Scientific Computing Exercise Set 1

Michael MacFarlane Glasow (12317217) & Anezka Potesilova (15884392) & Tycho Stam (13303147)

***Repository**: https://github.com/kingilsildor/CLS-Scientific_Computing-Assignment1*

I. Introduction

In this report, we study two fundamental partial differential equations (PDE): the one-dimensional wave equation and the two-dimensional diffusion equation.

The wave equation describes how waves travel through space of time [1]. Waves can be found everywhere in nature. These waves can be in the form of earthquakes, ocean waves hitting the coast, and sound waves. To understand how a simple wave behaves, we can eventually derive more complex formulations that are at the basis of modern technology, such as telecommunications or medical imaging.

The diffusion equation looks at how substances spread over time in a given medium. This can be heat, particles, or chemicals. Because of this, diffusion equations are used in various scientific fields, including physics, engineering, biology, and environmental science. By understanding how diffusion works, scientists can design more efficient systems, such as improving heat exchangers, or modelling biological systems in the human body.

The report is structured as follows: In the Introduction, we provide an overview of the 1D wave equation and 2D diffusion equation, highlighting their importance and relevance across various fields. The Theory section delves into the derivation, assumptions, and applications of both equations, as well as an introduction to the iterative methods used to solve the diffusion equation.

In the Method section, we describe the approach taken to solve the equations, focusing on the numerical methods and algorithms employed, and provide an explanation of how the results are obtained. The Results section presents the findings, including visual representations of the solutions and a comparison of convergence speeds for the iterative methods. Finally, in the Discussion, we analyse the results, assess the performance of the iterative methods, and explore practical implications. Throughout this report, we will frequently refer to the assignment paper as a basis for our analysis and discussions.

II. Theory

The wave equation can be derived from Hooke's Law and Newton's Second Law by considering an array of masses, connected by springs. When a spring is under tension, any small displacement creates a net force due to the difference in tension at neighbouring points. Newton's Second Law is applied to this small segment by equating the mass times acceleration to the net restoring force. This force is then expressed in terms of the change in displacement along the string, resulting in the wave equation:

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

For the diffusion equation, both the time-dependent and time-independent versions were studied. The diffusion equation can be derived from looking at the continuity equation [2]. A change in concentration within a system occurs due to the movement of material into and out of a given region. This means that any increase in concentration results from an inflow of material, while a decrease is caused by an outflow, making it so that no mass is created or destroyed. Fick's Law hereby states that the in and outflow in the system is proportional to the local concentration gradient, resulting in the diffusion equation:

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

Given that the PDE's are known, an approximation can be made by discretizing both the spatial and time domains. This numerical technique is called the finite-difference method.

For the diffusion equation, we also consider the time-independent version, i.e. Laplace equation:

$$\nabla^2 c = 0 \tag{3}$$

We implement and compare three iterative approaches: the Jacobi iteration, the Gauss-Seidel iteration, and Successive Over Relaxation (SOR). In doing so, we aim to measure the difference in convergence speed for the given algorithms.

III. Method

To approximate the wave equation a discretization has to first be made. For this approximation, a regular grid is used, whereby each $x \to i$ index is h distance apart, and each $t \to j$ index is k distance apart. Resulting in $u(x,t) \approx u(i,j)$. To approximate a second order derivative, a scheme is used that adds the forward and backward expansions:

$$\begin{gathered} f(x+\Delta x)+f(x-\Delta x) = 2f(x)+\Delta x^2 f''(x)+O(\Delta x^4) \\ \frac{f(x+\Delta x)-2f(x)+f(x-\Delta x)}{\Delta x^2} = f''(x)+O(\Delta x^2) \end{gathered} \tag{4}$$

The following approximating will arise for the second order derivatives, using the regular grid and equation 4:

$$\begin{gathered} \frac{\partial^2\Psi(x,t)}{\partial x^2} \approx \frac{\Psi(i+1,j)+\Psi(i-1,j)-2\Psi(i,j)}{\Delta x^2} \\ \frac{\partial^2\Psi(x,t)}{\partial t^2} \approx \frac{\Psi(i,j+1)+\Psi(i,j-1)-2\Psi(i,j)}{\Delta t^2} \end{gathered} \tag{5}$$

When substituting equation 5 into the wave equation, the following expression arises:

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

That can be rearranged into:

$$\begin{aligned} \Psi(i,j+1) = c^2\frac{\Delta t^2}{\Delta x^2}[&\Psi(i+1,j)+\Psi(i-1,j) \\ &-2\Psi(i,j)]-\Psi(i,j-1)+2\Psi(i,j) \end{aligned} \tag{6}$$

For the wave equation, a fixed boundary condition is applied in combination with a step method for iterations. For the initial conditions, the assumption is made that the string is at rest at $t=0$, that is, $\Psi'(x,t=0)=0$ and three different options for $\Psi(x,t=0)$ are used, as specified in the assignment. Finally, the propagation velocity c is set to 1.0 and $\Delta x = \Delta t = 0.001$.

Since we always need the two most recent time steps, we also need to specify Ψ at $t = \Delta t$, i.e. $\Psi(i,1)$. To do so, we use the fact that the string is at rest at $t = 0$, and using the central difference scheme for the first derivative,

$$\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x} = f'(x)+O(\Delta x^2) \tag{7}$$

we get that at $j=0$,

$$0 = \Psi'(i,j) = \frac{\Psi(i,j+1)-\Psi(i,j-1)}{2\Delta x} \tag{8}$$

and therefore $\Psi(i,j+1) = \Psi(i,j-1)$. Plugging this into equation 6, we get

$$\begin{aligned} \Psi(i,1) = c^2\frac{\Delta t^2}{2\Delta x^2}[&\Psi(i+1,0)+\Psi(i-1,0) \\ &-2\Psi(i,0)]+\Psi(i,0) \end{aligned} \tag{9}$$

For analysing the heat diffusion, the assignment specifies a two-dimensional square with a periodic boundary on the x-axis:

$$c(x=0,y;t) = c(x=1,y;t) \tag{10}$$

For the y-axis the following boundary conditions are used:

$$c(x,y=1;t)=1 \text{ and } c(x,y=0;t)=0 \tag{11}$$

Based on these conditions, the spatial and temporal domain are discretized as described in the assignment, and the following equation is formulated for solving of the diffusion:

$$c_{i,j}^{k+1} = \begin{cases} c_{i,j}^k + \frac{\delta t D}{\delta x^2}\Big(c_{i-1,j}^k + c_{i=1,j}^k + c_{i,j-1}^k \\ \quad + c_{i,j+1}^k - 4c_{i,j}^k\Big), & \text{if } i=N \\ c_{i,j}^k + \frac{\delta t D}{\delta x^2}\Big(c_{i=N-1,j}^k + c_{i+1,j}^k + c_{i,j-1}^k \\ \quad + c_{i,j+1}^k - 4c_{i,j}^k\Big), & \text{if } i=0 \\ 1, & \text{if } j=N \\ 0, & \text{if } j=0 \\ c_{i,j}^k + \frac{\delta t D}{\delta x^2}\Big(c_{i-1,j}^k + c_{i+1,j}^k + c_{i,j-1}^k \\ \quad + c_{i,j+1}^k - 4c_{i,j}^k\Big). & \text{otherwise} \end{cases} \tag{12}$$

The stability of the system can be measured by the following criteria:

$$\frac{4D\delta t}{\delta x^2} \leqslant 1 \tag{13}$$

We now progress to investigating the steady state of systems, and therefore turn to the time-independent, or Laplace, diffusion equation, given by

$$\nabla^2 c = 0. \tag{14}$$

We assume the same boundary conditions as before. Taking the same spatial discretisation as before and again applying the same 5-point stencil for the second order derivatives. From this, we can transform it into the following set of finite difference equations. [3]

$$\frac{1}{4}\left(c_{i+1,j}+c_{i-1,j}+c_{i,j+1}+c_{i,j-1}\right) = c_{i,j}. \tag{15}$$

It is important to note that the superscript of k is no longer present, meaning that the time-dependent behaviour is now suppressed. Now the three different ways to solve the above equation will be detailed.

We are able to derive the Jacobi method by reintroducing an iterative element k into the equation and follows from equation 12.

$$c_{i,j}^{(k+1)} = \frac{1}{4}\left(c_{i+1,j}^{(k)} + c_{i-1,j}^{(k)} + c_{i,j+1}^{(k)} + c_{i,j-1}^{(k)}\right) \tag{16}$$

We set the stability condition as follows.

$$\frac{\delta t\, D}{\delta x^2} = \frac{1}{4}. \tag{17}$$

As the iterative Jacobi method does not have a limit for the timesteps that it runs, it is limited by a stopping condition. This condition takes the change in one iteration and checks whether it falls below a certain value, which for the sake of this investigation lies at $\epsilon = 10^{-5}$.

$$\delta \equiv \max_{i,j} \left| c_{i,j}^{(k+1)} - c_{i,j}^{(k)} \right| < \epsilon. \tag{18}$$

An improved version of the Jacobi method is the Gauss-Seidel iteration method, where the updated values are used as soon as they are calculated, resulting in the following system.

$$c_{i,j}^{(k+1)} = \frac{1}{4}\left(c_{i+1,j}^{(k)} + c_{i-1,j}^{(k+1)} + c_{i,j+1}^{(k)} + c_{i,j-1}^{(k+1)}\right) \tag{19}$$

Empirically, the Gauss-Seidel method does not provide a big improvement when compared to the Jacobi method. Therefore, a further adjustment can be made to the system wherein an over-correction of the new iterate is included. This resulting system is called the Successive-Over-Relaxation (SOR) method.

$$c_{i,j}^{(k+1)} = \frac{\omega}{4}\left(c_{i+1,j}^{(k)} + c_{i-1,j}^{(k+1)} + c_{i,j+1}^{(k)} + c_{i,j-1}^{(k+1)}\right) + (1-\omega)\, c_{i,j}^{(k)}. \tag{20}$$

Here the parameter ω represents the relaxation factor, meaning the degree to which the new iterate is considered over the previous one.

In order to investigate how the iterative methods compare in the number of iterations needed for the respective systems to converge, they will be compared. In order to determine how the value of ω impacts the convergence of the system under the SOR method, an empirical investigation will determine the optimal relaxation factor. In order to investigate how the size of the grid impacts the optimal relaxation factor, an analysis will be implemented to determine how the resulting optimal ω changes when varying the grid size.

Furthermore, an alteration to the starting grid of $N = 50$ will be implemented. A square object with an area of 100 units squared will be placed at a central location on the grid. The exact location ranges from $(15, 15)$ to $(25, 25)$. The investigation will then focus on determining how the inclusion of the shape impacts the number of iterations needed to reach convergence, as well as the impact the shape has on the optimal omega.

IV. Results

Figure 1 illustrates the evolution of the system under different initial conditions. Whereby these conditions influence how the system behaves over time, while still following a periodic pattern.

In Figure 1a and 1b, the wave maintains a smooth and periodic structure, with oscillations that shift over time. Figure 1c shows a more localized wave, which spreads as time progresses. This is because the values are zero if they go outside the given bound.

Figure 2 illustrates the diffusion of heat in the two-dimensional system, where an initial localized temperature at the top of the square spreads smoothly as time progresses. In this figure, heat dissipates symmetrically across the system. As $N_{steps} \to \infty$, the system approaches equilibrium, where the temperature becomes uniform throughout.

To validate the simulation, the model was compared to the analytical solution:

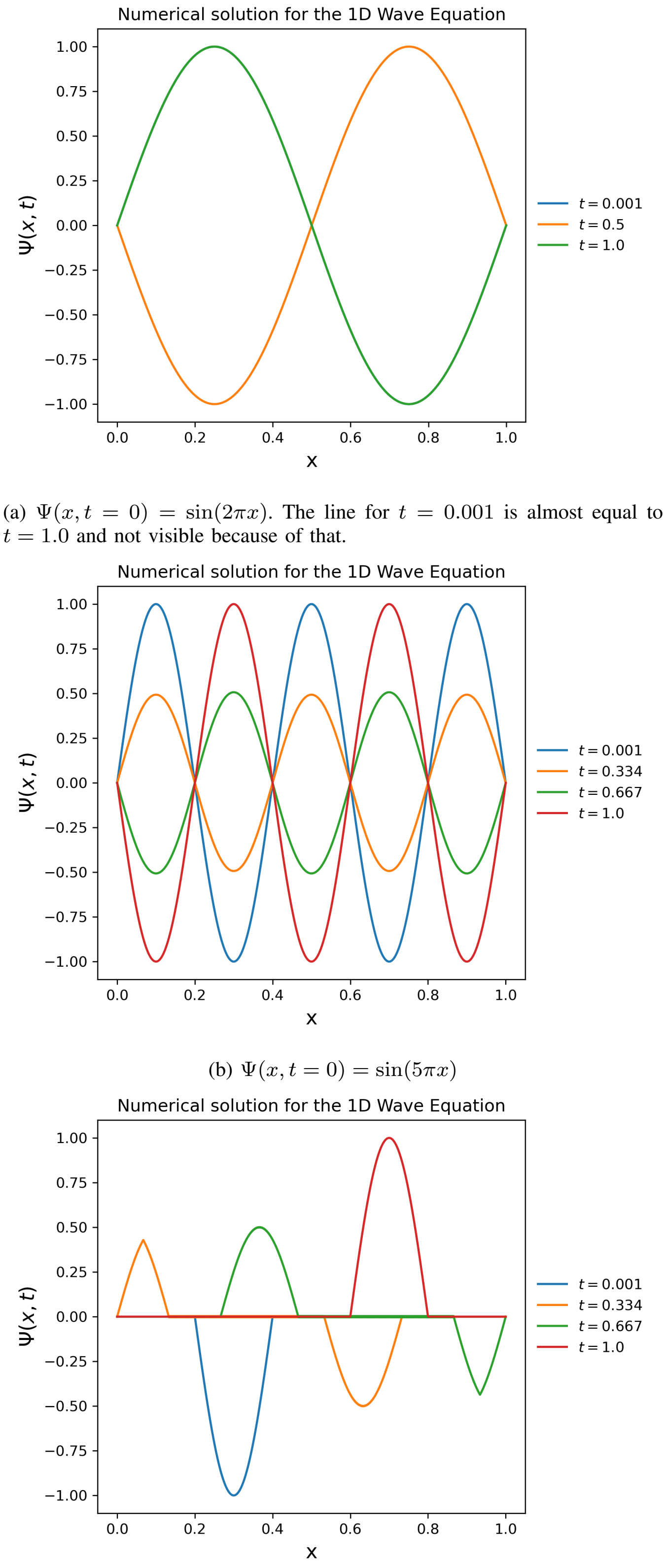
$$c(x,t) = \sum_{i=0}^{50} \operatorname{erfc}\left(\frac{1-x+2i}{2\sqrt{Dt}}\right) - \operatorname{erfc}\left(\frac{1+x+2i}{2\sqrt{Dt}}\right) \tag{21}$$

The comparison of which is shown in Figure 3. As shown in the figure, the simulation closely matches the analytical solution, particularly at $t = 1$, while the other time steps exhibit small deviations to the right.

Turning now to time-independent diffusion methods, in order to study the steady state of the system. The Jacobi method, the Gauss-Seidel method, and the Successive-over-Relaxation methods will be investigated with the Laplace equation.

Figure 4 highlights how the iterative methods compare in terms of the iterations needed to reach the convergence determined by the stopping criteria. Three separate values of ω were chosen initially, as they correspond to the range of expected ideal values for this system. [3] One can note that the SOR method with an $\omega = 1.9$ is the most effective at converging. One can also observe that as the ω value is increased, the rate of convergence increases. Compared to the Jacobi and Gauss-Seidel method the SOR proves to be more efficient by a considerable margin. It therefore is of interest to investigate empirically what the optimal omega value is, in order to further optimise the system.

Testing various values in the range of $1.7 < \omega < 2$, Figure 5 indicates that as the omega value increases from 1.7 upwards we can observe that the number of iterations



(a) $\Psi(x, t = 0) = \sin(2\pi x)$. The line for $t = 0.001$ is almost equal to $t = 1.0$ and not visible because of that.

(b) $\Psi(x, t = 0) = \sin(5\pi x)$

(c) $\Psi(x, t = 0) = \sin(5\pi x)$ for $1/5 < x < 2/5$, and $\Psi = 0$ elsewhere.

Fig. 1: Wave functions with the following parameters: $L = 1.0, T = 1.0, c = 1, \textit{spatial steps} = 1000, \textit{time steps} = 1000$.

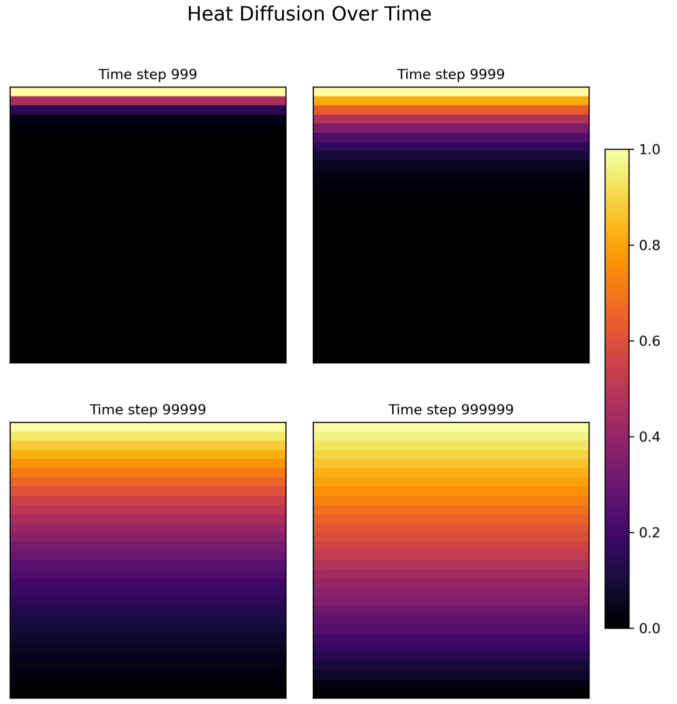


Fig. 2: Diffusion over a 2D grid with sizes of $N = 30$, diffusion coefficient $D = 1, \delta x = \frac{1}{N}, \delta t = 0.001$ and $N_{\text{steps}} = 1.000.000$.

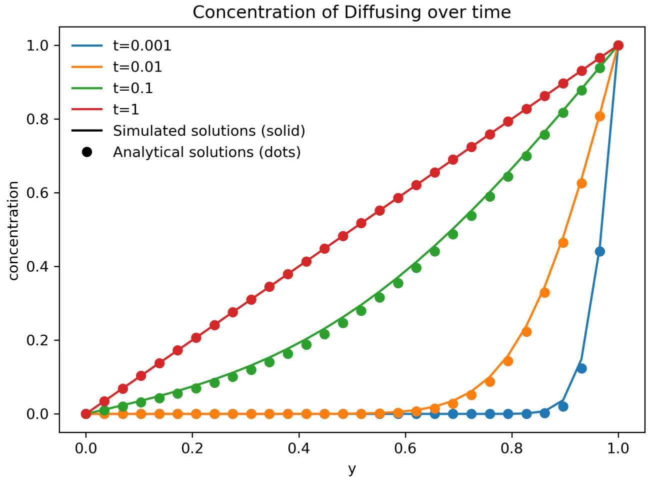


Fig. 3: Comparison between the analytical solution and the simulated solution for different time steps of the simulation. The analytical solution is hereby plotted as dots over the simulated one. The following parameters were used: $N = 30, D = 1, \delta x = \frac{1}{N}, \delta t = 0.001$ and $N_{\text{steps}} = 1.000.000$.

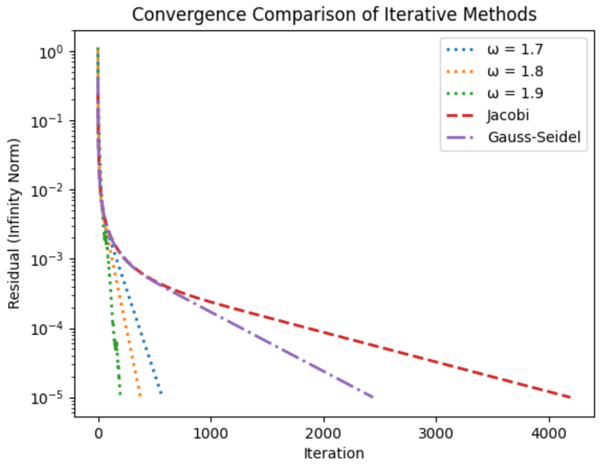


Fig. 4: Comparison of iterative method by number of iterations needed to reach convergence with gridsize of $N = 50$ and stopping criterion $\epsilon = 10^{-5}$.

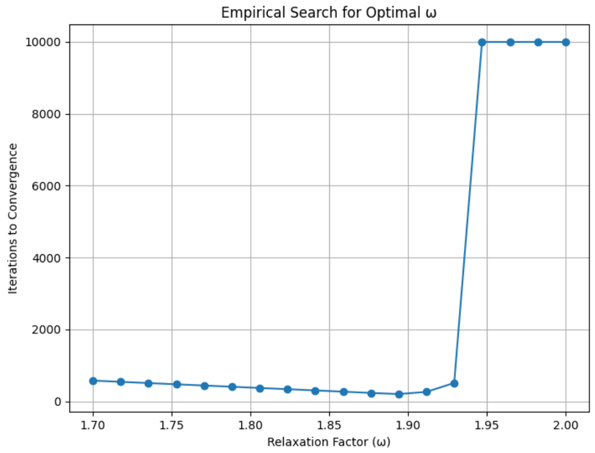


Fig. 5: Empirical search of optimal ω in terms of convergence. Gridsize of $N = 50$ and stopping criterion $\epsilon = 10^{-5}$.

steadily decreases until we reach $\omega \approx 1.9$. From there the number of iterations slowly increases until $\omega \approx 1.925$, after which the system no longer converges. The empirical optimal value is $\omega = 1.894$. The figure shows the number of iterations as $10,000$, which is due to that being the maximum limit of iterations that our simulation would allow. One would expect that the actual number of iterations needed to reach convergence, if at all, to be considerably higher. This is due to the fact that as the spectral radius approaches 1, the relative change in the residual occurs at very small increments, meaning that a vast quantity of iterations are needed.

Thus far only a grid size of $N = 50$ has been considered. However, the question of how the grid size may have an effect on the optimal omega is yet to be answered.

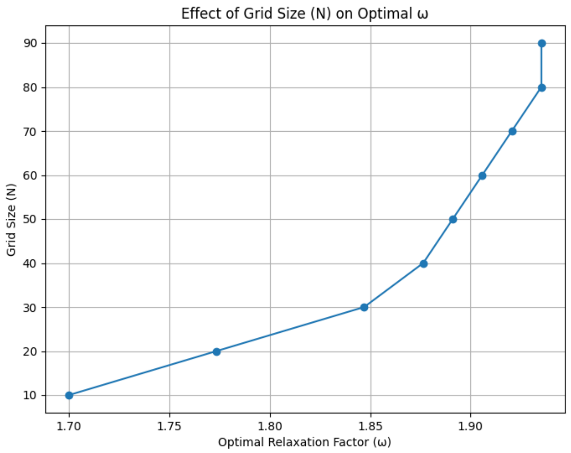


Fig. 6: Comparison of optimal ω to gridsize, with stopping criterion $\epsilon = 10^{-5}$.

The results from Figure 6 demonstrate that as the grid size increases, the optimal omega value needed for the optimal rate of convergence also increases. The previous result of $\omega = 1.894$ for grid size $N = 50$ is confirmed here. As we approach a very large grid size we find that ω approaches the value of 2. This can be explained by the fact that as the grid becomes larger, the diffusive process slows down in a relative sense, and the dominant eigenvalues of the iteration matrix approach 1, meaning that the SOR method compensates by using a larger relaxation factor.

So far, only a standard grid has been considered, namely a square grid that is completely homogeneous initially. However, in real-life settings we seldom encounter such ideal conditions. Therefore, implementing an intrusive structure that could influence diffusion in some manner is worth investigating.

As can be observed in Figure 7, the presence of an object in the middle of the grid leads to a decreased number of iterations needed to reach convergence when compared to the empty grid. One can observe a similar trend to Figure 5, where moving up from $\omega = 1.7$ through to $\omega = 1.9$ generally leads to a faster convergence rate. However, it can be argued that the system which includes a shape actually converges more quickly for the lower relaxation factors, and this effect balances out as the optimal omega values are reached. For the system which includes the shape, the optimal value was found to be $\omega = 1.8618$, somewhat lower than that of the homogeneous grid, which remained at $\omega = 1.8941$. As the higher relaxation factors are reached, we once more observe that both systems diverge; however, the grid that contains the shape does so at a faster rate than the empty

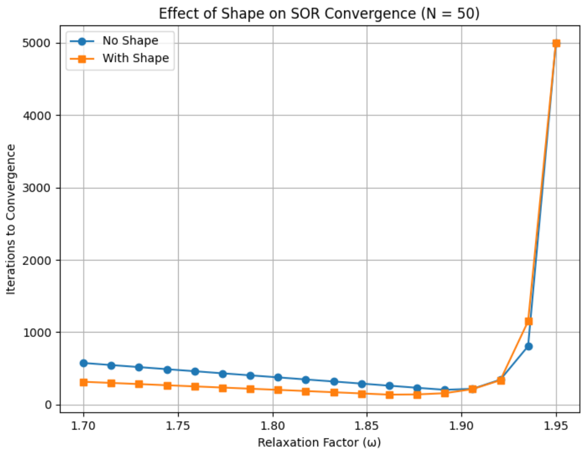


Fig. 7: Comparing optimal relaxation factors to convergence rate between empty grid and grid with square object at centre of grid.

grid.

V. Discussion

For the wave equation, the results are as expected, including periodic behaviour and overall smooth propagation of the wave. We picked the sizes of the spatial and time steps such that $\frac{\Delta t^2}{\Delta x^2} = 1$.

Upon testing, we were able to see that when the ratio of the spatial and time steps was further from 1, the method was unstable. An example of such a pair of steps is $\Delta x = 0.02$ and $\Delta t = 0.01$.

For the time-dependent diffusion equation, we were able to validate the simulation by comparing the model to the analytical solution. We also concentrated on optimisation of the code to ensure we can run many simulations effectively. We saw that the system went into a steady state eventually.

As for the time-independent diffusion equation, we were able to compare the performance of the three iterative methods by comparing the residuals versus the number of iterations. We showed that the SOR method converges the fastest followed by Gauss-Seidel. Even though the Gauss-Seidel method only provide a slight improvement when it comes to the convergence speed, one of its advantages is the fact that we no longer need to create and store a copy of the grid, but instead, we can do the perform the update in place.

On the other hand, the SOR method provides significantly faster convergence for carefully chosen value of ω. We showed that the optimal value of ω also depends on the gridsize and needs to be adjusted accordingly.

When it comes to the effectiveness of the implementation, we could have parallelised the Gauss-Seidel and SOR algorithms using the red-black ordering.

Lastly, we examined the behaviour of the SOR algorithm with a presence of an object in the domain. Our testing was limited to include one object of a square shape in the centre of the grid. We could study the influence of objects further by introducing objects of different shapes and changing their position, size and total number. This would give us more in-depth understanding of how the area and shape of the objects impact the diffusion.

References

[1] J. L. Davis, *Mathematics of wave propagation*. Princeton, New Jersey: Princeton University Press, 2000.

[2] A. Fick, “Ueber diffusion,” *Annalen der Physik*, vol. 170, no. 1, pp. 59–86, 1855. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.18551700105

[3] G. Závodszky, “Assignments for scientific computing,” PDF file, 2025, assignment sets covering topics such as the vibrating string, time-dependent and time-independent diffusion equations, iterative methods (Jacobi, Gauss-Seidel, SOR), diffusion-limited aggregation, and more.