

# Algorithms of Scientific Computing

## Discrete Cosine Transformation – Solution

### Exercise 1: Two-dimensional Cosine Transformation

We can find a term like

$$\tilde{G}_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n \cos\left(\frac{\pi k (n + \frac{1}{2})}{N}\right)$$

in the equation for  $\tilde{F}_{kl}$  after some small conversions:

$$\begin{aligned} \tilde{F}_{kl} &= \frac{1}{N \cdot M} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_{nm} \cos\left(\frac{\pi k (n + \frac{1}{2})}{N}\right) \cos\left(\frac{\pi l (m + \frac{1}{2})}{M}\right) \\ &= \frac{1}{M} \sum_{m=0}^{M-1} \underbrace{\frac{1}{N} \sum_{n=0}^{N-1} f_{nm} \cos\left(\frac{\pi k (n + \frac{1}{2})}{N}\right)}_{\tilde{G}_{km}} \cos\left(\frac{\pi l (m + \frac{1}{2})}{M}\right). \end{aligned}$$

Actually there are  $M$  of these terms, which we will call

$$\tilde{G}_{km} = \frac{1}{N} \sum_{n=0}^{N-1} f_{nm} \cos\left(\frac{\pi k (n + \frac{1}{2})}{N}\right), \quad m = 0, \dots, M-1,$$

With these  $\tilde{G}_{km}$  we can set

$$\tilde{F}_{kl} = \frac{1}{M} \sum_{m=0}^{M-1} \tilde{G}_{km} \cos\left(\frac{\pi l (m + \frac{1}{2})}{M}\right).$$

These sums can again be computed with the given procedure for each  $k = 0, \dots, N-1$ .

In total we get the following algorithm:

1. Compute all  $N$  values for all  $m = 0, \dots, M-1$ :

$$\tilde{G}_{km} = \frac{1}{N} \sum_{n=0}^{N-1} f_{nm} \cos\left(\frac{\pi k (n + \frac{1}{2})}{N}\right).$$

2. Use the results from all  $n = 0, \dots, N - 1$  to compute the  $N$  values

$$\tilde{F}_{kl} = \frac{1}{M} \sum_{m=0}^{M-1} \tilde{G}_{km} \cos \left( \frac{\pi l (m + \frac{1}{2})}{M} \right).$$

We can interpret this algorithm as a 1d transformation which is first applied column-wise on the 2d array and afterwards applied row-wise on the result values.

**Remark:**

1. We didn't use any special properties of the cosines. So we can apply the same method on all 2d transformations of the shape

$$F_{kl} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_{nm} \phi_k(n) \psi_l(m).$$

2. The extension on the 3d case and even higher-dimensional cases can be achieved in an analog way.

**Exercise 2: Discrete Cosine Transform**

**a) Show that the corresponding Fourier coefficients are real**

$$F_k = \frac{1}{2N} \sum_{n=-N+1}^N f_n \omega_{2N}^{-kn} \tag{1}$$

The proof is done in the following steps:

- ① Isolate the symmetry condition
- ② Insert the symmetry condition
- ③ Assemble terms to a sum over  $f_n$
- ④ Make terms "real"

$$\begin{aligned} F_k &= \frac{1}{2N} \sum_{n=-N+1}^N f_n \omega_{2N}^{-kn} \\ &\stackrel{\textcircled{1}}{=} \frac{1}{2N} \left( \sum_{n=-N+1}^{-1} f_n \omega_{2N}^{-kn} + f_0 \omega_{2N}^0 + \sum_{n=1}^{N-1} f_n \omega_{2N}^{-kn} + f_N \omega_{2N}^{-kN} \right) \\ &= \frac{1}{2N} \left( \sum_{n=1}^{N-1} f_{-n} \omega_{2N}^{kn} + f_0 e^0 + \sum_{n=1}^{N-1} f_n \omega_{2N}^{-kn} + f_N e^{-i2\pi kN/2N} \right) \\ &\stackrel{\textcircled{2}}{=} \frac{1}{2N} \left( \sum_{n=1}^{N-1} f_n \omega_{2N}^{kn} + f_0 + \sum_{n=1}^{N-1} f_n \omega_{2N}^{-kn} + f_N e^{-i\pi k} \right) \end{aligned}$$

$$\begin{aligned}
&\stackrel{\textcircled{3}}{=} \frac{1}{2N} \left( f_0 + \sum_{n=1}^{N-1} f_n \underbrace{(\omega_{2N}^{kn} + \omega_{2N}^{-kn})}_{=\omega_{2N}^{kn} + (\omega_{2N}^{kn})^* = 2\text{Re}\{\omega_{2N}^{kn}\}} + f_N e^{-i\pi k} \right) \\
&\stackrel{\textcircled{4}}{=} \frac{1}{2N} \left( f_0 + 2 \sum_{n=1}^{N-1} f_n \underbrace{\text{Re}\{e^{i2\pi kn/2N}\}}_{=\text{Re}\{\cos(\frac{\pi kn}{N}) + i \sin(\frac{\pi kn}{N})\}} + f_N (\cos(-\pi k) + i \sin(-\pi k)) \right) \\
&= \frac{1}{N} \left( \frac{1}{2} f_0 + \sum_{n=1}^{N-1} f_n \cos\left(\frac{\pi kn}{N}\right) + \frac{1}{2} f_N \cos(\pi k) \right) \in \mathbb{R} \quad \text{q.e.d.}
\end{aligned}$$

**b) Show that the  $F_k$  also have a symmetry:**

Since there are only cosine terms we assume (and get) an even symmetry:

$$\begin{aligned}
F_{-k} &= \frac{1}{N} \left( \frac{1}{2} f_0 + \sum_{n=1}^{N-1} f_n \cos\left(\frac{-\pi kn}{N}\right) + \frac{1}{2} f_N \cos(\pi k) \right) \\
&= \frac{1}{N} \left( \frac{1}{2} f_0 + \sum_{n=1}^{N-1} f_n \cos\left(\frac{\pi kn}{N}\right) + \frac{1}{2} f_N \cos(\pi k) \right) \\
&= F_k
\end{aligned}$$

Since all  $F_{-k} = F_k$ , we need the  $F_k$  only for  $k = 0, \dots, N$  for a Cosine Transform.

**c) Algorithm for the Cosine Transform**

The procedure  $\text{FFT}(\mathbf{f}, N)$  computes the correct coefficients, if we pass the  $N + 1$  data from field  $\mathbf{g}$  as a dataset of length  $2N$  with symmetry  $f_{-n} = f_n$ .

From equation (1) of the worksheet we know that  $\text{FFT}(\mathbf{f}, N)$  gets a dataset  $\mathbf{f}$  with indices  $n = -N + 1, \dots, N$ . We only have to compute the  $F_k$  for  $k = 0, \dots, N$ .

So, the algorithm looks like this:

1. Set  $\mathbf{f}[0] := \mathbf{g}[0] = f_0$   
For all  $n = 1, \dots, N - 1$ :  
Set  $\mathbf{f}[n] := \mathbf{g}[n] = f_n$   
Set  $\mathbf{f}[-n] := \mathbf{g}[n] = f_n$   
Set  $\mathbf{f}[N] := \mathbf{g}[N] = f_N$
2. Call  $\text{FFT}(\mathbf{f}, N)$
3. (Now the Fourier coefficients  $F_k$  are stored in the field  $\mathbf{f}$ )  
For all  $k = 0, \dots, N$ :  
Set  $\mathbf{g}[k] := \mathbf{f}[k] = F_k$

### Exercise 3: Fast Discrete Cosine Transform

The butterfly scheme is retrieved as usual:

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

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 lead to another Cosine 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 "even" symmetry. However, this is the transform shown in the lecture, known as Quarter-Wave-DCT, 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

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