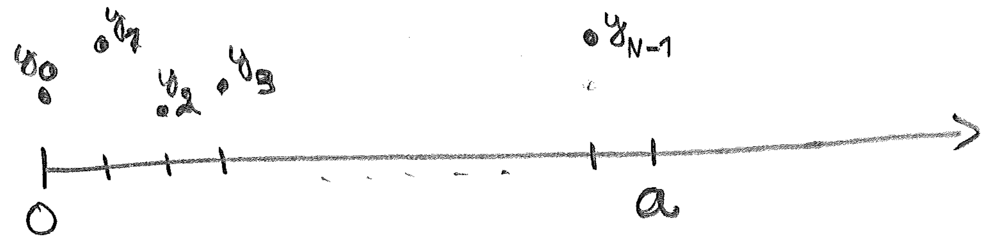
5. The Discrete Fourier Transform

5.1 Introduction and definitions

It happens often in practice that instead of knowing all of the values of a function in an interval $[0, a)$, we know the period a and N of its values regularly spaced over one period:

$$f\left(k \cdot \frac{a}{N}\right) = y_k, \quad k = 0, 1, 2, \dots, N-1.$$



Standardized to $T = [0, 1)$ we would have:

$$f(k/N) = y_k, \quad k = 0, 1, 2, \dots, N-1. \qquad (5.1)$$

The "signal" $f(t)$ is thus assumed to have been sampled at regularly spaced times separated by $1/N$ units (a/N units in $[0, a)$). We also assume that the Fourier series of f converges pointwisely to f and that at points of discontinuity

$$f(t_0) = \frac{1}{2}\left(f(t_0+) + f(t_0-)\right)$$

holds, where $f(t_0+) = \lim_{t \to t_0^+} f(t)$, $f(t_0-) = \lim_{t \to t_0^-} f(t)$.

It could also be the case that we know the function $f \in L_1(T)$ $(f \in L_1([0,a)))$ but cannot obtain the Fourier coefficients

$$\hat{f}(n) = \int_T f(t)\, e^{-2\pi i n t} \qquad \left(\hat{f}(n) = \frac{1}{a}\int_0^a f(t)\cdot e^{-2\pi i n t/a}\right)$$

analytically from the integral.

In both cases we are faced with the problem of finding approximative values for the Fourier coefficients $\hat{f}(n)$.

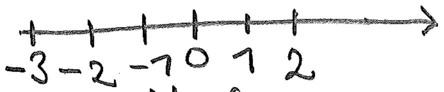
Notes. 1) There is no loss of generality in supposing that $f(0)$ is available in (5.7), because if we are given the "sampled" function values:

$$f(k/N + \alpha), \quad k = 0, 1, \dots, N-1,$$

for some constant α, then we may change the variable by $x = t - \alpha$ and use the data to compute Fourier coefficients for $f(x+\alpha)$.

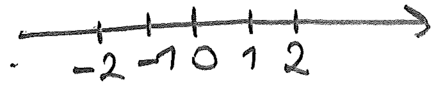
2) Given the N function values in (5.7) it is natural to try to compute N Fourier coefficients. Since $\hat{f}(n)$ tends to zero as $n \to \pm\infty$, we choose to compute $\hat{f}(n)$ for

N even: $n = -N/2, \dots, N/2 - 1,$



N = 6

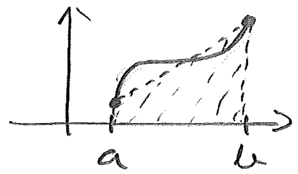
N odd: $n = -\frac{(N-1)}{2}, \dots, \frac{N-1}{2},$



N = 5

We can use the trapezoidal rule:

$$\int_a^b f(x)\,dx \approx (b-a)\,\frac{f(a)+f(b)}{2},$$



to approximate the integral $\int_T f(t)\,e^{-2\pi i n t}$ in N intervals of length $1/N$, assuming $f(1)=f(0)$:

$$\hat{f}(n) \approx \sum_{k=0}^{N-1} \frac{1}{2N}\left(f(k/N)\cdot e^{-2\pi i n \cdot k/N} + f\left(\tfrac{k+1}{N}\right)\cdot e^{-2\pi i n \cdot \frac{(k+1)}{N}}\right)$$

$$= \frac{1}{N}\cdot\sum_{k=0}^{N-1} f\left(\tfrac{k}{N}\right)\cdot e^{-2\pi i n \cdot \frac{k}{N}} = \frac{1}{N}\cdot\sum_{k=0}^{N-1} y_k\cdot e^{-2\pi i n \cdot \frac{k}{N}}$$

$$= \frac{1}{N}\cdot\sum_{k=0}^{N-1} y_k\cdot \omega_N^{-n\cdot k}, \qquad \omega_N := e^{2i\pi\cdot\frac{1}{N}}.$$

$$=: c_n'. \tag{5.2}$$

Thus we obtain the approximate formula

$$\hat{f}(n) \approx c_n', \quad \begin{cases} -N/2 \le n \le N/2 - 1, & N \text{ even}, \\ -\frac{(N-1)}{2} \le n \le \frac{N-1}{2}, & N \text{ odd}, \end{cases}$$

using the trapezoidal rule.

Another method to obtain approximations of the Fourier coefficients $\hat{f}(n)$ is to require that a trigonometrical polynomial interpolates f at the points k/N, $k=0,\dots,N-1$. Suppose that N is even and define

$$p(t) = \sum_{n=-N/2}^{N/2-1} c_n^N\cdot e^{2\pi i n t}. \tag{5.3}$$

We are then led to solve the linear system of order N,

$$\sum_{n=-N/2}^{N/2-1} c_n^N \cdot w_N^{n\cdot k} = y_k, \quad k=0,1,2,\dots,N-1. \qquad (*)$$

The functions involved are N-periodic, so we can translate the negative indices to the right by N:

$$\sum_{n=-N/2}^{-1} c_n^N w_N^{nk} = \sum_{l=N/2}^{N-1} c_{l-N}^N \cdot w_N^{k(l-N)} = \sum_{n=N/2}^{N-1} c_{n-N}^N \cdot w_N^{n\cdot k}.$$

By defining

$$Y_n = \begin{cases} c_n^N, & \text{if } 0 \le n \le \frac{N}{2}-1, \\ c_{n-N}^N, & \text{if } \frac{N}{2} \le n \le N-1, \end{cases}$$

the system $(*)$ is written as

$$\sum_{n=0}^{N-1} Y_n \cdot w_N^{n\cdot k} = y_k, \quad k=0,1,\dots,N-1.$$

This system can be solved explicitly. Let l be an integer between 0 and $N-1$:

$$\sum_{k=0}^{N-1} y_k w_N^{-k\cdot l} = \sum_{k=0}^{N-1}\sum_{n=0}^{N-1} Y_n w_N^{k(n-l)} = \sum_{n=0}^{N-1} Y_n \sum_{k=0}^{N-1} w_N^{k(n-l)}$$

The last sum is a geometric series:

$$\sum_{k=0}^{N-1} \omega_N^{k(n-\ell)} = \begin{cases} 0, & \text{if } \ell \neq n, \\ N, & \text{if } \ell = n. \end{cases} \qquad (**)$$

If $\underline{\ell = n}$ then $\omega_N^{k(n-\ell)} = \omega_n^0 = 1$, so clearly $(**)$ holds. If $\underline{\ell \neq n}$, then since $\omega_N^{(n-\ell)} = e^{2i\pi(n-\ell)/N}$,

$$\sum_{k=0}^{N-1} (\omega_N^{(n-\ell)})^k = \frac{(\omega_N^{(n-\ell)})^N - 1}{\omega_N^{(n-\ell)} - 1} = 0,$$

because $\omega_N^{(n-\ell)\cdot N} = e^{2i\pi\cdot(n-\ell)} = 1$. Hence

$$\sum_{k=0}^{N-1} y_k \omega_N^{-k\cdot\ell} = N Y_\ell,$$

and the unknowns Y_n are given by

$$Y_n = \frac{1}{N} \sum_{k=0}^{N-1} y_k \cdot \omega_N^{-n\cdot k}, \quad n = 0, 1, \ldots, N-1.$$

This is the same formula as (5,2)! After a change of indices we find that

$$c_n^N = c_n', \quad \begin{cases} -N/2 \leq n \leq \frac{N}{2} - 1, & N \text{ even}, \\ -\frac{(N-1)}{2} \leq n \leq \frac{N-1}{2}, & n \text{ odd}. \end{cases}$$

So integrating by the trapezoidal rule gives N approximate Fourier coefficients c_n' that are equal to the Fourier coefficients c_n^N of the trigonometrical polynomial (5,3) in the even N case, (and also when N is odd to the corresponding form of (5,3))$_g$ that interpolates f at the points $t_k = k/N$.

We have the equivalent formules

$$y_k = \sum_{n=0}^{N-1} Y_n \cdot w_N^{n\cdot k}, \quad k = 0,1,2,\dots,N-1,$$

$$Y_n = \frac{1}{N} \sum_{k=0}^{N-1} y_k \cdot w_N^{-n\cdot k}, \quad n = 0,1,2,\dots,N-1, \tag{5,4}$$

and the approximate Fourier coefficients are

$$\hat{f}(n) \approx c_n^N = c_n' = \begin{cases} Y_N, & \text{if } 0 \le n < \frac{N}{2}, \\ Y_{n+N}, & \text{if } \begin{cases} -\frac{N}{2} \le n < 0, & N \text{ even}, \\ -\frac{N}{2} < n < 0, & N \text{ odd}. \end{cases} \end{cases}$$

<u>Definition 5.1.</u> $\Pi_N = \{$all periodic sequences $F(m)$ with period N, i.e., $F(m+N) = F(m)$ $\forall m \in \mathbb{Z}\}$.

<u>Note</u>: It is enough to know $F(0), F(1), \dots, F(N-1)$ to know the whole sequence (or any other set of N consecutive integer values).

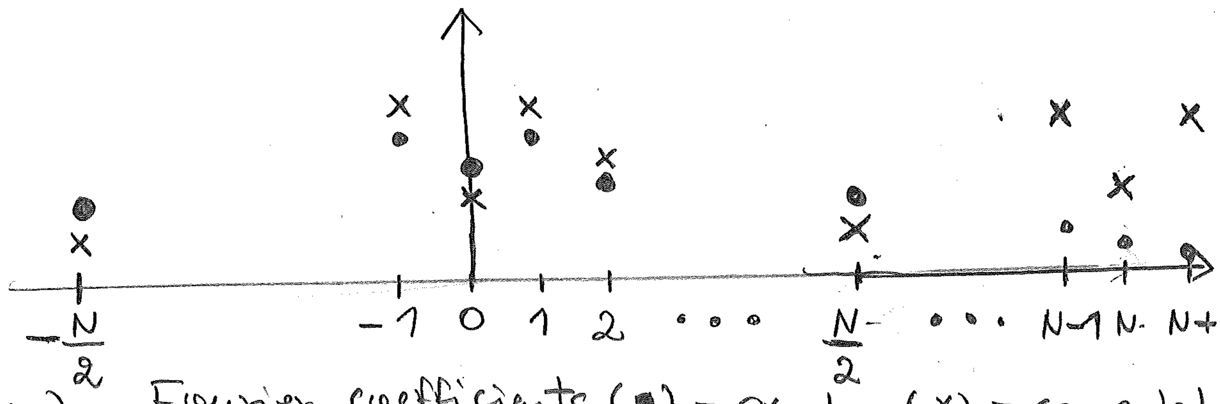
<u>Definition 5.2.</u> The <u>Fourier transform</u> of a sequence $F \in \Pi_N$ is given by

$$(\mathcal{F}F)(m) = \hat{F}(m) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) \cdot e^{-2\pi i m k/N}, \quad m \in \mathbb{Z}.$$

Warning 5.3. Sometimes the constant $\frac{1}{N}$ in front of the sum in Def. 5.2 is replaced by $1/\sqrt{N}$ or is omitted completely (Matlab), which affects the inversion formula.

Note. Notice, since y_k in the first formula in (5.4) stems from a function $f \in L^1(T)$ we can extend the vector to a sequence in Π_N, and thus Y_n in the second formula of (5.4) can be extended to a sequence in Π_N. Thus we have that $\{Y_n\}$ is the discrete Fourier transform of $\{y_k\}$. This means that the approximate Fourier coefficients $c_n^N = c_n'$ on page 72 also form a periodic sequence in Π_N. It is important to remember that c_n^N and c_n' are approximations for $\hat{f}(n)$, the n:th Fourier coefficient of $f \in L^1(T)$, only for

$$\begin{cases} -\frac{N}{2} \leq n < \frac{N}{2}, & \text{when } N \text{ is even,} \\ -\frac{(N-1)}{2} \leq n \leq \frac{N-1}{2}, & \text{when } N \text{ is odd.} \end{cases}$$



(N even). Fourier coefficients (•) = exact, (x) = computed.

Lemma 5.4. The sequence $\{\hat{F}(m)\}$ is periodic with the same period N as $\{F(k)\}$.

Proof. $$\hat{F}(m+N) = \frac{1}{N}\sum_{k=0}^{N-1} e^{-2\pi i(m+N)k/N}\cdot F(k)$$
$$= \frac{1}{N}\cdot\sum_{k=0}^{N-1} \underbrace{e^{-2\pi i k}}_{=1}\cdot e^{-2\pi i mk/N}\cdot F(k) = \hat{F}(m).$$

Thus $\{F\}\in\Pi_N \Rightarrow \{\hat{F}\}\in\Pi_N$. □

Theorem 5.5. The sequence $\{F(k)\}$ can be reconstructed from $\{\hat{F}(m)\}$ by the inversion formula
$$(\mathcal{F}^{-1}\hat{F})(k) = F(k) = \sum_{m=0}^{N-1} e^{2\pi i mk/N}\cdot\hat{F}(m), \quad k=0,1,\dots,N-1.$$

Note. No $\frac{1}{N}$ in front of the sum (Matlab puts the $1/N$ in front of the inversion formula instead).

Proof. (This calculation has been done earlier).
$$\sum_{m=0}^{N-1} e^{2\pi i mk/N}\cdot\left(\frac{1}{N}\sum_{\ell=0}^{N-1} e^{-2\pi i m\ell/N}\cdot F(\ell)\right) = \frac{1}{N}\sum_{\ell=0}^{N-1} F(\ell)\cdot\underbrace{\sum_{m=0}^{N-1} e^{\frac{2\pi i m(k-\ell)}{N}}}_{=\begin{cases}N, & \ell=k\\ 0, & \ell\neq k\end{cases}}$$
$$= F(k), \text{ for } k=0,1,\dots,N-1. \ □$$

Definition 5.6. The convolution $F*G$ of two sequences in Π_N is defined by
$$(F*G)(m) = \sum_{k=0}^{N-1} F(m-k)\cdot G(k).$$

Note: Some indices get outside of the set $\{0,1,\dots,N-1\}$, but we just use the periodicity of F and G to compute the corresponding values of $F(m-k)\cdot G(k)$.

Definition 5.7. The (ordinary) product of the sequences $F, G \in \Pi_N$ is defined by

$$(F \cdot G)(m) = F(m) \cdot G(m), \quad m \in \mathbb{Z}.$$

Theorem 5.8. For sequences F and G in Π_N we have the formulas

(i) $\widehat{(F \cdot G)} = \widehat{F} * \widehat{G}$,

(ii) $\widehat{(F * G)} = N \widehat{F} \cdot \widehat{G}$.

Proof. i) Let $H \in \Pi_N$ be defined by $H(m) = F(m) \cdot G(m)$, for $m \in \mathbb{Z}$. We compute the inverse Fourier transform of the sequence $\widehat{F} * \widehat{G}$ (denoted $\mathcal{F}^{-1}(\widehat{F} * \widehat{G})$):

$$\mathcal{F}^{-1}(\widehat{F} * \widehat{G})(m) = \sum_{n=0}^{N-1} e^{2\pi i m n/N} \cdot (\widehat{F} * \widehat{G})(n)$$

$$= \sum_{n=0}^{N-1} e^{2\pi i m n/N} \cdot \sum_{k=0}^{N-1} \widehat{F}(n-k) \cdot \widehat{G}(k)$$

$$= \sum_{k=0}^{N-1} \left(\sum_{n=0}^{N-1} e^{2\pi i m n/N} \cdot \widehat{F}(n-k) \right) \cdot \widehat{G}(k) = [n-k=\ell]$$

$$= \sum_{k=0}^{N-1} \left(\sum_{\ell=-k}^{N-k-1} e^{2\pi i m(k+\ell)/N} \widehat{F}(\ell) \right) \cdot \widehat{G}(k)$$

$$= \sum_{k=0}^{N-1} \underbrace{\left(\sum_{\ell=0}^{N-1} e^{2\pi i m \ell/N} \widehat{F}(\ell) \right)}_{=F(m)} \cdot e^{2\pi i m k/N} \cdot \widehat{G}(k)$$

$= F(m) \cdot G(m) = H(m)$, so $\widehat{(F \cdot G)} = \widehat{H} = \widehat{F} * \widehat{G}$.

ii) Use the direct $\mathcal{F}$-transform of $F * G$, homework. $\square$

<u>Definition</u>. A sequence $F \in \Pi_N$ is <u>even</u> if $F(-m) = F(m)$ for all $m \in \mathbb{Z}$, and <u>odd</u> if $F(-m) = -F(m)$ for all $m \in \mathbb{Z}$.

<u>Theorem</u>. If $\{\hat{F}(m)\} \in \Pi_N$ is the Fourier transform of $\{F(m)\} \in \Pi_N$, then:

(i) $\{\hat{F}(-m)\}$ is the Fourier transform of $\{F(-m)\}$,

(ii) $\{\overline{\hat{F}(-m)}\}$ ——— " ——— $\{\overline{F(m)}\}$,

(iii) $\{\overline{\hat{F}(m)}\}$ ——— " ——— $\{\overline{F(-m)}\}$.

<u>Proof</u>. Homework. □

<u>Theorem</u>. If $\{\hat{F}(m)\} \in \Pi_N$ is the Fourier transform of $\{F(m)\} \in \Pi_N$, then:

(i) F is even (odd) $\Leftrightarrow$ $\hat{F}$ is even (odd),

(ii) F is real $\Leftrightarrow$ $\hat{F}(-m) = \overline{\hat{F}(m)}$, $\forall m \in \mathbb{Z}$,

(iii) F is real and even $\Leftrightarrow$ $\hat{F}$ is real and even,

(iv) F is real and odd $\Leftrightarrow$ $\hat{F}$ is imaginary and odd.

<u>Proof</u>. The results are easily proved using the preceding theorem. □

<u>Note</u>: The results above also hold for the inverse Fourier transform (that is, $\hat{F}$ and F can change places in the theorems).

If we for the Fourier transform $\mathcal{F}$ in Def. 5.2 restrict ourself to $m = 0, \dots, N-1$, Then $\mathcal{F}$ can be interpreted as a linear operator from $\mathbb{C}^N$ to $\mathbb{C}^N$. Since

$$(\mathcal{F}F)(m) = \hat{F}(m) = \frac{1}{N}\Big(\underbrace{e^{-2\pi i m 0/N}}_{\omega_N^0}, \dots, \underbrace{e^{-2\pi i m (N-1)/N}}_{\omega_N^{-m(N-1)}}\Big) \cdot \begin{pmatrix} F(0) \\ F(1) \\ \vdots \\ F(m) \end{pmatrix}$$

we have the matrix representation Ω_N of $\mathcal{F}$:

$$\Omega_N = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N^{-1} & \omega_N^{-2} & \dots & \omega_N^{-(N-1)} \\ 1 & \omega_N^{-2} & \omega_N^{-4} & \dots & \omega_N^{-2(N-1)} \\ \vdots & & & & \\ 1 & \omega_N^{-(N-1)} & \omega_N^{-2(N-1)} & \dots & \omega_N^{-(N-1)^2} \end{bmatrix}.$$

The inverse operator $\mathcal{F}^{-1}$ in Theorem 5.5 has the matrix representation:

$$\Omega_N^{-1} = N \cdot \overline{\Omega}_N = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & \dots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \dots & \omega_N^{2(N-1)} \\ \vdots & & & & \\ 1 & \omega_N^{(N-1)} & \omega_N^{2(N-1)} & \dots & \omega_N^{(N-1)^2} \end{bmatrix}.$$

Thus $\mathcal{F}$ is also bijective.

5.2 The Fast Fourier Transform (=FFT)

Question 5.11. How many flops do we need to compute the Fourier transform $\hat{F}$ of $F \in \Pi_N$?

Flop = floating point operation (multiplication or addition).
1 Megaflop = 1 million flops/second (10^6),
1 Gigaflop = 1 billion flops/second (10^9).

Task 5.12. Compute $\hat{F}(m) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} e^{-2\pi i m k/N} \cdot F(k)$ with the minimum amount of flops (=quickly)

Denote now $\omega := e^{-2\pi i/N}$ and compute the coefficients $\omega^k = e^{-2\pi i k/N}$ only once, and store them in a table. Since $\omega^{k+N} = \omega^k$, we have

$$e^{-2\pi i m k/N} = \omega^{mk} = \omega^r,$$

where r is the remainder when we divide mk by N. Thus, only N numbers need to be stored. We can ignore the number of flops needed to compute the coefficients ω^{mk} (done in advance).

Trivial Solution 5.14. We need to compute N coefficients $\hat{F}(m)$ and each $\hat{F}(m)$ requires N multiplications and $N-1$ additions, which totals:

$$N(2N-1) = 2N^2 - N \approx 2N^2 \text{ flops}.$$

This number grows too fast with increasing N.

Fast Solution 5.15. Regroup the terms, using the symmetry. Start by doing even and odd coefficients separately:
Suppose for simplicity that N is even, $N = 2n$.
Then, for even m:

$$\hat{F}(2m) = \frac{1}{N}\sum_{k=0}^{N-1} w^{2mk}\cdot F(k)$$

$$= \frac{1}{N}\left(\sum_{k=0}^{n-1} w^{2mk}\cdot F(k) + \sum_{k=n}^{2n-1} w^{2mk} F(k)\right)$$

$$= \frac{1}{N}\left(\sum_{k=0}^{n-1}\left(w^{2mk}\cdot F(k) + w^{2m(k+n)}\cdot F(k+n)\right)\right)$$

$$= \frac{2}{N}\cdot\sum_{k=0}^{n-1} e^{-2\pi i mk/(N/2)}\cdot\frac{1}{2}\left(F(k)+F(k+n)\right)$$

This is a new discrete time periodic Fourier transform of the sequence $G(k) = \frac{1}{2}[F(k)+F(n+k)]$ with period $n = N/2$. For odd m:

$$\hat{F}(2m+1) = \frac{1}{N}\cdot\sum_{k=0}^{N-1} w^{(2m+1)\cdot k}\cdot F(k)$$

$$= \frac{1}{N}\left(\sum_{k=0}^{n-1} w^{2mk}\cdot w^{k}\cdot F(k) + \sum_{k=n}^{N-1} w^{2mk}\cdot w^{k}\cdot F(k)\right)$$

$$= \frac{1}{N}\cdot\sum_{k=0}^{n-1}\left(w^{2mk}\cdot w^{k}\cdot F(k) + w^{2m(k+n)}\cdot w^{(k+n)}\cdot F(k+n)\right)$$

$$= \frac{2}{N}\cdot\sum_{k=0}^{n-1} e^{-2\pi i mk/(N/2)}\cdot\frac{1}{2}\cdot e^{-i\pi k/(N/2)}\cdot\left(F(k) + \underbrace{w^{(2m+1)n}}_{-1}\cdot F(k+n)\right)$$

$$= \frac{2}{N}\sum_{k=0}^{n-1} e^{-2\pi i mk/(N/2)}\cdot\frac{1}{2}\cdot e^{-i\pi k/n}\cdot\left(F(k) - F(k+n)\right)$$

$$w^{(2m+1)n} = e^{-2\pi i(2m+1)n/2n} = e^{-\pi i(2m+1)} = -1$$

Thus we obtained a new discrete time periodic Fourier transform of the sequence $H(k) = \frac{1}{2} \cdot e^{-i\pi k/n} \cdot (F(k) - F(k+n))$ with period $n = \frac{N}{2}$.
Instead of one transform of order N we get two transforms of order $n = \frac{N}{2}$.

Number of flops: Computing the new transforms as in 5.14, page 78, we need the following flops:

Even: $n(2n-1) = \frac{N^2}{2} - \frac{N}{2}$ + n additions $= \frac{N^2}{2}$ flops.

Odd: Essentially the same amount, $\frac{N^2}{2} + \frac{N}{2}$ (n extra multiplications.

Total: $N^2 + \frac{N}{2} \approx N^2$.

Thus, this approximately halved the number of needed flops.

Repeat: Divide the new smaller intervals into two halfs, again and again. This is possible if $N = 2^k$ for some integer k, e.g., $N = 1024 = 2^{10}$.

Final conclusion: We get down to approximately

$$\frac{3}{2} \cdot 2^k \cdot k \text{ flops},$$

where $N = 2^k$, so $k = {}^2\log N$. Thus we obtain the following theorem for the Fast Fourier transform with radius 2:

Theorem 5.17. The Fast Fourier Transform with radius 2 outlined above needs approximately $\frac{3}{2} N \cdot {}^2\log N$ flops.

This is much smaller than $2N^2 - N$ for large N. For example if $N = 2^{10} = 1024$ we have

$$\frac{3}{2} \cdot N \cdot {}^2\log N \approx 15000 \ll 2000000 = 2N^2 - N.$$

Definition 5.18. Fast Fourier transform with

radius 2: split into 2 parts at each step $N = 2^k$
radius 3: " 3 " $N = 3^k$
radius m: " m " $N = m^k$

The FFT, presented by Cooley and Tukey in 1965, has had an immense impact on several branches of numerical analysis. It made it possible to compute Fourier transforms in practice.

The next example demonstrates how the FFT works in Matlab.

Example. Let $f \in C(T)$ be given by

$$f(x) = 1 + 2\cos(2\pi x) + 8 \cdot \sin(4\pi x) - 5 \cdot \cos(6\pi x).$$

We choose $N = 8$ and compute the vector y:

$$y_j = f(j/N), \quad j = 0, \ldots, 7,$$

$y = (-2, 13.94975, 1.0, -11.94975, 4.0, 4.05025, 1.0, -2.05025)$

Note: Matlab vectors are indexed starting with 1. The Fourier transform does not have $\frac{1}{N}$ in front of the sum, we have to divide the result by N.

Matlab: $X(k+1) = \sum_{n=0}^{N-1} y(n+1) \cdot e^{-2\pi i k n/N}$.

```
>> x=[0:1/8:1-1/8]

   x = 0     0.1250     0.2500     0.3750     0.5000     0.6250     0.7500
      0.8750

>> y=1+2*cos(2*pi*x)+8*sin(4*pi*x)-5*cos(6*pi*x)

   y = -2.0000    13.9497     1.0000   -11.9497     4.0000     4.0503
        1.0000    -2.0503

>> X=fft(y)

X = 8.00 8.00 -32.00i -20.00 0.00 -20.00 32.00i 8.00

>> c=fftshift(X)/8

c = 0.00 -2.50 4.00i 1.00 1.00 1.00 -4.00i -2.50
```

$C = [C_{-4}, C_{-3}, C_{-2}, C_{-1}, C_0, C_1, C_2, C_3]$

$$\begin{cases} \frac{a_0}{2} = C_0 = 1, \quad a_1 = C_1 + C_{-1} = 1 + 1 = 2 \\ a_2 = C_2 + C_{-2} = -4i + 4i = 0 \\ a_3 = C_3 + C_{-3} = -2{,}5 - 2{,}5 = -5 \end{cases}$$

$$\begin{cases} b_1 = i(C_1 - C_{-1}) = i(1-1) = 0 \\ b_2 = i(C_2 - C_{-2}) = i(-4i - 4i) = 8 \\ b_3 = i(C_3 - C_{-3}) = i(-2{,}5 - (-2{,}5)) = 0 \end{cases}$$

$$\therefore \underline{S_3} = \frac{a_0}{2} + \sum_{n=1}^{3} \left(a_n \cdot \cos(2\pi n t) + b_n \sin(2\pi n t)\right)$$

$$= \underline{1 + 2 \cdot \cos(2\pi t) + 8 \cdot \sin(4\pi t) - 5 \cdot \cos(6\pi t)}.$$

5.3 Computation of the Fourier Coefficients of a Periodic Function

We saw in section 5.1 that the coefficients of a periodic function that is a trigonometric polynomial can be obtained exactly using the trapezoidal rule. We will now investigate the error in the approximate Fourier coefficients of a function $f \in C(\mathbb{T})$.

Problem 5.19. Let $f \in C(\mathbb{T})$. Compute approximations to
$$\hat{f}(k) = \int_0^1 e^{-2\pi i k t} \cdot f(t)\,dt$$
as efficiently as possible.

To solve the problem we start by converting f into a periodic sequence in Π_N:

Conversion 5.20. Choose some $N \in \mathbb{Z}_+$ and define
$$F(m) = f\left(\frac{m}{N}\right), \quad \text{for } m \in \mathbb{Z},$$
(equidistant "sampling"). The periodicity of f makes F periodic with period N, $F \in \Pi_N$.

The next theorem gives an error estimate for $\hat{F}(m) - \hat{f}(n)$, where $\hat{F}$ is the discrete Fourier transform of the sequence F.

Theorem 5.21. (Error estimate). If $f \in C(\mathbb{T})$ and $\hat{f} \in \ell^1(\mathbb{Z})$, i.e. $\sum_{-\infty}^{\infty} |\hat{f}(k)| < \infty$, then

$$\hat{F}(m) - \hat{f}(m) = \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \hat{f}(m + k \cdot N).$$

Proof. Since $\sum_{-\infty}^{\infty} |\hat{f}(k)| < \infty$ Lemma 1.14 gives that the Fourier series converges uniformly to f, so for all t we have

$$f(t) = \sum_{j=-\infty}^{\infty} e^{2\pi i j t} \cdot \hat{f}(j).$$

Particularly for $t_k = k/N$ we have

$$f(t_k) = F(k) = \sum_{j=-\infty}^{\infty} e^{2\pi i k j/N} \cdot \hat{f}(j).$$

The definition of $\hat{F}$ gives:

$$\underline{\hat{F}(m)} = \frac{1}{N} \sum_{k=0}^{N-1} e^{-2\pi i m k/N} \cdot F(k)$$

$$= \frac{1}{N} \cdot \sum_{k=0}^{N-1} e^{-2\pi i m k/N} \cdot \sum_{j=-\infty}^{\infty} e^{2\pi i k j/N} \cdot \hat{f}(j)$$

$$\left(\sum_{-\infty}^{\infty} \text{ uniformly convergent} \right) = \frac{1}{N} \cdot \sum_{j=-\infty}^{\infty} \hat{f}(j) \underbrace{\left(\sum_{k=0}^{N-1} e^{2\pi i (j-m) k/N} \right)}_{= \begin{cases} N, & \text{if } (j-m)/N \text{ integer,} \\ 0, & \text{otherwise.} \end{cases}}$$

$$= \underline{\sum_{\ell=-\infty}^{\infty} \hat{f}(m + N \cdot \ell)}.$$

Thus we have

$$\hat{F}(m) = \hat{f}(m) + \sum_{\substack{l=-\infty \\ l \neq 0}}^{\infty} \hat{f}(m+N\cdot l). \quad \square$$

<u>Note</u>. If N is "large" and if $\hat{f}(m) \to 0$ "quickly" when $m \to \pm\infty$, then the error $\sum_{\substack{l=-\infty \\ l \neq 0}}^{\infty} \hat{f}(m+N\cdot l) \approx 0$.

<u>Corollary</u>. For a trigonometric polynomial of degree P,

$$f(t) = \sum_{n=-P}^{P} \hat{f}(n) \cdot e^{2\pi i n t},$$

the values of the approximate coefficients $\hat{F}(m)$ are exact when $N \geq 2P-1$. !

<u>Proof</u>. We have by Theorem 5.21:

$$\hat{F}(m) - \hat{f}(m) = \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \hat{f}(m+k\cdot N).$$

Assume now that $|m| \leq P$, (since $|\hat{f}(m)| = 0$ for $|m|>P$).

Then for $N \geq 2P+1$: $\begin{cases} m+k\cdot N \geq P+1, & \text{when } k \geq 1, \\ m+k\cdot N \leq -(P+1), & \text{when } k \leq -1, \end{cases}$

$$\Longrightarrow \quad \underline{\hat{F}(m) = \hat{f}(m), \quad \text{when } |m| \leq P \text{ and } N \geq 2P+1.} \quad \square$$

First Method 5.22. Suppose N is even and approximate:

i) $\hat{f}(m) \approx \hat{F}(m)$, if $|m| < N/2$,

ii) $\hat{f}(m) \approx \hat{F}(m)$, if $m = -N/2$,

iii) $\hat{f}(m) \approx 0$, if $|m| > N/2$ and $m = N/2$.

(Here $\hat{F}(m) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} e^{-2\pi i m k/N} \cdot F(k)$)

Error for the first method:

i) $|m| < N/2$: $|\hat{f}(m) - \hat{F}(m)| \leq \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} |\hat{f}(m + k \cdot N)|$,

ii) $m = -N/2$: error $= |\hat{f}(m) - \hat{F}(m)|$,

iii) $|m| > N/2$ and $m = N/2$: error $= |\hat{f}(m)|$,

We obtain the crude estimate:

$$\sup_{m \in \mathbb{Z}} |\hat{f}(m) - \hat{F}(m)| \leq \sum_{|m| \geq N/2} |\hat{f}(m)| \qquad (5.5)$$

First Method 5.22, Drawbacks:

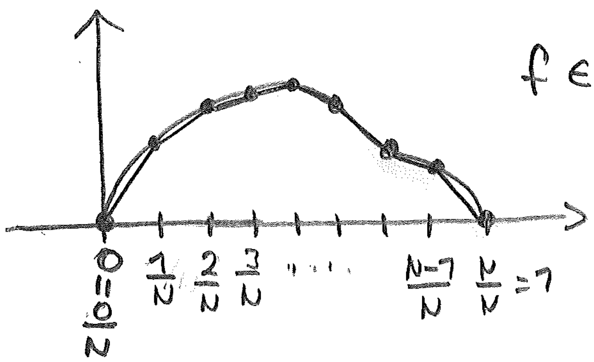
1) Large error, (If N is "small").
2) Inaccurate error estimate (5.5).
3) Error estimate based on $\hat{f}$ and not on f.

∴ We will investigate another method.

In the first Method 5.22 we applied the trapezoidal rule to the whole integrand when computing approximations to the Fourier coefficients

$$\hat{f}(n) = \int_0^1 f(t)\cdot e^{-2\pi i n t}\,dt, \qquad (*)$$

which meant that we approximated $Re(f(t)\cdot e^{-2\pi i n t})$ and $Im(f(t)\cdot e^{-2\pi i n t})$ with piecewise linear polynomials. In the second method we approximate just $f(t)$ with a piecewise linear polynomial using the points $y_k = f(k/N)$, $k = 0, 1, \dots, N$. $(y_0 = y_N)$.



Let $p(t)$ be the piecewise linear approximation of f. Then we can compute the Fourier coefficients $\hat{p}(n)$ and take them as approximations of $\hat{f}(n)$.

$$\hat{f}(n) \approx \hat{p}(n) = \int_0^1 p(t)\cdot e^{-2\pi i n t}\,dt$$

$$= \sum_{k=0}^{N-1} \int_{k/N}^{(k+1)/N} (a_k + b_k\cdot t)\cdot e^{-2\pi i n t}\,dt,$$

where

$$\begin{cases} a_k = (1+k)\cdot y_k - k\cdot y_{k+1}, \\ b_k = N\cdot(y_{k+1} - y_k), \qquad k = 0, \dots, N-1. \end{cases}$$

Computing the integrals we obtain, $(n \neq 0)$,

$$\hat{P}(n) = \mathrm{I} + \mathrm{II}$$

where

$$\underline{\underline{\mathrm{I}}} = \frac{N}{4n^2\pi^2}\left(e^{2\pi i n/N} - 1\right) \cdot \Big(\overbrace{\sum_{k=0}^{N-1} y_k \cdot e^{-2\pi i (k+1) n/N}}^{= e^{-2\pi i n/N} \cdot N \cdot \hat{F}(n)} - \underbrace{\sum_{k=0}^{N-1} y_{k+1} \cdot e^{-2\pi i (k+1) n/N}}_{= N \cdot \hat{F}(n)} \Big) = \underline{\underline{\frac{N^2}{4n^2\pi^2}\left(e^{2\pi i n/N} - 1\right)\left(e^{-2\pi i n/N} - 1\right) \cdot \hat{F}(n)}}$$

$$\underline{\underline{\mathrm{II}}} = -\frac{i}{2n\pi} \cdot \Big(\underbrace{\sum_{k=0}^{N-1} y_k \cdot e^{-2\pi i k n/N}}_{N \cdot \hat{F}(n)} - \underbrace{\sum_{k=0}^{N-1} y_{k+1} \cdot e^{-2\pi i (k+1) n/N}}_{N \cdot \hat{F}(n)} \Big) \underline{\underline{= 0.}}$$

Thus we have

$$\underline{\underline{\hat{P}(n)}} = \mathrm{I} = \frac{N^2}{4n^2\pi^2} \cdot \left(1 - e^{2\pi i n/N} - e^{-2\pi i n/N} + 1\right) \cdot \hat{F}(n)$$

$$= \frac{N^2}{2n^2\pi^2} \cdot \left(1 - \mathrm{Re}\left(e^{2\pi i n/N}\right)\right) \cdot \hat{F}(n)$$

$$= \frac{N^2}{2n^2\pi^2} \cdot \underbrace{\left(1 - \cos(2\pi n/N)\right)}_{2 \cdot \sin^2(\pi n/N)} \cdot \hat{F}(n)$$

$$= \frac{N^2 \cdot \sin^2(\pi n/N)}{n^2 \pi^2} \cdot \hat{F}(n)$$

$$\underline{\underline{= \left(\frac{\sin(\pi n/N)}{\pi \cdot n/N} \right)^2 \cdot \hat{F}(n), \quad n \neq 0.}}$$

$$\underline{\underline{\hat{P}(0) = \hat{F}(0).}}$$

Theorem 5.30. If we discretize f by the sequence $F(k) = f(k/N)$, $k = 0, \dots, N-1$, then compute $\hat{F}(m)$ and finally multiply $\hat{F}(m)$ by

$$\left(\frac{\sin(\pi m/N)}{\pi m/N}\right)^2,$$

then we get the Fourier coefficients $\hat{P}(m)$ of the function $P(t)$ which we get from f by linear interpolation at the points $t_k = k/N$,

$$\begin{cases} \hat{P}(m) = \left(\frac{\sin(\pi m/N)}{\pi m/N}\right)^2 \cdot \hat{F}(m), & m = 1, \dots, N-1, \\ \hat{P}(0) = \hat{F}(0). \end{cases}$$

Second Method 5.28. Construct F as in the First Method 5.22 and compute $\hat{F}$, (for instance with the FFT). Then the approximation of $\hat{f}(m)$ is

$$\begin{cases} \hat{f}(m) \approx \hat{P}(m) = \left(\frac{\sin(\pi m/N)}{\pi m/N}\right)^2 \cdot \hat{F}(m), \\ \hat{f}(0) \approx \hat{P}(0) = \hat{F}(0). \end{cases} \quad 1 \leq m \leq N-1,$$

where $P(t)$ is the piecewise linear function that interpolates f at the points $t_k = k/N$, $0 \leq k \leq N$.

We can obtain an error estimate based on f and P:

$$|\hat{f}(n) - \hat{P}(n)| = \left|\int_T (f(t) - P(t))\, e^{-2\pi i n t}\, dt\right|$$

$$\leq \int_T |f(t) - P(t)| \cdot 1 \cdot dt \leq \sup_{t \in T} |f(t) - P(t)| \cdot \int_T 1 \cdot dt$$

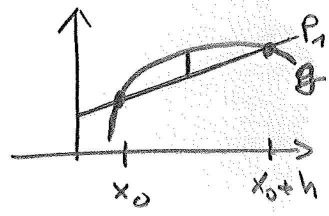
so

$$|\hat{f}(n) - \hat{P}(n)| \leq \sup_{t \in T} |f(t) - P(t)|. \qquad (5.6)$$

Suppose now that $f \in C^2(\mathbb{T})$. For a function $g \in C^2(I)$, where $I = [x_0, x_0+h]$ we have the following error estimate for linear interpolation:

$$\max_{x \in I} |g(x) - P_1(x)| \leq \max_{\xi \in I} \frac{|g''(\xi)|}{2!} \cdot \max_{x \in I} |(x-x_0)(x-(x_0+h))|$$

$$\leq \frac{h^2}{8} \cdot \max_{x \in I} |g''(x)|.$$

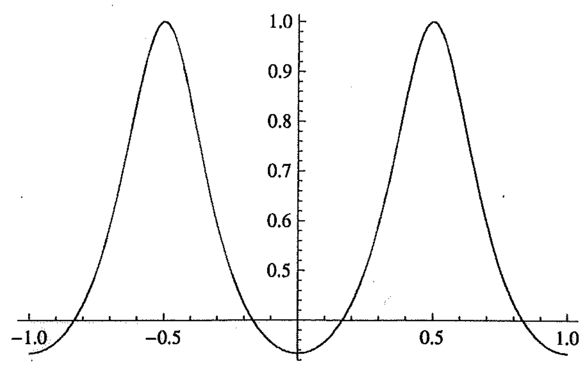


Thus, if $f \in C^2(\mathbb{T})$ we obtain an error estimate based on f'' and N:

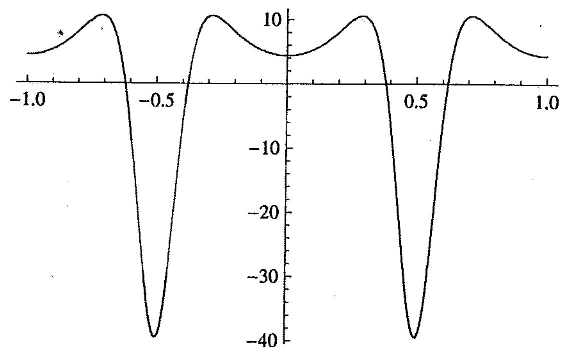
$$|\hat{f}(n) - \hat{p}(n)| \leq \frac{1}{8N^2} \cdot \sup_{t \in \mathbb{T}} |f''(t)| . \qquad (5.7)$$

Example. Let $f(t) = \dfrac{1}{2+\cos(2\pi t)}$ on $\mathbb{T}$. Then $f \in C^2(\mathbb{T})$ and $|f''(t)| \leq |f''(1/2)| = 4\pi^2$.

`Plot[f[t], {t, -1, 1}]`



`Plot[f''[t], {t, -1, 1}]`



Mathematica: NIntegrate gives the following approximations of $\hat{f}(0), \hat{f}(1)$ and $\hat{f}(2)$:

$\hat{f}(0) \approx 0.57735$, $\hat{f}(1) \approx -0.154701$ and $\hat{f}(2) \approx 0.041452$.

We take a numerical experiment in Matlab using the FFT with $N = 6, 8, 16, 32, 64, 128$ to compute approximations to $\hat{f}(0)$, $\hat{f}(1)$ and $\hat{f}(2)$ with First Method 5.22 and Second Method 5.28:

N	$\lvert\hat{f}(0)-\hat{F}(0)\rvert$	$\lvert\hat{f}(1)-\hat{F}(1)\rvert$	$\lvert\hat{f}(2)-\hat{F}(2)\rvert$	$\lvert\hat{f}(1)-\hat{P}(1)\rvert$	$\lvert\hat{f}(2)-\hat{P}(2)\rvert$	$\frac{\pi^2}{2N^2}$ (5.7)
6	0.000428	0.000855	0.002992	0.012857	0.011056	0.137078
8	$3 \cdot 10^{-5}$	$6 \cdot 10^{-5}$	$2 \cdot 10^{-4}$	0.007733	0.007678	0.077106
16	$3 \cdot 10^{-7}$	$5 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	0.001978	0.002088	0.019277
32	— " —	— " —	— " —	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.004819
64	— " —	— " —	— " —	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	0.001205
128	— " —	— " —	— " —	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	0.000307
	First Method 5.22			Second Method 5.28		

This example does not advocate the use of Second Method 5.28, if our goal is to compute good approximations of the Fourier coefficients $\hat{f}(n)$ to f!

Practical advice: If we want to compute approximations to $\hat{f}(-m), \dots, \hat{f}(0), \dots, \hat{f}(m)$ for a fixed m, apply First Method 5.22:

1°) Choose k_0 so that $N_{k_0} = 2^{k_0} > 2 \cdot m$

2°) Compute the FFT for $N_k = 2^k$, $k = k_0, k_0+1, \dots$ obtaining the approximations $\hat{F}_k(-m), \dots, \hat{F}_k(0), \dots, \hat{F}_k(m)$.

3°) Stop when $\max_{-m \le j \le m} \lvert \hat{F}_k(j) - \hat{F}_{k+1}(j) \rvert < \varepsilon$, for some suitably chosen precision $\varepsilon > 0$.

Suppose now that we have a function $f \in C(\mathbb{T})$ such that $\sum_{-\infty}^{\infty} |\hat{f}(n)| < \infty$. Then by Lemma 1.14 for all $t \in \mathbb{R}$:

$$f(t) = \sum_{m=-\infty}^{\infty} \hat{f}(m) \cdot e^{2\pi i m t}.$$

Choose $N = 2^k$, for some $k \in \mathbb{Z}_+$, and compute the sequence $\hat{F}(m)$, $m = 0, \ldots, N-1$, with the FFT. We know that the trigonometric polynomial

$$g(t) = \sum_{m=-N/2}^{N/2-1} c_m \cdot e^{2\pi i m t}$$

that interpolates f at $t_k = \frac{k}{N}$, $k = 0, \ldots, N-1$, has the Fourier coefficients $c_m = \hat{F}(m)$. Thus

$$g(t) = \sum_{m=-N/2}^{N/2-1} \hat{F}(m)\, e^{2\pi i m t} \underset{\text{Thm 5.27.}}{=} \sum_{m=-N/2}^{N/2-1} \Big(\hat{f}(m) + \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \hat{f}(m+k\cdot N) \Big) \cdot e^{2\pi i m t}.$$

$$= f(t) - \sum_{|m| > N/2} \hat{f}(m) \cdot e^{2\pi i m t} - \hat{f}(N/2) \cdot e^{2\pi i N/2 \cdot t}$$

$$+ \sum_{m=-N/2}^{N/2-1} \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \hat{f}(m+k\cdot N) \cdot e^{2\pi i m t},$$

So we obtain

$$|g(t) - f(t)| \leq \sum_{|m| > N/2} |\hat{f}(m)| + |\hat{f}(N/2)| + \sum_{m=-N/2}^{N/2-1} \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} |\hat{f}(m+k\cdot N)|$$

$$= 2 \cdot \Big(\sum_{|m| > N/2} |\hat{f}(m)| + |\hat{f}(N/2)| \Big).$$

Therefore we can formulate the following theorem on approximation of a periodic function by a trigonometric polynomial:

Theorem 5.33. Suppose that $f \in C(T)$ and that $\sum_{m=-\infty}^{\infty} |\hat{f}(m)| < \infty$. If $F(m)$, $m = 0, \dots, N-1$, are the approximations of $\hat{f}(m)$ computed by the FFT with N even ($N = 2^k$), then

$$|g(t) - f(t)| \leq 2 \cdot \left(\sum_{|m| > N/2} |\hat{f}(m)| + |\hat{f}(N/2)| \right)$$

for all $t \in \mathbb{R}$, where

$$g(t) = \sum_{m=-N/2}^{N/2-1} F(m) \cdot e^{2\pi i m t}.$$

Note: If $|\hat{f}(m)| \to 0$ rapidly we have a good approximation by $g(t)$. Better accuracy is achieved by increasing N.

5.4 One-Sided Sequences

We have treated periodic sequences in Π_N. In applications we are often interested in

A) Finite sequences: $A(0), A(1), \dots, A(N-1)$,

or

B) One-sided sequences: $A(n)$, $n \in \mathbb{Z}_+ = \{0, 1, 2, \dots\}$.

Notes: A finite sequence is a special case of a one-sided sequence: put $A(n) = 0$ for $n \geq N$.
A one-sided sequence is a special case of a two-sided sequence: put $A(n) = 0$ for $n < 0$.

Problem: These sequences are not periodic, so we cannot use FFT directly.

Notation: $\mathbb{C}^{\mathbb{Z}_+} = \{$all complex valued sequences $A(n), n \in \mathbb{Z}_+\}$

Definition 5.39. The convolution of two sequences $A, B \in \mathbb{C}^{\mathbb{Z}_+}$ is defined by

$$(A * B)(m) = \sum_{k=0}^{m} A(m-k) \cdot B(k), \quad m \in \mathbb{Z}_+.$$

Notation 5.40: The restriction of the sequence $A(k)$ to its first n terms is denoted by

$$A_{|n}(k) = \begin{cases} A(k), & 0 \le k < n, \\ 0, & k \ge n. \end{cases}$$

Lemma 5.41. $(A * B)_{|n} = (A_{|n} * B_{|n})_{|n}$

Proof. Suppose that $0 \le j < n$:

$$(A * B)(j) = \sum_{k=0}^{j} A(j-k) \cdot B(k) = \left[\begin{matrix} 0 \le k \le j < n \\ 0 \le j-k \le j < n \end{matrix} \right]$$

$$= \sum_{k=0}^{j} A_{|n}(j-k) \cdot B_{|n}(k) = (A_{|n} * B_{|n})(j)$$

Thus

$$(A * B)_{|n} = (A_{|n} * B_{|n})_{|n}. \quad \square$$

Notation: $A = O_n$ means that $A(k) = 0$ for $0 \le k < n$, that is $A|_n = 0$.

Lemma 5.43. If $A = O_n$ and $B = O_m$, then $A * B = O_{n+m}$.

Proof. Suppose $0 \le j < n+m$:

$$(A * B)(j) = \sum_{k=0}^{j} A(j-k)B(k) = \left[\begin{array}{l} \text{if } 0 \le k < m : B(k) = 0 \\ \text{if } m \le k \le j : 0 \le j-k < n, \\ \qquad \text{So } A(j-k) = 0 \end{array}\right.$$

$$= 0.$$

$\therefore$ $A * B = O_{n+m}$. $\square$

Computation of $A * B$ 5.44. (One-sided convolution)

We wish to compute $(A * B)|_n$:

1) Choose a number $N \ge 2n$ $(N = 2^k)$

2) Define

$$F(k) = \begin{cases} A(k), & 0 \le k < n, \\ 0, & n \le k < N, \end{cases}$$

and extend F to be periodic with period N.

3) Define

$$G(k) = \begin{cases} B(k), & 0 \le k < n, \\ 0, & n \le k < N, \end{cases}$$

and extend periodically: $G(k+N) = G(k)$.

Then, for all m, $0 \le m < n$:

$$\underbrace{(F * G)(m)}_{\text{periodic convolution}} = \sum_{k=0}^{N-1} F(m-k)\cdot G(k) = \left[\begin{array}{l} F(m-k)=0 \text{ if} \\ m-k=-1,-2,-3,\dots \\ -(N-n). \end{array}\right]$$

$$= \sum_{k=0}^{m} F(m-k)\cdot G(k)$$

$$= \sum_{k=0}^{m} A(m-k)\cdot B(k)$$

$$= (A * B)(m), \text{ one-sided convolution}$$

Theorem 5.45. We can compute $(A * B)|_n$ using

$$C \cdot n \cdot {}^2\log n, \quad C \text{ constant},$$

Flops by the following method:

Steps 1), 2) and 3) as in 5.44 above.

4) Use FFT to compute:

$$\hat{F} \cdot \hat{G} \left(= \frac{1}{N} \widehat{(F * G)} \right), \text{ (Theorem 5.8 (ii))}.$$

5) Use the inverse FFT to compute:

$$F * G \left(= N \cdot \mathcal{F}^{-1}(\hat{F} \cdot \hat{G}) \right).$$

Notes: A "naive" computation of $(A * B)|_n$ requires $C_1 \cdot n^2$ Flops, for some constant C_1. Use "naive" method for "small" n and "FFT-inverse FFT" for "large" n.

Example. Let $A=[1,2,3,4,5,6,7]$ and $B=[2,4,8,10,12,14]$. We want to compute $(A*B)_{16}$ using Matlab with the method described in Theorem 5.45.

```
>> A=[1 2 3 4 5 6 7];B=[2 4 8 10 12 14];n=6;N=16;   1)

>> F=[A(1:6) zeros(1,10)];G=[B(1:6) zeros(1,10)];   2),3)

>> Fhat=fft(F)/N;Ghat=fft(G)/N;   4)

>> AstarB=N*(ifft(Fhat.*Ghat)*N)   5)

AstarB = 2   8   22   46   82   132   168   188   176   142   84   0   0   0   0   0
```

$\underbrace{2 \;\; 8 \;\; 22 \;\; 46 \;\; 82 \;\; 132}_{(A*B)_{16}}$

```
>> help conv
   CONV Convolution and polynomial multiplication.
    C = CONV(A, B) convolves vectors A and B.  The resulting
    vector is length LENGTH(A)+LENGTH(B)-1.
    If A and B are vectors of polynomial coefficients, convolving
    them is equivalent to multiplying the two polynomials.


>> conv(A,B)

ans =   2   8   22   46   82   132   182   216   232   212   168   98

>> n=12;N=32;F=[A(1:7) zeros(1,25)];G=[B(1:6) zeros(1,26)];   1),2),3)

>> Fhat=fft(F)/N;Ghat=fft(G)/N;   4)

>> AstarB=N*(ifft(Fhat.*Ghat)*N)   5)

AstarB= 2   8   22   46   82   132   182   216   232   212   168   98
        0   0   0    0    0    0     0     0     0     0     0     0
        0   0   0    0    0    0     0     0
```

$\overbrace{2 \;\; 8 \;\; 22 \;\; 46 \;\; 82 \;\; 132 \;\; 182 \;\; 216 \;\; 232 \;\; 212 \;\; 168 \;\; 98}^{(A*B)_{12}}$

Notes: The Matlab-command conv can be used to compute convolutions. Observe again that Matlab does not have $\frac{1}{N}$ in front of the FFT, but has $\frac{1}{N}$ in front of the inversion formula ifft, contrary to our definitions.

5.5 The Polynomial Interpretation of a Finite Sequence

Problem 5.46. Compute the product of two polynomials:

$$p(x) = \sum_{k=0}^{n} a_k x^k \quad \text{and} \quad q(x) = \sum_{\ell=0}^{m} b_\ell x^\ell .$$

We have

$$p(x)\cdot q(x) = \left(\sum_{k=0}^{n} a_k x^k\right)\cdot\left(\sum_{\ell=0}^{m} b_\ell x^\ell\right)$$

$$= \sum_{j=0}^{m+n}\left(\sum_{\ell=0}^{j} a_{j-\ell}\cdot b_\ell\right)\cdot x^j$$

$$= \sum_{j=0}^{m+n} c_j \cdot x^j, \quad \text{where } c_j = \sum_{\ell=0}^{j} a_{j-\ell}\cdot b_\ell .$$

Theorem 5.47. i) Multiplication of two polynomials corresponds to a convolution of their coefficients:
If $p(x) = \sum_{k=0}^{n} a_k x^k$, $q(x) = \sum_{\ell=0}^{m} b_\ell x^\ell$, then

$$p(x)\, q(x) = \sum_{j=0}^{m+n} c_j \cdot x^j, \quad \text{where } c = a * b .$$

ii) Addition of two polynomials corresponds to addition of the coefficients:

$$p(x) + q(x) = \sum_{j=0}^{\max(m,n)} c_j \cdot x^j, \quad \text{where } c_j = a_j + b_j .$$

iii) Multiplication of a polynomial by a complex constant corresponds to multiplication of the coefficients by the same constant.

So we have:

Operation	Polynomial	Coefficients
Addition	$p(x)+q(x)$	$\{a_k+b_k\}_{k=0}^{\max(m,n)}$
Multiplication by $\lambda \in \mathbb{C}$	$\lambda \cdot p(x)$	$\{\lambda a_k\}_{k=0}^{n}$
Multiplication	$p(x)\,q(x)$	$(a * b)(k)$

Therefore there is a one-to-one correspondence between polynomials and finite sequences, which is used in computer computations of polynomials.

Note: Two conventions are in use: $p(x)=\sum_{j=0}^{n} a_j \cdot x^j$

1) first coefficient in the sequence is a_0,

2) —————— " —————— a_n.

Matlab uses convention 2).

Example: $p(x)=1+2x+3x^2+4x^3$, $q(x)=2-3x+5x^2$

```
>> p=[4 3 2 1];q=[5 -3 2];

>> r=conv(p,q)

   r = 20     3     9     5     1     2

>> P=[1 2 3 4 0 0 0 0];Q=[2 -3 5 0 0 0 0 0];

>> 8*(ifft((fft(P)/8).*(fft(Q)/8))*8)

ans = 2.0    1.0    5.0    9.0    3.0    20.0    0.0    0.0

>> P=[1 2 3 4 0 0];Q=[2 -3 5 0 0 0];

>> 6*(ifft((fft(P)/6).*(fft(Q)/6))*6)

ans = 2.0    1.0    5.0    9.0    3.0    20.0

>> [kvot,rest]=deconv(r,q)

kvot = 4.0    3.0    2.0    1.0
rest = 1.0e-14 * [0   -0.1776    0    0    0   -0.0444].
```

5.6 Formal Power Series and Analytic Functions

Definition 5.48. A Formal Power Series (FPS) is a sum of the type

$$\sum_{k=0}^{\infty} A(k) \cdot x^k$$

which need not converge for any $x \neq 0$. If it does converge it defines an analytic function in the region of convergence.

Example. a) $\sum_{k=0}^{\infty} \frac{x^k}{k!}$ converges for all $x \in \mathbb{C}$, the sum is e^x.

b) $\sum_{k=0}^{\infty} x^k$ converges for $|x|<1$, sum $= \frac{1}{1-x}$.

c) $\sum_{k=0}^{\infty} k! \cdot x^k$ converges for no $x \neq 0$.

All examples are formal power series. In cases a) and b) the power series define analytic functions in the region of convergence.

Notation. We denote

$$\tilde{A}(x) = \sum_{k=0}^{\infty} A(k) \cdot x^k.$$

Calculus with FPS 5.52. We imitate the rules for polynomials:

i) Addition of two FPS:s

$$\left(\sum_{k=0}^{\infty} A(k)\cdot x^k\right) + \left(\sum_{k=0}^{\infty} B(k)\cdot x^k\right) = \sum_{k=0}^{\infty} (A(k)+B(k))\cdot x^k.$$

ii) Multiplication of a FPS by a constant λ:

$$\lambda\cdot \sum_{k=0}^{\infty} A(k)\cdot x^k = \sum_{k=0}^{\infty} (\lambda A(k))\cdot x^k.$$

iii) Multiplication of two FPS:s by taking the convolution of the coefficients:

$$\left(\sum_{k=0}^{\infty} A(k)\cdot x^k\right)\left(\sum_{k=0}^{\infty} B(k)\cdot x^k\right) = \sum_{k=0}^{\infty} C(k)\cdot x^k,$$

where $C = A * B$.

Conclusion 5.54. There is a one-to-one correspondance between all FPS:s and all one-sided sequences $(= \mathbb{C}^{\mathbb{Z}_+})$.

Note. To compute $C_{IN} = [C(0), C(1), \dots, C(N-1), 0, 0, 0, \dots]$ we can compute $(A * B)_{IN}$ as described in 5.44 on page 95.

Problem 5.56. Given a FPS $\tilde{A}(x) = \sum_{k=0}^{\infty} A(k)\cdot x^k$, find the inverse FPS $\tilde{B}(x) = \sum_{k=0}^{\infty} B(k)\cdot x^k$, so that $\tilde{A}(x)\cdot\tilde{B}(x) = 1$, that is

$$\left(\sum_{k=0}^{\infty} A(k)\cdot x^k\right)\left(\sum_{l=0}^{\infty} B(k)\cdot x^k\right) = 1 + 0\cdot x + 0\cdot x^2 + \dots .$$

Notation. $\delta_0 = \{1, 0, 0, \dots\}$, the sequence in $\mathbb{C}^{\mathbb{Z}_+}$ that corresponds to the power series $1 + 0\cdot x + 0\cdot x^2 + \dots$, which is convergent with sum $= 1$.

Solution: $A * B = \delta_0 \iff \tilde{A}(x)\cdot\tilde{B}(x) = 1$

$$\begin{aligned}\tilde{A}(x)\cdot\tilde{B}(x) &= [A(0)B(0)]\cdot x^0 \\ &+ [A(0)B(1) + A(1)B(0)]\cdot x^1 \\ &+ [A(0)B(2) + A(1)\cdot B(1) + A(2)\cdot B(0)]\cdot x^2 \\ &+ \dots\end{aligned}$$

Thus recursively we get:

i) $A(0)\cdot B(0) = 1 \Rightarrow A(0) \neq 0$ and $B(0) = 1/A(0)$

ii) $A(0)\cdot B(1) + A(1)\cdot B(0) = 0 \Rightarrow B(1) = -A(1)\cdot B(0)/A(0)$

iii) $A(0)\cdot B(2) + A(1)\cdot B(1) + A(2)\cdot B(0) = 0 \Rightarrow$
$B(2) = -(A(1)B(1) + A(2)\cdot B(0))/A(0)$.

⋮

Theorem 5.58. The FPS $\tilde{A}(x)$ can be inverted if and only if $A(0) \neq 0$. The inverse FPS $[\tilde{A}(x)]^{-1}$ is obtained recursively as described above.

The recursive procedure requires a constant times N^2 Flops. If we need many coefficients the usage of FFT is a better method:

Theorem 5.59. Let $A(0) \neq 0$, and let $\tilde{B}(x)$ be the inverse of $\tilde{A}(x)$. Then, for every $k \geq 1$,

$$B_{|2k} = (B_{|k} * (2\delta_0 - A * B_{|k}))_{|2k}. \qquad (5.8)$$

Proof. Omitted.

Usage:

1) Compute $B_{|1} = \{\frac{1}{A(0)}, 0, \dots\}$
2) compute $B_{|2} = \{B(0), B(1), 0, 0, \dots\}$ by applying (5.8)
3) Compute $B_{|4} = \{B(0), B(1), B(2), B(3), 0, 0, \dots\}$ from (5.8)
4) Compute $B_{|8}$ using (5.8),

⋮

Convolutions are calculated in the way described in 5.44 on page 95.

Example. We wish to compute approximations of the 8 first coefficients in the inverse $\tilde{B}(x) = \sum_{k=0}^{\infty} B(k) \cdot x^k$ to $\tilde{A}(x) = \cos x = \sum_{k=0}^{\infty} (-1)^k \cdot \frac{x^{2k}}{(2k)!}$. Thus

$$A = \{1, 0, -\frac{1}{2}, 0, \frac{1}{4!}, 0, -\frac{1}{6!}, 0 \dots\}.$$

We apply formula (5.8):

```
>> l0=zeros(1,16);l0(1)=1;B1=zeros(1,16);B1(1)=1;

>> A=zeros(1,16);for j=1:8, A(2*j-1)=(-1)^(j+1)/gamma(2*j-1);end

>> temp=conv(A,B1);
>> B2=conv(B1,2*l0-temp(1:16));        } B12
>> B2(3:16)=0;

>> temp=conv(A,B2);
>> B4=conv(B2,2*l0-temp(1:16));        } B14
>> B4(5:16)=0;

>> temp=conv(A,B4);
>> B8=conv(B4,2*l0-temp(1:16));        } B18
>> B8(9:16)=0;
>> B8(1:8)

   ans = 1.0    0    0.5   0    0.208333...   0    0.0847222...    0
```

So we obtain the FPS:

$$\tilde{B}(x) = \frac{1}{\cos x} = 1 + \frac{1}{2}x^2 + 0{,}20833\ldots \cdot x^4 + 0{,}084722\ldots \cdot x^6 + \ldots,$$

$$= 1 + \frac{x^2}{2} + \frac{5}{24} \cdot x^4 + \frac{61}{720} \cdot x^6 + \ldots$$

Problem. Given a function $f(z)$ that is analytic in a region that contains the circle C centered at $z=z_0$ with radius r. How do we compute approximations to the coefficients c_j in its power series expansion:

$$f(z) = \sum_{j=0}^{\infty} c_j \cdot (z-z_0)^j \quad ?$$

By the theory of analytic functions we have:

$$c_k = \frac{f^{(k)}(z_0)}{k!} = \frac{1}{2\pi i} \oint_{|z-z_0|=r} \frac{f(z)}{(z-z_0)^{k+1}}\, dz$$

$$= \left[\begin{array}{l} z = z_0 + r \cdot e^{i\theta}, \quad 0 \le \theta \le 2\pi \\ \frac{dz}{d\theta} = i \cdot r \cdot e^{i\theta} \end{array} \right]$$

$$= \frac{1}{2\pi i} \int_0^{2\pi} \frac{f(z_0 + r \cdot e^{i\theta})}{(r \cdot e^{i\theta})^{k+1}} \cdot i \cdot r \cdot e^{i\theta}\, d\theta$$

$$= \frac{1}{2\pi} \int_0^{2\pi} \frac{f(z_0 + r \cdot e^{i\theta})}{r^k}\, e^{-ik\theta}\, d\theta .$$

Thus

$$r^k \cdot c_k = \frac{1}{2\pi} \int_0^{2\pi} f(z_0 + r \cdot e^{i\theta}) \cdot e^{-ik\theta}\, d\theta ,$$

So $r^k \cdot c_k$, $k \ge 0$, is the k:th Fourier coefficient of the 2π-periodic function $f(z_0 + r \cdot e^{i\theta})$, $0 \le \theta \le 2\pi$.

Example. Compute the coefficients in the Maclaurin series of the function $f(z) = \sin(\sin(\sin z))$, which is analytic in $\mathbb{C}$.

We can choose C to be the unit circle with $r=1$ and $z_0 = 0$. Thus the coefficients in the power series

$$f(z) = \sum_{k=0}^{\infty} c_k \cdot z^k$$

are given by: $c_k = \frac{1}{2\pi} \int_0^{2\pi} f(e^{i\theta}) \cdot e^{-ik\theta} \, d\theta$.

We compute them with the FFT in Matlab:

```
>> N=2048;theta=[0:2*pi/N:2*pi-2*pi/N];
>> F=sin(sin(sin(exp(i*theta))));
>> Ck=fft(F)/N;
>> Ck(1:11)

ans =

 -0.0 + 0.0i      1.000000000000000 - 0.000000000000000i

  0.0 - 0.0i     -0.500000000000000 + 0.000000000000000i

 -0.0 + 0.0i      0.275000000000000 - 0.000000000000000

  0.0 + 0.0i     -0.145039682539683 + 0.000000000000000i

 -0.0 - 0.0i      0.071254960317460 - 0.000000000000000i

  0.0 + 0.0i
```

So we conclude that the Maclaurin polynomial of order 5 is given by:

$$P_5(z) = z - \frac{z^3}{2} + \frac{11}{40} \cdot z^5.$$