Scientific Computation Project 2

Solution

March 24, 2025

Part 1

1(a)

The model equations are:

$$\begin{aligned}
\frac{dx_i}{dt} &= x_i\left(1 - x_i - \gamma\left(x_{i-2} + x_{i+1}\right)\right), \quad i = 2, 3 \ldots, n-2,\\
\frac{dx_0}{dt} &= x_0\left(1 - x_0 - \gamma\left(x_{n-2} + x_1\right) + \mu x_{n-3}\right),\\
\frac{dx_1}{dt} &= x_1\left(1 - x_1 - \gamma\left(x_{n-1} + x_2\right)\right),\\
\frac{dx_{n-1}}{dt} &= x_{n-1}\left(1 - x_{n-1} - \gamma\left(x_{n-3} + x_0\right)\right).
\end{aligned}$$

To find a non-trivial equilibrium solution, set the LHS to zero, so the first equation becomes:

$$\bar{x}_i\left(1 - \bar{x}_i - \gamma\left(\bar{x}_{i-2} + \bar{x}_{i+1}\right)\right) = 0.$$

We can remove the leading x_i and similarly remove the leading x_0 and x_{n-1} from the other equations which gives us a linear system, $\mathbf{A}\bar{\mathbf{x}} = \mathbf{1}$ that can be solved using `np.linalg.solve`. Then, to find the required maximum energy ratio, we solve the maximum growth problem for infinitesimal perturbations to the equilibrium solution. Let $\mathbf{D}$ be a diagonal matrix with $D_{ii} = \bar{x}_i$. Then, the perturbations must satisfy the following linear equations, $\frac{d\tilde{\mathbf{x}}}{dt} = -\mathbf{D}\mathbf{A}\tilde{\mathbf{x}}$. The maximum perturbation energy ratio at time t is given by σ_1^2 where σ_1 is the largest singular value of $\exp\left(-\mathbf{D}\mathbf{A}t\right)$. Let $\mathbf{M} = -\mathbf{D}\mathbf{A}$, and let $\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{W}^T$ be the SVD of $\mathbf{M}$. Then the first column of $\mathbf{W}$ is the initial condition that produces maximum growth. [Students were asked to give a 2-3 sentence description, so stating that they solved a linear system for $\bar{\mathbf{x}}$ and the maximum growth problem for the energy ratio and most dangerous initial condition would be sufficient].

1(b)

We can compute numerical solutions to the full problem using the initial condition ($\tilde{\mathbf{x}}_0$) and equilibrium solution computed as described above and . Specifically, we set the initial condition to be $\mathbf{x}_0 = \bar{\mathbf{x}} + \epsilon\tilde{\mathbf{x}}_0$ where ϵ is varied between 1e-2 and 1e-6. Then `simulate1` is used to compute solutions at $t = 50$ which are then used to compute the actual energy ratio. Figure 1(a) shows the relative difference between the perturbation energy ratios computed using the `simulate1` and `part1q1a`. We see that as ϵ decreases, the error decreases as well reaching a reasonably low level at $\epsilon = 10^{-6}$.

1(c)

When $\mu = 0$, $\mathbf{M}$ is a circulant matrix, and circulant matrices have orthogonal eigenvectors. As a result the solution to the maximum growth problem can be computed with the leading eigenvalue of $\mathbf{M}$. Here, *leading eigenvalue* means the eigenvalue with the most positive real part. let *ell* be this largest real part. Then, the maximum energy ratio at time t will be $\exp\left(2\ell t\right)$. Figure 1(b) compares the $\exp\left(2\ell t\right)$ with *eratio* computed using `part1q1a`. We see very good agreement, and the exponential increase in the maximum energy ratio can clearly be seen as well.

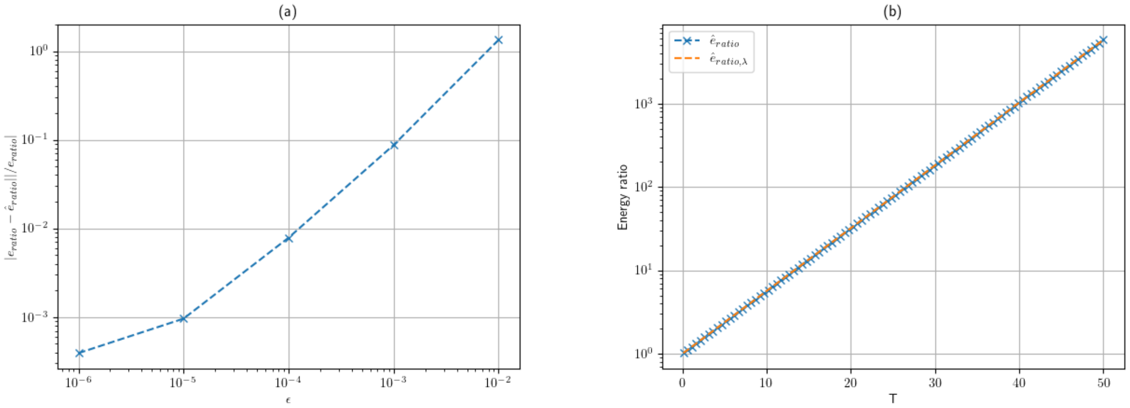

Figure 1: (a) Relative difference between maximum energy ratio computed using `simulate1` and `part1q1a`; (b) variation of max energy ratio with time for $\gamma = 1.2$, $\mu = 0$.

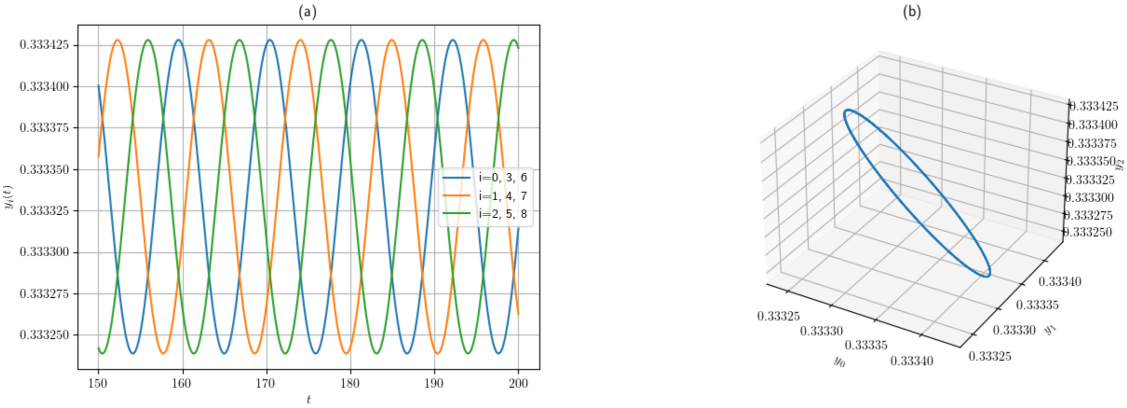

Figure 2: Results for case A. (a) time series; (b) phase plot.

2

First consider case A where $n = 9$. Here, the system display small-amplitude sinusoidal oscillations in time with period of oscillation ≈ 10.88 (figure 2). I have measured the period of oscillation simply by extracting the times between peaks and averaging, but Welch's method is an alternate approach that could be used. There is a simple relation between the components: $y_i(t) = y_{i+3}(t) = y_{i+6}(t)$, $i = 0, 1, 2$. The small magnitude of the oscillations suggests a connection to the linearized dynamics from question 1. Here, the eigenvalue of $\mathbf{M}$ (defined above) with largest real value is purely imaginary, $\lambda = 0.577i$. This corresponds to a period of $2\pi/0.577 \approx 10.89$ which is very close to what we observe in the simulation. The corresponding eigenvector follows the same pattern we observe in the simulation: $y_i = y_{i+3} = y_{i+6}$ for $i = 0, 1, 2$.

Case B ($n = 20$) is considerably more complicated. After the initial transient, there are large-amplitude oscillations between 0 and 1 with a frequency of 0.03 (estimated using Welch's method) until about $t = 1300$ when the oscillatory behavior becomes more complicated with multiple frequencies clearly visible (figure 3(a)). The variation across components also becomes more complicated. Careful examination of time series shows that for $t < 1300$, $y_i = y_{i+10}$ for $i = 0, 1, 2, ...$, however this coupling goes away at later times. Figure 3(b) shows $y_i(t)$ at $t = 8000$. We see there is a tendency for rapid changes from peaks at $y_i = 1$ to troughs at $y_{i+1} = 0$ with a sometimes more gradual increase up to the next peak. We can examine this behavior more carefully by plotting y_{i+1} vs. y_i for a range of times. Figure 3(c) shows this return map for times before the transition around $t = 1300$, and we see a loop that indicates: 1) periodic behavior, 2) when y_i is close to 1, y_{i+1} will be close to 0 (peak to trough); 3) when y_i is near 0, y_{i+1} can take on seemingly any value (trough towards peak); 4) for intermediate values of y_i, y_{i+1} will be either 0 or another intermediate value. For $t > 1300$, the phase plot does

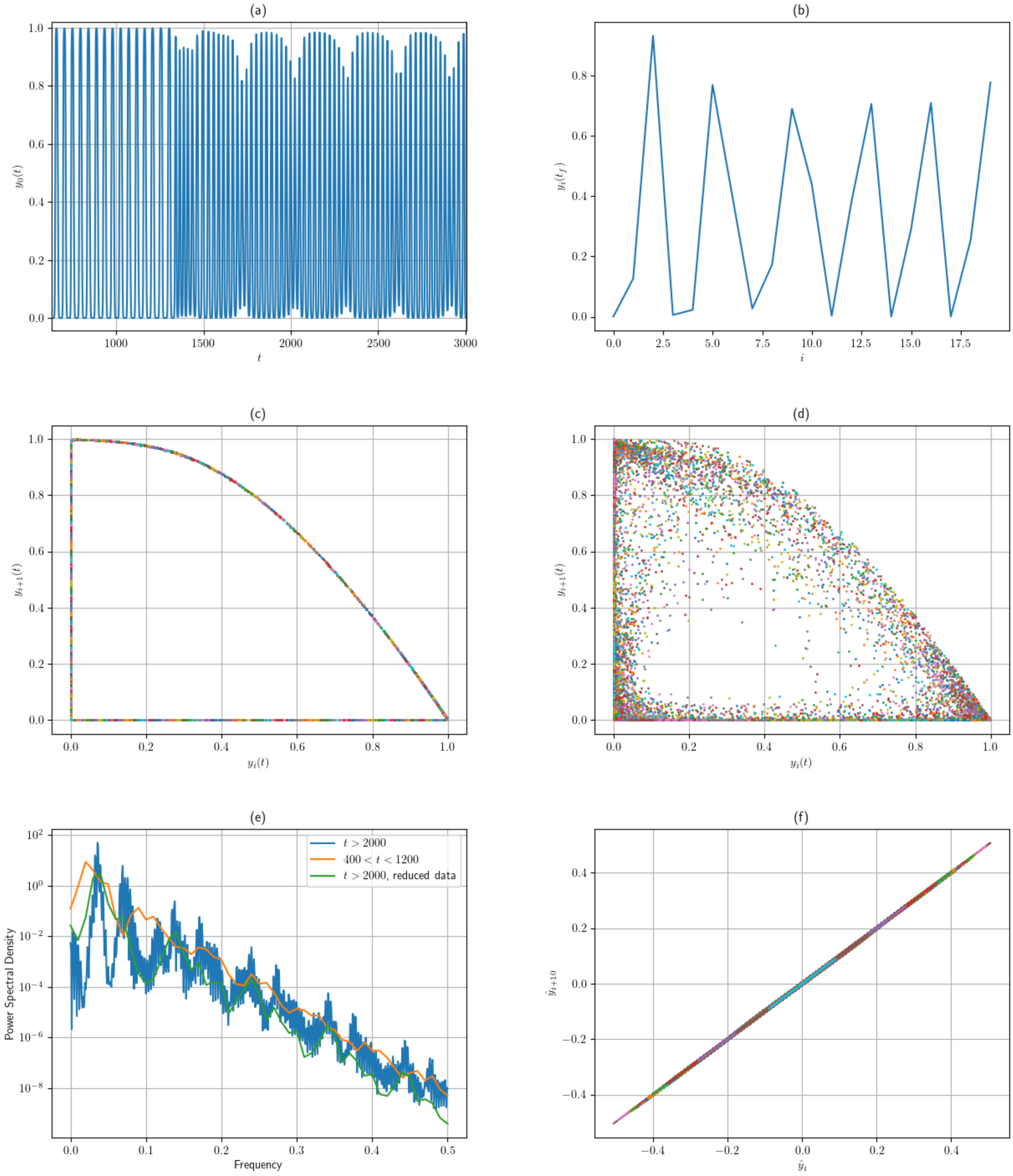

Figure 3: Results for case B. (a) time series for $i = 0$; (b) solution at $t = 8000$; (c) phase plot, y_{i+1} vs. y_i for $500 < t < 1200$ and $i = 0, 1, ..., 18$; (d) phase plot for $t > 1200$; (e) power spectral densities, $i = 0$; (f) Phase plot for reduced data, y_i vs. y_{i+10}, $i = 0, 1, ..., 9$

not fall on a simple loop, but there are strong similarities to the previous plot, and the behavior near $y_i = 0$ and $y_i = 1$ in particular remains similar to what we see in figure 3(c). Power spectral densities for $i = 0$ are shown in figure 3(e) (the PSDs for other i are very similar). The two curves for $t < 1200$ and $t > 2000$ are broadly similar – the result for $t < 1200$ looks somewhat like a smoothed version of the curve for larger times. The higher peaks at higher frequencies for the $t > 2000$ curve correspond to the complex temporal behavior in figure 3(a) observed for $t > 1300$. Applying PCA to the results for $t > 2000$ shows two dominant “modes” each accounting for about 30% of the variance. Carrying out dimensionality reduction with the first principal component and applying Welch’s method gives the “reduced data” curve in figure 3(e). This curve is qualitatively similar to the $t < 1200$ curve and suggests that similar dynamics can be found in both cases. Figure 3(f) shows y_i vs. y_{i+10} for the reduced data,

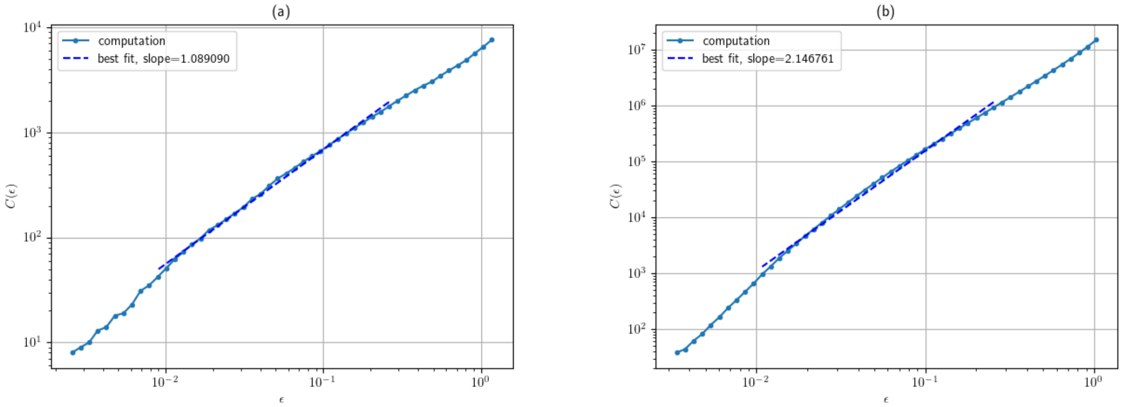

Figure 4: Correlation sum for case B. (a) $500 < t < 1200$, (b) $t > 1200$.

and the straight line shows that when we reduce data with the 1st principal component, the $y_i = y_{i+10}$ behavior for $t < 1300$ is recovered and is actually "hiding" in the more complex dynamics at later times. The dynamics for $t < 1300$ are clearly periodic and not chaotic, and this is is confirmed with the estimate of the correlation dimension ($d \approx 1$, figure 4(a)). However, for $t > 1300$, the correlation dimension is estimated to be slightly above two indicating weakly chaotic dynamics like we saw for the Lorenz system.

Moving to case C ($n = -20$), the time series for $y_0(t)$ shown in figure 4(a) shows oscillatory aperiodic behavior indicative of chaos. Figure 4(b) shows $y_i(t = 8000)$, and as in case B, there are sharp changes between peaks and troughs, but there also seems to be a more-gradual variation as well. The PSD for each component is shown in figure 4(c). There is a primary peak at a frequency of about 0.046, and a secondary peak at twice this frequency (about 0.092). These two peaks are what cause the "beating" behavior seen in figure 4(a)(plot $\sin(t) - 0.5\cos(2t)$ to see what I mean). A return map is shown in figure 4(d). While the boundaries of the map follow figure 3(c) quite closely, in general the map is much more complicated than what we see in case B. Despite this complexity, we can draw a useful general conclusion about how the components are related to each other. Extracting the component with the largest value at each time, and then plotting this component's index vs. time gives figure 4(e). As time increases, there is a tendency of the "peak index" to move from small i towards larger values until it reaches $i = 59$ and over a period of time jumps back and forth between the minimum and maximum indices before eventually starting another period of increase. We can think of this plot showing the location of a peak of a wave which tends to move from small i to large i and then jump back to small i. This wavelike behavior can be seen in the contour plot generated by the simulation code (plot not included here). Finally, this case is chaotic. The correlation dimension is estimated to be 3.69 which is larger than 2.

The correlation dimension was computed by randomly choosing six indices and then using time delays to shift the times used for the different indices to reduce correlation. Due to the randomness, the estimated dimension can vary from one run to the next. For case B, running 10 iterations, I had $d = 2.015 \pm 0.166$, while for case C, $d = 3.656 \pm 0.078$.

Part 2

1.

The code is below. The method is efficient as it replaces `np.linalg.solve` with `solve_banded` and all loops have been replaced with vectorized code.

2.

Figure 6 shows results for the accuracy and cost of the differentiation methods (implicit FD and 2nd-order FD. The test function is $f(x) = \exp[-\alpha(x-0.5)^2]$, and the top-left plot shows results for f_x. The error for the implicit FD method is different for points near to and far way from the boundary. Near the midpoint, the error decreases as $h \sim N^{-6}$ reaching grid-independence around $N = 1000$. The accuracy of the FD methods can also be compared via Fourier analysis. For the 2nd-order method, we have $(kh) \approx \sin(kh)$

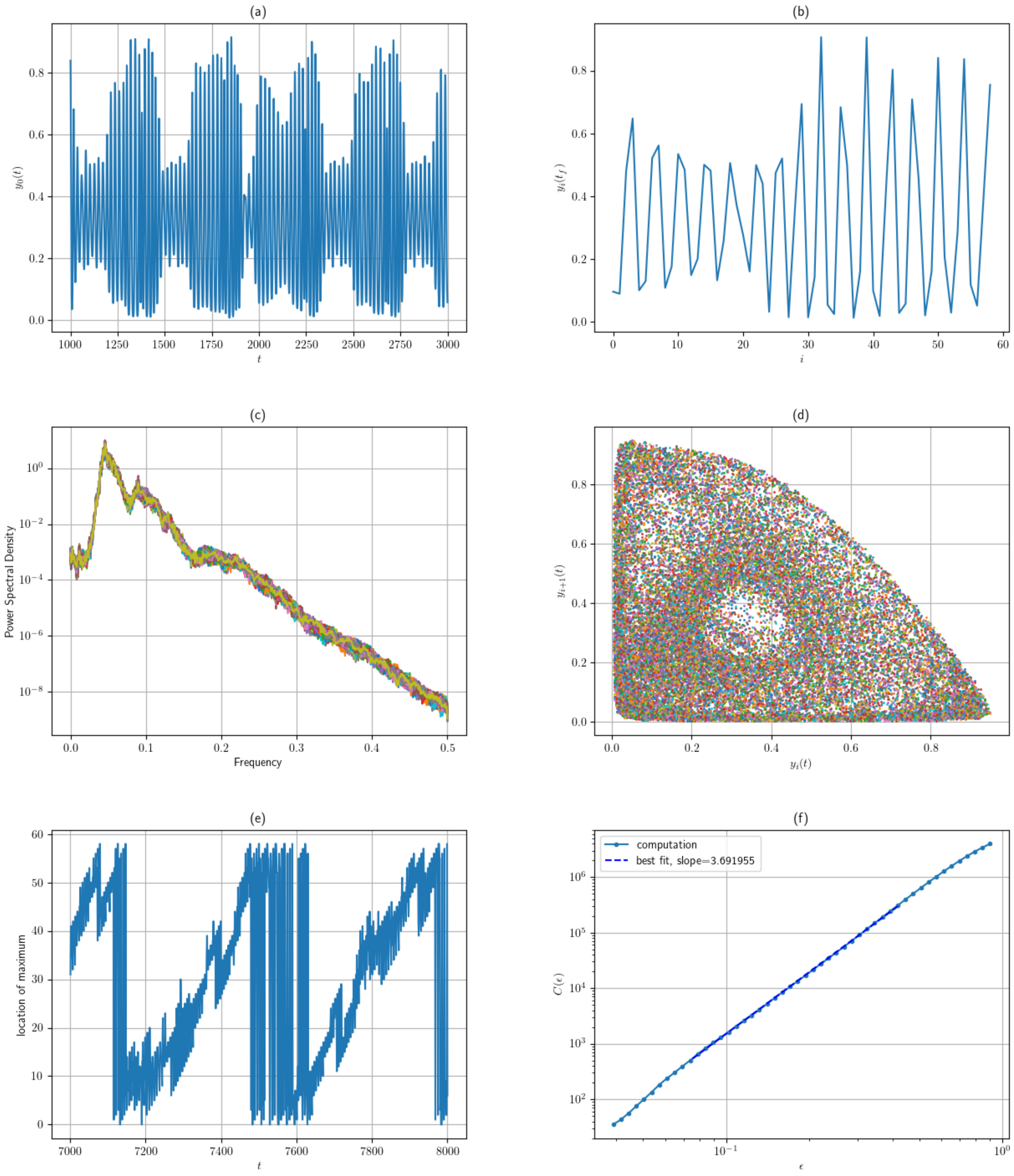

Figure 5: Results for case C. (a) time series for $i = 0$; (b) Solution at $t = 8000$; (c) power spectral densities for all i; (d) Return map, y_{i+1} vs. y_i; (e) index with peak value vs. time; (f) correlation sum.

while for the implicit method, after quite a lot of arithmetic, we have $(kh) \approx \frac{9/16+11/128\cos(kh)}{3/4+5/8\cos(kh)+1/32\cos(2kh)}$. These approximations are shown in figure 6(b), and as expected, the implicit scheme is much more accurate for high wavenumbers suggesting it is well-suited for multiscale problems. Note that the modified wavenumber can be numerically estimated simply by numerically differentiating $\sin(kx)$ with k or h varying. While the 2nd-order scheme is less accurate than the other methods, it is less expensive. The walltime for each method is shown in figure 6(c) with N varying. The behavior for small N is non-monotonic (probably due to memory effects), but we see that the 2nd-order method is about 3 times faster than the implicit method for $N \lesssim 100$. For larger N, the FD methods both follow quadratic trends (a least-squares fit for the implicit method is shown in the figure). From lecture, we know that the cost of the 2nd-order method is $O(N)$, and here the method is applied M times with $M = N$ so the total cost is $O(N^2)$. Similarly, from lecture we know the cost of solve_banded is $O(N)$ and using the same reasoning,

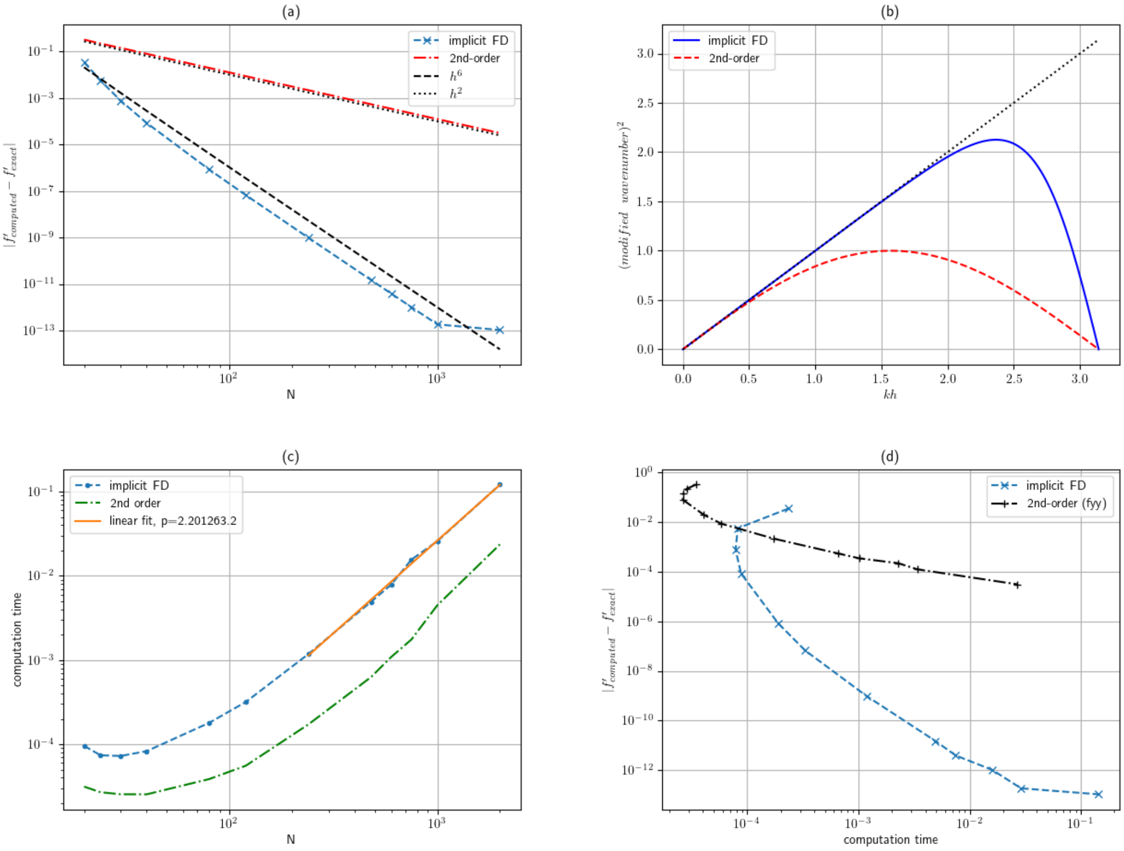

Figure 6: Results for part 2 question 2; (a) error for f_{xx}, (b) Fourier analysis results, (c) wall time, (d) Error vs. wall time

the total cost here will also be $O(N^2)$. However the big-O notation suppresses the leading coefficients in these costs. For the 2nd-order method, the cost of the first derivative is $2N + const$ and the cost of the 2nd derivative is $4N + const$. The cost of solving a $k-$ banded system is $\approx 2kN + const$ and the cost of constructing the RHS is roughly $6N$ giving a total cost of about $34N + const$. The theoretical cost ratio should then be about $34/6 \approx 5.6$. For $N > 100$, the walltime ratios range between 5.3 and 8.4 showing fair agreement with my rough estimate.

Figure 6(d) compares the efficiency of the two methods by examining the variation of the average error with the walltime. The question to consider is, which method is faster for a given error? For errors larger than ≈ 0.01, the 2nd-order method is faster. For smaller errors, it is clear that the implicit method is superior.