Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
M.Sc. J.-M. Gallard, M.Sc. M. Obersteiner

SS 2018
Worksheet 4
02.05.2018

Algorithms of Scientific Computing

Exercise 1: Fast Discrete Cosine Transform

The butterfly scheme is retrieved as usual:

$$\begin{aligned}
F_k &= \frac{1}{2N}\sum_{n=-N+1}^{N} f_n \omega_{2N}^{-kn} = \frac{1}{2}\left(\frac{1}{N}\sum_{n=-N/2+1}^{N/2} f_{2n}\omega_{2N}^{-2kn} + \frac{1}{N}\sum_{n=-N/2+1}^{N/2} f_{2n-1}\omega_{2N}^{-k(2n-1)}\right) \\
&= \frac{1}{2}\Bigg(\underbrace{\frac{1}{N}\sum_{n=-N/2+1}^{N/2} f_{2n}\omega_N^{-kn}}_{=:G_k} + \underbrace{\frac{1}{N}\sum_{n=-N/2+1}^{N/2} f_{2n-1}\omega_N^{-kn}}_{=:H_k}\omega_{2N}^{k}\Bigg) \\
&= \frac{1}{2}\left(G_k + \omega_{2N}^{k} H_k\right) \\
F_{k+N} &= \frac{1}{2}\left(G_{k+N} + \omega_{2N}^{k+N} H_{k+N}\right) = \frac{1}{2}\left(G_k - \omega_{2N}^{k} H_k\right)
\end{aligned}$$

For the datasets $g_n := f_{2n}$ and $h_n := f_{2n-1}$, respectively, we can try to find other symmetries:

$$g_{-n} = f_{2(-n)} = f_{-2n} = f_{2n} = g_n$$

The "even" data also shows an even symmetry and therefore leads to another Cosine Transform but with half length.

Analogously, for the data with odd indices:

$$h_{-n} = f_{2(-n)-1} = f_{-2n-1} = f_{2n+1} = f_{2(n+1)-1} = h_{n+1}$$

Again we get an "even" symmetry. This is the transform shown in the lecture, known as Quarter-Wave-DCT, again with half length.

Exercise 2: DFT and Least Squares Approximation

Since the Euclidean norm is defined by a scalar product, the function giving the error $E(\alpha_0, \ldots, \alpha_{N-1}) = ||\mathbf{f} - \Phi_N(x)||_2^2$ is a quadratic function that attains its extremum where $\nabla E(\alpha_0, \ldots, \alpha_{N-1})$ vanishes.

Given that the k-th partial derivative is given by

$$\frac{\partial E}{\partial a_k} = \sum_{n=0}^{N-1} \left[e^{-i2\pi nk/N} \left(f_n - \sum_{p=0}^{N-1} \alpha_p e^{i2\pi np/N} \right) \right], \tag{1}$$

we set each partial derivative to zero and obtain

$$\sum_{n=0}^{N} \left[e^{-i2\pi nk/N} \left(f_n - \sum_{p=0}^{N-1} \alpha_p e^{i2\pi np/N} \right) \right] = 0. \tag{2}$$

Rearranging the terms gives us the set of N equations

$$\sum_{n=0}^{N-1} f_n \omega_N^{-nk} = \sum_{p=0}^{N-1} \alpha_p \underbrace{\sum_{n=0}^{N-1} \omega_N^{n(p-k)}}_{N\delta_N(p-k)}, \tag{3}$$

where $\omega_N = e^{i2\pi/N}$ and $\delta_N(k)$ is known as the modular Kronecker delta defined by

$$\delta_N(k) = \begin{cases} 1 & \text{if } k = 0 \text{ or a multiple of } N, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

.

Next, we will show that $\sum_{n=0}^{N-1} \omega_N^{n(p-k)} = N\delta_N(p-k)$. ω_N^k are complex exponentials that for $k = 0, \dots, N-1$ are the Nth roots of unity.

$$(\underbrace{\omega_N^k}_{:=z})^N = \left(e^{i2\pi k/N} \right)^N = e^{i2\pi k} = 1 \tag{5}$$

We can rewrite $z^N - 1 = 0$ as

$$z^N - 1 = (z-1)\left(z^{N-1} + z^{N-2} + \dots + z + 1 \right) = (z-1) \sum_{n=0}^{N-1} z^n = 0. \tag{6}$$

A product is zero, if one of the factors evaluates zero. We have a case distinction.

1. $z = \omega_N^{p-k}$ where $p-k$ is not a multiple of N. Then $z \neq 1$. The product can only evaluate zero if the second term evaluates zero. Hence

$$\sum_{n=0}^{N-1} z^n = \sum_{n=0}^{N-1} \left(\omega_N^{p-k} \right)^n = 0 \tag{7}$$

2. $z = \omega_N^{p-k}$ where $p-k$ is a multiple of N. Then $\omega_N^{p-k} = 1$ and

$$\sum_{n=0}^{N-1} z^n = \sum_{n=0}^{N-1} \left(\omega_N^{p-k} \right)^n = \sum_{n=0}^{N-1} 1 = N. \tag{8}$$

Using this result in equation (3) we obtain

$$\sum_{n=0}^{N-1} f_n \omega_N^{-nk} = N\alpha_k, \tag{9}$$

for $k = 0, \ldots, N-1$. We get that α_ks correspond exactly to the DFT coefficients

$$\alpha_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n \omega_N^{-nk}. \tag{10}$$

The least squares error is minimised when the coefficients of the trigonometric polynomial correspond to the DFT coefficients.

The FFT algorithm can be used to efficiently compute the α_k.

Exercise 3: Fast Discrete Sine Transform

The butterfly scheme is retrieved as usual:

$$\begin{aligned}
F_k &= \frac{1}{2N} \sum_{n=-N+1}^{N} f_n \omega_{2N}^{-kn} = \frac{1}{2} \left(\frac{1}{N} \sum_{n=-N/2+1}^{N/2} f_{2n} \omega_{2N}^{-2kn} + \frac{1}{N} \sum_{n=-N/2+1}^{N/2} f_{2n-1} \omega_{2N}^{-k(2n-1)} \right) \\
&= \frac{1}{2} \Bigg(\underbrace{\frac{1}{N} \sum_{n=-N/2+1}^{N/2} f_{2n} \omega_N^{-kn}}_{=:G_k} + \underbrace{\frac{1}{N} \sum_{n=-N/2+1}^{N/2} f_{2n-1} \omega_N^{-kn}}_{=:H_k} \omega_{2N}^{k} \Bigg) \\
&= \frac{1}{2} \left(G_k + \omega_{2N}^{k} H_k \right) \\
F_{k+N} &= \frac{1}{2} \left(G_{k+N} + \omega_{2N}^{k+N} H_{k+N} \right) = \frac{1}{2} \left(G_k - \omega_{2N}^{k} H_k \right)
\end{aligned}$$

For the datasets $g_n := f_{2n}$ and $h_n := f_{2n-1}$, respectively, we can try to find other symmetries:

$$g_{-n} = f_{2(-n)} = -f_{-2n} = -f_{2n} = -g_n$$

The "odd" data also shows an odd symmetry and therefore lead to another Sine Transform but with half length.

Analog for the data with odd indices:

$$h_{-n} = f_{2(-n)-1} = f_{-2n-1} = -f_{2n+1} = -f_{2(n+1)-1} = -h_{n+1}$$

Again we get an "odd" symmetry. However, this is the transform shown in the lecture, known as Quarter-Wave-DST, again with half length.

For a dataset with the symmetry constraint $f_{-n} = -f_{n+1}$ we get accordingly

$$g_{-n} = f_{2(-n)} = f_{-2n} = -f_{2n+1} = -h_{n+1}$$

and

$$h_{-n} = f_{-2n-1} = f_{-2n+1} = -f_{2n+2} = -f_{2n-1} = -g_{n+1}$$