Scientific Computation II HW3

Michael Nameika

March 2023

Chapter 12 Problems

2. Write a MATLAB program to implement (6.8) and (6.9) and construct the differentiation matrix D_N associated with an arbitrary set of distinct points $x_0, \ldots, x_N$. Combine it with **gauss** to create a function that computes the matrix D_N associated with Legendre points in $(-1, 1)$. Print results for $N = 1, 2, 3, 4$.

Implementing (6.8) and (6.9) in MATLAB (see attached diffMatTest.m file), we find the following for $N = 1, 2, 3, 4$:

```
D =

     0
```

```
D =

   -0.8660    0.8660
   -0.8660    0.8660
```

```
D =

   -1.9365    2.5820   -0.6455
   -0.6455   -0.0000    0.6455
    0.6455   -2.5820    1.9365
```

```
D =

   -3.3320    4.8602   -2.1088    0.5806
   -0.7576   -0.3844    1.4707   -0.3287
    0.3287   -1.4707    0.3844    0.7576
   -0.5806    2.1088   -4.8602    3.3320
```

7. Use the FFT in N points to calculate the first twenty Taylor series coefficients of $f(z) = \log(1 + \frac{1}{2}z)$. What is the asymptotic convergence factor as $N \to \infty$? Can you explain this number?

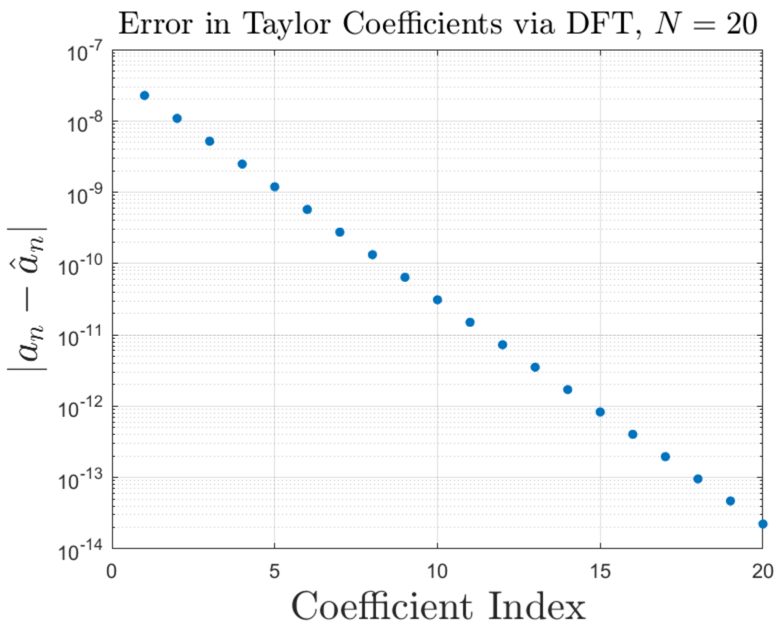
Notice for a function $f(z)$, we may consider Taylor expanding on the unit circle such that

$$f(e^{i\theta_n}) = \sum_{k=0}^{N-1} a_k e^{i\theta_k}$$

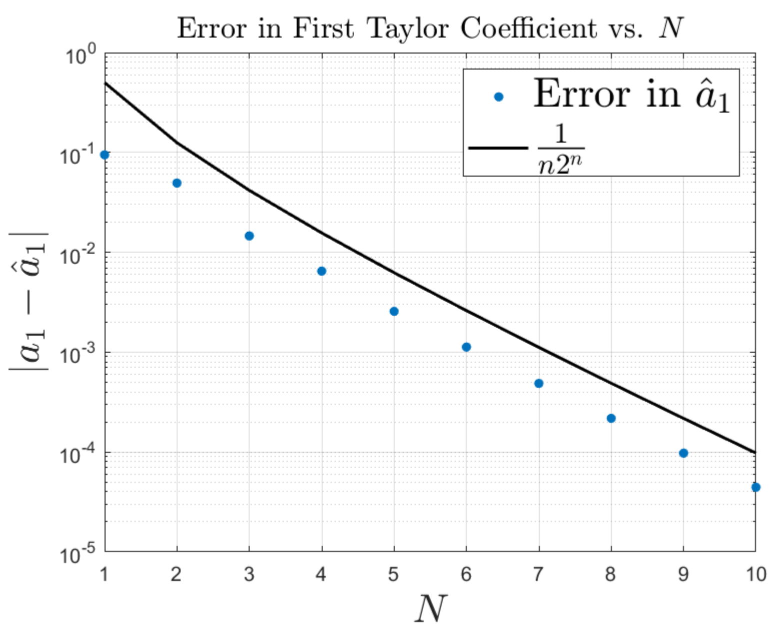
Notice that this is the discrete Fourier transform of the Taylor coefficients. Then by using the inverse DFT, we have

$$a_n = \frac{1}{N} \sum_{k=0}^{N-1} f(\theta_k) e^{-i\theta_k n}$$

Implementing this into MATLAB (see attached fourierTaylorTest.m script), we find the following plot of the error in the first 20 coefficients:



We also find the error in the first coefficient a_1 for $1 \leq N \leq 10$:



Notice that it appears the asymptotic convergence factor is roughly $\frac{1}{n2^n}$. Notice that this is the same convergence factor as the Taylor coefficients for $\log(1 + \frac{1}{2}z)$. That is, the Taylor coefficients computed via the DFT converges to the true value as $N \to \infty$ at the same rate as the Taylor coefficients converge to 0.

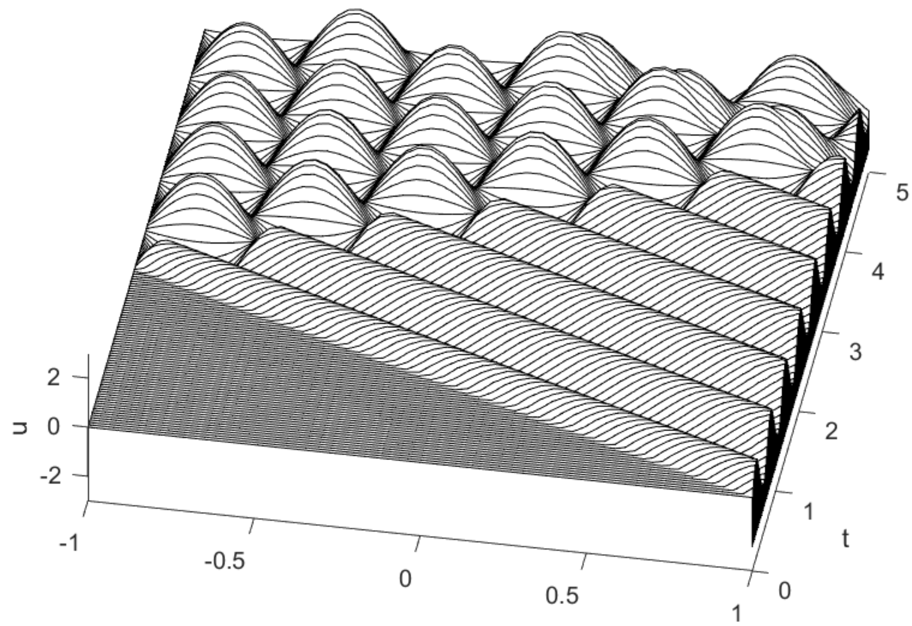
Chapter 13 Problems

3. Modify Program 19 (p. 82) to solve $u_{tt} = u_{xx}$ as before but with initial and boundary conditions

$$u(x, 0) = 0, \quad u_x(-1, t) = 0, \quad u(1, t) = \sin(10t)$$

Produce an attractive plot of the solution for $0 \leq t \leq 5$.

Implementing this problem in MATLAB (see attached myp19.m file), we find the following solution plot for $0 \leq t \leq 5$:



4. The time step in Program 37 is specified by $\Delta t = 5/(N_x + N_y^2)$. Study this discretization theoretically and, if you like, numerically, and decide: is this the right choice? Can you derive a more precise stability limit on Δt?

For the stability region of the leapfrog method, we have that (along with the fourier discretization), $\Delta t < 4/N_x$. From the Chebyshev differentiation, we have that $\Delta t < \frac{9}{N_y^2}$. That is, we require that

$$\Delta t < \min\left\{\frac{9}{N_y^2}, \frac{4}{N_x}\right\}$$

Since $\frac{4}{N_y^2+N_x} < \min\left\{\frac{9}{N_y^2}, \frac{4}{N_x}\right\}$, a more precise stability limit would be

$$\Delta t < \frac{4}{N_y^2 + N_x}$$

I think

Chapter 14 Problems

1. Determine the first five eigenvalues of the problem

$$u_{xxxx} + u_{xxx} = \lambda u_{xx}, \quad u(\pm 2) = u_x(\pm 2) = 0, \quad -2 < x < 2,$$

and plot the corresponding eigenvectors.

To begin, we must transform this problem from $-2 \leq 2$ to $-1 \leq x \leq 1$ by the transformation $t = x/2$, we have that the nth order differentiation matrices must be multiplied by $1/2^n$ since $d/dtu(t) = 1/2d/dxu(x)$. Implementing this into MATLAB, we find the following figure of the first five eigenvalues and eigenvectors: