

# Scientific Computing Exercise Set 3

Ya'gel Schoonderbeek(10791213) & Zaida Rivai (12124303)

## I. INTRODUCTION

The wave equation and the diffusion equation are one of the most commonly known Partial Differential Equations (PDE's). These two equations have been studied and used in many different fields of research, for example in studies on sound, heat, fluid dynamics or electrodynamics.

In this paper, the wave equation is used in a numerical approach to study the eigenmodes of a drum. A famous question posed within the music world is:

*"Can you hear the shape of the drum?"*

Put into scientific terms, this would be similar to:

*"Can the membrane shape can be identified uniquely from its eigenvalue spectrum?"*

The first part of this paper examines this question for three different shapes of membranes: a square, a rectangle and a circle. For each shape, the eigenmodes are simulated and studied by finding numerical solutions to the wave equation. The second part of this paper studies the diffusion of a nutrient in a source-sink model. Direct methods are used based on the five-point stencil to find the steady-state solution.

## II. THEORY

### A. Eigenmodus of drums or membranes of different shapes

The properties of an ideal circular drum membrane (drum head) of uniform thickness can be modeled by vibrations of a circular membrane [1]. Due to resonance, the drum head can store vibrational energy moving its surface in a characteristic pattern of standing waves at certain vibration frequencies [1].

The evolution of these vibrations can be described by the wave equation in two dimensions, which is given as follows:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \quad (1)$$

where  $\partial/\partial t$  is the partial time derivative,  $u = u(x, y, t)$  the height of the drum membrane (which describes which parts is vibrating),  $c^2$  a constant and  $\nabla^2$  the 2-dimensional Laplace operator [3].

Specifically, we are interested in a solution where time and space dependencies are independent with respect to each other, i.e. a solution of the form

$$u(x, y, t) = v(x, y)T(t) \quad (2)$$

where  $v(x, y)$  describes the spatial behaviour and  $T(t)$  the time behaviour. Inserting this into the two dimensional wave equation and moving all t-dependencies to the left and all  $x, y$  dependencies to the right results in:

$$\frac{1}{c^2 T(t)} \frac{\partial^2 T(t)}{\partial t^2} = \frac{1}{v(x, y)} \nabla^2 v(x, y) \quad (3)$$

Important to note is the fact that the left side depends on different variables than the right side, respectively  $t$  and  $x, y$ . This relation of two functions depending on different variables requires both sides to be equal to some constant  $K$ . Rewriting this for the left side leads to

$$\frac{\delta^2 T(t)}{\delta t^2} = K c^2 T(t) \quad (4)$$

which is a commonly know PDE. It's solution depends on the value of  $K$ . First of all,  $K < 0$  results in an oscillating solution:

$$T(t) = A \cos(c\lambda t) + B \sin(c\lambda t) \quad (5)$$

where  $\lambda^2 = -K$ .  $\lambda > 0$  can be assumed without loss of generality.

If  $K = 0$ , the solution is either constant or a linear function of  $x$  and  $y$ . Due to the boundary conditions, it has to be  $v = 0$ , which is not an interesting case.

Lastly, for  $K > 0$ , the solutions grows or decays exponentially, which is also not interesting for this particular problem.

Having found a solution for the left hand side of eq. 3, let us examine the right hand side, which gives:

$$\nabla^2 v(x, y) = Kv(x, y) \quad (6)$$

Solutions to eq 6 are given by eigenmodes  $v$  and eigenfrequencies  $\lambda$ . By discretising eq. 6 using Taylor expansions, one obtains the numerical approximation

$$\begin{aligned} \nabla^2 v_{i,j} &\approx v_{i-1,j} + v_{i+1,j} + v_{i,j-1} + v_{i,j+1} - 4 \cdot v_{i,j} \\ &= Kv_{i,j} \end{aligned} \quad (7)$$

where  $c_{i,j}$  is the concentration modelled on position  $x = i \cdot dx$  and  $y = j \cdot dy$  in the computational grid.

This can be interpreted as a set of linear equations for each gridpoint in the computational domain. These linear equations can be written in the form of a matrix  $M$ , which satisfies

$$Mv = Kv, \quad (8)$$

where  $v$  is a vector containing the discretised grid points of  $v(x, y)$  and  $K$  is the constant from *eq. 6*.

The equation  $Mv = Kv$  describes a general eigenvalue problem. In the next section III, the construction of this matrix  $M$  for the drum membranes will be further described. The eigenvalues of a system correspond to the eigenfrequencies of the system, which is the frequency at which the drum membrane would be observed to be oscillating (in absence of any damping or driving force). The set of all eigenvalues of  $v$  is known as the spectrum of  $v$ , which describes the sound of a drum. The lowest eigenvalue, corresponding to the lowest frequency or the lowest note, is known to be the keynote.

Thus, modelling the eigenfrequencies provides insight on the tone of the drum. The frequency of a wave can be described by:

$$f = 0.5n\frac{v}{L} \quad (9)$$

in which  $f$  is the frequency in Hz,  $v$  is the speed of the wave in  $m/s$ ,  $n$  is the  $n^{th}$  wave and  $L$  is the length of the domain. For the keynote,  $n = 1$ , the eigenfrequency  $f$  is expected to decrease for an increasing drum length  $L$ .

### B. Direct methods for solving steady-state problems

In previous personal research, the steady-state solution of the diffusion equation for a source-sink model was studied with iterative methods. The 2-dimensional diffusion equation is given by

$$\frac{\partial c}{\partial t} = \nabla^2 c \quad (10)$$

with  $\partial/\partial t$  being the partial time derivative,  $c = c(x, y, t)$  the concentration of a nutrient and  $\nabla^2$  the two dimensional Laplace operator.

The steady-state solution is constant over time, i.e.:

$$\frac{\partial c}{\partial t} = 0 = \nabla^2 c \quad (11)$$

Discretising this formula results in

$$\begin{aligned} \nabla^2 c_{i,j} &\approx c_{i-1,j} + c_{i+1,j} + c_{i,j-1} + c_{i,j+1} - 4 \cdot c_{i,j} \\ &= 0 \end{aligned} \quad (12)$$

where  $c_{i,j}$  is the concentration modelled on position  $x = i \cdot dx$  and  $y = j \cdot dy$  in the computational grid. Equation 12 can be translated to a set of linear equations, similar to the ones described in the previous section. These linear equations can be written in the form of a matrix  $M$ , for which  $Mc = b$ . The construction of this matrix is described in the section III.

### 1) Direct methods compared to iterative methods:

Before we turn to the implementation of the direct method, it is useful to reflect on the reasons to use direct methods. As described by Heath [2], there are a couple of differences between direct methods and iterative methods, which are shortly summarised here:

- 1) A direct method does not require an initial estimate of the solution, but therefore can also not be improved by a good one;
- 2) Direct methods produce high accuracy in the numerical solution, which might be unnecessary delay if only low accuracy is desired;
- 3) Iterative methods often depend on special properties and only show quick convergence for properly conditioned systems. Direct methods are more robust in both senses;
- 4) Iterative methods usually require less work if convergence is rapid. However, often various parameters must be computed to achieve this
- 5) Iterative methods mostly require little explicit storage and hence are good when the matrix can be easily implemented;
- 6) Iterative methods are less readily embodied in standard software packages

What is important to take away is that direct methods generally provide a more stable method to calculate the numerical solution. In addition, although they require much more memory, direct methods might also be quicker in achieving the same accuracy compared to iterative methods. Therefore, it is interesting and relevant to study their implementation.

## III. METHODS

### A. Two dimensional wave equation

For this research, three different shapes for the membrane were modelled: a square domain of  $L \times L$ , circle domain with radius  $L/2$  and a rectangle domain of  $2L \times L$ . For most simulations, the length was set at  $L = 1$  and the spatial differences at  $\delta x = 0.02$ . Different parameter settings will be mentioned for particular simulations. In all models, the boundaries are fixed at 0. For example,

for the square domain, this translates to the boundary conditions

$$v(0, y) = v(1, y) = v(x, 0) = v(x, 1) = 0 \quad (13)$$

*1) Construction of  $M$ :* Here, the construction of the matrix  $M$  is illustrated by the means of an example for the square domain in a small grid of  $4 \times 4$ . The computational grid  $V$  is given by

$$V = \begin{bmatrix} v_{00} & v_{01} & v_{02} & v_{03} \\ v_{10} & v_{11} & v_{12} & v_{13} \\ v_{20} & v_{21} & v_{22} & v_{23} \\ v_{30} & v_{31} & v_{32} & v_{33} \end{bmatrix}$$

In order to solve the eigenvalue problem, we transform our  $4 \times 4$  grid  $V$  into a vector  $v$  of length 16 by filling in the points by going down from left to right. In a general case, the  $n \times n$  matrix is transformed into a vector of length  $n^2$ . This results in:

$$v \equiv [v_{00} \ v_{01} \ v_{02} \ v_{03} \ v_{10} \ v_{11} \dots \ v_{32} \ v_{33}]^T \quad (14)$$

Now, we want to translate the equations resulting from *eq. 7* into a matrix  $M$  of size  $16 \times 16$  (or  $n^2 \times n^2$  in the general case). First of all, consider a non-boundary point  $v_j$  that must obey *eq. 7*. For the  $j^{th}$  row of matrix  $M$ , we see that

- The element of the diagonal to be  $-4$
- The elements left & right of this diagonal element need to be 1, representing the left and right neighbor of the grid
- The elements positioned 4 places to the right and 4 places to the left of the diagonal element are set at 1 as well. For the general case, this is  $n$  (the amount of columns of grid  $V$ ).

Summarised, we obtain:

$$M_j \equiv [0 \ \dots \ 1 \ 0 \ 0 \ 1 \ -4 \ 1 \ 0 \ 0 \ 1 \ \dots \ 0] \quad (15)$$

In addition, we must consider all boundary points, which are fixed. The row of matrix  $M$  related to this point does not depend on the other points. Therefore, only the element on the diagonal is nonzero. The element of the diagonal of this row is set at  $-4$  in line with the other points. Thus, if element  $v_i$  is a point on the edge, the  $i^{th}$  row of matrix  $M$  is defined as

$$M_i \equiv [0 \ \dots \ 0 \ -4 \ 0 \ \dots \ 0] \quad (16)$$

where the nonzero part is positioned on the  $i^{th}$  column.

This all can be rewritten in the form  $Mv = Kv$ , thus our eigenvalue problem becomes:

$$\begin{bmatrix} -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} v_{00} \\ v_{01} \\ v_{02} \\ v_{03} \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{20} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{30} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} \quad (17)$$

$$= K \begin{bmatrix} v_{00} \\ v_{01} \\ v_{02} \\ v_{03} \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{20} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{30} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} \quad (18)$$

Note that for this example of a  $4 \times 4$  grid, most rows of  $M$  contain only a nonzero element on the diagonal. However, for a  $n \times n$  grid with  $n \gg 4$ , most points are non-boundary points and this would not be the case.

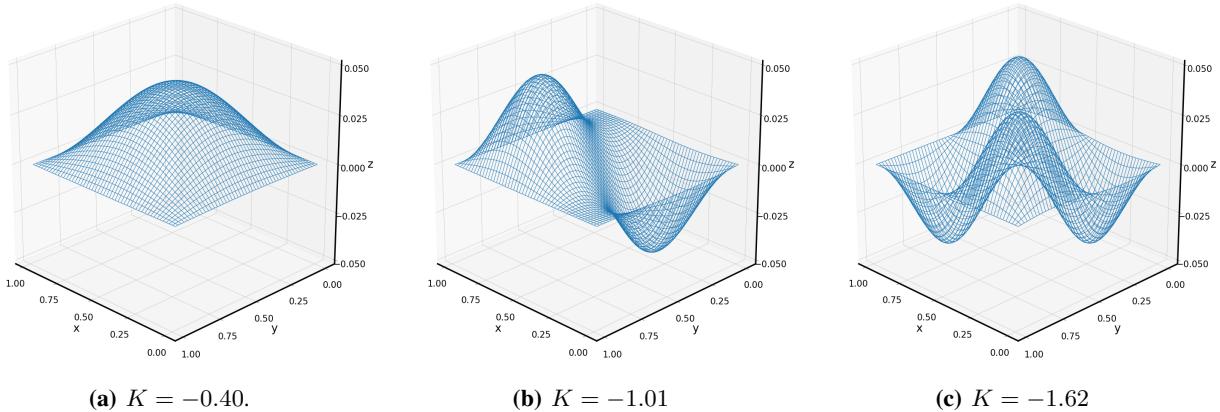
For the circle domain, a computational grid of  $N_x \times N_x$  was generated, which was transformed to a vector of size  $N_x^2$  with  $N_x = L/\delta x$ . It was calculated which points were inside the circle. For those points, the corresponding matrix row was constructed similar to the example above (*eq. 15*). The other points were treated as boundary points, for which the matrix row is the same as *eq. 16*.

The matrix  $M$  for the rectangular domain was constructed in the same manner. However, as the grid  $V$  is now of size  $N_x \times 2N_x$ , vector  $v$  of size  $2N_x^2$  and matrix  $M$  of size  $2N_x^2 \times 2N_x^2$ . Note that upper and lower neighboring points are now found  $2N_x$  to the right and left from the diagonal element.

*2) Python packages for eigenvalue problem:* In this research, calculating the eigenvalues and eigenvectors of the three different membrane shapes was done by use of the Scipy package in Python. First, a benchmark test was done with function *scipy.linalg.eig*, which returns eigenvalues and eigenvectors for a general matrix. Unfortunately, *scipy.linalg.eigh* could not be used as matrix  $M$  as described above is not symmetric, which would provide a speed up in the computational time. However, the matrix is obviously sparse, which is why it is possible to use *scipy.sparse.linalg.eigs*. The computational time associated of both function was studied and is discussed below.

### B. Direct method for solving steady-state problems

For this particular study, the computational domain was taken to be a circle with radius  $L = 1$ . A square of



**Fig. 1:** Three dimensional plots of eigenmodes of a drum membrane in the shape of a square corresponding to the three lowest absolute eigenvalues. The size of the computational domain was set at length  $L = 1$  with a grid blocks of size  $dx = 0.02$ . As the absolute value of the eigenvalue increases, the amount of peaks increases as well. The eigenmodes appear to always consist an even amount of peaks, apart from the eigenmode with a single peak.

size  $2N_x \times 2N_x$  was generated with  $N_x = L/\delta x$ ,  $\delta x = 0.015$ . For each point it was checked whether it was inside the domain. On the boundary of this disk, a sink was modelled, i.e.  $c(x, y, t) = 0$  for all  $x, y$  on the boundary of the disk. On point  $x = 0.6, y = 1.2$  a source was modelled, i.e.  $c(0.6, 1.2, t) = 1$ . All points outside of the circle were also kept at zero, thus treated as boundary points.

Now, let us take a look at the construction of matrix  $M$ , for which  $Mc = b$  describes eq. 11. Similar to the example described in the previous section, the discretised grid  $C(x, y)$  is transformed to vector  $c$  by filling in all elements from left to right, top to bottom.

For a point  $c_i$  at the boundary of the disk (or outside the disk), the concentration is fixed at 0 and does not depend on neighboring points. Therefore, similar to the previous section, the row  $M_i$  contains only a nonzero element on the diagonal, which is set at -4. This also applies to the point at (0.6, 1.2). For all other points, equation 12 applies, which results in rows as described in the previous section by equation 15.

Looking at eq. 12, vector  $b$  must be set at 0. However, we have introduced a source point, for which the concentration of the nutrient is kept at 1. Its corresponding element in vector  $b$  must also be 1.

#### IV. RESULTS

### A. Eigenmodes of drums or membranes of different shapes

Figure 1 shows the modelled shape of a vibrating square drum membrane for the three smallest eigenvalues  $K$ . As seen in eq. 5 for increasing absolute eigenvalues  $K$ , the frequency of the vibrating membrane increases.

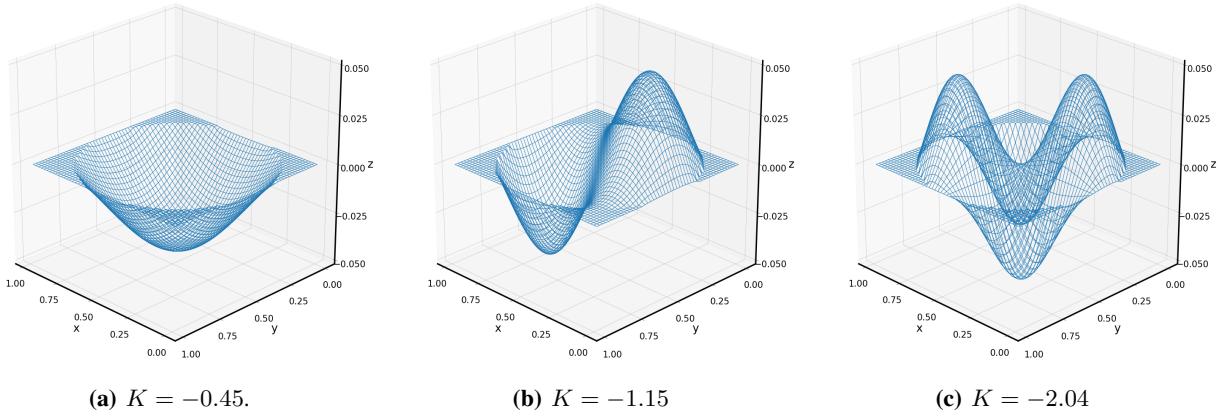
Thus, the key note and the first two note higher than the key note are plotted. For the lowest absolute eigenvalue, i.e. 0.40, the vibrating membrane contains only one peak. As the absolute eigenvalue increases (1.01&1.62), the amount of waves increases as well. It appears that the square membrane has only eigenmodes with an even amount of peak, except for the eigenmode with a single peak.

A similar behaviour was found for a circular domain, see figure 2. Note that the absolute eigenvalues are slightly larger than those of the square domain. This is logical, as the membrane surface is smaller, its eigenmodes will be vibrating more frequently, which corresponds with higher eigenvalues.

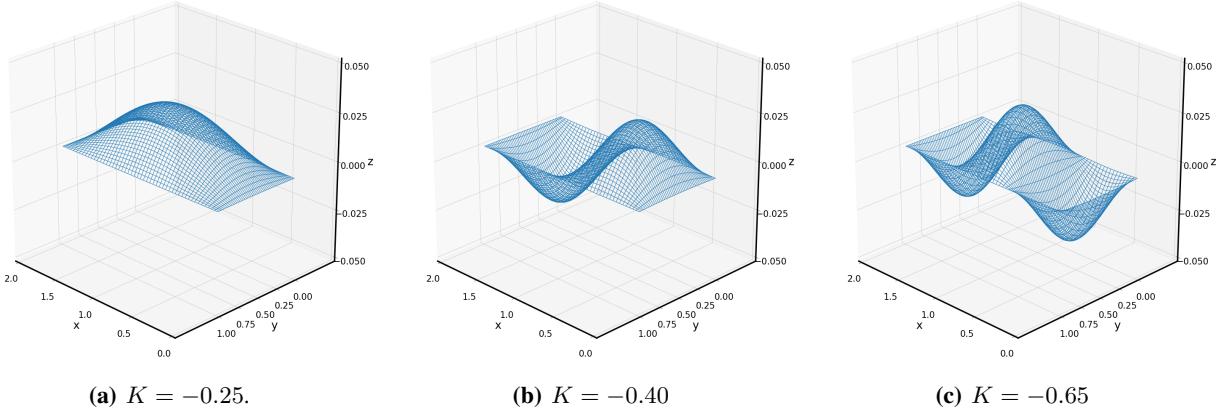
The rectangle membrane, plotted in figure 3, shows somewhat different behaviour. First of all, its absolute eigenvalues are significantly smaller, which again is logical as its surface is twice as big as the square membrane. In addition, the rectangular domain has an eigenmode with 3 peaks, which is not the case for the square- and circle membrane. This hints at a fundamental difference in eigenmodes, as the rectangle domain seems to support odd amount of waves.

## *B. Algorithms based on regular matrices versus sparse matrices*

As mentioned before, the matrix  $M$  is clearly sparse, which is why it is possible to use algorithms based on sparse matrices. This might lead to a considerable speed up of our computational time. Figure 4 shows a comparison in the computational time for calculating the eigenvalues and eigenvectors using both an algorithm based on general matrices as well as one based on sparse



**Fig. 2:** Three dimensional plots of eigenmodes of a drum membrane in the shape of a circle corresponding to the three lowest absolute eigenvalues. The size of the computational domain was set at length  $L = 1$  with a grid blocks of size  $dx = 0.02$ . As the absolute value of the eigenvalue increases, the amount of wave increases as well. Similar to the square-shaped drum (see fig. 1), the eigenmodes appear to always consist an even amount of peaks, apart from the eigenmode with a single peak. As this membrane is slightly smaller than the square membrane, its eigenmodes are quicker oscillating, which is why the absolute eigenvalues are larger than those of the square membrane.



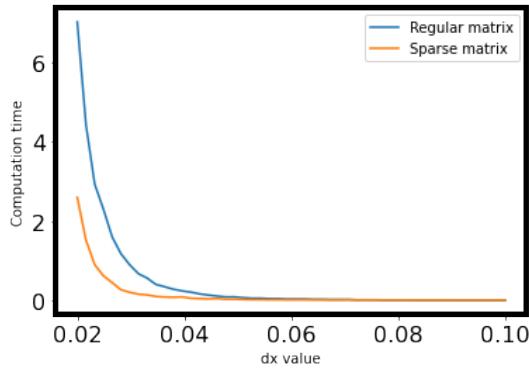
**Fig. 3:** Three dimensional plots of eigenmodes of a drum membrane in the shape of a rectangle. The size of the computational domain was set at length  $L = 1$  with a grid blocks of size  $dx = 0.02$ . The long side is of the rectangle is of size  $2L$ . As the absolute value of the eigenvalue increases, the amount of wave increases as well. Unlike the square-shaped and circle-shaped drum (see fig. 1 & 2), this drum membrane has an eigenmode with 3 peaks. The observed absolute eigenvalues are also significantly smaller than those of the other two shapes.

matrices. Shown in both plots, is that the algorithm based on sparse matrices is faster than the one based on general matrices. This difference only becomes observable for lower values of  $\delta x$ , which corresponds with more discretisation steps and thus much larger matrices. For most simulations done in this research, where  $\delta x = 0.02$ , the computational time can be reduced roughly from 6 to 2 seconds. Such a speed up is notable, but not essential for this particular study. However, it must be noted that for bigger simulations, i.e. larger  $L$  or smaller  $\delta x$ , it is highly recommended to use an algorithm based on sparse matrices.

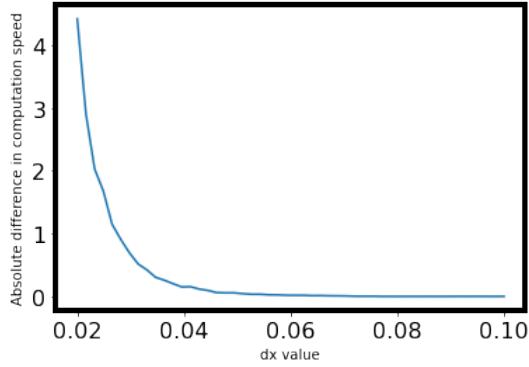
### C. The dependency of eigenvalues $K$ on length $L$

In figure 5, the smallest absolute eigenvalue, which corresponds to the key tone, is plotted for increasing  $L$  for the different membrane shapes. The amount of discretisation steps is fixed, i.e. for  $n$  steps the spatial differences are set at  $\delta x = L/n$ . As expected, the smallest absolute eigenvalue decreases for larger  $L$ . As the membrane gets bigger, its vibrations become less frequent and thus its smallest eigenvalue decreases.

Furthermore, figure 5a and figure 5b show the influence of the amount of discretisation steps on the eigenvalues. Shown is that for both 30 and 50 discretisation



(a) Absolute computational time



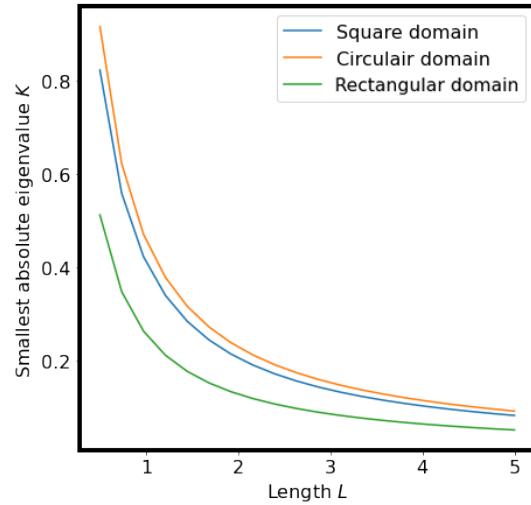
(b) Relative difference in computational time

**Fig. 4:** Plots of the computational time of the algorithm based on regular matrices and the algorithm based on sparse matrices for varying finite spatial difference  $\delta x$ . The upper plot shows the absolute computational time of each algorithm and the lower plot shows the relative difference. The algorithm based on sparse matrices only shows notable speed up for  $\delta x \leq 0.04$ . These results were found by simulating the square membrane for  $L = 1$ . The finite spatial difference  $\delta x$  relates to the grid size of  $M$  (eq. 8) by  $N_x = L/\delta x$ .

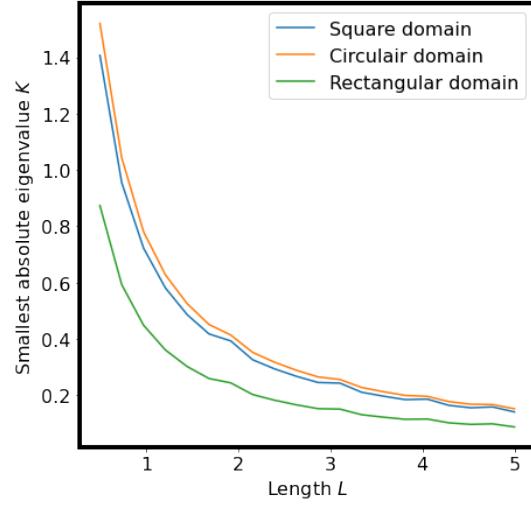
steps the eigenvalues decrease as the size of the shape increases and the behaviour of the square and circular domain are for both discretisation steps more similar than the rectangular domain.

#### D. Direct methods for solving steady state problems

In the last part of this paper, we try to use direct methods solving steady state problems. Figure 6 shows the behaviour of the steady-state solution of the source-sink model as described in the previous section. In the figure 6a, the concentration is plotted linearly and it is seen that this concentration decays rapidly when moving away from the source. This is logical, as the concentration is introduced by a single point and is able to diffuse in all directions. The right plot 6b shows the concentration on a logarithmic scale, which illustrates the speed of this decay. Due to time limits for this



(a) 50 discretisation steps



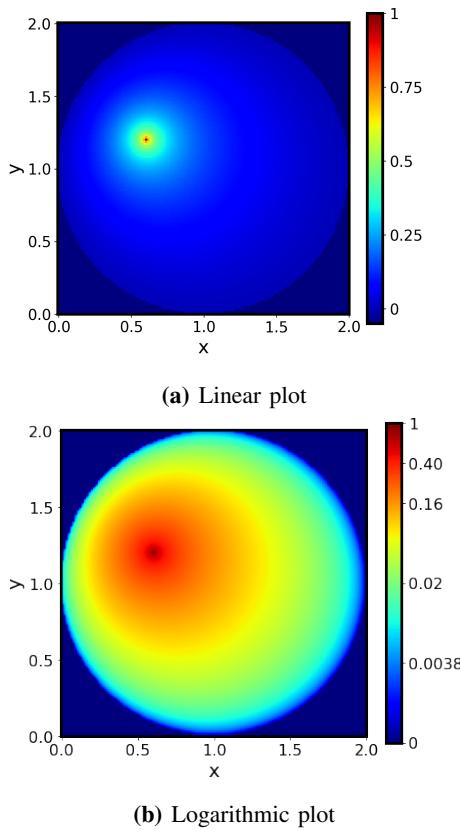
(b) 30 discretisation steps

**Fig. 5:** The smallest eigenvalue corresponding to different sizes of  $L$  for each modelled membrane shape. The smallest eigenvalue relates to the key tone of the membrane. As  $L$  increases, the smallest eigenvalue decreases. This is expected as a larger membrane allows vibrations of less frequent oscillations. This is found for both 30 and 50 discretisation steps.

research, it was not possible to compare the robustness, computational time and other properties of this direct method to iterative models.

## V. DISCUSSION

One prime aim of this paper was research the question *Can the membrane shape can be identified uniquely from its eigenvalue spectrum?* Three different membrane shapes were modelled, for which the eigenmodes were numerically calculated by the two-dimensional wave equation: a square, a circle and a rectangular. The observed eigenvalues of the circle membrane and the square membrane were very similar. However, the observed



**Fig. 6:** Plots of the steady-state solution of the diffusion of a nutrient in a source-sink model. The modelled domain is a circle with radius  $L = 1$  in a computational domain of  $2L \times 2L$ . The boundary of the circle is a sink, i.e.  $c = 0$  and the point at  $x = 0.6$ ,  $y = 1.2$  is a source, i.e.  $c = 1$ . In each plot, the concentration of the nutrient is shown, which is kept at 0 outside the circle. As only a single point introduces nutrient to the system, which diffuses away from the source in all directions, the concentration decays quickly when moving away from the source. The logarithmic plot illustrates the speed of this decay.

eigenvalues for the rectangle membrane significantly differed from the other two membranes. In addition, for the rectangle an eigenmode with 3 peaks was found in contrast to the other two membranes. Thus, it appears that a rectangle membrane can be identified from a square or circle membrane. The latter two seem to be more difficult to distinguish from each other.

In further examination of modelling these drum membranes, the computational speed of two algorithms for the eigenvalue problem were tested: one general method and one based on sparse matrices. Significant speed up was observed only for small discretisation steps, thus large matrices.

A last result on the drum membranes was found on the behaviour of the smallest absolute eigenvalue

corresponding to different lengths of the shape. For each shape, the smallest eigenvalue decreased for increasing length  $L$  in the same manner. This general behaviour did not change by changing the amount of discretisation steps: the eigenvalues were only amplified.

In the last part of this paper, we used a direct method to solve the steady-state solution of a source-sink model. According to the results, the steady-state solution describes a rapid decay of the concentration of the nutrient away from the single source point in the domain. Unfortunately, due to insufficient time, it was not possible to compare this method with an iterative method.

## REFERENCES

- [1] Nakhlé H Asmar. *Partial differential equations with Fourier series and boundary value problems*. Courier Dover Publications, 2016.
- [2] Michael T Heath. *Scientific computing: an introductory survey*, volume 80. SIAM, 2018.
- [3] Jaap Kaandorp. Scientific computing lecture nodes. 2020.