Universität Tübingen
Mathematisches Institut
Dr. Yanyan Shi
Dr. Yoann Le Hénaff

Tübingen, April 29, 2025

2. Exercise Sheet for Algorithms in Numerical Mathematics

Exercise 8: Let f be continuous and 2π-periodic with absolutely summable Fourier coefficients $(\widehat{f}(n))_{n\in\mathbb{Z}}$. Their approximation by the midpoint rule is

$$\widetilde{f}_N(n) = \frac{1}{N}\sum_{j=0}^{N-1} f(t_j)e^{-int_j} \quad \text{with} \quad t_j = \frac{2j+1}{2}\cdot\frac{2\pi}{N}.$$

Show the aliasing formula

$$\widetilde{f}_N(n) = \sum_{l=-\infty}^{\infty} (-1)^l \widehat{f}(n+lN).$$

Exercise 9: Let

$$u_N(x) = \sum_{n=-N/2}^{N/2-1} \widehat{u}_N(n)e^{inx}, \qquad b_N(x) = \sum_{n=-N/2}^{N/2-1} \widehat{b}_N(n)e^{inx}$$

be trigonometric interpolation polynomials to 2π-periodic continuous functions $u(x)$ and $b(x)$. Then for $x_j = j\frac{2\pi}{N}$ it holds

$$\|u_N - b_N\|_{L^2}^2 = \frac{1}{N}\sum_{j=0}^{N-1} |u(x_j) - b(x_j)|^2.$$

Note: First apply the Parseval's equation from §4 and then the equation from §1.

Exercise 10: Given a circulant matrix

$$A = \begin{pmatrix} a_0 & a_1 & \cdots & a_{N-2} & a_{N-1} \\ a_{N-1} & a_0 & \ddots & \ddots & a_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_2 & \ddots & \ddots & a_0 & a_1 \\ a_1 & a_2 & \cdots & a_{N-1} & a_0 \end{pmatrix}$$

Show that the eigenvalues of A are the Fourier coefficients $\hat{a}_0, \dots, \hat{a}_{N-1}$ (with $\hat{a}_k = (\mathcal{F}_N a)_k$) and the eigenvectors are the columns of the Fourier matrix $\mathcal{F}_N = (w_N^{jk})_{j,k=0}^{N-1}$. Provide a fast algorithm for solving the linear system of equations $Ax = b$.

Programming Exercise 2: The two-dimensional discrete Fourier transform $\mathcal{F}(X) = \hat{X}$ of $X = (x_{m,n}) \in \mathbb{C}^{M\times N}$ is defined by

$$\hat{x}_{k,l} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x_{m,n} w_M^{mk} w_N^{nl}$$

and the convolution $X * Y \in \mathbb{C}^{M \times N}$ by

$$(X * Y)_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} y_{k-m,l-n}$$

($k = 0, \ldots, M-1, l = 0, \ldots, N-1$). As in the one-dimensional case, the following holds:

$$X * Y = \mathcal{F}^{-1}(\mathcal{F}(X) \cdot \mathcal{F}(Y)),$$

where the multiplication on the right-hand side is again to be understood component-wise, and

$$\mathcal{F}^{-1}(X) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} w_M^{-mk} w_N^{-nl}$$

is the inverse transform. Implement the two-dimensional FFT by first applying your code from PE1 (or even the ones provided from packages) to the rows and then to the columns of X. Then, efficiently compute the convolution of the matrix `double(rgb2gray(imread('input_image.jpg')))` in Matlab or `using Images; Float64.(Gray.("input_image.jpg"))` in Julia and the matrix stored in the file 'convolution_matrix.dat'. The result C can be saved as an image file using the command `imwrite(C,'output_image.jpg');` in Matlab or `save("output_image.png",C)` in Julia. The image file and the convolution matrix can be found on the exercise homepage.
Hint: You can show the image with `imshow(cast(C,'uint8'))` in Matlab and `using ImageView; imshow(C)` in Julia.

Solutions are discussed on Tuesday 29.04.2025.
Tutor: Georgios Vretinaris - if you have question just come to my office (C3P16) or write me an email.