Scientific Computation Project 2

Leon Wu 01190736

March 27, 2020

Part 1

1.1)

- The three main operations for updating A and B were vectorized. The Rsum (for A for example) can be reduced down to a dot product, vectorizing across p and nlist[n]. The case where the index = n can be subtracted at the end to allow the operation to be vectorized.
- Similar vectorization steps were done for Bfac and Asum, indexing using a list mlist[m] instead of looping through these indices. B was updated in a similar way.
- From plots of the recovered data with a grid search of the parameters p, λ and niter, a higher value for niter usually gives a better image. (There is randomness since the algorithm takes random permutations of the points each iteration).
- Also higher values of p for the values that I tested seemed to give better results. The optimal value for λ seems to lie around $\lambda = 3$.
- However, increasing p and niter both increase the computation time. The values I used in figure 1 showing the repaired data were $\lambda = 2$, niter=100, p=10.
- To determine the best parameters for the repair, a bigger and finer grid search is required. There is likely to also be correlations between the parameters (eg dependance of λ on p) which must be carefully investigated.

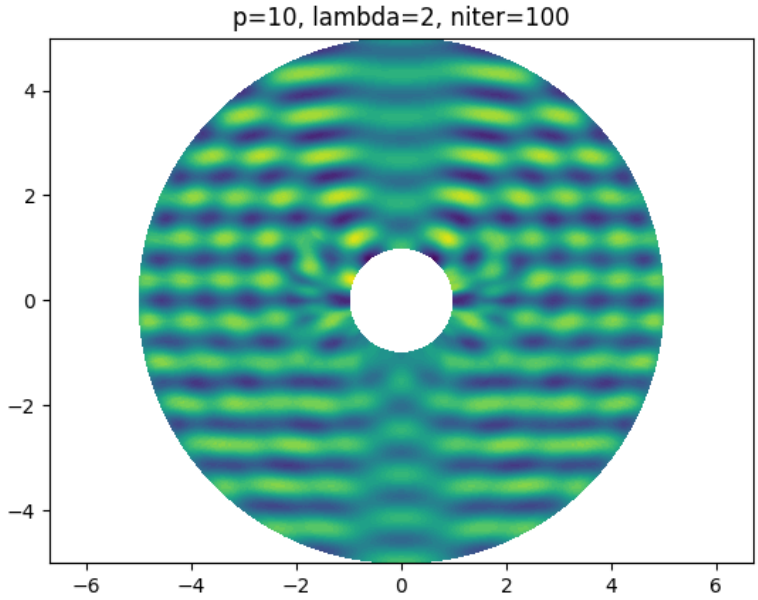

Figure 1: Repaired data

1.2i))

- The equation is identified as the wave equation in polar coordinates. The equation can be solved by computing Hankel functions of the first kind evaluated at $r\omega$ and $r_0\omega$. The solution is truncated at 100 terms - including more terms will increase the accuracy of the solution, but past 100 terms there is not much change in the solution. My method takes the initial condition given in the data and computes the relevant Hankel functions.

1.2ii)

- Figures 2, 3 and 4 show the contour plots of the data at the relevant θ values. There are correlations both in the t and r direction that must be considered. Also, the contour plot corresponding to $\theta = 3\pi/3$ shows more constant heights over time for a fixed r, ie the waves are not oscillating much compared to the other 2 plots.

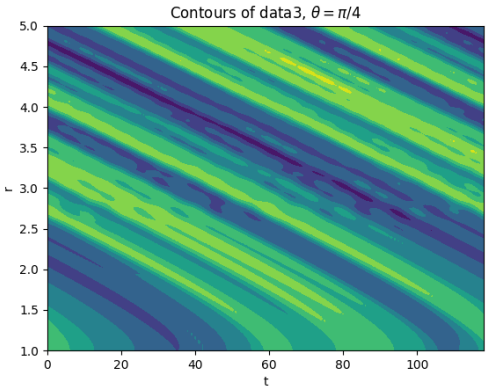

Figure 2:

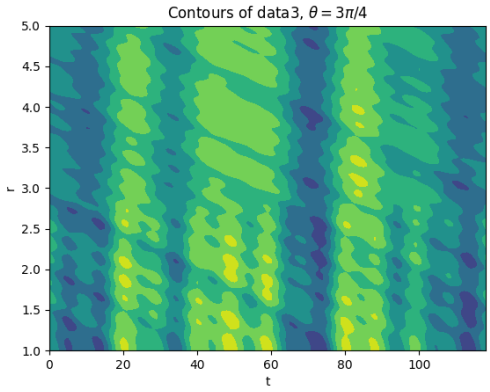

Figure 3:

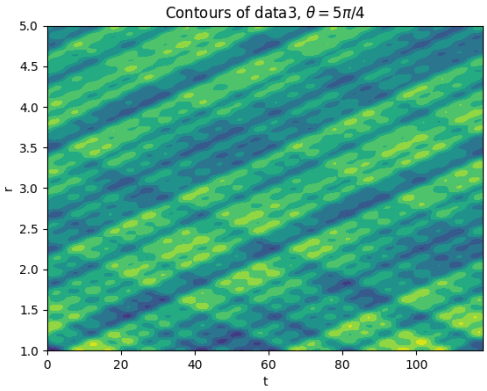

Figure 4:

- Figure 5 shows the heights at one particular value or r for the 3 θ values. The heights are oscillatory, with $\theta = \pi/4$ having lower amplitude waves. Also, it looks as if the series at $3\theta = \pi/4$ and $5\theta = \pi/4$ are negatively correlated, suggesting that one is at the bottom of a wave and the other the top of a closeby wave.

- It is worth noting the waves oscillate around 0, with the mean of the heights at each r point across time equal to approximately 0.

- Figure 6 shows an autocorrelation plot of the same heights. Notably for $\theta = \pi/4$, the shape of the plot shows that for nearby times, the heights of the waves are highly correlated to eachother. As you get to about 45 timesteps away, the heights become very negatively

correlated with eachother, meaning that at points 50 timesteps away the heights are expected to be very different from eachother. This correlation increases to nearly 1 again at about 80 timesteps away.

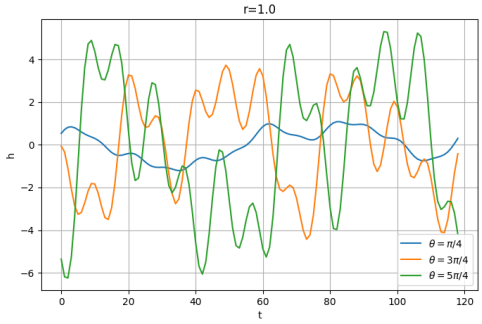

Figure 5:

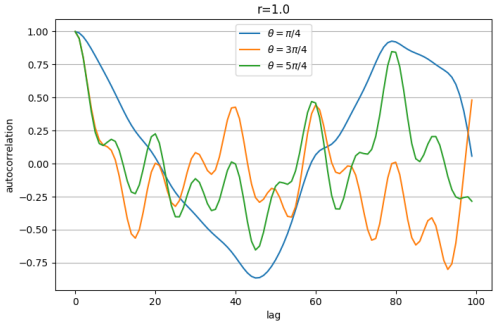

Figure 6:

- Computing the power spectrum of these timeseries using Welch's method, figure 7 shows that the timeseries corresponding to $\theta = \pi/4$ has one high-power oscillation at a lower frequency, as seen on the previous plots with the height oscillating slowly. The other two timeseries have other peaks at higher frequencies, meaning there are some faster oscillations. There are both made up of three main peaks.
- For other fixed values of r we get different dynamics. For example figure 8 shows at $r = 5$ the waves are very similar to eachother.

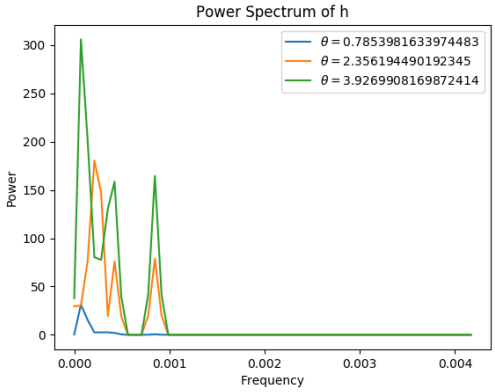

Figure 7:

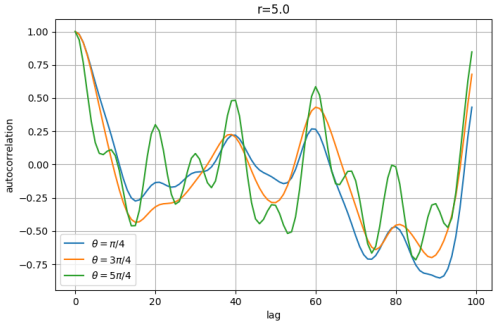

Figure 8:

- Taking the variance of the heights at each point in time in figure 9, we see that $\theta = \pi/4$ has small amplitude and low frequency oscillations, $\theta = 3\pi/4$ has medium amplitude and high frequency oscillations, and $\theta = 5\pi/4$ has high amplitude and high frequency oscillations.
- Figure 10 shows an autocorrelation plot for a timeseries in r for a fixed t. It shows that as you move further away in radius at a fixed time, the autocorrelation decreases, since the waves interfere with eachother so that at far away points in space the waves don't have much relationship with eachother.
- Figure 11 also shows the varaince of the series for $\theta = \pi/4$ increasing as the radius increases, indicating that the heights are less oscillatory nearer to the centre.

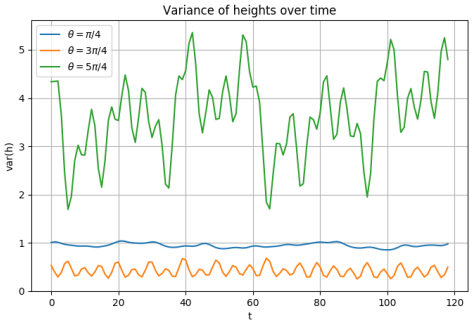

Figure 9:

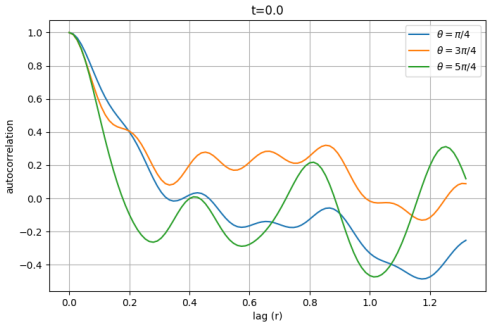

Figure 10:

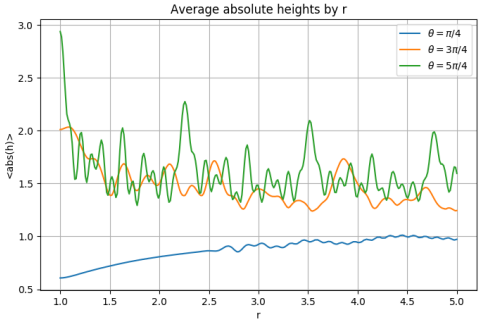

Figure 11:

1.3)

- The MxNxT input can be reduced in three steps. The first step is to slice along one axis of the 3darray and perform an SVD decomposition on each the resulting 2darray slices.
- Slicing along all three dimensions, figures 12, 13, and 14 show the cumulative sums of the singular values for all slices. The black line vertical refers to the point at which the singular values account for 99.9% of the maximum value of the cumulative sum. When the data is sliced in the first dimension, this happens at $p = 12$.
- The arrays U, S and Vh from the SVD of each slice are first truncated to the relevant $p = 12$ portions needed to reconstruct a rank-p approximation of the input. Then, they are stacked on eachother, resulting in 3 arrays of size MxNxp, MxpxT, Mxp. The information stored in these arrays can be said to store 99.9% of the 'information' of the original data (according to the singular values). It could be that the square of the singular values could describe the amount of 'information' retained better, (since these values link to the amount of variance explained), but I will stick with the singular values here.
- The first two arrays (size MxNxp, MxpxT) can be reduced further using SVD in the same way, but slicing in the p dimension. Figures 15, 16 show again the number of singular values needed to encode 99.9% of the information (q=69 and r=12 respectively).
- The output arrays are now of size pxMxq, pxqxN, pxq, pxMxr, pxrxT, pxr, Mxp. The number of elements of the input is 10317300, and for the output it is 552600, meaning that the size of the reduced data is 5.356% of the original.
- The amount of information kept at each SVD loop is about 99.9%, so multiplying 99.9% twice (since the last two SVDs are separate from eachother) gives a final accuracy of around 99.8%.
- To reconstruct this, the process is simply reversed. p, q and r are inferred from the sizes of the input arrays. To compute the matrices from their SVD, the relevant arrays are simply multiplied together as in the definition of the SVD. The decompositions U_m, S_m, Vh_m of each slice m of H are computed in a loop of size p (S is already known). Then each slice m of H is computed using its relevant decomposition.
- Figures 17 and 18 show one time slice of the original data and my reconstructed data after reduction, respectively. By eye it is extremely hard to tell any differences, which is consistent with the 99.8% accuracy quoted. Depending on the use for this data, the accuracy / amount of storage saved can easily be adjusted by changing the values of p, q, r. Increasing these values will improve the quality but increase the memory used, and decreasing will do the opposite.

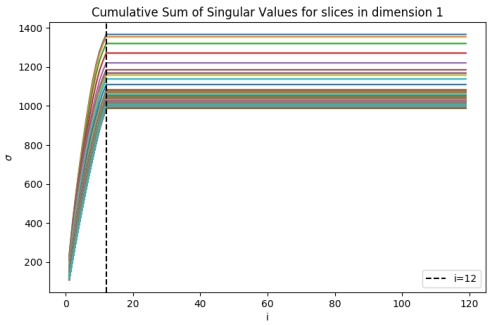

Figure 12:

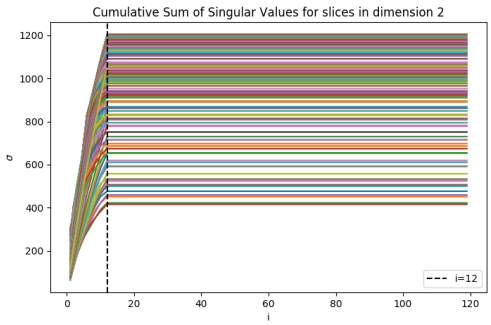

Figure 13:

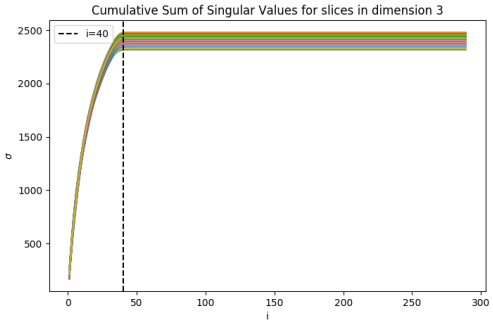

Figure 14:

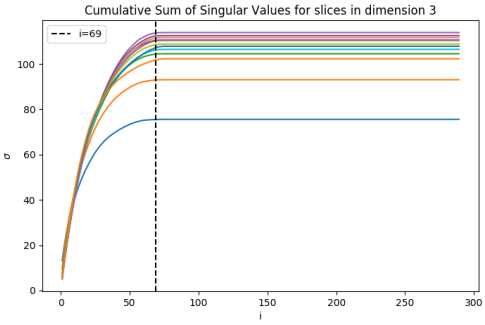

Figure 15:

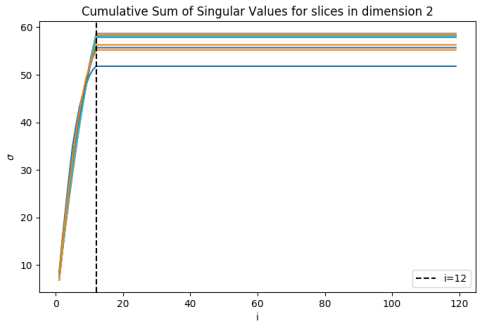

Figure 16:

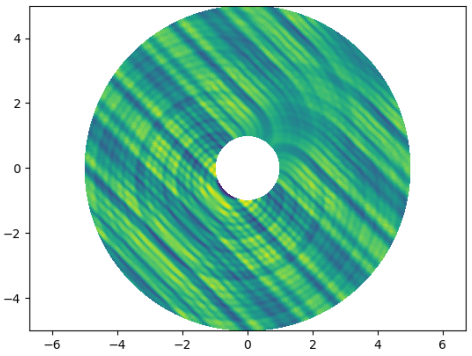

Figure 17:

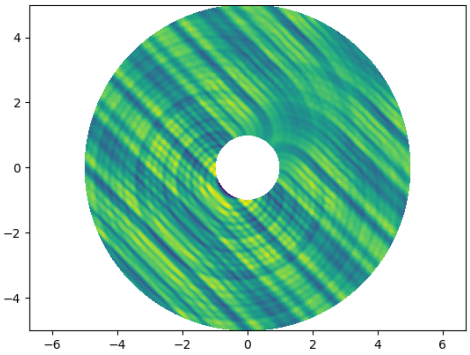

Figure 18:

Part 2

2.1)

- The cost of newdiff is $O(N)$, where N=number of grid points, since the construction of A and b when solving $Ax = b$ is $O(N)$. Solving the system is $O(N)$ since a banded solver is used. For the case of the tridiagonal system, one such banded solver is the Thomas algorithm, which is $O(N)$. Figure 19 demonstrates this, with the loglog plot showing a linear fit with coefficient 0.96, showing that it is roughly linear in time.
- The timing ratios in figure 20 show the compact scheme is roughly 7 times slower than the old scheme for a range of Nx values.
- Figure 21 shows the average absolute error for increasing Nx. The error of the old scheme reduces with h^2, whereas the new scheme reduce with roughly h^6.
- The error of the new difference formula is compared to the 2nd order centered formula for a fixed Nx. Figure 22 shows the error of the 2 schemes for a high frequency sin wave oscillation. The error is much lower for the compact scheme. This is expected, especially for high wavenumbers as in this equation.
- It is clear from the timings (from the fact they are both linear) and the accuracy plots that using the new scheme gives better timings for a fixed accuracy. Therefore in most cases, the new scheme should be used over the old scheme.

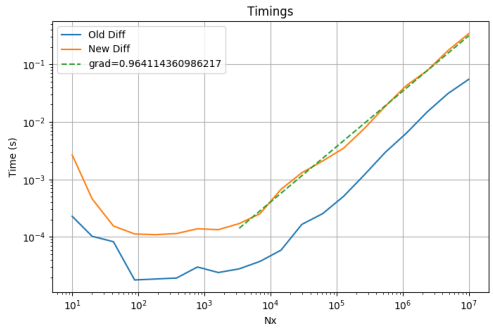

Figure 19:

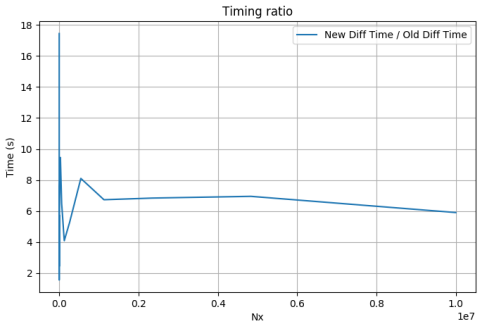

Figure 20:

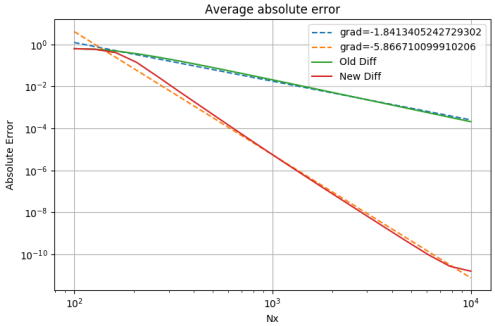

Figure 21:

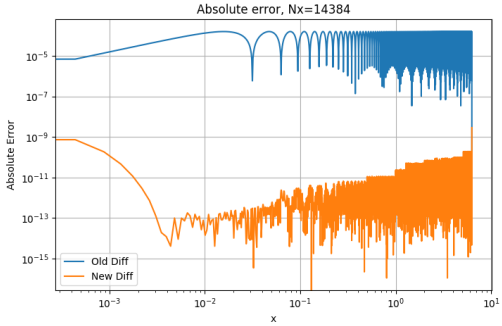

Figure 22:

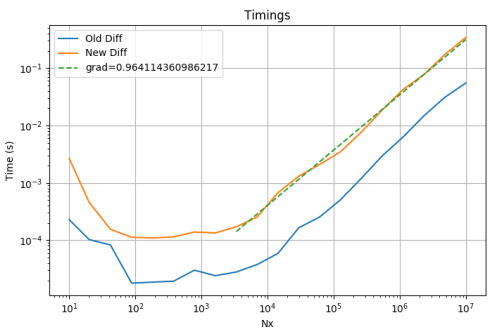

Figure 23:

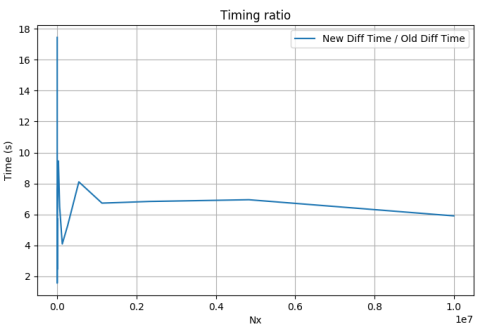

Figure 24:

2.2)

- Figures 25, 26 and 27 show contour plots for the solutions at $\kappa = 1.5, 1.7, 2$ respectively. Walking forward in time there seems to be no obvious repeating patterns of the solution; the solution looks aperiodic. Figure 29 plots the solution at three points in space over time, which again shows aperiodic behaviour as the amplitudes oscillate without a clear pattern. The purple wave is at the boundary which is why it is periodic (because of the boundary conditions).
- Figure 28 shows the mean of the points over x. There is a general increase in the solution as you move towards the centre of the domain in space.
- Figure 30 shows the mean of the points over time. Again, this looks quite aperiodic as there is not obvious pattern.
- Figures 31 shows a plot of $x(t)$ against $x(t-\tau)$ for mean of the points in space (This plot is similar for all κ in the relevant range). This again suggests chaotic behaviour since the points don't seem to repeat. The relevant correlation sums are plotted in figures 32, 33 and 34. The value of τ was set to 1/5 of the dominant time period, found by applying Welch's method to the time series to find the power spectrum. The number of lagged points to compare to was chosen as 3. The fractal dimension is estimated at just above 2 for all the values of κ. All in all this suggests (weakly) chaotic behaviour of the solution in time.
- (g can be seen to have similar dynamics in general compared to f.)
- For data3, figures 35 and 36 show fractal dimensions at two points in space. These vary around 2. However from the previous analysis of this data, we know it has some periodicity in time and space. Therefore this system is not chaotic, but not far from chaotic since the fractal dimension is high and the system is only wekaly periodic.

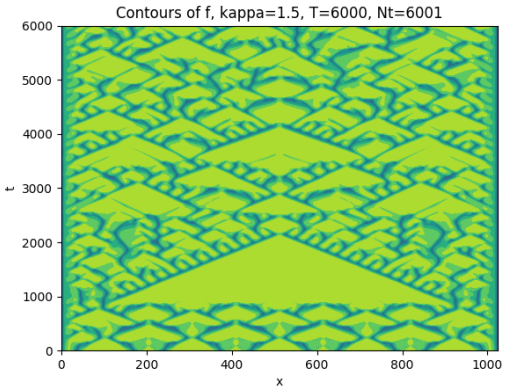

Figure 25:

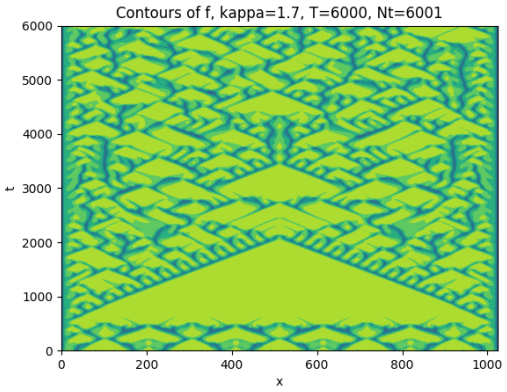

Figure 26:

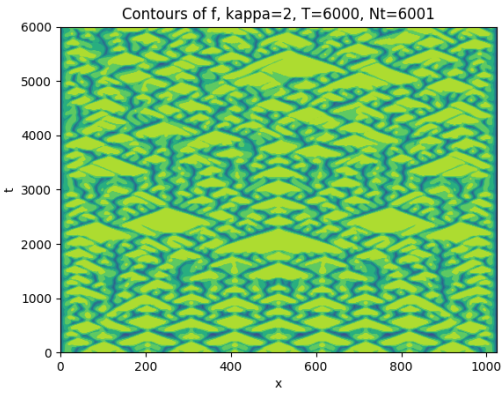

Figure 27:

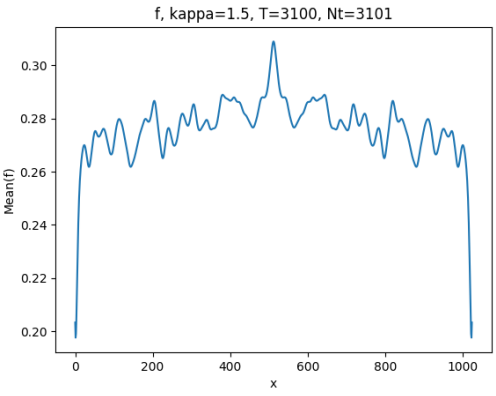

Figure 28:

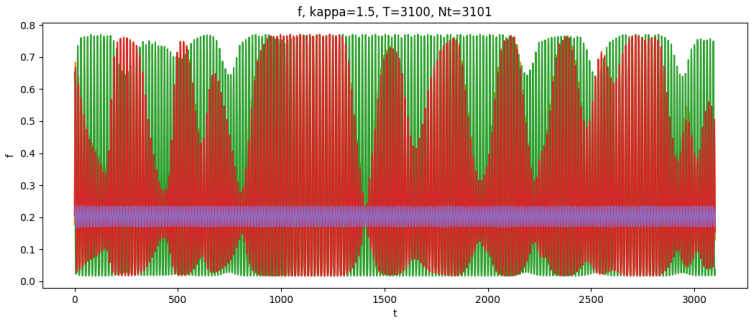

Figure 29:

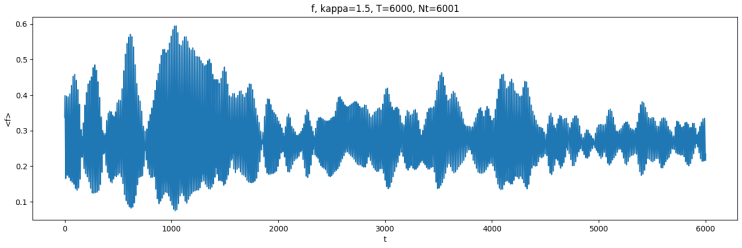

Figure 30:

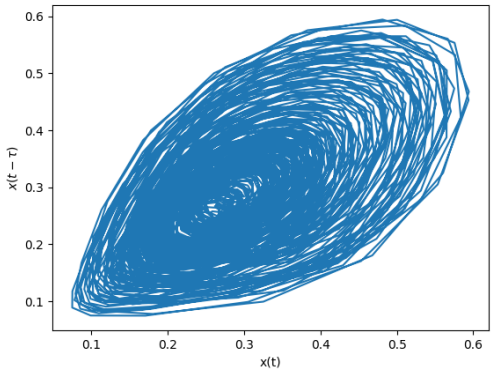

Figure 31:

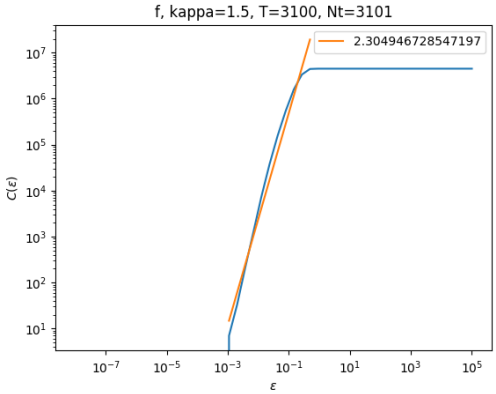

Figure 32:

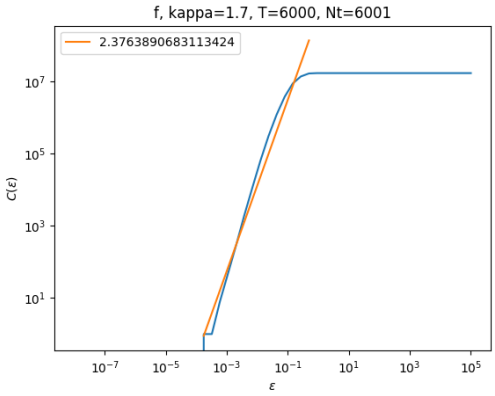

Figure 33:

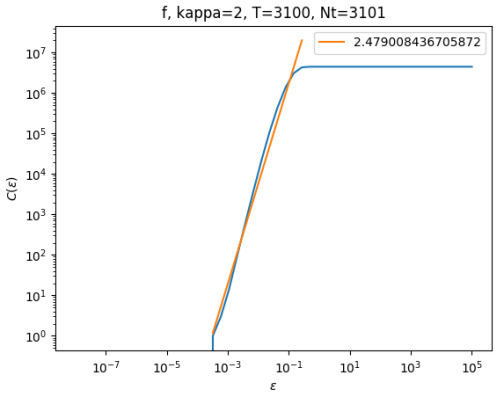

Figure 34:

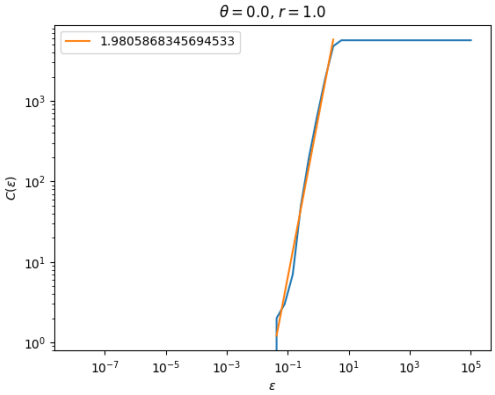

Figure 35:

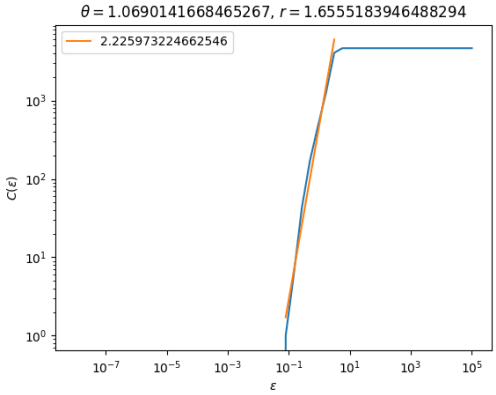

Figure 36:

2.3)

- The sensitivity to the pertubations can be analyszed using Lyapunov exponents. Taking the steady state as the initial condition, step forward in time to get solutions for a large period of time.
- Then, set an amplitude for spatially sinusoidal pertubations to this steady state. Apply the pertubation to the this state, and run the solution for the same time frame.
- Compare the solutions at each time step by taking the euclidean distance between solutions. Then, use a least-squares fit to estimate the Lyapunov exponent (*2).
- Repeat this process for different amplitudes of the pertubation. The Lyapunov exponent can then be plotted against the amplitudes to see the dependance. This quantifies the sensitivity of the steady state to these pertubations. Then, for example, we could see for what amplitudes the steady state returns back to its original state.
