

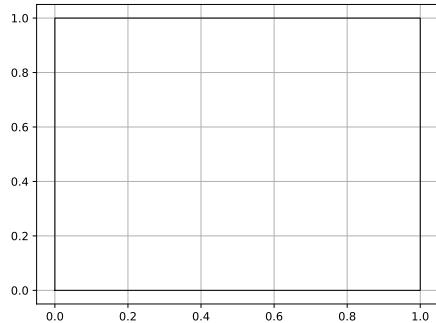
TFY4235 Numerical Physics Assignment 1

Jostein Kløgetvedt

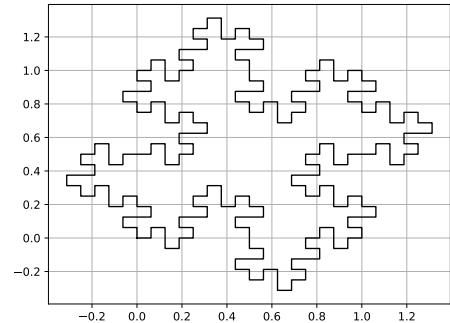
February 2022

Fractal

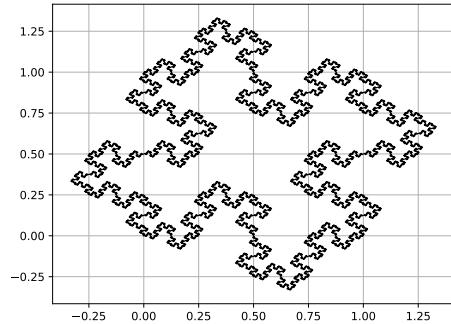
The Koch-fractal is generated by dividing each side into eight new pieces. All of the side-lengths are either vertically or horizontally oriented. We start with a square of length $L = 1$. Figure 1 shows the quadratic Koch fractal for different generations $l = 0, 2, 4$ and 6 .



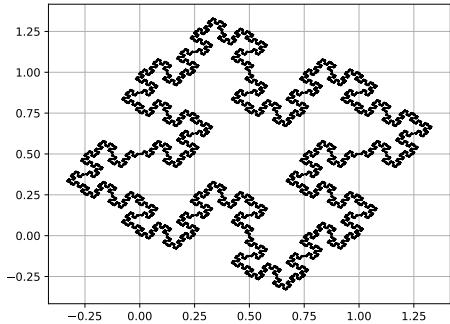
(a) $l = 0$



(b) $l = 2$



(c) $l = 4$



(d) $l = 6$

Figure 1: The quadratic Koch fractal for generation $l = 0, 2, 4$ and 6 . The initial square is set to have sides of length $L = 1$.

Grid

When defining a square lattice constant δ do we see that the generator that splits up a line into eight new lines does not change the initial two points. This means that the gridpoints with a grid lattice adjusted to a fractal generation of $l = l_{max}$ will also fall into place for fractals with $l < l_{max}$. Since the line distance is reduced by $s \rightarrow s/4$ for each time can we set $\delta = \frac{L}{4^{l_{max}}}$. By the phrase, 'a grid adjusted to a fractal of l_{max} ', do I mean that there are no points inbetween each corner of the fractal, so the shortest line in the fractal matches with the grid constant δ . We also see that the grid length and height increases for each generation. The additional distance d can be calculated as follows

$$d = \sum_{i=1}^{l_{max}} \frac{L}{4^i}. \quad (1)$$

Classifying the grid points

We will look at two methods for determining whether a grid point is inside, outside or on the boundary. The first method that I used is based on complex integration. The idea is to find a complex function $f(z)$ that models a pole in the point P and integrate that function along the fractal boundary. The xy -plane will now represent a complex plane where the y -axis is the imaginary axis. A line integral of $f(z)$ along the contour will then yield zero if a point is located outside the fractal, and $2\pi i$ if it's inside. The function takes the simple form

$$f(z) = \frac{1}{z - P}. \quad (2)$$

With this function the closed line integral can be computed as

$$\oint_C f(z) dz = \sum_{i=1}^n \int_{S_i} \frac{1}{z - P} dz = \sum_{i=1}^{n-1} \left(\log\left(\frac{V_{i+1} - P}{V_i - P}\right) \right) + \log\left(\frac{V_0 - P}{V_{n-1} - P}\right), \quad (3)$$

where S_i are the line segments of the fractal and V_i, V_{i+1} are the vertices of that line segment. Before doing this calculation do we have to check whether the point P is on the boundary. This method was taken from [1].

The other method is based on making a ray (a line) that starts at the point in focus, and goes towards infinity. By calculating how many times the ray intersects the fractal can we determine whether the point is inside or outside the boundary. If the number of intersections is odd, then the point is inside, and even if it's outside the boundary. The method causes some errors if a intersected line segment is collinear to the ray. This problem is handled by first check whether the point is on the boundary, and if not, move it by a small amount in a direction perpendicular to the ray such that they do not intersect anymore. For instance, if the ray goes along the y -axis, then the point is moved to the right by $\delta/2$ so it is inbetween points of the fractal. This method works fine for this particular fractal but would have been more complicated if the boundary had a different shape.

To compare these methods do we look at computation time for different sets of grids and fractals. The results are shown in table 1. Here l is the generation of the fractal we are looking at and l_{max} is the grid adjusted for a fractal of generation l_{max} . We see that the complex integration (CI) method scales badly for larger values of l_{max} comparing to the ray method. This may be because, in the CI-method, we have to calculate a line integral for every point in the grid. This becomes a long a process as the number of fractal corners increases by eight and the number of grid points increases approximately by fifteen. On the other hand, the ray method uses Python's Numpy features more

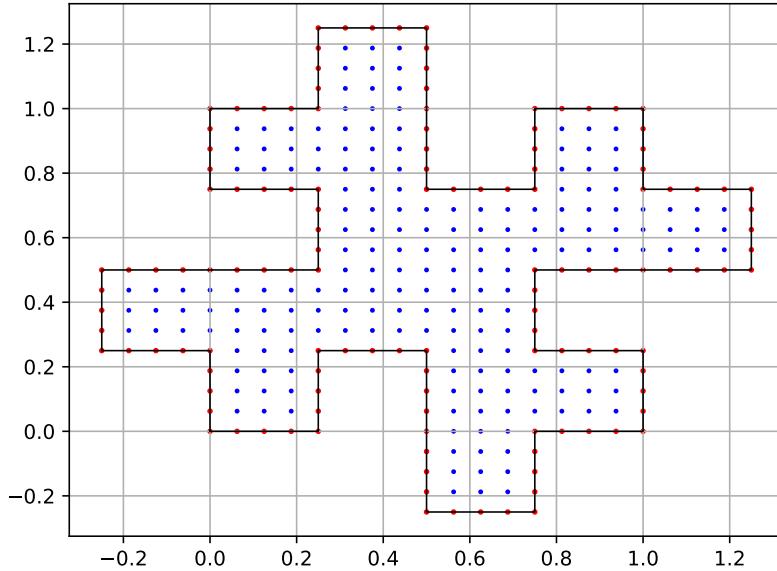


Figure 2: The Koch-fractal for $l=1$ with a grid adjusted for $l_{max}=2$. The red dots are on the boundary while the blue ones are inside.

frequently to classify points that are outside or on the boundary, and only make a for-loop where it counts intersections for points that are inside or not so easily categorized. An example of how the grid points look like inside the fractal is shown in figure 2.

Table 1: Computation times, in seconds.

(l, l_{max})	Ray-method	CI-method
(2,2)	7,45	1,02
(2,3)	7,48	2,77
(3,3)	7,67	14,4
(2,4)	8,37	31,0
(3,4)	11,4	230,2
(4,4)	35,5	1846

Eigenmodes

Assuming that the area enclosed by the fractal is filled with an elastic membrane, it will behave as a fractal drum. We shall investigate how the eigenmodes- and frequencies look like. The equation in focus is the Fourier transformed wave equation, the so-called Helmholtz equation. The equation takes the form

$$-\nabla^2 U(\mathbf{x}, \omega) = \frac{\omega^2}{v^2} U(\mathbf{x}, \omega). \quad (4)$$

A 5-point stencil is used to discretize the 2D-Laplacian operator in the equation. The boundary values are set to zero as there are no oscillations at the boundary. The eigenmodes and values are then found with the function `scipy.sparse.linalg.eigs` since the matrix storing the coefficients in the 5-points stencil is stored as a sparse matrix. The ten smallest eigenfrequencies, $\frac{\omega}{v}$, are shown in table 2 for a Koch-fractal of generation $l=3$. The computation is done on a grid adjusted for a fractal with $l_{max} = 4$. Contour plots of the corresponding eigenmodes are shown in figure 3 and 4. From the table can we see some degeneracy, for instance are the eigenvalues for the second and third excited states similar. This corresponds with the contour plots in 3b and 3c where we can see that the modes are identical, except for a rotation of ninety degrees. Note that beige colour in the contour plots represent approximately zero vibrations. The contour and 3D-plots, which are not shown here, seems to be confirmed by the figures in the article written by Sapoval et al. as mentioned in the project description [2].

Table 2: Smallest eigenvalues ω/v for a fractal with $l=3$, in units m^{-1} .

Mode	1	2	3	4	5	6	7	8	9	10
Value	9.331	13.917	13.917	14.177	14.263	14.920	14.921	17.446	18.663	19.197

Large fractal generation l

The largest value l that I computed the finite difference matrix was for $l=5$ with a grid of $l_{max} = 5$. The finite difference matrix was stored as a sparse matrix. For this value of l , the matrix had the size of 44,0 MB, which took 3.1% of the available memory. The available memory, as printed out in the script, was approximately 1,42 GB. This is consistent with what the task manager showed during a test run.

In my case, it was not the lack of memory that stopped me from going higher, but the computation time. Both classifying the grid points in a fractal and setting up the finite difference matrix are time consuming processes. For instance, the computer used over one hour each to classify and setting up the matrix for $l = 5$. A fractal of generation $l = 5$ has approximately $1 \cdot 10^6$ grid points inside the fractal which means that the finite difference matrix would have had $1 \cdot 10^{12}$ elements had it not been for the sparse data structure. This amounts to several TB of data. To compare, for $l = 6$ are there approximately $20 \cdot 10^6$ grid points inside the fractal, which shows how fast the problem scales in both memory and time.

The challenges to this problem are very clear. The datapoints needed to solve the problem scales incredibly fast. One possible way to solve this problem is to have a more dynamic grid where one for instance choose to have a less dense grid for the points far from the boundary. Another alternative is to make use of symmetry. As we can see from the plots are all eigenfunctions symmetric or antisymmetric about one or more axis. Thus, one should be able to determine the eigenmodes for the whole fractal by only looking at one half, or one quarter of the fractal.

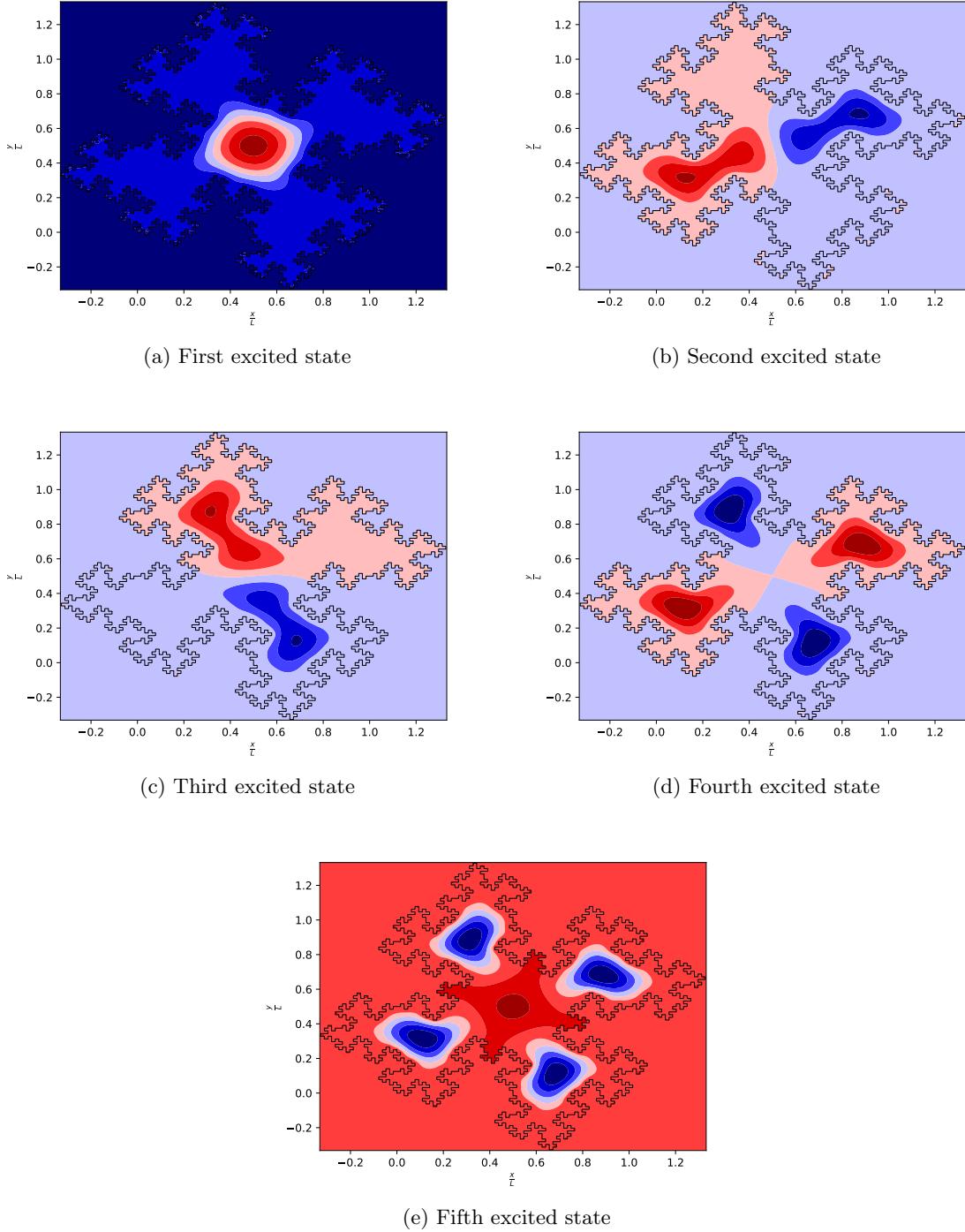


Figure 3: Contour plots of the first to fifth eigenmodes for the Koch-fractal with $(l, l_{max})=(3,4)$.

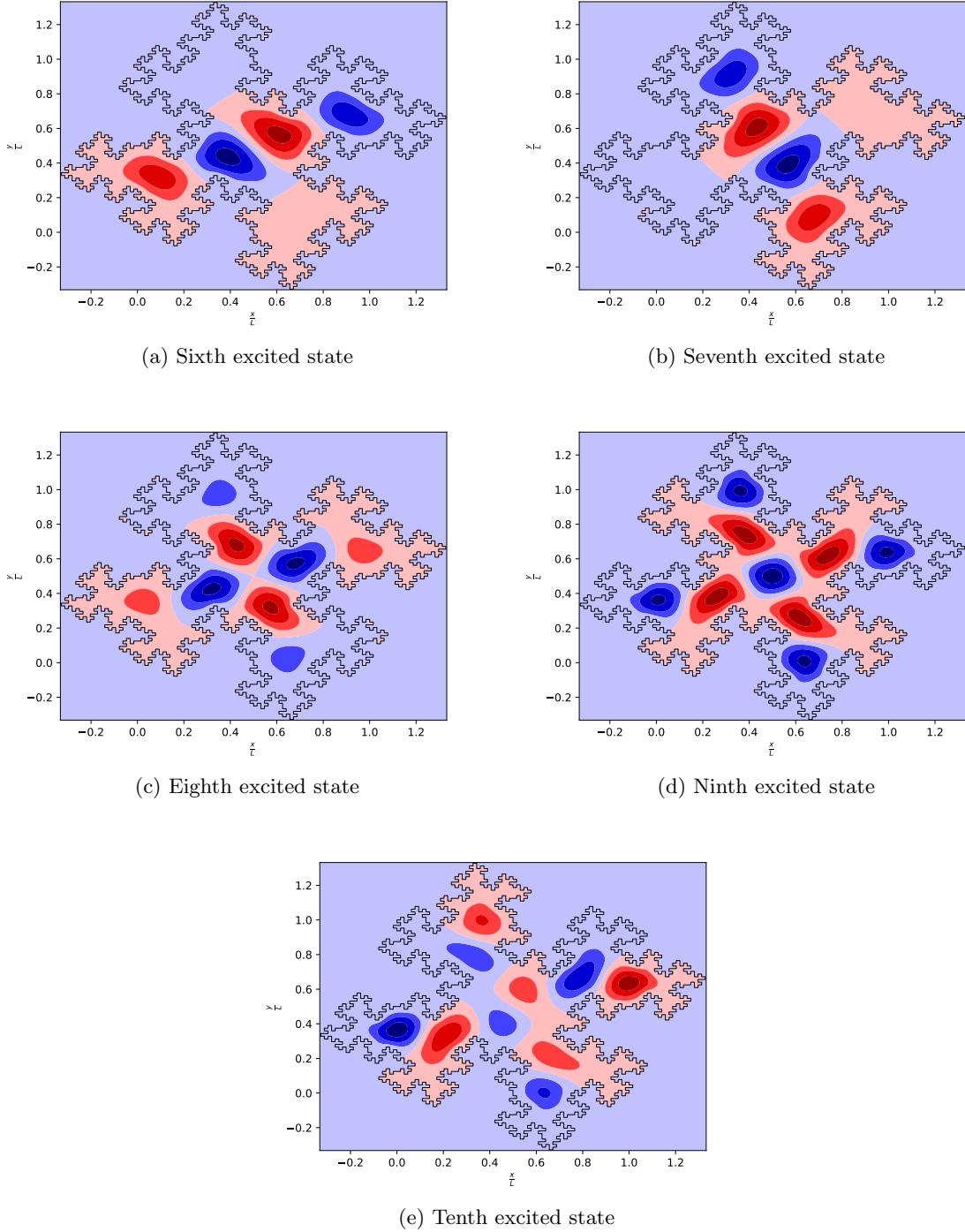


Figure 4: Contour plots of the sixth to tenth eigenmodes for the Koch-fractal with $(l, l_{max})=(3,4)$.

Scaling of $\Delta N(\omega)$

The function $N(\omega)$ is known as the integrated density of states. For a smooth boundary can it be expressed by the Weyl conjecture where it goes as $A\omega^2/4\pi$ for large frequencies. The fractal does not have a smooth surface so the suggested conjecture is the Weyl-Berry conjecture having the same first term and the second term goes as $\sim \omega^d$. This means that $\log(\Delta N(\omega))$, being the difference between the fractal and the smooth case, goes as $\sim d \log(\omega)$. d is here the dimension of the perimeter of the drum. Computing $\Delta N(\omega)$ for several eigenfrequencies makes us capable of computing the slope of the straight line for the log-curve. A figure showing $\Delta N(\omega)$ for several values of (l, l_{max}) is shown in figure 5. Here we have used the hundred smallest eigenvalues for all sets of (l, l_{max}) . The figure shows that all l 's, except for $l=2$, move along approximately the same line. The calculated values of d , for each set (l, l_{max}) , is shown in table 3. The values are calculated by fitting a straight line to the acquired data.

Table 3: Computed values of the scaling d for different sets of (l, l_{max}) .

(l, l_{max})	(2,3)	(3,3)	(3,4)	(4,4)	(4,5)	(5,5)
d	1,413	1,512	1,531	1,550	1,547	1,553

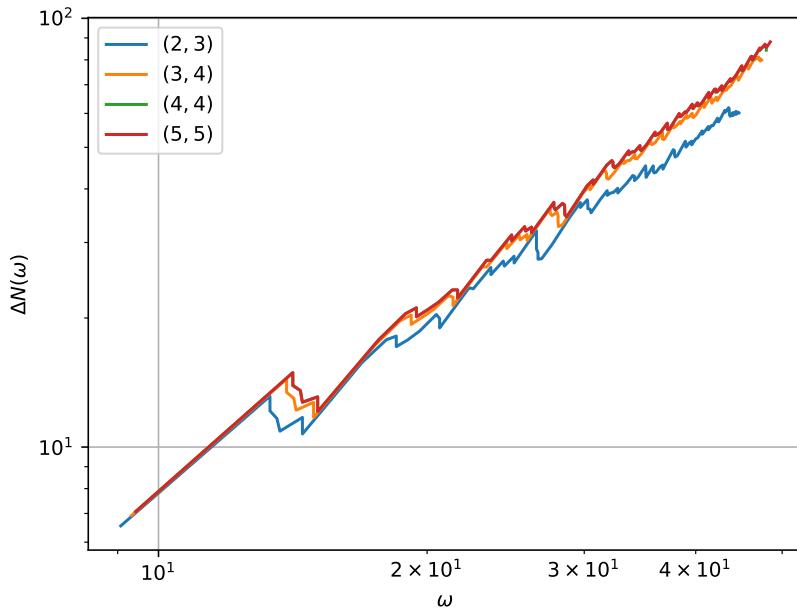


Figure 5: Loglog-plot of $\Delta N(\omega)$ for several values of (l, l_{max}) . The plot is of the hundred smallest eigenvalues ω .

For lower values of l we can see that the value of d is lower. This happens because fractals with low values of l do not have as many unique eigenmodes so the eigenvalues will be very similar, e.g.

a high degree of degeneracy, for higher modes. We can see this effect from the formula

$$\Delta N(\omega) = \frac{A}{4\pi} \omega^2 - N(\omega). \quad (5)$$

$N(\omega)$ will increase in the same way for every new mode, but if ω increases very little for each new mode then $\Delta N(\omega)$ will flatten out and even decrease. This effect is shown in figure 6 where $\Delta N(\omega)$ is plotted for the lowest five hundred eigenvalues for $l = 2$ and the lowest seven hundred eigenvalues for $l = 3$.

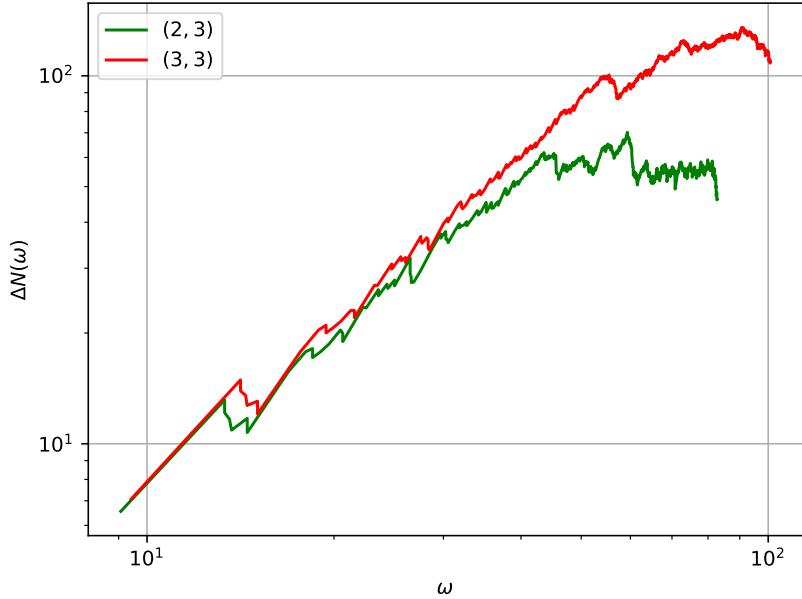


Figure 6: Loglog-plot of $\Delta N(\omega)$ for $(l, l_{max}) = (2, 3)$ and $(3, 3)$ for the smallest five and seven hundred eigenvalues ω .

It is difficult to estimate the scaling of $\Delta N(\omega)$ for $l = \infty$ since the value of d increases for each new mode l and does not stabilize for such low l 's. However, we will make an attempt. If we compare each (l_{max}, l_{max}) mode do we see that d increases by a decreasing amount. For instance, from $(3,3)$ to $(4,4)$ increases d by 0.038, while from $(4,4)$ to $(5,5)$ does it increase by 0.003. This may be because the flattening-effect, as described above, affects the higher generations less. These results suggest that the value of d for $l = \infty$ approaches a finite value.

By assuming that d stops growing after a while, will we use the trial function

$$d(l) = \frac{a}{b + e^{-l}}, \quad (6)$$

to fit our data. This is a model for logistic growth. Here, a and b are parameters meant to fit our datapoints from the table 3. Using the function `scipy.optimize.curve_fit` to fit the data, we find a curve that might be a good approximation. Figure 7 shows both the datapoints and the fitted curve. Extrapolating the curve to $l = \infty$ we find a value of $d = 1.568$. From the graph can we see that

there is an abrupt stop in the growth of d around $l = 6$. It is the exponential decay in the trial function that is the cause of this. The obtained estimate relies on our chosen trial function. It is difficult to predict how d will increase for higher l , but the results suggest that a value close to 1.55 might be true since $l = 4$ and 5 almost has the same value for d and one might even say that it has converged to a number. However, it would be instructive to gather more data and find values of d for e.g. $l = 6$ and 7.

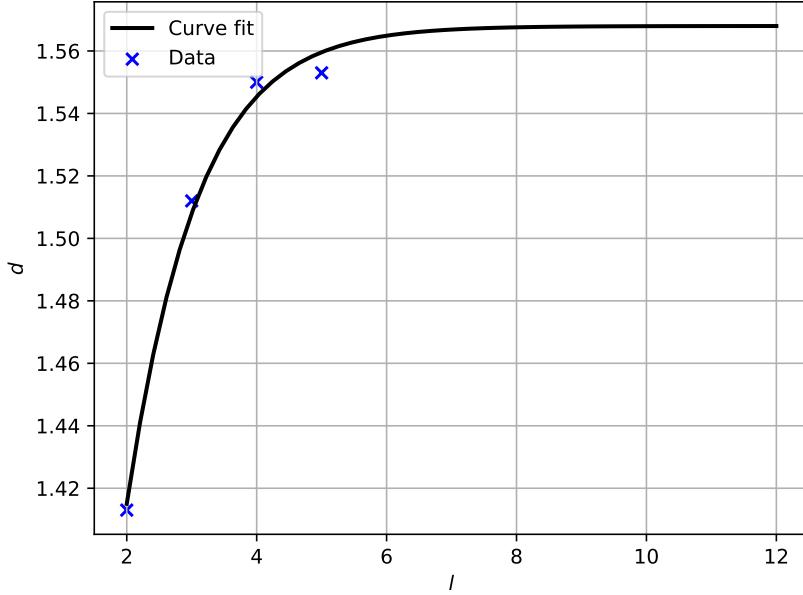


Figure 7: Plot of the data points from table 3, with $l = l_{max}$, and the fitted curve using the trial function shown in equation (6).

Higher order difference scheme

The higher order (HO) difference scheme used to approximate the laplacian operator is

$$U''(x) = \frac{-U(x + 2\Delta x) + 16U(x + \Delta x) - 30U(x) + 16U(x - \Delta x) - U(x - 2\Delta x)}{12(\Delta x)^2} + \mathcal{O}(\Delta x^4). \quad (7)$$

Since this stencil have points outside the fractal close to the boundary do we need to use another stencil for those points. Close to the boundary do we use

$$U''(x) = \frac{11U(x - \Delta x) - 20U(x) + 6U(x + \Delta x) + 4U(x + 2\Delta x) - U(x + 3\Delta x)}{12(\Delta x)^2} + \mathcal{O}(\Delta x^3), \quad (8)$$

where $x - \Delta x$ is a boundary point. It is the opposite case if $x + \Delta x$ is the boundary point. We are using these schemes both along the x - and y -direction.

The Helmholtz equation is solved for a Koch-fractal with $l = 3$ on a grid adjusted for a fractal with $l_{max} = 4$. Figures of the ten first eigenmodes are shown in figures 8 and 9 and the corresponding eigenfrequencies are shown in table 4. We see that the plots are very similar, except for some places where red and blue have switched colours. This is not a problem since it is the same mode, just anti-symmetric from each other. Comparing the eigenfrequencies we see a good agreement between what was found for the lower order scheme. Here all the eigenfrequencies have decreased by a very small amount. The highest deviation is of order 0.05%, which is a small value. These results may suggest that a second order difference scheme gives high enough precision for this particular problem.

Table 4: Smallest eigenvalues ω/v for a fractal with $l=3$, using HO, in units m^{-1} .

Mode	1	2	3	4	5	6	7	8	9	10
Value	9.329	13.910	13.911	14.171	14.256	14.916	14.917	17.441	18.657	19.192

Biharmonic equation

The biharmonic equation in 2D is of the form

$$\lambda W(\mathbf{x}, \omega) = \nabla^4 W(\mathbf{x}, \omega) = \frac{\partial^4 W}{\partial x^4} + \frac{\partial^4 W}{\partial y^4} + 2 \frac{\partial^4 W}{\partial y^2 \partial x^2}. \quad (9)$$

To solve the biharmonic equation numerically do we use the difference scheme for the laplacian twice. Adopting the notation $W(x_i, y_j) = W_{i,j}$, and denoting the step size as h , the central difference scheme takes the form

$$(\nabla^4 W(x))_{i,j} \approx \frac{1}{h^4} (W_{i+2,j} - 8W_{i+1,j} + 10W_{i,j} - 8W_{i-1,j} + W_{i-2,j} + W_{i,j+2} - 8W_{i,j+1} + 10W_{i,j} - 8W_{i,j-1} + W_{i,j-2} + 2(W_{i+1,j+1} + W_{i+1,j-1} + W_{i-1,j+1} + W_{i-1,j-1})). \quad (10)$$

For the points close to the boundaries will we get a difference scheme where one point is outside the boundary. To solve this do we use the Neumann condition $\partial_n W(\mathbf{x}) = 0$. For instance, if we look along the x -axis and the point $W_{i+2,j}$ is outside the boundary do we get

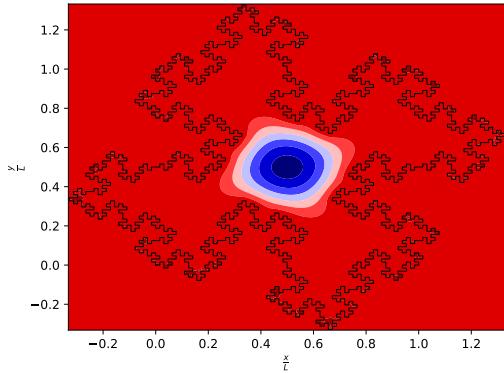
$$\frac{W_{i+2,j} - W_{i,j}}{2h} = 0 \implies W_{i+2,j} = W_{i,j}. \quad (11)$$

Combining this equation with the stencil given above, the first row in equation (10), do we find the scheme along the x -axis when the point is next to the boundary

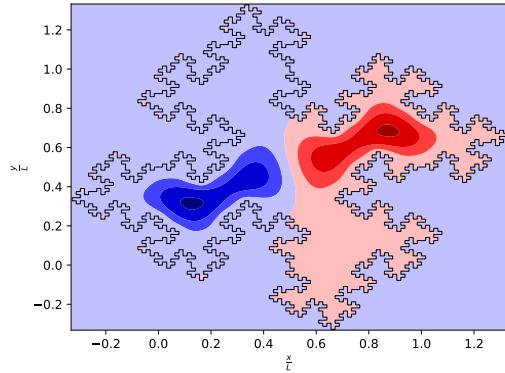
$$(\nabla^4 W(x))_{i,j} \approx \frac{11W_{i,j} - 8W_{i-1,j} + W_{i-2,j}}{h^4} + \text{other terms}, \quad (12)$$

where we have employed the Dirichlet condition, namely that $W_{i+1,j} = 0$ in this case. In order for this method to work do we need atleast three gridpoints between each boundary on the fractal. This is achieved for values of the fractal and grid $(l, l_{max})=(3,4)$, as we did for the Helmholtz equation, both in the regular and higher order difference scheme.

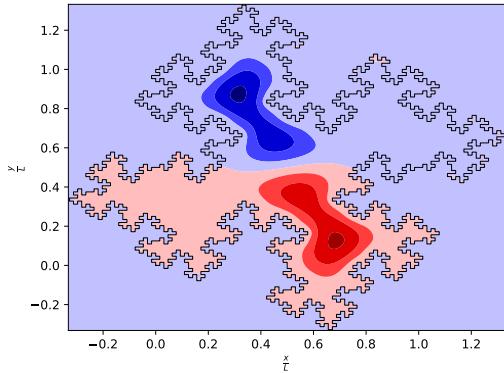
In table 5 can we see the ten smallest eigenvalues to the biharmonic equation, using $(l, l_{max})=(3,4)$. Here have we computed the fourth root of the eigenvalue, to compare with what was obtained for the



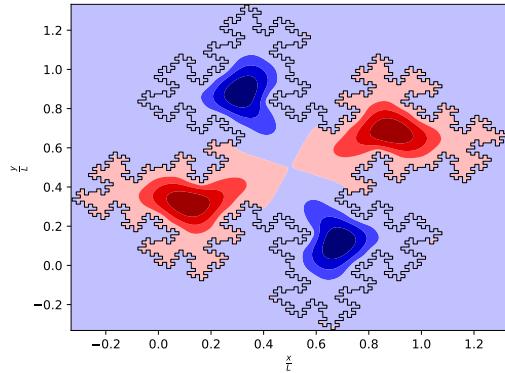
(a) First excited state



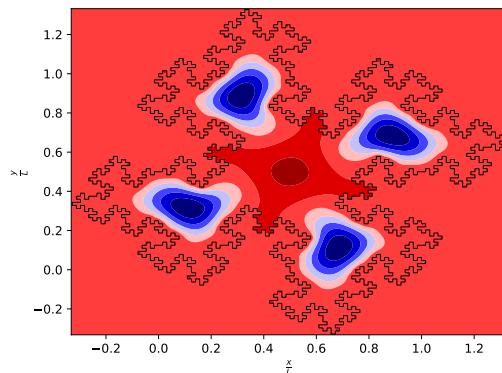
(b) Second excited state



(c) Third excited state

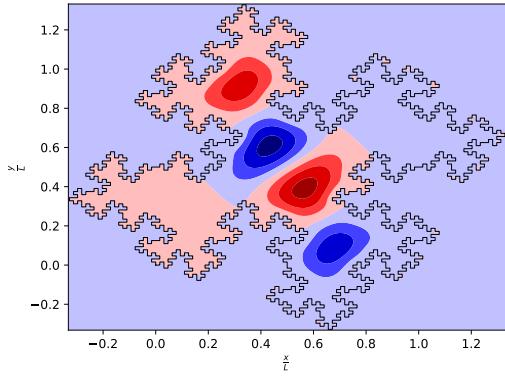


(d) Fourth excited state

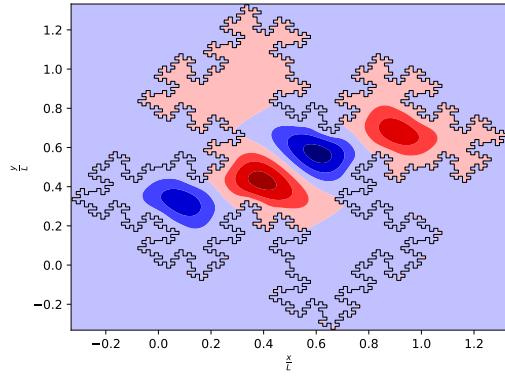


(e) Fifth excited state

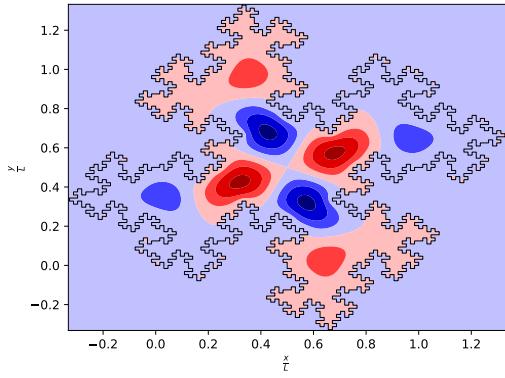
Figure 8: Contour plots of the first to fifth eigenmodes for the Koch-fractal with $(l, l_{max})=(3,4)$, using a higher order difference scheme.



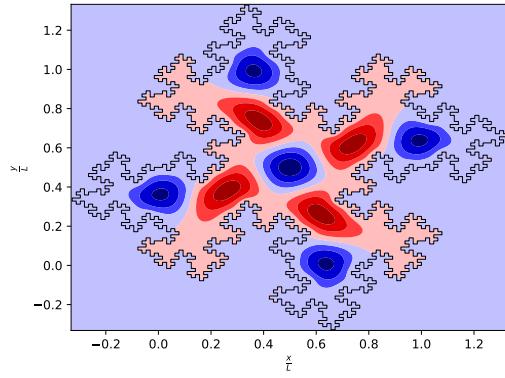
(a) Sixth excited state



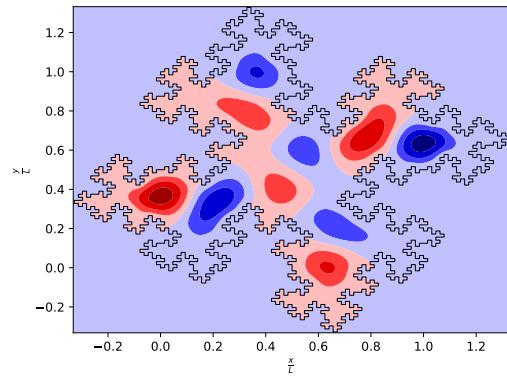
(b) Seventh excited state



(c) Eighth excited state



(d) Ninth excited state



(e) Tenth excited state

Figure 9: Contour plots of the sixth to tenth eigenmodes for the Koch-fractal with $(l, l_{max})=(3,4)$, using a higher order difference scheme.

Helmholtz equation. The eigenvalues are much higher in this case. The corresponding eigenmodes are shown in figures 10 and 11.

The eigenmodes exhibit also in this case some degree of degeneracy. Looking at the eigenvalues we can see that the second and third state are the same state, and perhaps the same yields the fourth, fifth, sixth and seventh state though their eigenvalues differ a bit. This could be due to numerical errors.

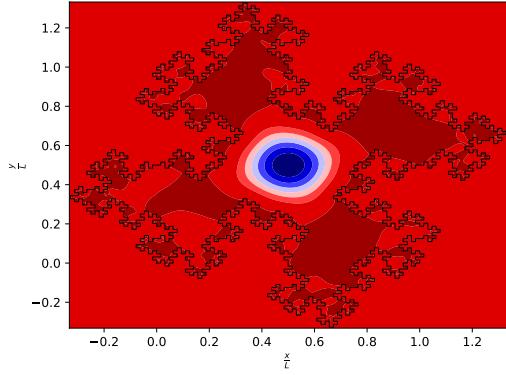
Now, comparing the eigenmodes to the Helmholtz case we see that the eigenmodes are in general different, except for the ground state which is almost identical. However, we can see some similarities between the structure of each eigenmode. For instance can we see similarities between the second, third, eighth and ninth state. This is expected, since in reality do we know that vibrations on a membrane are not completely different from vibrations on a clamped thin plate, which is described with the biharmonic equation.

Table 5: Smallest eigenvalues $\lambda^{\frac{1}{4}}$ for a fractal with $l=3$, for the biharmonic equation, in units m^{-1} .

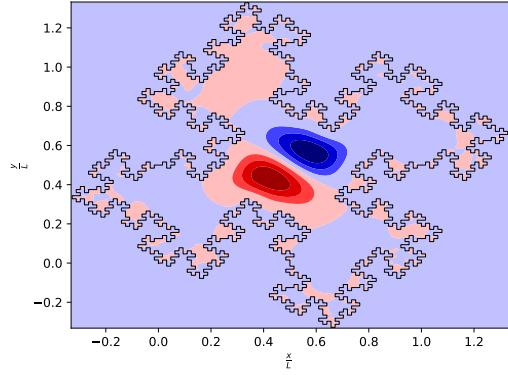
Mode	1	2	3	4	5	6	7	8	9	10
Value	13.108	18.484	18.511	20.569	20.668	20.929	20.961	22.060	24.134	24.689

References

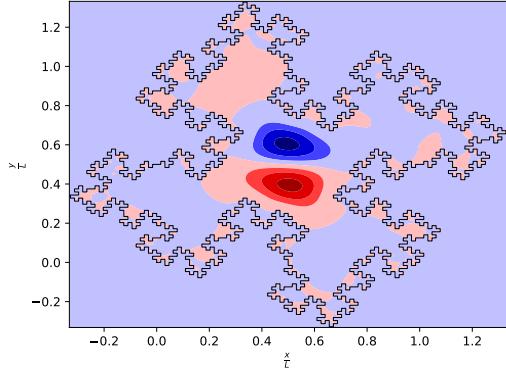
- [1] <https://www.linkedin.com/pulse/short-formula-check-given-point-lies-inside-outside-polygon-ziemecki/>
- [2] Ingve Simonsen, Department of Physics, NTNU. *TFY4235 Computational Physics Assignment 1: Vibrations of Fractal Drums.*
http://web.phys.ntnu.no/~ingves/Teaching/TFY4235/Assignments/TFY4235_Assignment_01.pdf



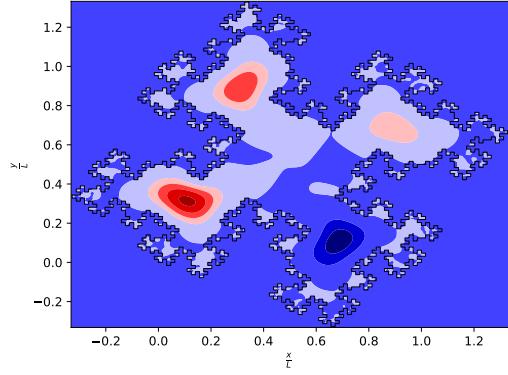
(a) First excited state



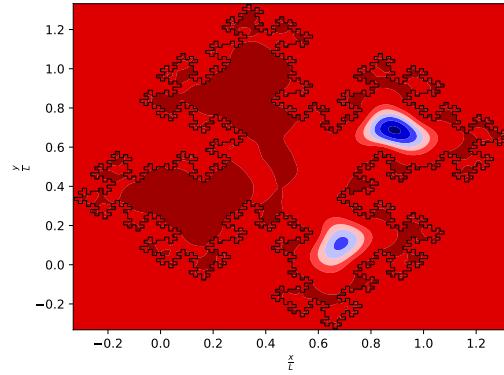
(b) Second excited state



(c) Third excited state



(d) Fourth excited state



(e) Fifth excited state

Figure 10: Contour plots of the first to fifth eigenmodes for the Koch-fractal with $(l, l_{max})=(3,4)$, for the Biharmonic equation.

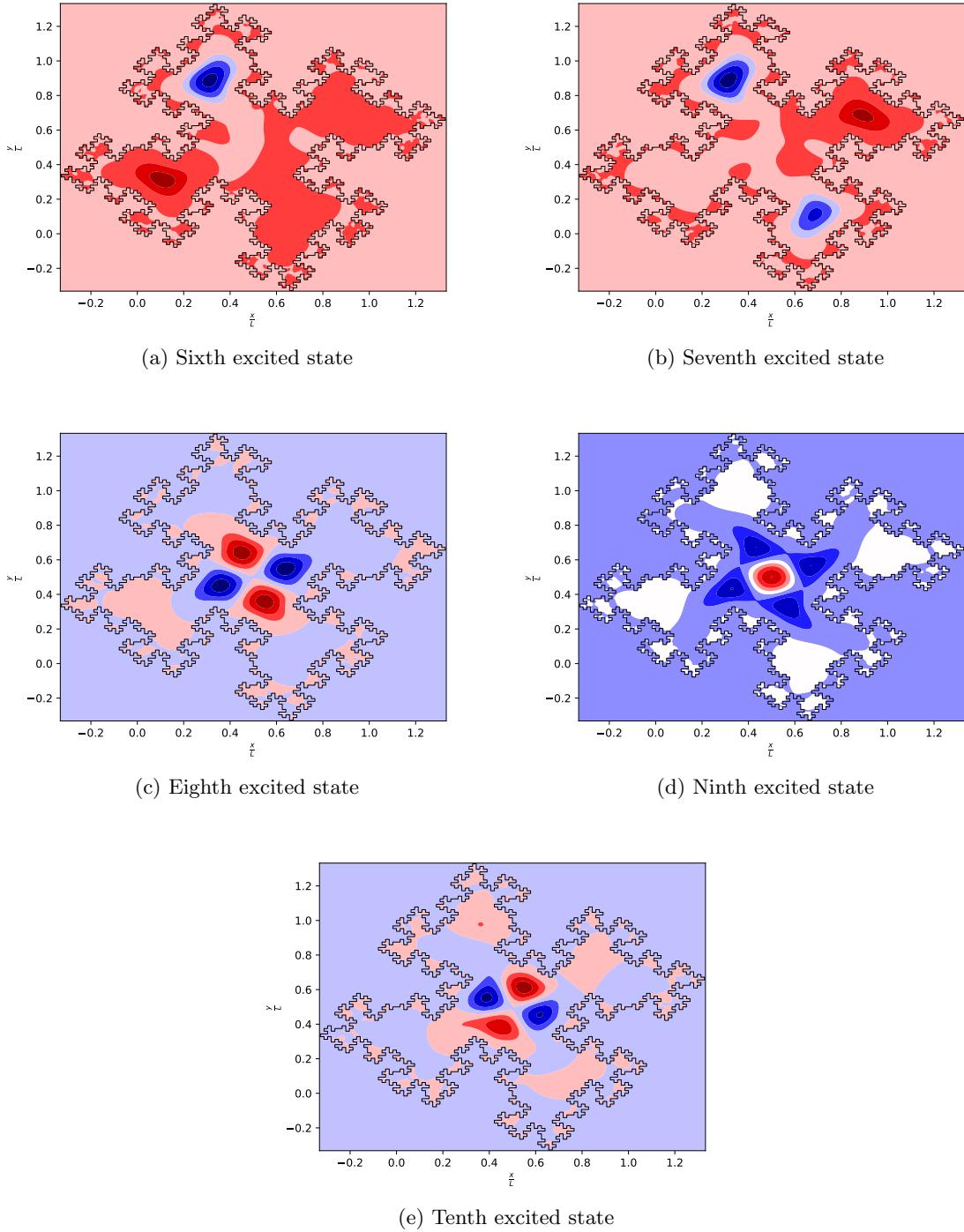


Figure 11: Contour plots of the sixth to tenth eigenmodes for the Koch-fractal with $(l, l_{max})=(3,4)$, for the Biharmonic equation.