Scientific Computing Exercise Set 1

Peter Voerman (11749547) & Leon Kuiper (12366943)

I. Introduction

PARTIAL differential equations can be very difficult and sometimes even impossible to solve analytically. Luckily it is possible to make approximations and solve these approximations using computers. This method is used in a wide range of fields; from computational fluid dynamics used in the production of aeroplanes and formula 1 cars to the field of computational finance where PDEs are used to model price movements of stocks and other financial products.

In this paper we will focus on two PDEs in particular: the wave equation and the diffusion equation. The wave equation is an equation that relates the change of the amplitude of a string in time to the change when moving along the string in space. The diffusion equation relates the change of concentration in time to the change of concentration in space.

In particular we will investigate how these numerical approximations perform for the wave equation and the time dependent diffusion equation. Furthermore, we will use various methods to find the steady state of the time independent diffusion equation to see how they perform. Finally, we will experiment with adding sinks to the diffusion grid to see how they affect the simulations.

II. Theory

A. Wave equation

The wave equation is

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

where the solution $\Psi(x,t)$ is the vibration amplitude as a function of position x and time t. The boundary conditions are $\Psi(x=0,t)=0$ and $\Psi(x=L,t)=0$, with L the length of the string.

If we want to implement the wave equation in a computer program, we need to discretize the equation, resulting in

$$\Psi(x,t) \equiv u(i,j), \tag{2}$$

where the solution now depends on the integers i and j. i goes from 0 to the amount of discretized parts of the string and j goes from 0 to the amount of timesteps. The first derivatives of the discretized equation are

$$\begin{aligned} \frac{\partial \Psi}{\partial x} &\equiv \frac{u(i+1,j)-u(i,j)}{\Delta x} \\ \frac{\partial \Psi}{\partial x} &\equiv \frac{u(i,j)-u(i-1,j)}{\Delta x}. \end{aligned} \tag{3}$$

The second derivatives are

$$\begin{aligned} \frac{\partial^2 \Psi}{\partial x^2} &\equiv \frac{u(i+1,j)+u(i-1,j)-2u(i,j)}{(\Delta x)^2} \\ \frac{\partial^2 \Psi}{\partial t^2} &\equiv \frac{u(i,j+1)+u(i,j-1)-2u(i,j)}{(\Delta t)^2}. \end{aligned} \tag{4}$$

Filling these in the original wave equation gives the new discretized wave equation

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

We set L = 1 for simplicity and divided the string in N intervals, such that $\Delta x = L/N$.

When implementing the time stepping with the two most recent time points to compute the next one using equation 5, we get

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

B. Time dependent diffusion equation

Another differential equation that is often difficult to solve analytically is the diffusion equation given by

$$\frac{\partial c}{\partial t} = D\nabla^2 c, \tag{7}$$

with $c(x,y;t)$ the concentration as a function of the coordinates x and y and time t and the diffusion constant D.

This equation by itself can not be solved numerically. Just like the wave equation, it has to be discretized first. In order to discretize this equation, the following definition is used:

$$c(i\delta x, j\delta y, k\delta t) = c^k_{i,j}. \tag{8}$$

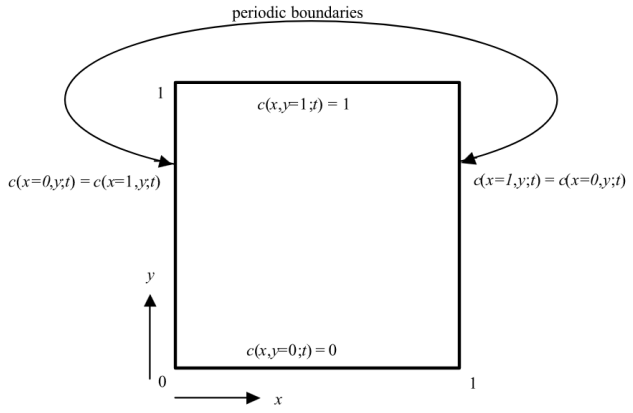


Fig. 1: The boundary conditions used in this paper.

In this definition δx and δy are defined as the width and height of the grid divided by the number of discretization points and δt as the simulation time divided by the number of timesteps. i, j and k are integers that go from 0 to the number of discretization points in the x and y dimensions for i and j and from 0 to the number of timesteps for k.

The discretization of the diffusion equation can be derived in a similar way to the wave equation. By applying the same approximations as the ones used in the wave equation, the derivatives become

$$\frac{\partial c}{\partial t} = \frac{c_{i,j}^{k+1} - c_{i,j}^{k}}{\delta t} \tag{9}$$

$$\frac{\partial^2 c}{\partial x^2} = \frac{c_{i+1,j}^{k} + c_{i-1,j}^{k} - 2c_{i,j}^{k}}{\delta x^2} \tag{10}$$

$$\frac{\partial^2 c}{\partial y^2} = \frac{c_{i,j+1}^{k} + c_{i,j-1}^{k} - 2c_{i,j}^{k}}{\delta y^2}. \tag{11}$$

Filling in these derivatives into equation 7 then allows for the calculation of the concentration at subsequent timesteps with

$$c_{i,j}^{k+1} = c_{i,j}^{k} + \frac{\delta t D}{\delta x^2}\Big(c_{ib+1,j}^{k} + c_{i-1,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k} - 4c_{i,j}^{k}\Big). \tag{12}$$

In this paper, a periodic boundary condition will be used along the x axis. The boundaries at the top and bottom of the grid are set to 1 and 0 respectively. A representation of these boundary conditions is shown in figure 1. fig

Due to these boundary conditions, the left side of the grid, the cells border the ones on the right side of the grid due to the periodic boundary conditions. If the grid goes from $i = 0$ to $i = N$, this leads to the following equation at the left side of the grid:

$$c_{0,j}^{k+1} = c_{0,j}^{k} + \frac{\delta t D}{\delta x^2}\Big(c_{1,j}^{k} + c_{N,j}^{k} + c_{0,j+1}^{k} + c_{0,j-1}^{k} - 4c_{0,j}^{k}\Big). \tag{13}$$

Since the top and bottom of the grid have constant values of 1 and 0 respectively, during simulation only the values of the cells from $j = 1$ to $j = N - 1$ need to be calculated.

Using equation 13 is only stable if

$$\frac{4\delta t D}{\delta x \delta y} \leq 1. \tag{14}$$

This means that a grid 1x1 in size divided into 100 grid points in either axis with a diffusion constant D of 1, the timesteps have to be smaller than or equal to $2.5 \cdot 10^{-7}$.

C. Time independent diffusion equation

If only the steady state of a concentration profile is of interest, the time independent diffusion equation can be used. This equation is given by

$$\nabla^2 c = 0. \tag{15}$$

Since the time derivative is set to 0, this equation can be interpreted as the steady state solution of the time depentent diffusion equation.

By combining this equation with equations 10 and 11, we find the following equation:

$$\frac{1}{4}(c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1}) = c_{i,j}. \tag{16}$$

Note that the superscript k is no longer present since we're no longer interested in the time aspect of the concentration.

If we now use k to denote the number of iterations, we can find the steady state by using

$$c_{i,j}^{k+1} = \frac{1}{4}(c_{i+1,j}^{k} + c_{i-1,j}^{k} + c_{i,j+1}^{k} + c_{i,j-1}^{k}). \tag{17}$$

This method is called Jacobi iteration and it is nothing more than using the maximum allowed timestep given by equation 14.

Since we are no longer interested in the time aspect of the diffusion but instead in the steady state, we need to set a stopping condition. In this paper, the stopping condition

$$\delta = \max_{i,j} |c_{i,j}^{k+1} - c_{i,j}^{k}| < 10^{-5} \tag{18}$$

was used.

A more efficient method for finding the steady state is called Gauss-Seidel iteration. With this method, values

calculated in the $(k+1)$th iteration are used to calculate values further along in the grid. This results in the equation

$$c_{i,j}^{k+1} = \frac{1}{4}(c_{i+1,j}^{k} + c_{i-1,j}^{k+1} + c_{i,j+1}^{k} + c_{i,j-1}^{k+1}). \quad (19)$$

It is important to remember the boundary conditions when using this method, so on the borders of the grid, Jacobi iteration was still used while the other cells were calculated using Gauss-Seidel iteration.

One final improvement on both these methods is called successive over relaxation. In this method the calculation of the next cells is given by

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

This method should converge for $0 < \omega < 2$. When $\omega = 1$, the equation is the same as the one for Gauss-Seidel iteration. The optimal ω can depend on factor such as the grid size and on possible objects in the grid, so it may by prudent to search for a good value of ω before doing large simulations.

III. Methods

A. Wave equation

We implemented equation 6 with the initial conditions that the string is at rest at $t = 0$, $\Psi'(x, t = 0) = 0$, and the following three starting positions of the string:

(i) $\Psi(x, t = 0) = sin(2\pi x)$
(ii) $\Psi(x, t = 0) = sin(5\pi x)$
(iii) $\Psi(x, t = 0) = sin(5\pi x)$ if 1/5 $< x <$ 2/5, else $\Psi = 0$

Furthermore, we set $c = 1$,the time step size $\Delta t = 0.001$, amount of intervals the string is divided to $N = 1000$ and we stopped the simulation after $t_{max} = 1$. Due to the method requiring the position of the string two timesteps back, the starting position of the string was also used for the second timestep. Because the timesteps are so small, the potential effect this had on the evolution of the string was negligible.

B. Time dependent diffusion equation

For all calculations a 100x100 grid was used of which the top row was set to a constant value of 1 and the bottom row was set to a value of 0. The subsequent timesteps were calculated using equation 12. At a few timesteps, 2D plots were made of the concentration to show the progression of the diffusion. The concentration was also compared to the analytical solution given by

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

C. Time independent diffusion equation

For the time independent diffusion equation, all three methods described above were used to compute the steady state of the grid. The amount of time it took to reach the steady state was compared to find out how efficient the methods were compared to each other. Multiple values of ω were also tested to find the best one. Finally, we experimented by adding sinks in the grid which had a constant value of 0 to see how they affected the simulation times.

IV. Results

A. Wave equation

The the wave equation was solved numerically for a piece of string connected to a wall on both sides with three starting positions. First, with starting position $\Psi(x, t = 0) = sin(2\pi x)$, see Figure 2. Secondly, the simulation was done on a string with starting position $\Psi(x, t = 0) = sin(2\pi x)$, see Figure 3. In these simulations we can see the conservation of energy. Finally, the string was simulated with starting position $\Psi(x, t = 0) = sin(5\pi x)$ if 1/5 $< x <$ 2/5, else $\Psi = 0$, see Figure 4. Here we can see that constructive interference in the waves is occurring.

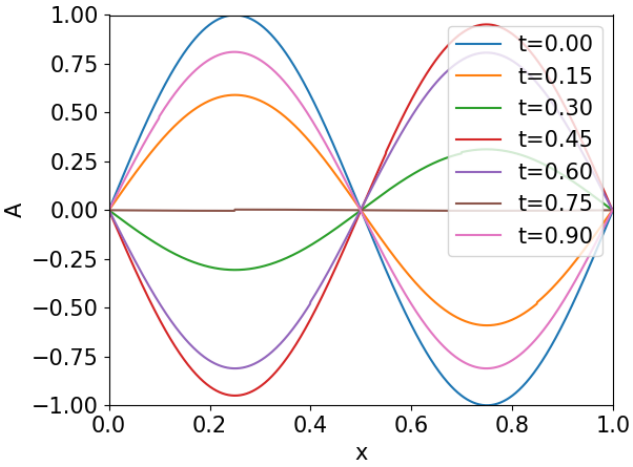


Fig. 2: The development of the string with time step $\Delta_t = 0.001$, string length $L = 1$ and the string was divided in $N = 1000$ intervals. The state of the same string is plotted for several times. The starting position of the string was $\Psi(x, t = 0) = sin(2\pi x)$.

B. Time dependent diffusion equation

The numerical solution of the time dependent diffusion equation is compared to the analytical solution in figure 5. Both solutions are shown at various timesteps

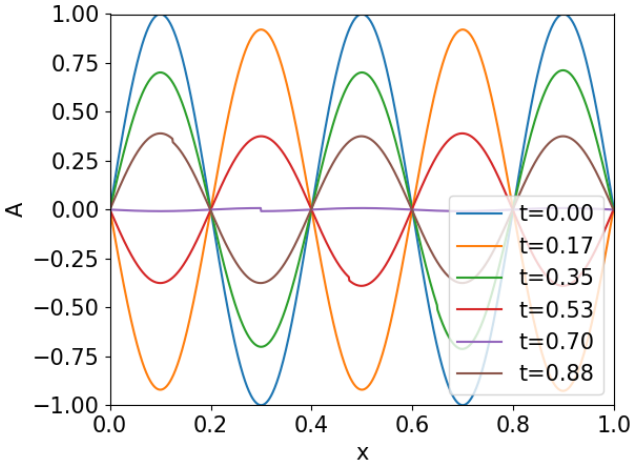


Fig. 3: The development of the string with time step $\Delta_t = 0.001$, string length $L = 1$ and the string was divided in $N = 1000$ intervals. The state of the same string is plotted for several times. The starting position of the string was $\Psi(x, t = 0) = sin(5\pi x)$.

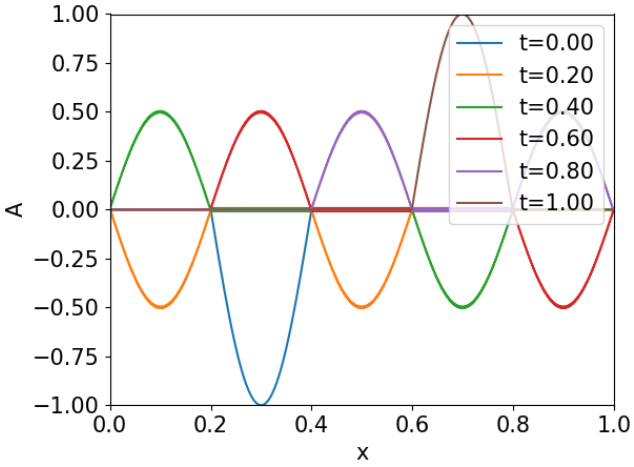


Fig. 4: The development of the string with time step $\Delta_t = 0.001$, string length $L = 1$ and the string was divided in $N = 1000$ intervals. The state of the same string is plotted for several times. The starting position of the string was $\Psi(x, t = 0) = sin(5\pi x)$ if 1/5 $< x <$ 2/5, else $\Psi = 0$.

and the solid and dotted lines represent the numerical and analytical solutions respectively.

Figure 6 shows the progression of the diffusion through various timesteps. The simulation starts with a row with a concentration of 1 at the top and all other cells with a concentration of 0. During the simulation the concentration of those cells changes according to the diffusion equation.

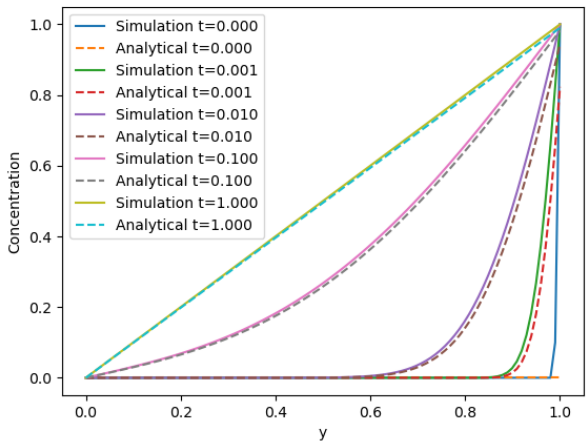
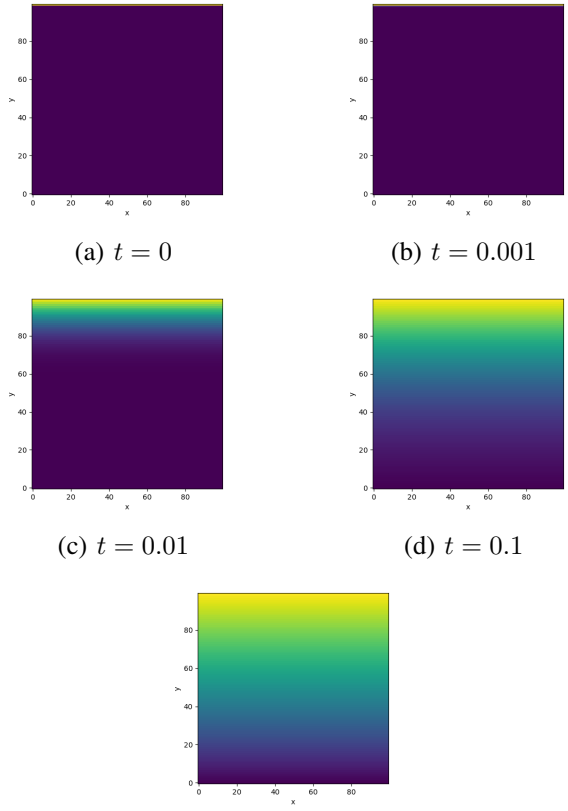


Fig. 5: The numeric solution of the time dependent diffusion equation compared to the analytical solution during various timesteps.



(a) $t = 0$ (b) $t = 0.001$

(c) $t = 0.01$ (d) $t = 0.1$

(e) $t = 1$

Fig. 6: The progression of the diffusion shown at various timesteps.

C. Time independent diffusion equation

Jacobi iteration, Gauss-Seidel iteration and successive over relaxation are compared in figure 7. Every line

shows the progression of the δ parameter during the simulation. For the methods where the line goes up, the simulation didn't converge.

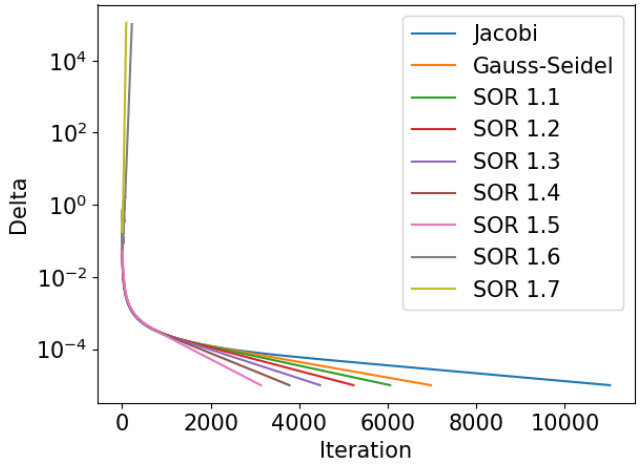


Fig. 7: The progression of the delta parameter during the calculations of the steady state solution using various methods.

Figures 8 and 9 show the results when sinks were added to the grid. Two sinks were added and in these regions the concentration was always set to 0. The coordinates of the sinks were (20,40),(70,80) and (70, 80), (80, 96) (in the format (x1, x2),(y1, y2)).

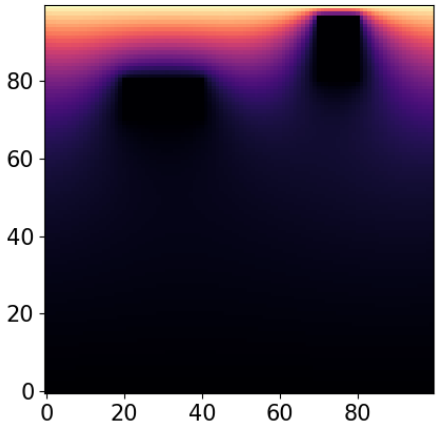


Fig. 8: The configuration of the two sinks that were added to study the effect of sinks on the number of iterations and ω.

V. Discussion

A. Wave equation

The laws of physics were abided by with the numerical implementation of the wave equation. We saw that the maximum amplitude and the period of the wave remained the same (Figure 2 & 3), therefore there was conservation of energy. We furthermore saw constructive interference happening, see Figure 4.

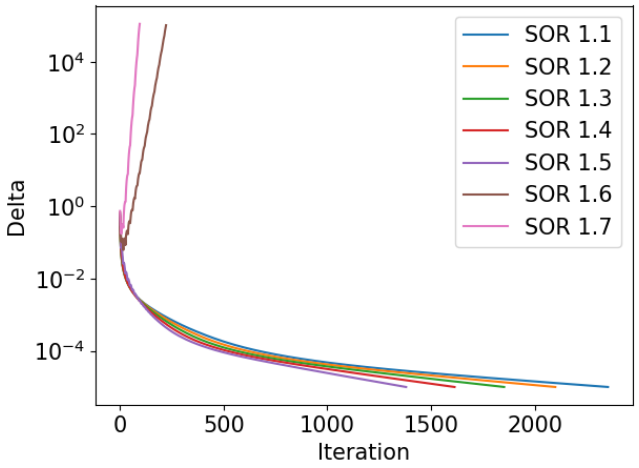


Fig. 9: The progression of the delta parameter during the calculations of the steady state solution using various methods. This time two sinks were added in the configuration that can be seen in Figure 8.

B. Time dependent diffusion equation

The comparison between the numerical and analytical solution of the time dependent diffusion equation shows that the numerical solution follows the progression of the analytical solution although there is some difference between them.

The progression of the diffusion is as expected. The concentration at the bottom of the grid rises as time passes and the distribution is symmetric along the x-axis.

C. Time independent diffusion equation

For the calculation of the steady state of the grid, Jacobi iteration needed the most iterations to reach the steady state. Gauss-Seidel iterations needed approximately half the iterations and successive over relaxation with the optimal value for ω needed approximately 25% of the iterations needed during the Jacobi iteration.

The best value for ω we found was 1.51. This is outside of the predicted range in the exercise, so there might be a mistake in the code we were not able to find. This value was the best for a grid size of 25, 50, 75 and 100.

When sinks were added, successive over relaxation needed even less iterations to reach a steady state. This can be explained by the fact that less cells changed

each iteration which allowed the stopping condition to be reached earlier.

VI. Conclusion

Solving partial differential equations is often a difficult task. However, it is sometimes possible to use numerical approximations in order to find a solution. In this paper, we used some of these approximations to find solutions to the wave equation and the diffusion equation.

The wave equation was solved numerically and the system abided by the laws of physics. Even with more complicated waves, the amplitude and period behaved as expected.

We compared the numerical approximation to the analytical solution belonging to specific boundary conditions. The numerical solutions did differ slightly from the analytical solution, but they were close enough to give a good impression of how the diffusion progressed through time.

For the time independent diffusion equation, we compared three methods: Jacobi iteration, Gauss-Seidel iteration and successive over relaxation. We found that successive over relaxation needed the fewest iterations to reach a steady state, although this method did not converge for all values of ω.

Finally we experimented by adding sinks to the grid. These sinks meant that an analytical solution should be even harder to find, but for the numerical methods, it made the amount of iterations needed to solve the equations even less than the number of iterations needed without a sink.

All these different simulations show that numerical approximations can often serve as a good alternative to analytical solutions and can sometimes even be a better way of arriving at a solution.