Introduction to the FFT

1 Introduction

Assume that we have a “signal” \( f(x) \) where \( x \) denotes “time”. We would like to represent \( f(x) \) as a linear combination of functions \( e^{2\pi iax} \) (or, equivalently, \( \sin(2\pi ax) \) and \( \cos(2\pi ax) \)) of frequency \( a \). Note that \( \hat{a} = 2\pi a \) is often called circular frequency. The frequency is the number of periods per unit interval, the circular frequency is the number of periods per interval of length \( 2\pi \). Here we will always use frequencies and have \( 2\pi \) in all exponents.

The resulting coefficients \( \hat{f}(a) \) describe “how strong frequency \( a \) is present in the signal” and are interesting in many applications. Often one wants to apply certain “filters” which are easier to apply in the “frequency domain”. So one first applies the Fourier transformation, then applies some other transformation, and then performs an inverse Fourier transformation.

This approach is attractive since there exists a Fast Fourier Algorithm which takes only \( O(N \log(N)) \) operations. This is in particular useful for convolution operations.

We will always assume that the function \( f(x) \) is complex valued. In fact, the Fourier data \( \hat{f} \) will be complex valued even for a real valued function \( f(x) \).

2 Different cases of “Fourier transforms”

There are four basic cases, depending on whether one has periodic or nonperiodic data, or continuous or discrete data. In each case the correspondence of \( f \) and \( \hat{f} \) is stated for the case where \( f, \hat{f} \) are square integrable or summable. Note that Fourier transforms can also be defined for more general functions and even distributions.

2.1 Different conventions for definition of “Fourier transform”

We will always put \( 2\pi \) in the exponent for consistency. Often Fourier transforms are defined without \( 2\pi \) in the exponent, giving factors involving \( 2\pi \) in the definition of the Fourier transform, its inverse, or both. Also, often \( i \) is replaced with \( -i \) in the definitions. Furthermore, for discrete data factors of \( \frac{1}{N} \) or \( h \) occur which can be split up in different ways in the definition of the Fourier transform and its inverse.

These different definitions are trivially related to each other. But one has to carefully check what the definition of “Fourier transform” is in a specific book or software package to make sure that one obtains correct results.

2.2 Periodic functions

Let \( f : \mathbb{R} \to \mathbb{C} \) be a 1-periodic function, i.e., \( f(x) = f(x + 1) \) for all \( x \in \mathbb{R} \). We want to represent this function in the form \( f(x) = \sum_{k \in \mathbb{Z}} \hat{f}_k e^{2\pi ikx} \) using coefficients \( \hat{f}_k \), \( k \in \mathbb{Z} \). If \( \int_0^1 |f(x)|^2 \, dx < \infty \) there are unique coefficients \( (\hat{f}_k)_{k \in \mathbb{Z}} \) with \( \sum_{k \in \mathbb{Z}} |\hat{f}_k|^2 < \infty \) and we have the following relations:

\[
\hat{f}_k = \int_0^1 f(x) e^{-2\pi ikx} \, dx \iff f(x) = \sum_{k \in \mathbb{Z}} \hat{f}_k e^{2\pi ikx} \quad \text{Inversion (1)}
\]

\[
\int_0^1 |f(x)|^2 \, dx = \sum_{k \in \mathbb{Z}} |\hat{f}_k|^2 \quad \text{Parseval (2)}
\]
\[ q(x) = \int_{0}^{1} f(t)g(x-t)\, dt \iff \hat{q}_k = \hat{f}_k\hat{g}_k \]

**Proof:** Assume first that only finitely many \( \hat{f}_k \) are nonzero, and that \( f(x) = \sum_{k \in \mathbb{Z}} \hat{f}_k e^{2\pi ikx} \). Then the result follows since

\[
\int_{0}^{1} e^{2\pi jx} e^{-2\pi ikx} \, dx = \begin{cases} 
1 & \text{if } j = k \\
c'[e^{2\pi i(j-k)x}]^2 & \text{if } j \neq k
\end{cases}
\]

This also implies Parseval’s identity (2). By using a density argument one can conclude that (1), (2) hold for all \( f \) with \( \int_{0}^{1} |f(x)|^2 \, dx < \infty \).

For the convolution formula we have

\[ \hat{q}_k = \int_{x=0}^{1} \int_{t=0}^{1} f(t)g(x-t)\, dt\, e^{-2\pi ikx} \, dx = \int_{t=0}^{1} \int_{y=-t}^{1-t} f(t)g(y)e^{-2\pi ky}e^{-2\pi ikt} \, dy\, dt = \hat{f}_k\hat{g}_k \]

by changing the order of integration and a change of variables from \( x \) to \( y = x - t \). \( \square \)

### 2.3 Discrete data of periodic functions

We consider the values \( f_j \) of a 1-periodic function at the points \( \frac{j}{N} \) for \( j \in \mathbb{Z} \), i.e., we have \( f_j = f_{j+N} \).

We want to find a function \( \tilde{f}(x) \) of the form

\[ \tilde{f}(x) = \sum_{k=0}^{N-1} \hat{f}_k e^{2\pi ikx} \]

which passes through the given points, i.e., \( \tilde{f}(\frac{j}{N}) = f_j, j = 0, \ldots, N-1 \). For given values \( (f_0, \ldots, f_{N-1}) \) there exists a unique solution \((\tilde{f}_0, \ldots, \tilde{f}_{N-1})\) and we have the following relations:

\[
\hat{f}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-2\pi ikj/N} \iff f_j = \sum_{k=0}^{N-1} \hat{f}_k e^{2\pi ikj/N} \quad \text{Inversion} \quad (6)
\]

\[
\frac{1}{N} \sum_{j=0}^{N-1} |f_j|^2 = \sum_{k=0}^{N-1} |\hat{f}_k|^2 \quad \text{Parseval} \quad (7)
\]

\[ q_j = \frac{1}{N} \sum_{\ell=0}^{N-1} f_{\ell} g_{j-\ell} \iff \hat{q}_k = \hat{f}_k\hat{g}_k \quad \text{Convolution} \quad (8)
\]

**Proof of (6):** Assume that the right equation in (6) holds. Then

\[
\frac{1}{N} \sum_{\ell=0}^{N-1} \sum_{j=0}^{N-1} f_{\ell} e^{2\pi i(\ell-k)j/N} = \sum_{\ell=0}^{N-1} \frac{1}{N} \sum_{j=0}^{N-1} e^{2\pi i(\ell-k)j/N} = \frac{1}{N} \sum_{j=0}^{N-1} e^{2\pi i(k-\ell)j/N} = \frac{1}{N} (q^N - 1)/(q - 1)
\]

If \( \ell = k \) we have \( \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{2\pi i(k-k)j/N} = 1 \). If \( \ell \neq k \) we have \( \sum_{j=0}^{N-1} e^{2\pi i(k-\ell)j/N} = \frac{1}{N} (q^N - 1)/(q - 1) \) with \( q = e^{2\pi i k/N} \), hence \( q^N = 1 \) and the sum is zero. \( \square \)

The function \( \tilde{f}(x) \) in (5) uses values of \( k \) in the range \( 0, \ldots, N-1 \). Usually one prefers to have \( k \) values in a symmetric range about zero since the frequency of \( e^{ikx} \) is \( |k| \), and we would like to include all frequencies below a certain bound. It turns out that we can construct a different interpolating function \( \tilde{p}(x) \) using \((\tilde{f}_0, \ldots, \tilde{f}_{N-1})\). The key observation is that \( e^{2\pi inx} = 1 \) at all points \( x = \frac{k}{N} \). Hence we can divide some terms in (5) by \( e^{2\pi Nkx} \) and still have an interpolating function.
**N is odd:** Here we divide the terms in (5) with $k = \frac{N-1}{2}, \ldots, N - 1$ by $e^{2\pi i N x}$ and obtain

$$
\hat{p}(x) = \sum_{k=0}^{(N-1)/2} \hat{f}_k e^{2\pi i k x} + \sum_{k=(N-1)/2+1}^{N-1} \hat{f}_k e^{2\pi i (k-N) x} = \sum_{k=-(N-1)/2}^{(N-1)/2} \hat{f}_k e^{2\pi i k x}
$$

(9)

if we define $\hat{f}_k := \hat{f}_{k+N}$ for $k < 0$. In other words, we use the coefficients

$$
\begin{array}{cccccccc}
-N/2 & -N/2 + 1 & \ldots & 0 & 1 & \ldots & N/2 - 1 & N/2 \\
\hat{f}_{(N-1)/2+1} & \hat{f}_{N-1} & \hat{f}_0 & \hat{f}_1 & \ldots & \hat{f}_{N/2-1} & \frac{1}{2} \hat{f}_{N/2}
\end{array}
$$

Using $e^{it} = \cos t + i \sin t$ we can rewrite $\hat{p}(x)$ in the form

$$
\hat{p}(x) = a_0 + \sum_{k=1}^{(N-1)/2} \left( a_k \cos(2\pi k x) + b_k \sin(2\pi k x) \right).
$$

If all $f_j$ are real, then $a_k, b_k$ are real.

**N is even:** Here we divide the terms in (5) with $k = \frac{N}{2} + 1, \ldots, N - 1$ by $e^{2\pi i N x}$, and for $k = N/2$ we replace $\hat{f}_{N/2} e^{2\pi i (N/2) x}$ with $\frac{1}{2} \hat{f}_{N/2} e^{2\pi i (N/2) x} + \frac{1}{2} \hat{f}_{N/2} e^{2\pi i (-N/2) x} e^{-2\pi i N x}$ yielding

$$
\hat{p}(x) = \sum_{k=-N/2+1}^{N/2-1} \hat{f}_k e^{2\pi i k x} + \frac{1}{2} \hat{f}_{N/2} \left( e^{-2\pi i (N/2) x} + e^{-2\pi i (-N/2) x} \right)
$$

(10)

if we define $\hat{f}_k := \hat{f}_{k+N}$ for $k < 0$. In other words, we use the coefficients

$$
\begin{array}{cccccccc}
-N/2 & -N/2 + 1 & \ldots & 0 & 1 & \ldots & N/2 - 1 & N/2 \\
\frac{1}{2} \hat{f}_{N/2} & \hat{f}_{N/2+1} & \ldots & \hat{f}_{N-1} & \hat{f}_0 & \hat{f}_1 & \ldots & \hat{f}_{N/2-1} & \frac{1}{2} \hat{f}_{N/2}
\end{array}
$$

Using $e^{it} = \cos t + i \sin t$ we can rewrite $\hat{p}(x)$ in the form

$$
\hat{p}(x) = a_0 + \sum_{k=1}^{N/2-1} \left( a_k \cos(2\pi k x) + b_k \sin(2\pi k x) \right) + a_{N/2} \cos(2\pi \frac{N}{2} x).
$$

(11)

If all $f_j$ are real, then $a_k, b_k$ are real.

We can use (5) to define the Fourier coefficients $(\hat{f}_0, \ldots, \hat{f}_{N-1})$, but we should use (9), (10) to interpret them in terms of frequencies. The coefficients $\hat{f}_j$ with $j$ close to 0 or $N - 1$ correspond to “low frequency” components of $f$, the coefficients $\hat{f}_j$ with $j$ close to $N/2$ correspond to high frequency components. We also see that given data with a spacing $d = \frac{1}{N}$ allows us to determine frequencies up to $N/2 = \frac{1}{2d}$. This frequency $\frac{1}{2d}$ is called the Nyquist frequency.

**Matlab:** The function `fft` in Matlab implements $N F_N$ and the function `fft` implements $N^{-1} F_N$. Assume we have the Matlab vectors $f$ and $\text{fhat}$ with

$$
f = [f_0, \ldots, f_{N-1}], \quad \text{fhat} = [\hat{f}_0, \ldots, \hat{f}_{N-1}]
$$

then

$$
\text{fft}(f) = N [\hat{f}_0, \ldots, \hat{f}_{N-1}], \quad \text{ifft}(\text{fhat}) = N^{-1} [f_0, \ldots, f_{N-1}].
$$

Note that Matlab vectors of length $N$ are indexed by $1, \ldots, N$ (instead of $0, \ldots, N - 1$).
2.4 Manipulating the Fourier vector

Recall that we should interpret the Fourier vector \((\hat{f}_0, \ldots, \hat{f}_{N-1})\) as follows:
For odd \(N = 2n + 1\) the vector contains the coefficients of the functions
\[
\hat{f}_0 \quad \hat{f}_1 \quad \ldots \quad \hat{f}_n \quad \hat{f}_{N-n} \quad \ldots \quad \hat{f}_{N-1}
\]
For even \(N = 2n\) the vector contains the coefficients of the functions
\[
\hat{f}_0 \quad \hat{f}_1 \quad \ldots \quad \hat{f}_{n-1} \quad \hat{f}_n \quad \frac{\hat{f}_{N-(n-1)}}{2} \quad \hat{f}_{N-(n-1)} \quad \ldots \quad \hat{f}_{N-1}
\]

Extending the Fourier vector by zeros: The Fourier vector \((\hat{f}_0, \ldots, \hat{f}_{N-1})\) corresponds to a function \(\hat{p} \in \mathcal{T}_n\) for \(N = 2n + 1\), and \(\hat{p} \in \mathcal{T}^\text{cos}_n\) for \(N = 2n\). Note that the same function is represented by the longer Fourier vector of length \(M > N\) where we insert zeros for the higher frequency terms:
For odd \(N = 2n + 1\) we obtain the vector
\[
(\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_n, 0, \ldots, 0, \hat{f}_{n+1}, \ldots, \hat{f}_{2n})
\]
\(M - N\) zeros
For even \(N = 2n\) we obtain the vector
\[
(\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{n-1}, \frac{1}{2}\hat{f}_n, 0, \ldots, 0, \frac{1}{2}\hat{f}_n, \hat{f}_{n+1}, \ldots, \hat{f}_{2n})
\]
\(M - N - 1\) zeros
This operation allows us to evaluate the function \(\hat{p}(x)\) at \(M > N\) equidistant points using \(F_M^{-1}\), yielding an interpolation on a finer grid of the original data.

Truncating the Fourier vector: The Fourier vector \((\hat{f}_0, \ldots, \hat{f}_{M-1})\) corresponds to a function \(\hat{p}(x)\). Sometimes we want to truncate the Fourier vector by removing the higher frequency terms, so that we get a Fourier vector of length \(N < M\) corresponding to a function \(\hat{q} \in \mathcal{T}_n\) for \(N = 2n + 1\), and \(\hat{q} \in \mathcal{T}^\text{cos}_n\) for \(N = 2n\).
For odd \(N = 2n + 1\) we obtain the vector
\[
(\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{n-1}, \hat{f}_n + \hat{f}_{M-n}, \hat{f}_{M-n+1}, \ldots, \hat{f}_{M-1})
\]
For even \(N = 2n\) we obtain the vector
\[
(\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{n-1}, \hat{f}_n + \frac{\hat{f}_{M-n}}{2}, \frac{\hat{f}_{M-n}}{2}, \hat{f}_{M-n+1}, \ldots, \hat{f}_{M-1})
\]
Let \(x_j = \frac{j}{M}, j = 0, \ldots, M-1\). Note that \((\hat{q}(x_j))_{j=0,\ldots,M-1}\) is the best discrete least squares approximation of the given data \((f_j)_{j=0,\ldots,M-1}\) using \(\mathcal{T}_n\) for \(N = 2n + 1\), or using \(\mathcal{T}^\text{cos}_n\) for \(N = 2n\).
For \(M \gg N\) the function \(\hat{q}(x)\) approximates the truncated Fourier series for the function \(f(x)\) as can be seen by comparing (1) and (6). The truncated Fourier series is the best continuous least squares approximation of \(f(x)\) using \(\mathcal{T}_n\) for \(N = 2n + 1\), or using \(\mathcal{T}^\text{cos}_n\) for \(N = 2n\).

2.5 The cosine transform

In some applications it is useful to consider even or odd data, which allow to use series with only cosine or sine functions, respectively. The cosine transform is useful in image processing. There are a continuous version and two discrete versions of the cosine transform.
Continuous cosine transform  Here we consider a 2-periodic function \( f \) (i.e., \( f(x) = f(x+2) \)) which is even (i.e., \( f(x) = f(-x) \)). We assume that \( \int_0^1 |f(x)|^2 \, dx \) is finite. Using the results of section 2.2 we can then find a unique coefficients \( a_k \) \((k = 0, 1, 2, \ldots)\) and \( b_k \) \((k = 1, 2, \ldots)\) such that

\[
f(x) = a_0 + \sum_{k=1}^{\infty} \left( a_k \cos(\pi k x) + b_k \sin(\pi k x) \right)
\]

As \( f(x) \) is even the uniqueness implies that all \( b_k = 0 \). The resulting transform \( f \mapsto (2a_0, a_1, a_2, \ldots) =: (\hat{f}_0, \hat{f}_1, \hat{f}_2, \ldots) \) is called cosine transform and has the following properties:

\[
\hat{f}_k = 2 \int_0^1 f(x) \cos(\pi k x) \, dx \iff f(x) = \frac{1}{2} \hat{f}_0 + \sum_{k=1}^{\infty} \hat{f}_k \cos(\pi k x) \tag{12}
\]

\[
2 \int_0^1 |f(x)|^2 \, dx = \frac{1}{2} |\hat{f}_0|^2 + \sum_{k=1}^{\infty} |\hat{f}_k|^2 \tag{13}
\]

\[
q(x) = \int_{-1}^1 f(t) g(x-t) \, dt \iff \hat{q}_k = \hat{f}_k \hat{g}_k \tag{14}
\]

The integrals on the left in (12), (13) could also be written as \( \int_{-1}^1 (\ldots) \), but they simplify to \( 2 \int_0^1 (\ldots) \) since \( f(x) \) is even. In the sum on the left in (14) we cannot make this simplifications as \( f(t)g(x-t) \) is not an even function of \( t \).

Note that the cosine transform of real data is real.

Discrete even data: interval-boundary and interval-center case  There are two possibilities for samples of an even, 2-periodic function: Either the samples are taken at the points \( \frac{j}{N} \), or at the midpoints \( \frac{j+1/2}{N} \). The two situations are shown here for the case \( N = 4 \):

\[
\begin{array}{cccccccc}
F_4 & F_3 & F_2 & F_1 & F_0 & F_1 & F_2 & F_3 & F_4 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
-1 & 0 & 1 & & & & & & \\
\end{array}
\]

\[
\begin{array}{cccccccc}
f_3 & f_2 & f_1 & f_0 & f_0 & f_1 & f_2 & f_3 & f_4 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
-1 & 0 & 1 & & & & & & \\
\end{array}
\]

Note that in both cases we obtain discrete sequences \( F_j, f_j \) with period 8, i.e., \( F_{j+8} = F_j, f_{j+8} = f_j \) for all \( j \in \mathbb{Z} \). In the first case we have \( N+1 = 5 \) different values \( F_0, \ldots, F_N \) whereas in the second case we have only \( N = 4 \) different values \( f_0, \ldots, f_{N-1} \).

Discrete cosine transform, values at endpoints of intervals  Here we consider a 2-periodic function \( u \) (i.e., \( u(x) = u(x+2) \)) which is even (i.e., \( u(x) = u(-x) \)) and which we sample at the endpoints \( \frac{j}{N} \) of the intervals \( \left[ \frac{j}{N}, \frac{j+1}{N} \right] \) \( j = 0, \ldots, N \). We denote the resulting discrete values at \( \frac{j}{N} \) by \( F_j \) for \( j \in \mathbb{Z} \). Therefore \( F_j \) satisfies \( F_j = F_{j+2N}, F_{-j} = F_j \) for \( j \in \mathbb{Z} \).
Using the results of section 2.3 we can find a unique interpolating function

\[
\hat{P}(x) = a_0 + \sum_{k=1}^{N-1} \left( a_k \cos(\pi k x) + b_k \sin(\pi k x) \right) + a_N \cos(\pi N x)
\]

But since the data are even in \([-1,1]\) the function \(\hat{P}(x)\) must have the same property (using the uniqueness) and all \(b_k = 0\). The resulting transform \((F_0, \ldots, F_N) \mapsto (2a_0, a_1, \ldots, a_N) =: (\hat{F}_0, \ldots, \hat{F}_N)\) is called **discrete cosine transform**. With the notation \(\sum_{j=\alpha}^{\beta} C_j := \frac{1}{2} C_\alpha + \left( \sum_{j=\alpha+1}^{\beta-1} C_j \right) + \frac{1}{2} C_\beta\) (which is reminiscent of the trapezoid rule) we have the following properties:

\[
\hat{F}_k = \frac{2}{N} \sum_{j=0}^{N} F_j \cos\left(\frac{\pi k j}{N}\right) \quad \Longleftrightarrow \quad F_j = \frac{2}{N} \sum_{k=0}^{N} \hat{F}_k \cos\left(\frac{\pi k j}{N}\right) \tag{15} \text{ Inversion}
\]

\[
\frac{2}{N} \sum_{j=0}^{N} |F_j|^2 = \sum_{k=0}^{N} |\hat{F}_k|^2 \tag{16} \text{ Parseval}
\]

\[
Q_j = \frac{1}{N} \sum_{\ell=-N}^{N} F_\ell G_{j-\ell} = \frac{1}{N} \sum_{\ell=-N}^{N} F_\ell \tilde{G}_{j-\ell} \quad \Longleftrightarrow \quad \tilde{Q}_k = \hat{F}_k \hat{G}_k \tag{17} \text{ Convolution}
\]

In (17) \(F_j, G_j\) are understood as infinite \(2N\) periodic sequences. The sums on the left in (15), (16) could also be written as \(\frac{1}{N} \sum_{\ell=-N}^{N} F_\ell\), but they simplify to \(\frac{2}{N} \sum_{j=0}^{N} F_j\) since \(f_j\) is even, i.e., \(f_{-j} = f_j\). In the sum on the left in (17) we cannot make this simplifications as \(f_\ell G_{j-\ell}\) is not even with respect to \(\ell\).

Note that the discrete cosine transform of real data is real.

**Discrete cosine transform, values at centers of intervals**  Here we consider a 2-periodic function \(u\) (i.e., \(u(x) = u(x + 2)\)) which is even (i.e., \(u(x) = u(-x)\)) and which we sample at the midpoints \(\frac{j+1/2}{N}\) of the intervals \([\frac{j}{N}, \frac{j+1}{N}]\) \(j = 0, \ldots, N-1\). We denote the resulting discrete values at \(\frac{j+1/2}{N}\) by \(f_j\) for \(j \in \mathbb{Z}\).

Therefore \(f_j\) satisfies \(f_j = f_{j+2N}\), \(f_{j-1} = f_j\) for \(j \in \mathbb{Z}\).

Using the results of section 2.3 we can find a unique interpolating function

\[
\hat{\tilde{P}}(x) = a_0 + \sum_{k=1}^{N-1} \left( a_k \cos\left(\frac{\pi k j}{N}\right) + b_k \sin\left(\frac{\pi k j}{N}\right) \right) + b_N \sin(\pi N x)
\]

(as the data are in the midpoints we have to shift (11), giving sine instead of cosine for the highest frequency). But since the data are even in \([-1,1]\) the function \(\hat{\tilde{P}}(x)\) must have the same property (using the uniqueness) and all \(b_k = 0\). The resulting transform \((f_0, \ldots, f_{N-1}) \mapsto (2a_0, a_1, \ldots, a_{N-1}) =: (\hat{f}_0, \ldots, \hat{f}_{N-1})\) is called **discrete cosine transform** and has the following properties:

\[
\hat{f}_k = \frac{2}{N} \sum_{j=0}^{N-1} f_j \cos\left(\frac{\pi k j + 1/2}{N}\right) \quad \Longleftrightarrow \quad f_j = \frac{1}{2} \hat{f}_0 + \sum_{k=1}^{N-1} \hat{f}_k \cos\left(\frac{\pi k j + 1/2}{N}\right) \tag{18} \text{ Inversion}
\]

\[
\frac{2}{N} \sum_{j=0}^{N-1} |f_j|^2 = \left(\frac{1}{2}\right) |\hat{f}_0|^2 + \sum_{k=1}^{N-1} |\hat{f}_k|^2 \tag{19} \text{ Parseval}
\]
\[ Q_j = \frac{1}{N} \sum_{\ell=-N}^{N-1} f_{\ell} g_{j-\ell} \quad \iff \quad \hat{Q}_k = \begin{cases} \hat{f}_k \hat{g}_k & k = 0, \ldots, N - 1 \\ 0 & k = N \end{cases} \quad \text{(Convolution 20)} \]

\[ q_j = \frac{1}{N} \sum_{\ell=-N}^{N-1} f_{\ell} G_{j-\ell} \quad \iff \quad \hat{q}_k = \hat{f}_k \hat{G}_k, \quad k = 0, \ldots, N - 1 \quad \text{(21)} \]

Note that there are two cases for the convolution: You can convolve \( f_j \) with another even, interval-center sequence \( g_j \) which gives an even, interval-boundary sequence \( Q_j \). You can also convolve \( f_j \) with an even, interval-boundary sequence \( g_j \) which gives an even, interval-center sequence \( q_j \). The cosine transforms \( \hat{Q}_k \), \( \hat{G}_k \) of the interval-boundary sequences \( Q_j \), \( G_j \) are defined by (15). In (20), (21) \( f_j \), \( g_j \), \( G_j \) are understood as infinite 2\(N\)-periodic sequences.

The sums on the left in (18), (19) could also be written as \( \frac{1}{N} \sum_{\ell=-N}^{N-1} \), but they simplify to \( \frac{2}{N} \sum_{j=0}^{N-1} \) since \( f_j \) is “even”, i.e., \( f_{-j-1} = f_j \). In the sum on the left in (20) we cannot make this simplifications as \( f_{\ell} g_{j-\ell} \) is not “even” with respect to \( \ell \).

Note that the discrete cosine transform of real data is real.

**Image Compression**  The two dimensional version of the discrete Fourier transform is frequently used in image processing. The “image” is a function defined on \([0,1]^2\) which gives the brightness at each point. A discretized image consists of “pixels” obtained by dividing \([0,1]^2\) in \(N\) by \(N\) subsquares, with a brightness value given in the center of each subsquare. By using a symmetric extension of the image values to \(\mathbb{R}^2\) one avoids jumps at the edges, yielding faster convergence of interpolating functions and truncated series.

**Matlab:** The function \texttt{dct} implements the discrete cosine transform for interval-centered data as defined here, but with a factor \(\sqrt{N/2}\) and a different factor for \(\hat{f}_0\): Assume the Matlab vector \(f\) contains the data values, i.e., \(f = [f_0, \ldots, f_{N-1}]\). Then \texttt{dct}(\(f\)) gives the vector

\[
\texttt{dct}([f_0, \ldots, f_{N-1}]) = \sqrt{\frac{N}{2}} [2^{-1/2} \hat{f}_0, \ldots, \hat{f}_{N-1}]
\]

### 3 Fast Fourier Transform

The *discrete Fourier transform* is the mapping \((f_0, \ldots, f_{N-1}) \mapsto (\hat{f}_0, \ldots, \hat{f}_{N-1})\) for the case of periodic, discrete data. A straightforward implementation of the formulas

\[
f_j = \sum_{k=0}^{N-1} e^{2\pi i jk/N} \hat{f}_k, \quad \hat{f}_k = \frac{1}{N} \sum_{j=0}^{N-1} e^{-2\pi i jk/N} f_j \quad \text{(22)}
\]

would involve \(O(N^2)\) operations. However, for special values of \(N\) it is possible to give a much faster algorithm which requires only \(O(N \log N)\) operations. The most important case is where \(N\) is a power of 2.

Note that the only difference between the two equations in (22) is a minus sign and a factor of \(\frac{1}{N}\). Hence it is sufficient to describe the algorithm for one direction, e.g., \(F_N: (\hat{f}_0, \ldots, \hat{f}_{N-1}) \mapsto (f_0, \ldots, f_{N-1})\). Here \(F_N\) corresponds to a matrix containing powers of \(w_N := e^{2\pi i/N}\).

Assume that \(N\) is even. Then we can split up the sum in even and odd terms:

\[
f_j = \sum_{k=0}^{N-1} w_N^{jk} \hat{f}_k = \sum_{k=0}^{N/2-1} w_N^{(2k)} \hat{f}_{2k} + \sum_{k=0}^{N/2-1} w_N^{(2k+1)} \hat{f}_{2k+1} = \sum_{k=0}^{N/2-1} w_N^{jk} \hat{f}_{2k} + w_N^{j} \sum_{k=0}^{N/2-1} w_N^{jk} \hat{f}_{2k+1}
\]
using \( w_{N/2} = w_N^2 \). Note that \( \sum_{k=0}^{N/2-1} w_{N/2}^j \hat{f}_{2k} \) for \( j = 0, \frac{N}{2} - 1 \) is exactly \( F_{N/2} \hat{f}^e \) where \( \hat{f}^e := (\hat{f}_0, \hat{f}_2, \ldots, \hat{f}_{N-2})^\top \) contains the even components of the vector \( \hat{f} \). For \( j = N/2, \ldots, N-1 \) we can write \( j = N/2 + j_1 \) and have \( w_{N/2}^j = w_{N/2}^{j_1} \) because of \( w_{N/2} = 1 \). The second sum can be similarly expressed in terms of the odd components \( \hat{f}^o := (\hat{f}_1, \hat{f}_3, \ldots, \hat{f}_{N-1})^\top \). Since \( w_{N/2}^N = -1 \) we get

\[
f = \left( \begin{array}{c} y^e + y^o \\ y^e - y^o \end{array} \right), \quad \text{where} \quad y^e := F_{N/2} \hat{f}^e, \quad y^o := \left( \begin{array}{c} w_N \\ \vdots \\ w_{N/2-1} \end{array} \right) F_{N/2} \hat{f}^o
\]

(23)

Therefore a transform \( F_N \) of length \( N \) requires two transforms \( F_{N/2} \) of length \( N/2 \) and \( N \) additional operations (additions and multiplications). Let \( c_N \) denote number of operations needed for performing \( F_N \), then (23) implies \( c_N = 2c_{N/2} + N \). Since \( F_1 f = f \) we have \( c_1 = 0 \) and therefore \( c_N = Nm = N \log_2 N \) for \( N = 2^m \).