Scientific Computing Exercise Set 1

Daniel Bezema (13919881) & Magda Karavangeli (14712164)

I. Introduction

The study of partial differential equations (PDEs) plays a critical role in many fields of science and engineering, from physics to finance. In this paper, we explore the behavior of several types of PDEs, including the vibrating string equation, time-dependent and time-independent diffusion equations. We also investigate numerical methods for solving the time-dependent equations, specifically the Jacobi, Gauss-Seidel and Successive Over Relaxation (SOR) methods.

The vibrating string equation describes the motion of a string, which is modeled as a continuous system with one spatial dimension. Some of the applications of this equation is the transport of voltage [3], optical [4] or sound waves [1] and flow rate of a liquid or gas in a pipe. The time-dependent diffusion equation describes the behavior of a concentration gradient over time, while the time-independent diffusion equation characterizes the behavior of the gradient at equilibrium. The time-dependent diffusion equation has applications in various fields, such as physics [6], chemistry, and biology . In physics, it is used to describe the diffusion of atoms or molecules in a material. In chemistry, it is used to describe the transport of chemical species in a solution [2]. In biology, it is used to describe the diffusion of molecules through a cell membrane or tissue [5].

To solve the time independent equation, we examine the Jacobi iteration, Gauss-Seidel iteration, an improved iterative method from the Jacobi iteration. Additionally, we investigate the SOR method, which is a more efficient algorithm for solving these equations that involves tuning a relaxation factor to achieve faster convergence.

Overall, the present study of these equations and solution methods provides insight into the behavior of physical systems and engineering problems, and can inform the development of more accurate and efficient models for future applications.

II. Methods

A. One-dimensional wave equation

The one-dimensional wave equation describes the propagation of a wave along a one-dimensional medium. The mathematical formulation of the one-dimensional wave equation is described in equation 1.

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

where $\psi(x,t)$ is the amplitude wave at position x and time t, and c is a fixed non-negative real coefficient.

The wave equation involves partial derivatives, which must be approximated at discrete grid points using a finite difference method. The first step is to assume the numerical scheme $\psi(x,t) \equiv u(i,j)$. Next, using forward and backward differences, we derive the discrete wave equation.

Forward difference: $u^i(x+\Delta x) = u^i(x) + \frac{\delta u^i(x)}{\delta x}(\Delta x) + \frac{1}{2}\frac{\delta^2 u^i(x)}{\delta x^2}(\Delta x)^2 + O\left(\Delta x^3\right)$

Backward difference: $u^i(x-\Delta x) = u^i(x) + \frac{\delta u^i(x)}{\delta x}(-\Delta x) + \frac{1}{2}\frac{\delta^2 u^i(x)}{\delta x^2}(-\Delta x)^2 + O\left(-\Delta x^3\right)$

After removing the higher order terms, the equation becomes,

$$u^i(x+\Delta x)+u^i(x-\Delta x) = u^i(x)+\frac{\delta u^i(x)}{\delta x}\Delta x+\frac{1}{2}\frac{\delta^2 u^i(x)}{\delta x^2}\Delta x^2 + u^i(x) + \frac{\delta u^i(x)}{\delta x(\Delta x)} + \frac{1}{2}\frac{\delta^2 u^i(x)}{\delta x^2}(-\Delta x)^2 \tag{2}$$

Summing both expansions, $\frac{\partial^2 \psi}{\partial x^2}$ and $\frac{\partial^2 \psi}{\partial t^2}$ can be calculated. For simplification, we replace x by the counter j and x+x, which describes the next position and x-x becomes j-1 and it describes the previous position.

$$\frac{\partial^2 \psi}{\partial x^2} \equiv \frac{u(i+1,j)+u(i-1,j)-2u(i,j)}{(\Delta x)^2} \tag{3}$$

and similarly,

$$\frac{\partial^2 \psi}{\partial t^2} \equiv \frac{u(i,j+1)+u(i,j-1)-2u(i,j)}{(\Delta t)^2} \tag{4}$$

Following is the substitution of equations 3 and 4 to the equation 1.

$$\frac{u(i,j+1)+u(i,j-1)-2u(i,j)}{(\Delta t)^2} = c^2\frac{u(i+1,j)+u(i-1,j)-2u(i,j)}{(\Delta x)^2} \tag{5}$$

The final expression used is described by the equation **??**.

$$u(i,j+1) = c^2 \frac{u(i+1,j) + u(i-1,j) - 2u(i,j)}{(\Delta x)^2} - u(i,j-1) + 2u(i,j) \quad (6)$$

Boundary conditions are necessary to fully specify the problem. They are approximated at discrete grid points. In the case of a string fixed at both ends, the boundary conditions are:

$$\psi(0,t) = 0 \quad \text{and} \quad \psi(L,t) = 0$$

where L is the length of the string.

Assuming that the string is at rest at $t = 0$,i.e. $\Psi'(x, t = 0) = 0$ we start with deriving the very first step of the wave equation using first order forward and backward Taylor series.

$$u_j(t + \Delta t) = u(t) + \frac{\partial u_j(t)}{\partial t}\Delta t \quad (7)$$

$$u_j(t - \Delta t) = u(t) + \frac{\partial u_j(t)}{\partial t}(-\Delta t) \quad (8)$$

Subtracting the equations 7 and 8 we end up at,

$$\frac{\partial u_j(t)}{\partial t} = \frac{u_j(t + \Delta t) - u_j(t - \Delta t)}{2\Delta t}$$

When $t = 0$, $\frac{\partial u_j(t)}{\partial t} = 0$

$$\frac{\partial u_j(t)}{\partial t} = \frac{u_j(t + \Delta t) - u_j(t - \Delta t)}{2\Delta t} = 0$$

Finally we get that:

$$u_j(t - \Delta t) = u_j(t + \Delta t)$$

Which using then becomes,

$$u(i,j-1) = u(i,j+1) \quad (9)$$

Substituting this result in equation 6 and rewriting yields:

$$u(i,j+1) = 1/2c^2 \frac{u(i+1,j) + u(i-1,j) - 2u(i,j)}{(\Delta x)^2} + u(i,j) \quad (10)$$

The initial conditions implemented for the time stepping are described by the following equations 11, 12, and 13. Experiments where performed accordingly.

$$\psi(x, t = 0) = sin(2\pi x) \quad (11)$$

$$\psi(x, t = 0) = sin(5\pi x) \quad (12)$$

$$\psi(x, t = 0) = sin(5\pi x) if 1/5 < x < 2/5, else \psi = 0 \quad (13)$$

For all experiments we used C=1, time step of 0.001, length of string equal to 1 and 100 iterations.

B. The Time Dependent Diffusion Equation

The two-dimensional diffusion equation is described from equation 14 and 15.

$$\frac{\partial C}{\partial t} = D\nabla^2 c \quad (14)$$

$$\frac{\partial C}{\partial t} = D(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}) \quad (15)$$

where $C(x, y; t)$ is the concentration of the diffusing species with spatial coordinates x,y and t is time and D is the diffusion coefficient.

The equation can be solved using various numerical methods. These methods discretize the concentration field and solve the equation for each time step. The solutions obtained from these methods can be used to predict the behavior of the system over time.

The spatial and temporal domains are then separated. Intervals of length $deltax$ are used to split the x- and y-axes. If we assume that there is an interval of N in both the x- and y-directions, then $\delta x = 1/N$, and $x = i\delta x, y = j\delta x$ where $i, j \in (0, 1, 2, \ldots N)$. Additionally, we make the assumption that δt and $k \in \mathbb{N}$ are small time intervals.

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

Similarly to the wave equation, finite differences were used to approximate the diffusion equation, more specifically a combination of forward and backward differences.

$$\frac{\delta^2 c}{\delta x^2} = \frac{u^i_{j+,k} + u^i_{j-1,k} - 2u^i_{j,k}}{\Delta x^2}$$

where k is now the y-direction, similarly

$$\frac{\delta^2 c}{\delta y^2} = \frac{u^i_{j,k+1} + u^2_{j,k-1} - 2u^i_{j,k}}{\Delta y^2}$$

For the approximation of t we need a first order approximation, only forward difference was used up to first derivation.

$$c_{i,k}(t + \Delta t) = c_{i,k}(t) + \frac{\delta c_{i,k}(t)}{\delta t} * \Delta t + O\left(-\Delta t^2\right) \quad (16)$$

The higher order terms are ignored and the equation is rearranged. For simplicity and using lattice notations, we substitute $t + \Delta t$ with j + 1. After these changes we arrive at equation 17.

$$\frac{\delta c}{\delta t} = \frac{u_{j,k}^{i+1} - u_{j,k}^{i}}{\Delta t}. \tag{17}$$

Substituting equation 17 to equation 15 and assuming that $\Delta x^2 = \Delta y^2$ we arrive at equation 18.

$$c_{i,j}^{k+1} = c_{i,j}^{k} + \frac{\delta t D}{\delta x^2}\left(c_{i+1,j}^{k} + c_{i-1,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k} - 4c_{i,j}^{k}\right) \tag{18}$$

Boundary conditions are an important part of the solution, for that reason we assume the following boundary conditions. $c(x, y = 1; t) = 1$ and $c(x, y = 0; t) = 0$. More specifically, on the top of our system, the concentration is always equal to one, and on the bottom, the concentration is always equal to zero. In the x-direction we assume periodic boundary conditions, $c(x = 0, y; t) = c(x = 1, y; t)$.

This means that for $x = 1$ equation 18 will be adapted to

$$c_{i,j}^{k+1} = c_{i,j}^{k} + \frac{\delta t D}{\delta x^2}\left(c_{0,j}^{k} + c_{i-1,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k} - 4c_{i,j}^{k}\right) \tag{19}$$

and for the case where $x = 0$ to

$$c_{i,j}^{k+1} = c_{i,j}^{k} + \frac{\delta t D}{\delta x^2}\left(c_{i+1,j}^{k} + c_{n,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k} - 4c_{i,j}^{k}\right) \tag{20}$$

where n is the width of the domain.

In order to examine the correctness of our simulation, the analytical solution is used. For $t \to \infty$ in the time-independent diffusion equation, the concentration profile is a straight line.

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

So the comparison between the results produced and the analytical solution can provide further insight into the correctness of our method. The analytical solution is described in equation 21.

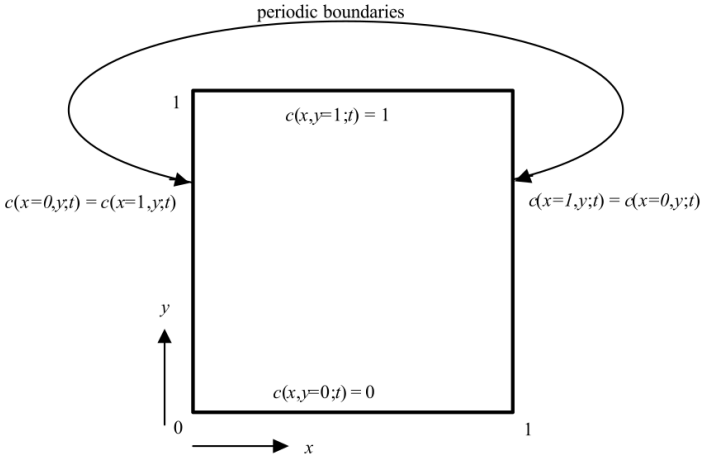

Fig. 1: A schematic of the computational domain for the 2D diffusion equations

$$c(y, t) = \sum_{i=0}^{\infty} \operatorname{erfc}\left(\frac{1 - y + 2i}{2\sqrt{Dt}}\right) - \operatorname{erfc}\left(\frac{1 + y + 2i}{2\sqrt{Dt}}\right) \tag{21}$$

C. The Time Independent Diffusion Equation

In the case that there is no transient behavior and all time derivatives are zero, the diffusion of the system is time independent. The general form of the time-independent diffusion equation is the Laplace equation 22.

$$\nabla^2 c = 0 \tag{22}$$

The boundary conditions, as well as the spatial discretization remain the same. The equation for this methods is described in equation 23.

$$\frac{1}{4}\left(c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1}\right) = c_{i,j} \tag{23}$$

The superscript k has been removed, as the time behavior is no longer relevant. There are several methods available to solve the equations, but only iterative methods were used in this discussion. The system first takes the form $Ax = b$. Iterative methods have the advantage of being potentially efficient and requiring less memory than direct methods, and they can be relatively easily parallelized.

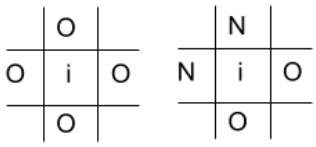

Fig. 2: Difference between Jacobi (left) and Gauss-Seidel (right). Given some cell i, surrounding neighbours are taken from the old O or new state N

a) The Jacobi Iteration: The Jacobi iteration is described by the equation 24 and is derived from equation 23 by using the equality $\frac{\delta t D}{\delta x^2} = \frac{1}{4}$.

$$c_{i,j}^{k+1} = \frac{1}{4}\left(c_{i+1,j}^{k} + c_{i-1,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k}\right) \tag{24}$$

Because the equation driving the Jacobi iteration is time independent, a stopping criterion is necessary to determine the amount of iterations to run. In this case we consider whether the solution has converged,i.e. there is no notable difference between successive states. It is assumed that this happens when the maximum difference observed between two iterations is below a specified tolerance level ϵ which in our case is 10^{-5}.

$$\delta \equiv \max_{i,j} \left| c_{i,j}^{k+1} - c_{i,j}^{k} \right| < \epsilon \tag{25}$$

The Jacobi iteration represents merely the solution of the time-dependent problem with the largest time step permitted. This implies the need for more effective iterative techniques.

b) The Gauss-Seidel Iteration: The Gauss-Seidel method, commonly referred to as the Liebmann method in numerical linear algebra, is an iterative technique used to resolve a system of linear equations. It is comparable to the Jacobi technique, but ultimately needs less time steps. The solution for next state is shown in the equation 26.

$$c_{i,j}^{k+1} = \frac{1}{4}\left(c_{i+1,j}^{k} + c_{i-1,j}^{k+1} + c_{i,j+1}^{k} + c_{i,j-1}^{k+1}\right) \tag{26}$$

Equations 25 and 26 have one important difference. The values at $i-1$ and $j-1$ are taken from the next state $k+1$ instead of the current one k. This is the additional benefit of the Gauss-Seidel method; it uses newly calculated values as soon as they are calculated. This difference can be made visual with Figure 2.

c) Successive Over Relaxation: The last method used is Successive Over Relaxation which is an improved version of the Gauss-Seidel iteration. The method is described by equation 27.

$$c_{i,j}^{k+1} = \frac{\omega}{4}\left(c_{i+1,j}^{k} + c_{i-1,j}^{k+1} + c_{i,j+1}^{k} + c_{i,j-1}^{k+1}\right) + (1-\omega)c_{i,j}^{k} \tag{27}$$

It is similar to Gauss-Seidel with one important difference. The method uses a mixture between the values in the old and the new state. This freedom in choosing a preference for old and new values can be favourable for certain problems that benefit from either one of these relaxations. It is easy to see from equation 27 that for $\omega = 1$ we receive the Gauss-seidel method. With increased ω more preference is given to new values. Conversely, decreased ω yields increased preference for old values. For values greater than two or smaller than one the method becomes unstable. This is summarized in the following:

- $0 < \omega < 1$: under relaxation
- ω = 1 : Gauss-Seidel
- $1 < \omega < 2$: over relaxation
- $1 > \omega > 2$: unstable

III. Results

A. Vibrating string

The discrete wave equation (6) was used in order to calculate the amplitude of a wave for a given point x at time t. At time $t = 0$ the amplitude of the wave at point x is given by one of the initial conditions described by equations 11, 12 and 13. For the first time-step the initial condition is advanced by applying equation 10. For subsequent time-steps equation 6 is used for calculating the amplitude at each point x.

In Figures 3, 4 and 5 the advancement of the initial conditions using the wave equation is shown. Different time-steps are visualised with different coloured lines. The color black denotes the start of the simulation at $t = 0$. Yellow denotes the maximum time $t = 1$.The simulation is advanced with $\Delta t = 0.001$ at every time-step, however only every 25th time-step is plotted. It is demonstrated that the waves' amplitude oscillate between two states as time is evolved. For initial condition 11 it takes exactly one second for the wave to return to the initial condition. State 12 oscillates faster and returns approximately two and a half times within one second. Finally, condition 13 oscillates only halfway in one second.

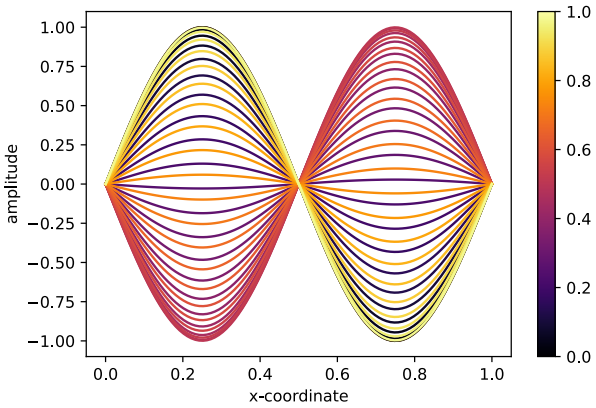

Fig. 3: Results of the wave equation with initial condition $\psi(x, t = 0) = sin(2\pi x)$. Different colors represent different time steps.

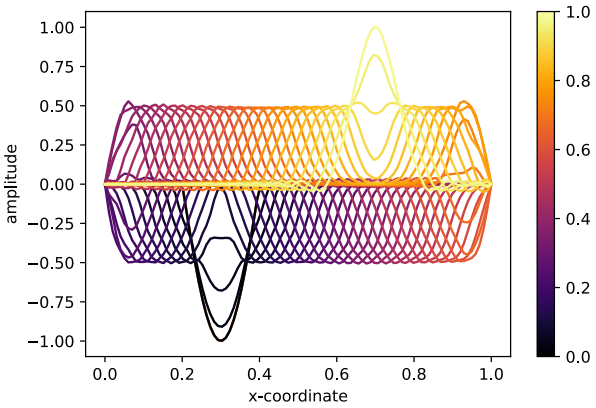

Fig. 5: Results of the wave equation with initial condition $\psi(x, t = 0) = sin(5\pi x)$ with if $1/5 < x < 2/5$, else $\psi = 0$. Different colors represent different time steps.

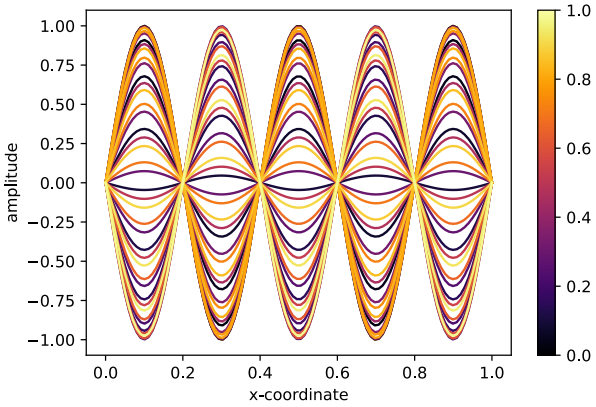

Fig. 4: Results of the wave equation with initial condition $\psi(x, t = 0) = sin(5\pi x)$. Different colors represent different time steps.

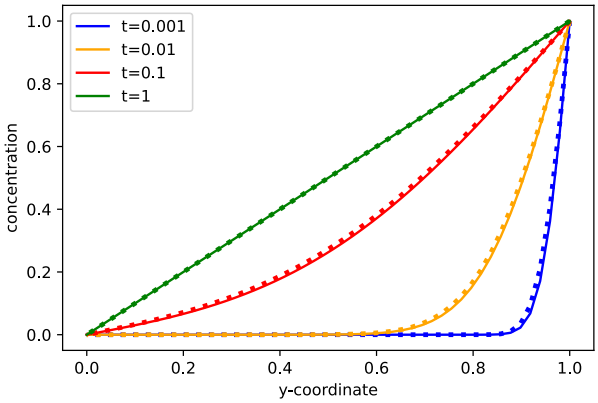

Fig. 6: Comparison of analytical solution with results from the simulation at different times t. Analytical solution is plotted as straight line, simulation as dotted. $\Delta t = 0.0001$

B. Time Dependent Diffusion Equation

The implementation of the time dependent diffusion equation is tested for correctness by comparing the results of the simulation to analytic solutions. For the time stepping of the diffusion equation $\Delta t = 0.0001$ is used. The results are shown in Figure 6 .

It is apparent that for each of the plotted times t, the solution from the simulation is very close to the expected analytical solution. However for all $t < 1$ the numerical solution slightly overestimates the analytical one. This is due to the discrete nature of the method. Even with a relatively small Δt of 0.0001, the difference in advancement of the concentration in the diffusion between the numerical method and the analytical one is noticeable.

In Figure 7, the time development of the concentration in the 2D domain is shown. Different colors represent different concentration levels as depicted by the accompanying color bar. As time progresses the concentration gradually diffuses across the plane. At $t = 1$ the concentration profile shows a linear decrease in the y-axis as is expected from Figure 6.

C. Time Independent Diffusion Equation

The time independent diffusion equation is simulated using three different iterative methods for comparison. In theory, the best method would be Successive Over Relaxation (SOR), while the slowest would be the Jacobi. This hypothesis is tested through several experiments that test the error and effectiveness of each of the methods. First, in figure 8 the output of each method is compared to

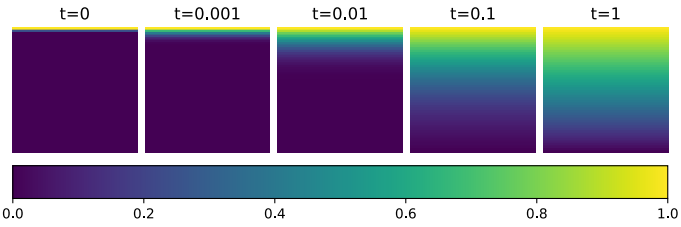

Fig. 7: Concentration using different time steps = 0, 0.001, 0.01, 0.1, and 1 shown from left to right. Color bar is included for the concentration of the diffusion process

the analytical result provided by equation 6.This is done by applying each of the methods to a plane of size $N = 50$ and running them until the stopping criterion form equation 25 is met. For the SOR method a ω of 1.5 is used. It is observed that the SOR iteration is closest to the analytical result, followed by Gauss-Seidel and lastly by Jacobi. This confirms the intuition as provided by the theory.

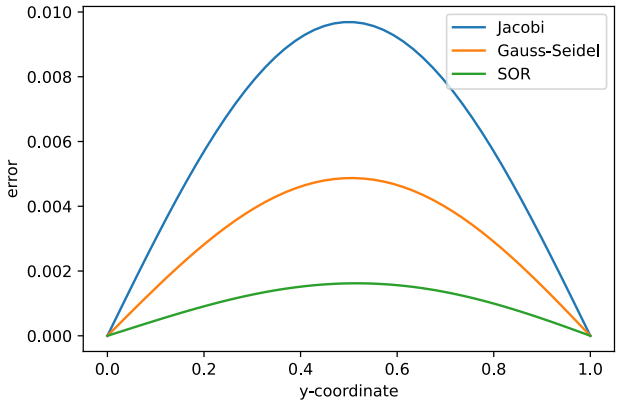

Fig. 8: Comparison of the results produced from the different iteration methods and with the analytical solution

In Figure 9 the amount of iterations until the stopping criterion is shown for each of the iterative methods. It is observed that SOR is, in addition to being more accurate, the fastest method with 1000 iterations. Gauss-Seidel follows second with roughly 2500 and Jacobi last with more than 4000. In addition Figure 10 shows the comparison to the analytical solution of the SOR method in isolation for different values of ω. It is observed that with increasing ω the error from the analytical solution becomes smaller. Lastly, Figure 11 shows the effect of increasing ω on the amount of iterations needed until convergence. It is clear that an increased ω significantly reduces the amount of iterations.

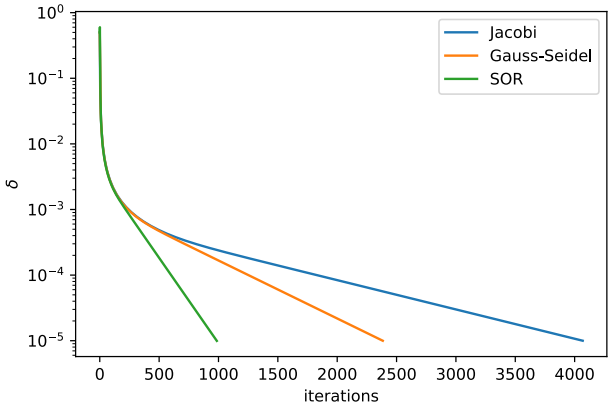

Fig. 9: Comparison of the convergence between the different iterative methods. $\omega = 1.5$

The difference in performance between these methods can be explained by the degree to which old or current state values are used in computing the next state. Jacobi solely makes use of old values in computing the values of the new state. All the values in the plane need to be updated before the newly computated values are used. It is therefore fairly slow. Jacobi does use newly computed values as soon as they are computed and it is therefore faster. SOR (with $\omega > 1$) gives even more weight to the newly computed values and it therefore converges the fastest. From these experiments it can be concluded that SOR is the most effective numerical method for solving the two-dimensional diffusion equation. For this reason, the rest of the report will focus on this method.

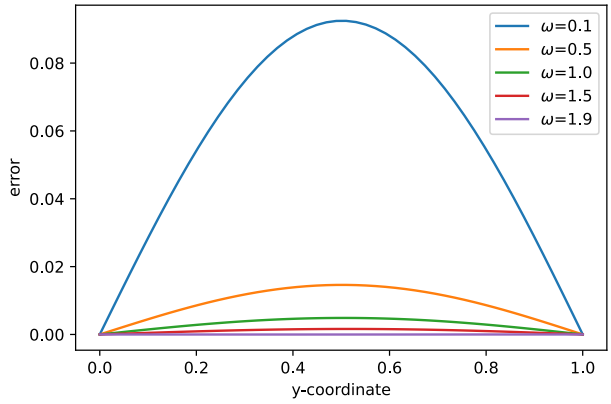

Fig. 10: Comparison of the accuracy of the solution using SOR for different ω

From this point on a number of sinks (up to two) is added to the computational domain that was defined in Figure 1. These sinks add a new condition to the

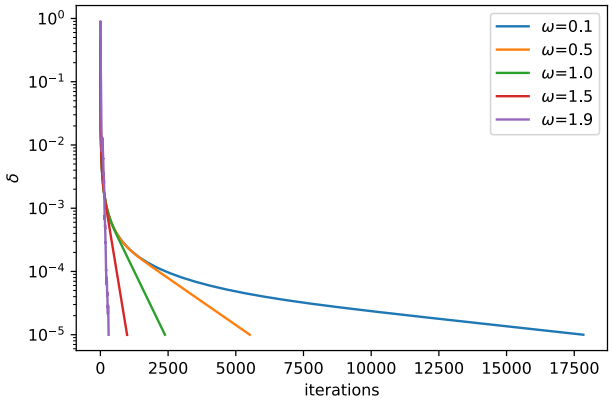

Fig. 11: Comparison of the convergence of the solution using SOR for different ω

computational domain; at the location of the sink the concentration is always zero. The position and size of these sinks are depicted in Figure 12. The size of the sinks scales with the size of the domain N.

In order to determine the most optimal value for ω an optimizing experiment is performed. As a minimizing measure the change between states δ is used (equation 25). Subsequently the SOR method is run on a domain with varying N and number of sinks. The method is run for only 100 iterations after which the last δ is reported. Because the minimizing measure is different from the error from the analytical solution that was shown in Figure 10, the optimal ω's will slightly deviate from the optimal value of 1.9 in that figure. This measure was chosen because the previous analytical solution does not account for the presence of sink objects. The results of these experiments are summarized in Table I.

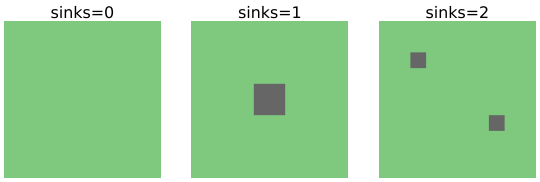

Fig. 12: Position of sinks in the computational domain.

N	30	40	50	60	70
Sink 0	1.8206	1.8484	1.8569	1.8761	1.8954
Sink 1	1.7938	1.8232	1.8466	1.8644	1.8876
Sinks 2	1.7865	1.8229	1.8429	1.8605	1.8789

TABLE I: Changes in ω while the number of N and number of sinks change

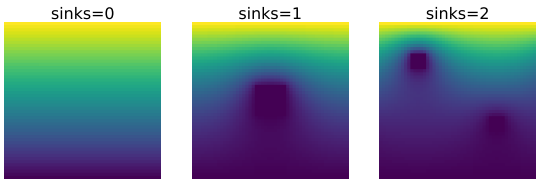

Fig. 13: Diffusion using SOR ($\omega = 1.85$) on planes $N = 50$ including sinks.

The results of the optimal ω experiment for changing domain sizes N and addition of sinks is summarized in Table I. It appears that as the domain size increases, the value of ω increases as well. Conversely, as the number of sinks increases, the value of ω decreases.

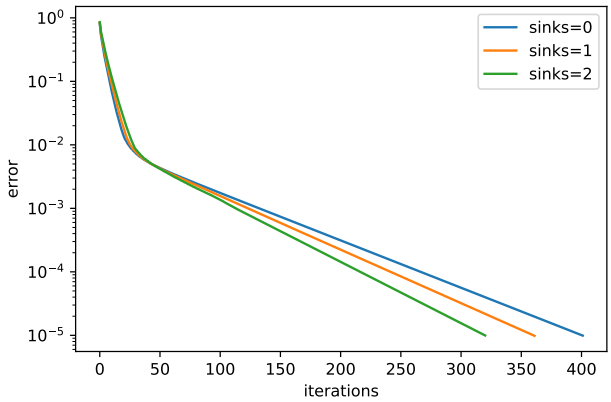

Fig. 14: Convergence of SOR (ω) for different amount of sinks in the domain. $N = 50$,$\omega = 1.85$

Lastly, the effect of the presence of sinks on the convergence and the final solution is discussed. Figure 14, shows the convergence of SOR for the different amount of sinks ($N = 50$ and $\omega = 1.85$). As the sinks increase in quantity the system needs less iterations to converge. The system without sinks converges after more than 400 iterations. For one sink it reaches convergence around 350. The domain with two sinks reaches convergence in less than 350. This effect can be explained by the definition of the sinks. The sinks effectively act as drains for the solution, meaning that as the solution evolves over time, regions with higher concentrations of the solution will tend to flow towards the sinks and be removed from the computational domain. With the addition of sinks, this drainage effect is stronger and the number of iterations required for the SOR method to converge to a given tolerance is reduced. Also, because the cells at the location of the sinks will never change, the change level δ will start out lower if the total area of sinks is higher.

This effect can also be clearly observed from the final stable states shown in Figure 13.

IV. Discussion

Concluding, the wave equation is used to study the amplitude of the waves in a system. It is found that waves oscillate between the same states. In a system with diffusion, as time passes the concentration of the diffused element increases. It is apparent from the experiments performed that the results of the time dependent diffusion method are very similar to the analytical solution of the diffusion equation. Overall, the theory of time-dependent diffusion equation provides a powerful tool for understanding the behavior of systems. By numerically solving this equation, predict the evolution of concentration fields along time can be predicted and insights into the behavior of the system can be gained.

The problem of time independent diffusion can be solved with multiple iterative methods the best being SOR as was concluded from experiments. SOR can reach solutions very close to the analytical one as the factor ω is increased. This makes it an accurate and reliable iteration method. It is important to include that after a certain value the system becomes unstable. For the specific problem of the 2D time independent diffusion, the optimal ω is found to lie between 1.7 and 2. The exact optimal value, however, is dependent on the size of the specific domain and the existence of objects and their specific shape and size. For our experiments the optimal value of ω lies around 1.85 in all systems.

The stability and accuracy of the numerical solution must also be carefully considered. The numerical method used to approximate the partial derivatives must be stable, meaning that small perturbations in the solution do not grow uncontrollably.The accuracy of the numerical method depends on the order of the finite difference approximation and the size of the grid spacing. To ensure accuracy, the grid spacing must be small enough to capture the details of the solution, but not too small to introduce numerical errors.

Limitations in this research was computational power which was limiting in the aspect of studying a larger system, for example adding a larger amount of sinks. The computational power used was enough for the systems to converge.

For future research we can incorporate objects with insulating material in the domain which will prevent or decelerate the diffusion. Also the addition of differently shaped objects could be interesting. This can have many different practical applications [7].

References

[1] Santiago Esteban Perez Bergliaffa, Katrina Hibberd, Michael Stone, and Matt Visser. Wave equation for sound in fluids with vorticity. *Physica D: Nonlinear Phenomena*, 191(1-2):121–136, 2004.

[2] Gregory J McRae, William R Goodin, and John H Seinfeld. Numerical solution of the atmospheric diffusion equation for chemically reacting flows. *Journal of Computational Physics*, 45(1):1–42, 1982.

[3] Willard L Miranker. Periodic solutions of the wave equation with a nonlinear interface condition. *IBM Journal of Research and Development*, 5(1):2–24, 1961.

[4] Kaido Reivelt and Peeter Saari. Localized wave solutions of the scalar homogeneous wave equation and their optical implementation. *arXiv preprint physics/0309079*, 2003.

[5] RG Safranyos and Stanley Caveney. Rates of diffusion of fluorescent molecules via cell-to-cell membrane channels in a developing tissue. *The Journal of cell biology*, 100(3):736–747, 1985.

[6] F Villars. Adiabatic time-dependent hartree-fock theory in nuclear physics. *Nuclear Physics A*, 285(2):269–296, 1977.

[7] Alexey Zhukov, Armen Ter-Zakaryan, Ekaterina Bobrova, Igor Bessonov, Andrey Medvedev, Vitaly Mukhametzyanov, and Alexey Poserenin. Evaluation of thermal properties of insulation systems in pitched roofs. In *E3S Web of Conferences*, volume 91, page 02047. EDP Sciences, 2019.