Problem Set 8

18.330 Intro to Numerical Analysis (MIT, Spring 2019)

Due: May 2. To be submitted *online* on Stellar.

Problem 1. Fixing the Gibbs phenomenon. (25 points)

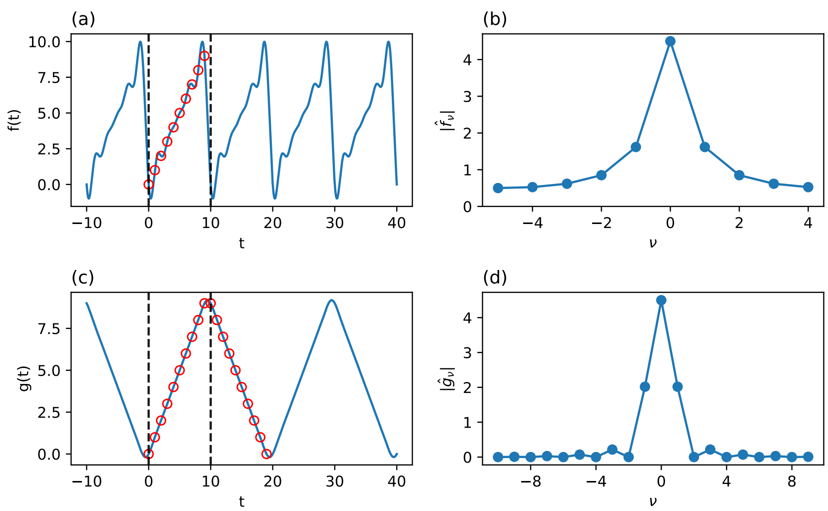

Figure 1: Gibbs phenomenon in the FFT for periodic and mirrored even periodic samples. (a) IDFT interpolating function for the samples (red circles) shows Gibbs phenomenon (rapid oscillations) near the discontinuity. (b) Absolute value of the DFT coefficients $\hat{f}_\nu$ is nonzero everywhere, reflecting the high frequency components responsible for the Gibbs phenomenon. (c) Mirrored samples (red circles). The IDFT interpolant shows no Gibbs phenomenon. (d) The absolute value of the DFT coefficents for the mirrored data. Most are very close to zero.

Recall from our discussion in class the Gibbs phenomenon: Taking the Fourier transform of a discontinuous function, or the DFT of discontinuous samples leads to rapid oscillations near the discontinuity (see Figure 1). At the same time, this leads to most of the DFT coefficients being nonzero, even for slowly varying functions. Here, you will examine one way of mitigating this issue by cleverly mirroring the function of interest such that discontinuities at the end of of the interval of periodicity are removed. This method is used, for instance, in the JPEG algorithm for image compression, and its continuous analogue will be used in the Clenshaw-Curtis quadrature rule.

(a) Define the mirrored sample vector g_n from N samples f_n using

$$g_n = f_n, \qquad n = 0, \ldots, N-1$$
$$g_{N+k} = f_{N-1-k}, \qquad k = 0, \ldots, N-1.$$

This samples an *even* periodic function using $2N$ samples. Express the discrete Fourier transform of g_n in terms of cosines only. How many independent coefficients $\hat{g}_\nu$ are there?

(b) The expression you found in (a) is one variant of the *Discrete Cosine Transform*. Use the DFT again to derive a formula for the inverse discrete cosine transform.

(c) Implement the DCT you found in part (a) in Julia and reproduce the plots from Figure 1 (b) and (d) using $f_n = n$ with $n = 0, \ldots, 9$.

(d) Comment on the usefulness of the DCT for lossy compression algorithms in spectral space.

Problem 2. Fourier Transforms in higher dimensions. (25 points)

Compute the n-D Fourier transform of the following functions

(a) $f(x,y,z) = e^{-\sigma_x x^2/2 - \sigma_y y^2/2 - \sigma_z z^2/2}$

(b) $f(x,y,z) = \frac{e^{-\kappa\|\mathbf{r}\|}}{\|\mathbf{r}\|}$, where $\|\mathbf{r}\| = \sqrt{x^2+y^2+z^2}$ and $\kappa > 0$

(c) $f(\mathbf{x}) = e^{-\mathbf{x}^\top A\mathbf{x}}$, where A is a $n \times n$ symmetric positive definite matrix.

Problem 3. Solving special linear systems with the FFT. (25 points)

An $n \times n$ circulant matrix C takes the form

$$C = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix},$$

which is fully defined by its first column $\mathbf{c} = (c_0, c_1, \dots, c_{n-1})^\top$, and the other columns correspond to cyclic permutations of the first one. Circulant matrices often occur in numerical discretizations of differential operators on periodic domains. For instance, a finite difference second derivative matrix with stepsize h is defined by $\mathbf{c} = (2, -1, 0, \dots, 0 - 1)/h^2$ (Write it down to see that it's true!).

(a) Show that the linear system of equations

$$C\mathbf{x} = \mathbf{b}, \tag{1}$$

where $\mathbf{b}$ is arbitrary, can be written as a convolution with the first column $\mathbf{c}$ of the circulant matrix C.

(b) Let $\hat{\mathbf{a}} = F\mathbf{a}$ be the discrete Fourier transform of $\mathbf{a}$ with $F_{\nu n} = e^{-2\pi i \nu n/N}/\sqrt{N}$. Using this DFT, derive a formula for the solution $\mathbf{x}$ of Eq. (1) that can be efficiently computed using the Fast Fourier Transform.

(c) Implement your algorithm in Julia and compare its runtime to the backslash operator (\) for some $n \in [10, 10^4]$. You may use the `fft` and `ifft` functions from the `FFTW` package. Use random $\mathbf{c}$ and $\mathbf{b}$.

(d) Try to solve a circulant system with $n = 10^7$ using the backslash operator and with your FFT algorithm. Which algorithm will still work?

Problem 4. More on circulant matrices. (25 points)

As you saw in the previous problem, linear systems involving circulant can be solved both extremely quickly and circulant matrices "act" like sparse matrices: all we need is their first column to completely specify them.

Here you will investigate some additional interesting properties of circulant matrices.

(a) The eigenvalues and eigenvectors of circulant matrices can be computed exactly. Show that if the $n \times n$ matrix C is circulant, then

$$\mathbf{v}_j = \begin{pmatrix} 1 \\ \zeta_j^1 \\ \zeta_j^2 \\ \vdots \\ \zeta_j^{n-1} \end{pmatrix}, \qquad \zeta_j = e^{2\pi i j/n}, \qquad j = 0, \dots, n-1,$$

are its eigenvectors. What are the corresponding eigenvalues?

(b) What is the relationship between the matrix $V = (\mathbf{v}_0 \mid \cdots \mid \mathbf{v}_{n-1})$ and the DFT matrix F from Problem 3 (b)?

(c) Express the eigenvalue formula you found in part (a) using a discrete Fourier transform. Come up with an algorithm that uses the FFT to efficiently compute the eigenvalues of circulant matrices.

(d) Implement your algorithm from part (c) in Julia and compare the result to the builtin `eigvals` function from the `LinearAlgebra` package (which often uses a variant of the QR algorithm). Then compare computational timings using $n \in [10, 10^3]$ and random $\mathbf{c}$.