) = 1$ boundary condition as stated in equation 3, and for the sink (bottom) $c(x, y = 0; t) = 0$. Since the source and sink have a fixed concentration, we eliminated those pixels in the range of the iteration. Moreover, we implemented the periodic boundaries (eq. 4) for each side of the domain. For a grid of 50x50, at the left boundary, the diffusion equation (equation 9) looks like this:

$$c_{0,j}^{k+1} = c_{0,j}^{k} + \frac{\partial t D}{\partial x^2}(c_{1,j}^{k} + c_{49,j}^{k} + c_{0,j+1}^{k} + c_{0,j-1}^{k} - 4c_{0,j}^{k})$$

As a result, the range of the width stays the same as the indices of the grid, whereas the range of the height has become one index smaller both at the start and stop condition. Note: we implemented an extra if statement immediately after the for loops to check if there is an object at the gridpoint for section III-C.

A simulation has been made to analyze the discretized two-dimensional time dependent diffusion equation (eq. 2) using the explicit finite difference formulation (eq. 9). Moreover, we compared the simulation to the analytic solution (eq. 6), which is illustrated in figure 4.

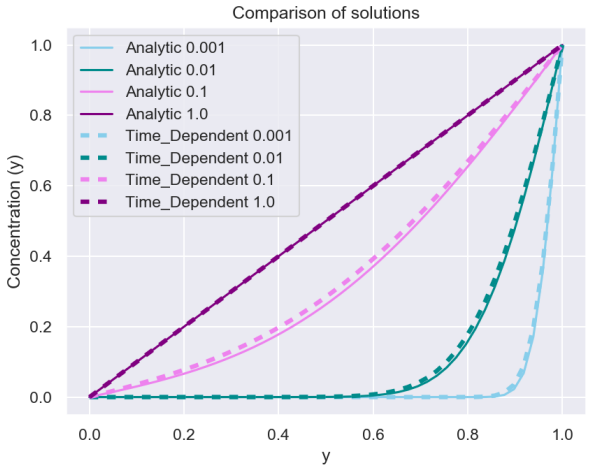

Fig. 4: A comparison of the analytic solution of the concentration as a function of the y-coordinate, and the time dependent solution for $t = 0.001, 0.01, 0.1, 1.0$. Here, $D = 1$ and the initial and boundary conditions are as in the text.

The concentration of the analytic solution for different times is quite comparable to the simulation. For $t = 1.0$, both solutions are linear. This is due to the concentration profile for $t \to \infty$ (eq. 7), which is derived from the time-independent diffusion equation (eq. 11). When looking at the other timesteps, the simulation deviates from the analytic solution, which is caused by the discretized space and time points, and the stability of the scheme (eq. 10). This determines the maximum time step δt that we could take.

Besides that, the simulation is shown in figure 5. As the time increases, the resource diffuses among the grid. This is caused by the finite difference scheme which is dependent on δt (eq. 9). We used $\delta t = 0.0001$, to achieve a stable scheme.

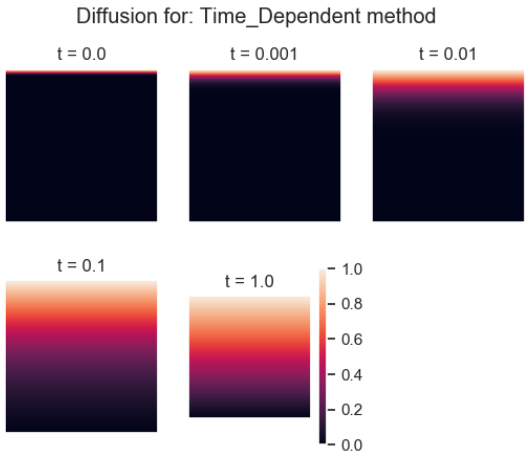

Fig. 5: The 2D domain of the diffusion for the time dependent method. The color represents the concentration at each point. Several plots have been made to show the state of the system at times $t = 0, 0.001, 0.01, 0.1,$ and 1. $\delta t = 0.0001, D = 1$, with a grid size of 50 x 50.

For the animated MP4 of the time dependent diffusion equation until equilibrium, see our files folder. The transient behaviour is clearly shown.

C. The time independent diffusion equation

The Jacobi iteration, the Gauss-Seidel iteration, and the successive over-relaxation (SOR) were implemented

with $N = 50$. The linear dependence of the concentration on y is shown in figure 6. A few representative values for ω are chosen and the optimal ω, that we derive in the next step is also shown. Again, the concentration profile for $t \to \infty$ causes the linearity (eq. 7), which is derived from the Laplace equation (eq. 11).

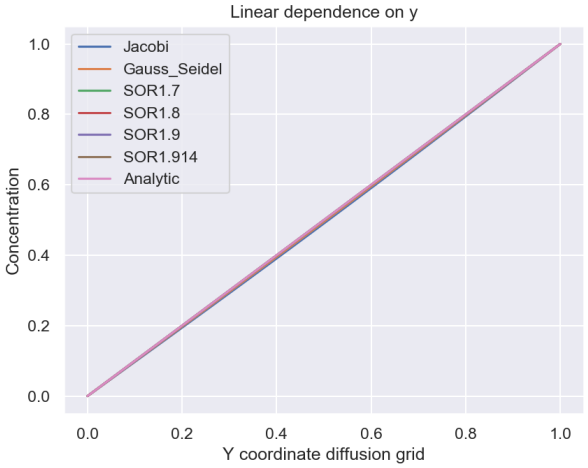

Fig. 6: The linear dependence of the concentration on y of the Jacobi iteration, the Gauss-Seidel iteration, different successive over-relaxations (SOR) with $\omega = 1.7, 1.8, 1.9,$ and 1.914, respectively, and the analytical solution. $N = 50$.

By comparing the difference in concentration, we see a clear distinction in each of the methods in figure 7. As we can see, the Jacobian iteration performs worse than the Gauss-Seidel iteration and SOR. The SOR with $\omega = 1.914$ encompasses the same result as the analytic solution. All in all, the results show indeed how the methods perform. The Gauss-Seidel (eq. 15) is an improvement over the Jacobian (eq. 13), because during the iteration the new value is used as soon as it has been calculated. Evenso, the SOR (eq. 16) is an improvement over the Gauss-Seidel, as it uses an over-correction of the new iterate. In figure 8 we have zoomed in on the linear dependence of the SOR methods. Here, we can see some interesting behavior, because we can see that the highest point in the lines shifts to the right.

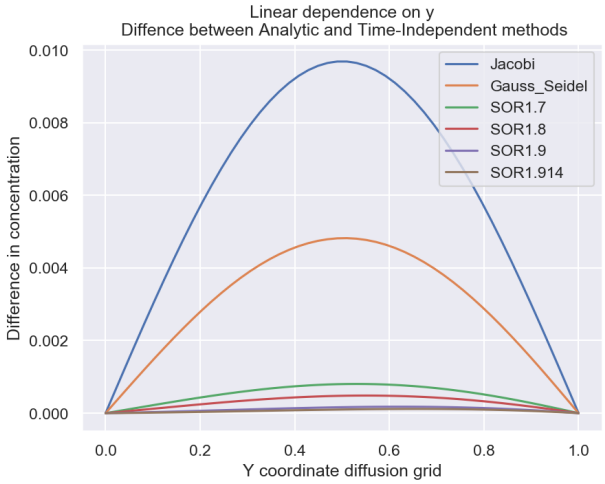

Fig. 7: The difference of the linear dependence of the concentration on y of the Jacobi iteration, the Gauss-Seidel iteration, different successive over-relaxations (SOR) with $\omega = 1.7, 1.8, 1.9,$ and 1.914, respectively, and the analytical solution. $N = 50$.

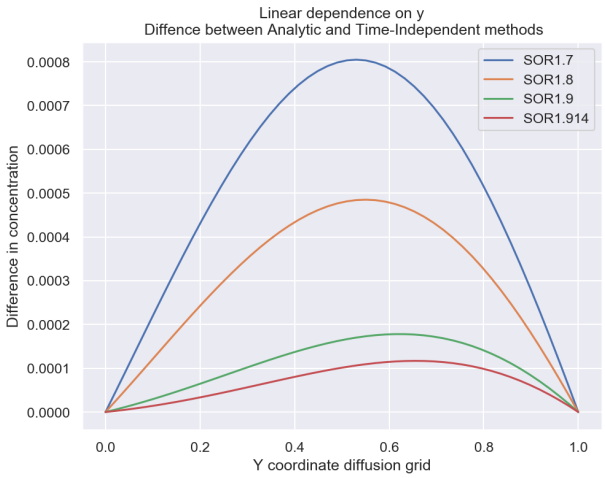

Fig. 8: The difference of the linear dependence of the concentration on y of different successive over-relaxations (SOR) with $\omega = 1.7, 1.8, 1.9,$ and 1.914, respectively, and the analytical solution. $N = 50$.

In figure 9, the convergence measure δ in equation 14 is shown. The δ depends on the number of iterations k for each of the methods. The higher the iteration becomes, the smaller the convergence rate. This convergence rate is the maximum delta at which the methods performs best. The Jacobi method takes the longest to iterate over

the simulation, because the method is only calculated with the previous time step. Moreover, the Gauss-Seidel method is some faster, since the equation already takes a new value of the iteration as soon as it has been calculated. By an over-correction of the new iterate, the SOR outperforms the other methods. Since the optimal ω lies somewhere in the interval $1.7 < \omega < 2.0$, we also calculated the SOR of $\omega = 1.7, 1.8,$ and $, 1.9,$ respectively. Moreover, with an ω of 1.913 or 1.914, the SOR converges the fastest, with 147 steps.

Besides that, in figure 10, the dependence of the optimal ω (i.e. the one that minimizes the number of iterations) on various gridsizes is illustrated. The optimal ω is 1.914 for a grid size of 50 by 50, which indeed lies in the interval $1.7 < \omega < 2.0$. We can see that the ω increases when the gridsize increases.

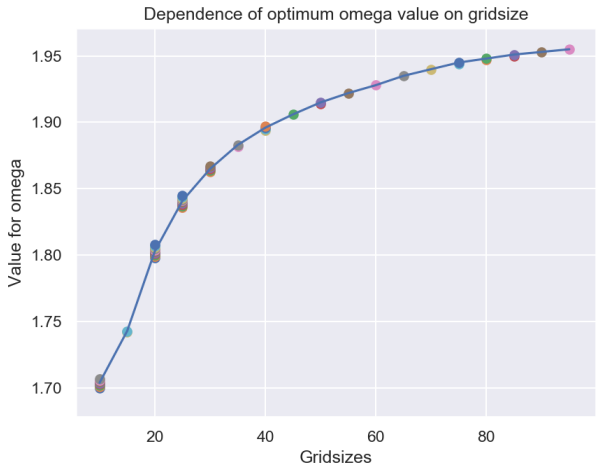

Fig. 10: The dependence of the optimal ω value on the grid size N.

Now, we included a square object. We assume that the object is a sink for the diffusion concentration (i.e. the concentration is zero everywhere on the object). This is shown in figure 11.

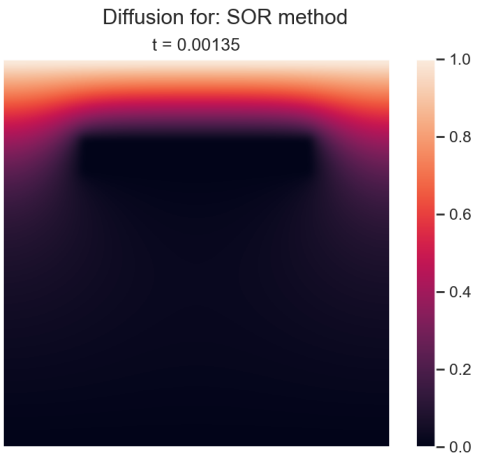

Fig. 11: The 2D domain of the diffusion for the time independent method with a square object of 5x30 grid-points. The color represents the concentration at each point, in which the object is a sink. $\delta t = 0.0001, D = 1$, with a gird size of 50 x 50.

	Steps (ω 1.914)	Optimal ω	Steps (optimal ω)
Normal	147	1.914	147
1 Sink	140	1.873	94
2 Sinks	152	1.881	106

TABLE I: Dependence of omega on the presence of objects that are sinks during the diffusion.

In table I we can see that the objects have an influence on the amount of steps for the optimal omega we calculated previously. The new optimal omega is calculated. And it becomes a lot lower for the diffusion with more sinks. Also the amount of steps is lower for both. The diffusion with two sinks becomes a bit harder, which makes sense when looking at the diffusion pattern that we see in figure 12.

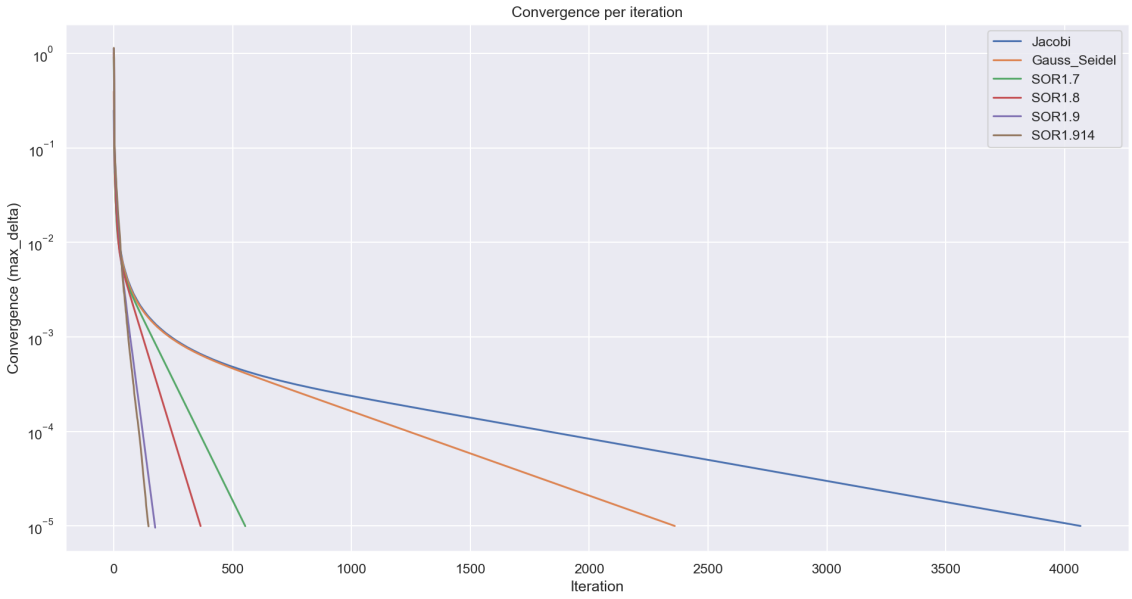

Fig. 9: The convergence per iteration of the Jacobi, Gauss-Seidel, and SOR method with $\omega = 1.7, 1.8, 1.9, 1.914$. $N = 50$.

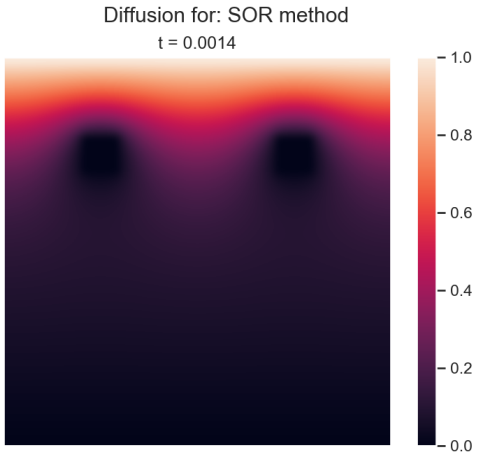

Fig. 12: The 2D domain of the diffusion for the time independent method with two rectangular objects of 5x5 gridpoints. The color represents the concentration at each point, in which the object is a sink. $\delta t = 0.0001, D = 1$, with a grid size of 50 x 50.

IV. Discussion

The problems at hand were the numerical solutions of a one-dimensional wave equation, which described the vibrations of a uniform string, and of a two-dimensional diffusion equation, which described the diffusion of a resource through a grid.

With the right δt and δx, the discretized wave equation showed marvelous results to the vibrating string with fixed boundary conditions. Saving the animation could be optimized, since it took us so much time to generate MP4s.

The time dependent diffusion equations showed a linear dependence of the concentration on y. It is also almost perfect, when compared to the analytic solution, but it takes a lot of time steps to get to the equilibrium.

The time independent methods are all solved faster than the time dependent equation, but they differ among each other. The Jacobi and Gauss-Seidel methods take quite long to converge. The SOR method is fast, and when optimized (with the right omega) even faster.

When comparing the three time independent methods to the analytic solution, we see that the difference in the linear dependence on y is minimal (at max 0.01 for the Jacobi method and 0.00008 for the SOR method). Furthermore, when zooming in on the differences of

the SOR with different omega's, we can see that with the optimal (fastest) ω, the difference with the analytic solution is also smallest.

The difference of various SOR methods illustrates an interesting shift in figure 8. More research could be done to investigate this.

Last but not least, it would be interesting to hinder the diffusion with insulating material and see the influences on the spread of the resource and on the optimal ω.

While not really part of the exercises, we would have like to also parallelize the computation with something like the JIT-compiler Numba, to be able to calculate the different states faster and thus to investigate more things in less time.

V. References

We'd like to refer to the assignments of this years course Scientific Computing at the UvA in Amsterdam, and the lecture slides for most of the theory.