Chapter 5

The Discrete Fourier Transform

We have studied four types of Fourier transforms:

i) Periodic functions on $\mathbb{R} \Rightarrow \hat{f}$ defined on $\mathbb{Z}$.

ii) Non-periodic functions on $\mathbb{R} \Rightarrow \hat{f}$ defined on $\mathbb{R}$.

iii) Distributions on $\mathbb{R} \Rightarrow \hat{f}$ defined on $\mathbb{R}$.

iv) Sequences defined on $\mathbb{Z} \Rightarrow \hat{f}$ periodic on $\mathbb{R}$.

The final addition comes now:

v) f *a periodic sequence* (on $\mathbb{Z}$) $\Rightarrow \hat{f}$ *a periodic sequence.*

5.1 Definitions

Definition 5.1. $\Pi_N = \{$all periodic sequences $F(m)$ with period N, i.e., $F(m+N) = F(m)\}$.

<u>Note</u>: These are in principle defined for all $n \in \mathbb{Z}$, but the periodicity means that it is enough to know $F(0), F(1), \ldots, F(N-1)$ to know the whole sequence (or any other set of N consecutive (= på varandra följande) values).

Definition 5.2. The **Fourier transform** of a sequence $F \in \Pi_N$ is given by

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

Warning 5.3. *Some people replace the constant $\frac{1}{N}$ in front of the sum by $\frac{1}{\sqrt{N}}$ or omit it completely. (This affects the inversion formula.)*

Lemma 5.4. *$\hat{F}$ is* periodic *with the same period N as F.*

Proof.

$$\begin{aligned}\hat{F}(m+N) &= \frac{1}{N}\sum_{\text{one period}} e^{-\frac{2\pi i(m+N)k}{N}}F(k) \\ &= \frac{1}{N}\sum_{\text{one period}} \underbrace{e^{-2\pi ik}}_{=1} e^{-\frac{2\pi imk}{N}}F(k) \\ &= \hat{F}(m). \quad \square\end{aligned}$$

Thus, $\boxed{F \in \Pi_N \Rightarrow \hat{F} \in \Pi_N}$.

Theorem 5.5. *F can be reconstructed from $\hat{F}$ by the inversion formula*

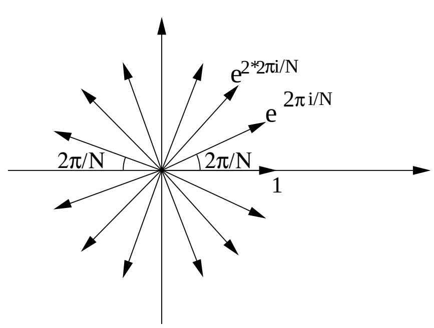
$$F(k) = \sum_{m=0}^{N-1} e^{\frac{2\pi imk}{N}}\hat{F}(m).$$

Note: No $\frac{1}{N}$ in front here.
Note: Matlab puts the $\frac{1}{N}$ in front of the inversion formula instead!

Proof.

$$\sum_m e^{\frac{2\pi imk}{N}}\frac{1}{N}\sum_l e^{-\frac{2\pi iml}{N}}F(l) = \frac{1}{N}\sum_{l=0}^{N-1}F(l)\underbrace{\sum_{m=0}^{N-1}e^{\frac{2\pi im(k-l)}{N}}}_{=\begin{cases}N, & \text{if } l=k\\ 0, & \text{if } l\neq k\end{cases}} = F(k)$$

We know that $\left(e^{\frac{2\pi i}{N}}\right)^N = 1$, so $e^{\frac{2\pi i}{N}}$ is the N:th root of 1:



We add N numbers, whose absolute value is one, and who point symmetrically in all the different directions indicated above. For symmetry reasons, the sum must be zero (except when $l = k$). (You always jump an angle $\frac{2\pi(k-l)}{N}$ for each turn, and go $k - l$ times around before you are done.)

Definition 5.6. The **convolution** $F * G$ of two sequences in Π_N are defined by

$$(F * G)(m) = \sum_{\text{one period}} F(m-k)G(k)$$

(Note: Some indeces get out of the interval $[0, N-1]$. You must use the *periodicity* of F and G to get the corresponding values of $F(m-k)G(k)$.).

Definition 5.7. The (ordinary) **product** $F \cdot G$ is defined by

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

Theorem 5.8. *$(\widehat{F \cdot G}) = \hat{F} * \hat{G}$ and $(\widehat{F * G}) = N\hat{F} \cdot \hat{G}$ (note the extra factor N).*

Proof. Easy. (Homework?)

Definition 5.9. $(RF)(n) = F(-n)$ (**reflection** operator).

As before: The inverse transform = the ususal transform plus reflection:

Theorem 5.10. *$\check{F} = N(\widehat{RF})$ (note the extra factor N), where $\hat{}$ = Fourier transform and $\check{}$ = Inverse Fourier transform.*

Proof. Easy. We could have avoided the factor N by a different scaling (but then it shows up in other places instead).

5.2 FFT=the Fast Fourier Transform

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

FLOP=FLoating Point Operation={multiplication or addition or combination of both}.

1 Megaflop = 1 million flops/second (10^6)
1 Gigaflop = 1 billion flops/second (10^9)

(Used as speed measures of computers.)

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

Good Idea 5.13. *Compute the coefficients* $\left(e^{-\frac{2\pi i}{N}}\right)^k = \omega^k$ only once, *and store them in a table. Since* $\omega^{k+N} = \omega^k$*, we have* $e^{-\frac{2\pi imk}{N}} = \omega^{mk} = \omega^r$ *where* $r =$ *remainder when we divide* mk *by* N*. Thus, only* N *numbers need to be stored.* *<u>Thus</u>: We can ignore the number of flops needed to compute the coefficients* $e^{-\frac{2\pi imk}{N}}$ *(done in advance).*

Trivial Solution 5.14. *If we count multiplication and addition separetely, then we need to compute* N *coefficients (as* $m = 0, 1, \ldots, N-1$*), and each coefficient requires* N *muliplications and* $N-1$ *additions. This totals*

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

This is too much.

Brilliant Idea 5.15. *Regroup (=omgruppera) the terms, using the* symmetry. *Start by doing* even *coefficients and* odd *coefficients separetely:*
Suppose for simplicity that N *is* even*. Then, for* even m*, (put* $N = 2n$*)*

$$\begin{aligned}
\hat{F}(2m) &= \frac{1}{N}\sum_{k=0}^{N-1}\omega^{2mk}F(k) \\
&= \frac{1}{N}\Big[\sum_{k=0}^{n-1}\omega^{2mk}F(k) + \underbrace{\sum_{k=n}^{2n-1}\omega^{2mk}F(k)}_{\text{Replace } k \text{ by } k+n}\Big] \\
&= \frac{1}{N}\left[\sum_{k=0}^{n-1}\omega^{2mk}F(k) + \omega^{2m(k+n)}F(k+n)\right] \\
&= \frac{2}{N}\sum_{k=0}^{n-1} e^{-\frac{2\pi imk}{(N/2)}}\frac{1}{2}\left[F(k)+F(k+n)\right].
\end{aligned}$$

This is a new *discrete time periodic Fourier transform of the sequence* $G(k) = \frac{1}{2}[F(k) + F(n+k)]$ *with* $\boxed{\text{period } n = \frac{N}{2}}$.
A similar computation (see Gripenberg) shows that the odd *coefficients can be computed from*

$$\hat{F}(2m+1) = \frac{1}{n}\sum_{k=0}^{n-1} e^{-\frac{2\pi imk}{n}} H(k),$$

where $H(k) = \frac{1}{2}e^{-\frac{i\pi k}{n}}\left[F(k) - F(k+n)\right]$. *Thus, instead of* one *transform of order* N *we get* two *transforms of order* $n = \frac{N}{2}$.
Number of flops: Computing the new transforms by brute force (as in 5.14 on page 105) we need the following flops:
Even: $n(2n-1) = \frac{N^2}{2} - \frac{N}{2} + n$ *additions* $= \frac{N^2}{2}$ *flops.*
Odd: The numbers $e^{-\frac{i\pi k}{n}} = e^{-\frac{2i\pi k}{N}}$ *are found in the table already computed.*
We essentially again need the same amount, namely $\frac{N^2}{2} + \frac{N}{2}$ *(n extra multiplications).*
Total: $\frac{N^2}{2} + \frac{N^2}{2} + \frac{N}{2} = N^2 + \frac{N}{2} \approx N^2$. *Thus, this* approximately halfed *the number of needed flops.*

Repeat 5.16. *Divide the new smaller transforms into two halfs, and again, and again. This is possible if* $N = 2^k$ *for some integer* k*, e.g.,* $N = 1024 = 2^{10}$.

Final conclusion: After some smaller adjustments we get down to

$$\frac{3}{2}2^k k \text{ flops.}$$

Here $N = 2^k$, so $k = \log_2 N$, and we get

Theorem 5.17. *The* Fast Fourier Transform *with* radius 2 *outlined above needs approximately* $\frac{3}{2}N\log_2 N$ *flops.*

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

$$\frac{3}{2}N\log_2 N \approx 15000 \ll 2000000 = 2N^2 - N.$$

Definition 5.18. Fast Fourier transform with

$$\begin{cases} \text{radius } 2: & \text{split into 2 parts at each step} \quad N = 2^k \\ \text{radius } 3: & \text{split into 3 parts at each step} \quad N = 3^k \\ \text{radius } m: & \text{split into } m \text{ parts at each step} \quad N = m^k \end{cases}$$

Note: Based on *symmetries*. "The same" computations repeat themselves, so by combining them in a clever way we can do it quicker.
Note: The FFT is *so fast* that it caused a minor revolution to many branches of numerical analysis. It made it possible to compute Fourier transforms in practice.
Rest of this chapter: How to use the FFT to compute the *other* transforms discussed earlier.

5.3 Computation of the Fourier Coefficients of a Periodic Function

Problem 5.19. *Let $f \in C(\mathbb{T})$. Compute*

$$\hat{f}(k) = \int_0^1 e^{-2\pi ikt} f(t)dt$$

as efficiently as possible.

Solution: Turn f into a periodic sequence and use FFT!

Conversion 5.20. *Choose some $N \in \mathbb{Z}$, and put*

$$F(m) = f(\frac{m}{N}), \quad m \in \mathbb{Z}$$

(equidistant "sampling"). The periodicity of f makes F periodic with period N. Thus, $F \in \Pi_N$.

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

$$\hat{F}(m) - \hat{f}(m) = \sum_{k \neq 0} \hat{f}(m + kN).$$

Proof. By the inversion formula, for all t,

$$f(t) = \sum_{j \in \mathbb{Z}} e^{2\pi ijt} \hat{f}(j).$$

Put $t_k = \frac{k}{N} \Rightarrow$

$$f(t_k) = F(k) = \sum_{j \in \mathbb{Z}} e^{\frac{2\pi ikj}{N}} \hat{f}(j)$$

(this series converges uniformly by Lemma 1.14). By the definition of $\hat{F}$:

$$\begin{aligned} \hat{F}(m) &= \frac{1}{N} \sum_{k=0}^{N-1} e^{-\frac{2\pi imk}{N}} F(k) \\ &= \frac{1}{N} \sum_{j \in \mathbb{Z}} \hat{f}(j) \qquad \underbrace{\sum_{k=0}^{N-1} e^{\frac{2\pi i(j-m)k}{N}}}_{=\begin{cases} N, & \text{if } \frac{j-m}{N} = \text{integer} \\ 0, & \text{if } \frac{j-m}{N} \neq \text{integer} \end{cases}} \\ &= \sum_{l \in \mathbb{Z}} \hat{f}(m + Nl). \end{aligned}$$

Take away the term $\hat{f}(m)$ $(l = 0)$ to get

$$\hat{F}(m) = \hat{f}(m) + \sum_{l \neq 0} \hat{f}(m + Nl).$$

Note: If N is "large" and if $\hat{f}(m) \to 0$ "quickly" as $m \to \infty$, then the error

$$\sum_{l \neq 0} \hat{f}(m + Nl) \approx 0.$$

First Method 5.22. *Put*

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

ii) $\hat{f}(m) \approx \frac{1}{2}\hat{F}(m)$ *if* $|m| = \frac{N}{2}$ *(N even)*

iii) $\hat{f}(m) \approx 0$ *if* $|m| > \frac{N}{2}$.

Here ii) is not important. *We could use* $\hat{f}(\frac{N}{2}) = 0$ *or* $\hat{f}(\frac{N}{2}) = \hat{F}(m)$ *instead. Here*

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

Notation 5.23. *Let us denote (note the extra star)*

$$\sum_{|k| \leq N/2}^{*} a_k = \sum_{|k| \leq N/2} a_k$$

= the usual sum of a_k if N odd (then we have exactly N terms), and

$$\sum_{|k| \leq N/2}^{*} a_k = \begin{array}{l}\textit{a sum where the first and last terms have been} \\ \textit{divided by two (these are the same if the sequence} \\ \textit{is periodic with period } N\textit{, there is ``one term too} \\ \textit{many'' in this case).}\end{array}$$

First Method 5.24 (Error)**.** *The first method gives the error:*

i) $|m| < \frac{N}{2}$ *gives the error*

$$|\hat{f}(m) - \hat{F}(m)| \leq \sum_{k \neq 0} |\hat{f}(m + kN)|$$

ii) $|m| = \frac{N}{2}$ *gives the error*

$$|\hat{f}(m) - \frac{1}{2}\hat{F}(m)|$$

iii) $|m| > \frac{N}{2}$ *gives the error* $|\hat{f}(m)|$.

we can simplify *this into the following crude (="grov") estimate:*

$$\boxed{\sup_{m\in\mathbb{Z}}|\hat{f}(m) - \hat{F}(m)| \leq \sum_{|m|\geq N/2}^{*} |\hat{f}(m)|} \tag{5.1}$$

(because this sum is $\geq$ *the actual error).*

First Method 5.25 (Drawbacks)**.**

1° *Large error.*

2° *Inaccurate error estimate* (5.1)*.*

3° *The error estimate based on* $\hat{f}$ *and not on* f*.*

We need a better method.

Second Method 5.26 (General Principle)**.**

1° *Evaluate* t *at the points* $t_k = \frac{k}{N}$ *(as before),* $F(k) = f(t_k)$

2° *Use the sequence* F *to construct a new function* $P \in C(T)$ *which "approximates"* f*.*

3° *Compute the Fourier coefficients of* P*.*

4° *Approximate* $\hat{f}(n)$ *by* $\hat{P}(n)$*.*

For this to succeed we must choose P *in a smart way. The final result will be quite simple, but for later use we shall derive* P *from some "basic principles".*

Choice of P **5.27.** *Clearly* P *depends on* F*. To simplify the computations we require* P *to satisfy (write* $P = P(F)$*)*

A) P *is linear:* $P(\lambda F + \mu G) = \lambda P(F) + \mu P(G)$

B) P *is translation invariant: If we translate* F*, then* $P(F)$ *is translated by the same amount: If we denote*

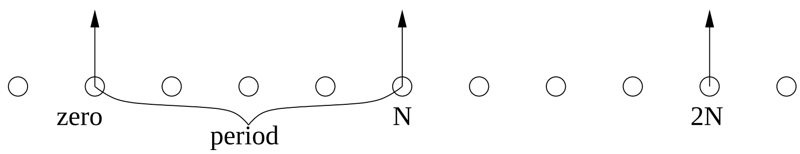
$$\begin{aligned} (\tau_j F)(m) &= F(m-j), \text{ then} \\ P(\tau_j F) &= \tau_{j/N} P(F) \end{aligned}$$

(j *discrete steps* $\Longleftrightarrow$ *a time difference of* j/N*).*

This leads to simple computations: We want to compute $\hat{P}(m)$ (which we use as approximations of $\hat{f}(m)$) Define a δ-sequence:

$$D(n) = \begin{cases} 1, & \text{for } n = 0, \pm N, \pm 2N, \dots \\ 0, & \text{otherwise.} \end{cases}$$

Then



$$(\tau_k D)(n) = \begin{cases} 1, & \text{if } n = k + jN,\ j \in \mathbb{Z} \\ 0, & \text{otherwise,} \end{cases}$$

so

$$[F(k)\tau_k D](n) = \begin{cases} F(k), & n = k + jN \\ 0, & \text{otherwise.} \end{cases}$$

and so

$$\boxed{F = \sum_{k=0}^{N-1} F(k)\tau_k D}$$

Therefore, the principles A) and B) give

$$\begin{aligned} P(F) &= \sum_{k=0}^{N-1} F(k)P(\tau_k D) \\ &= \sum_{k=0}^{N-1} F(k)\tau_{k/N} P(D), \end{aligned}$$

Where $P(D)$ is the approximation of $D =$ "unit pulse at time zero" D.

We denote this function by p. Let us transform $P(F)$:

$$\begin{aligned}
(\widehat{P(F)})(m) &= \int_0^1 \sum_{k=0}^{N+1} F(k)(\tau_{k/N}p)(s)e^{-2\pi ism}ds \\
&= \sum_{k=0}^{N+1} F(k) \int_0^1 e^{-2\pi ism} p(s-\frac{k}{N})ds \quad (s-\frac{k}{N}=t) \\
&= \sum_{k=0}^{N+1} F(k) \int_{\substack{\text{one}\\\text{period}}} e^{-2\pi im(t+\frac{k}{N})}p(t)dt \\
&= \sum_{k=0}^{N+1} F(k)e^{-\frac{-2\pi imk}{N}} \underbrace{\int_{\substack{\text{one}\\\text{period}}} e^{-2\pi imt}p(t)dt}_{\hat{p}(m)} \\
&= \hat{p}(m) \underbrace{\sum_{k=0}^{N+1} F(k)e^{-\frac{2\pi imk}{N}}}_{=N\hat{F}(m)} \\
&= N\hat{p}(m)\hat{F}(m).
\end{aligned}$$

We can get rid of the factor N by replacing p by Np. This is our approximation of the "pulse of size N at zero"

$$\begin{cases} N, & n = 0 + jN \\ 0, & \text{otherwise.} \end{cases}$$

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

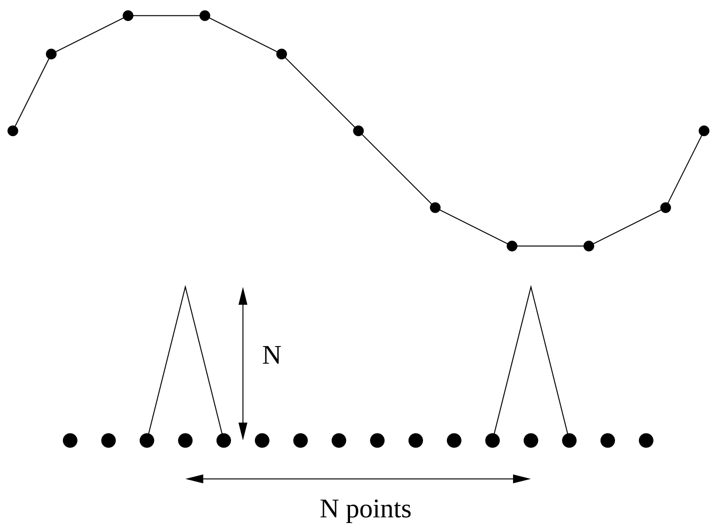
$$\hat{f}(m) \approx \hat{F}(m)\hat{p}(m),$$

where $\hat{p}$ is the Fourier transform of the function that we get when we apply our approximation procedure to the sequence

$$ND(n) = \begin{cases} N, & n = 0(+jN) \\ 0, & \textit{otherwise.} \end{cases}$$

Note: The complicated proof of this simple method will pay off in a second!

Approximation Method 5.29. *Use any type of translation-invariant interpolation method, for example* splines. *The simplest possible method is* linear interpolation*: If we interpolate the pulse ND in this way we get Thus,*



$$p(t) = \begin{cases} N(1 - N|t|), & |t| \le \frac{1}{N} \\ 0, & \frac{1}{N} \le |t| \le 1 - \frac{1}{N} \end{cases}$$

(periodic extension)

This is a periodic version of the kernel. A direct computation gives

$$\hat{p}(m) = \left(\frac{\sin(\pi m/N)}{\pi m/N} \right)^2 .$$

We get the following interesting theorem:

Theorem 5.30. *If we first discretize f, i.e. we replace f by the sequence $F(k) = f(k/N)$, the compute $\hat{F}(m)$, and finally multiply $\hat{F}(m)$ by*

$$\hat{p}(m) = \left(\frac{\sin(\pi m/N)}{\pi m/N} \right)^2 .$$

then we get the Fourier coefficients for the function which we get from f by linear interpolation at the points $t_k = k/N$.

(This corresponds to the computation of the Fourier integral $\int_0^1 e^{-2\pi imt} f(t)dt$ by using the trapetsoidal rule. Other integration methods have similar interpretations.)

5.4 Trigonometric Interpolation

Problem 5.31. *Construct a good method to approximate a periodic function $f \in C(T)$ by a* trigonometric polynomial

$$\sum_{m=-N}^{N} a_m e^{2\pi imt}$$

(a finite *sum, resembles inverse Fourier transformation).*

Useful for numerical computation etc.
Note: The earlier "Second Method" gave us a *linear interpolation*, not trigonometric approximation.
Note: This trigonometric polynomial has only finitely many Fourier coefficients $\neq 0$ (namely $a_m, |m| \leq N$).
Actually, the "First Method" gave us a trigonometric polynomial. There we had

$$\begin{cases} \hat{f}(m) \approx \hat{F}(m) & \text{for } |m| < \frac{N}{2}, \\ \hat{f}(m) \approx \frac{1}{2}\hat{F}(m) & \text{for } |m| = \frac{N}{2}, \\ \hat{f}(m) \approx 0 & \text{for } |m| > \frac{N}{2}. \end{cases}$$

By inverting this sequence we get a trigonometric approximation of f: $f(t) \approx g(t)$, where

$$g(t) = \sum_{|m| \leq N/2}^{*} \hat{F}(m) e^{2\pi imt}. \tag{5.2}$$

We have *two* different *errors*:

i) $\hat{f}(m)$ is replaced by $\hat{F}(m) = \frac{1}{N}\sum_{k=0}^{N-1} f(\frac{k}{N}) e^{\frac{2\pi ikm}{N}}$,

ii) The inverse series was truncated to N terms.

Strange fact: These two errors (partially) *cancel* each other.

Theorem 5.32. *The function g defined in* (5.2) *satisfies*

$$g(\frac{k}{N}) = f(\frac{k}{N}), \quad n \in \mathbb{Z},$$

i.e., g interpolates *f at the points t_k (which were used to construct first F and then g).*

PROOF. We defined $F(k) = f(\frac{k}{N})$, and

$$\hat{F}(m) = \sum_{|k|\le N/2}^{*} F(k) e^{-\frac{2\pi i m k}{N}}.$$

By the inversion formula on page 103,

$$\begin{aligned} g(\frac{k}{N}) &= \sum_{|m|\le N/2}^{*} \hat{F}(m) e^{\frac{2\pi i m k}{N}} \text{ (use periodicity)} \\ &= \sum_{m=0}^{N-1} \hat{F}(m) e^{\frac{2\pi i m k}{N}} \\ &= F(k) = f(\frac{k}{N}) \quad \Box \end{aligned}$$

Error estimate: How large is $|f(t)-g(t)|$ between the mesh points $t_k = \frac{k}{N}$ (where the error is zero)? We get an estimate from the computation in the last section. Suppose that $\hat{f} \in \ell^1(\mathbb{Z})$ and $f \in C(T)$ so that the inversion formula holds for all t (see Theorem 1.37). Then

$$f(t) = \sum_{m\in\mathbb{Z}} \hat{f}(m) e^{2\pi i m t}, \quad \text{and}$$

$$\begin{aligned} g(t) &= \sum_{|m|\le N/2}^{*} \hat{F}(m) e^{2\pi i m t} \quad \text{(Theorem 5.21)} \\ &= \sum_{|m|\le N/2}^{*} \left[\hat{f}(m) + \sum_{k\ne 0} \hat{f}(m+kN) \right] e^{2\pi i m t} \\ &= f(t) - \sum_{|m|\ge N/2}^{*} \hat{f}(m) e^{2\pi i m t} + \sum_{|m|\le N/2}^{*} \sum_{k\ne 0} \hat{f}(m+kN) e^{2\pi i m t}. \end{aligned}$$

Thus

$$|g(t) - f(t)| \le \sum_{|m|\ge N/2}^{*} |\hat{f}(m)| + \underbrace{\sum_{|m|\le N/2}^{*} \sum_{k\ne 0} |\hat{f}(m+kM)|}_{=\sum^{*}_{|l|\ge N/2} |\hat{f}(l)|} = 2 \sum_{|m|\ge N/2}^{*} |\hat{f}(m)|$$

(take $l = m + kN$, every $|l| > \frac{N}{2}$ appears one time, no $|l| < \frac{N}{2}$ appears, and $|l| = \frac{N}{2}$ two times).

This leads to the following theorem:

Theorem 5.33. *If* $\sum_{m=-\infty}^{\infty} |\hat{f}(m)| < \infty$*, then*

$$|g(t) - f(t)| \le 2 \sum_{|m|\ge N/2}^{*} |\hat{f}(m)|,$$

where

$$\begin{array}{rcl} g(t) & = & \sum_{|m|\le N/2}^{*} \hat{F}(m) e^{2\pi i m t}, \quad \text{and} \\ \hat{F}(m) & = & \frac{1}{N} \sum_{|m|\le N/2}^{*} e^{-\frac{2\pi i m k}{N}} f(\frac{k}{N}). \end{array}$$

This is nice if $\hat{f}(m) \to 0$ *rapidly as* $m \to \infty$*. Better accuracy by increasing* N*.*

5.5 Generating Functions

Definition 5.34. The **generating function** of the sequence $J_n(x)$ is the function

$$f(x,z) = \sum_n J_n(x) z^n,$$

where the sum over $n \in \mathbb{Z}$ or over $n \in \mathbb{Z}_+$, depending on for which values of n the functions $J_n(x)$ are defined.

Note: We did this in the course on *special functions*. E.g., if J_n = Bessel's function of order n, then

$$f(x,z) = e^{\frac{x}{2}(z-1/2)}.$$

Note: For a fixed value of x, this is the "mathematician's version" of the Z-transform described on page 101.
Make a change of variable:

$$\begin{array}{rcl} z = e^{2\pi i t} \Rightarrow f(x, e^{2\pi i t}) & = & \sum_{n\in\mathbb{Z}} J_n(x) (e^{2\pi i t})^n \\ & = & \sum_{n\in\mathbb{Z}} J_n(x) e^{2\pi i n t}, \end{array}$$

Comparing this to the inversion formula in Chapter 1 we get

Theorem 5.35. *For a fixed* x*, the* n*:th Fourier coefficient of the function* $t \mapsto f(x, e^{2\pi i t})$ *is equal to* $J_n(x)$*.*

Thus, we can *compute* $J_n(x)$ by the method descibed in Section 5.3 to compute the coefficients $a_n = J_n(x)$ (x = fixed, n varies):

1) Discretize $F(k) = f(x, e^{\frac{2\pi ik}{N}})$

2) $\hat{F}(m) = \frac{1}{N} \sum^{*}_{|k|\leq N/2} e^{-\frac{2\pi imk}{N}} F(k)$

3) $\hat{F}(m) - J_n(x) = \sum_{k\neq 0} a_{m+kN}$, (Theorem 5.21)

where $a_{m+kN} = J_{m+kN}(x)$.

5.6 One-Sided Sequences

So far we have been talking about *periodic* sequences (in Π_N). Instead one often wants to discuss

A) *Finite sequences* $A(0), A(1), \ldots, A(N-1)$ or

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

Note: 5.36. *A* finite *sequence is a special csse of a one-sided sequence: put* $A(n) = 0$ *for* $n \geq N$.

Note: 5.37. *A one-sided sequence is a special case of a two-sided sequence: put* $A(n) = 0$ *for* $n < 0$.

<u>Problem</u>: These extended sequences are *not periodic*. $\Rightarrow$ We cannot use the Fast Fourier Transform directly.

Notation 5.38. $\mathbb{C}^{\mathbb{Z}_+} = \{$*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

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

<u>Note</u>: The summation boundaries are the natural ones if we think that $A(k) = B(k) = 0$ for $k < 0$.

Notation 5.40.

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

Thus, this restricts *the sequence* $A(k)$ *to the* n *first terms.*

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

Proof. Easy.

Notation 5.42. *$A = 0_n$ means that $A(k) = 0$ for $0 \le k < n-1$, i.e., $A_{|n} = 0$.*

Lemma 5.43. *If $A = 0_n$ and $B = 0_m$, then $A * B = 0_{n+m}$.*

Proof. Easy.

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

1) *Choose a number $N \ge 2n$ (often a power of 2).*

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, period N.

3) *Define*

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

periodic extension: $G(k+N) = G(k)$.

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

$$\begin{aligned} \underbrace{(F * G)(m)}_{\text{periodic convolution}} &= \sum_{k=0}^{N-1} F(m-k)G(k) \\ &= \sum_{k=0}^{m} F(m-k)G(k) \\ &= \sum_{k=0}^{m} A(m-k)G(k) = \underbrace{(A * B)(m)}_{\text{one-sided convolution}} \end{aligned}$$

Note: Important that $N \ge 2n$.
Thus, this way we have computed the n first coefficients of $(A * B)$.

Theorem 5.45. *The method described below allows us to compute $(A * B)_{|n}$ (=the first n coefficients of $A * B$) with a number of FLOP:s which is*

$$C \cdot n \log_2 n, \text{ where } C \text{ is a constant.}$$

Method: 1)-3) same as above

4) Use FFT to compute

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

5) Use the inverse FFT to compute

$$F*G = \frac{1}{N}(\hat{F}\cdot\hat{G})\check{}$$

Then $(A*B)_{|n} = (F*G)_{|n}$.

Note: A "naive" computation of $A*B_{|n}$ requires $C_1\cdot n^2$ FLOPs, where C_1 is another constant.
Note: Use "naive" method if n small. Use "FFT-inverse FFT" if n is large.
Note: The rest of this chapter *applies* one-sided convolutions to different situations. In all cases the method described in Theorem 5.45 can be used to compute these.

5.7 The Polynomial Interpretation of a Finite Sequence

Problem 5.46. *Compute the product of two polynomials:*

$$p(x) = \sum_{k=0}^{n} a_k x^k \qquad q(x) = \sum_{l=0}^{m} b_l x^l.$$

Solution: Define $a_k = 0$ for $k > 0$ and $b_l = 0$ for $l > m$. Then

$$\begin{aligned}
p(x)q(x) &= \underbrace{\left(\sum_{k=0}^{\infty} a_k x^k\right)\left(\sum_{l=0}^{\infty} b_l x^l\right)}_{\text{sums are actually finite}} \\
&= \sum_{k,l} a_k b_l x^{k+l} \quad (k+l=j,\ k=j-l) \\
&= \sum_{j} x^j \sum_{l=0}^{j} a_{j-l} b_l = \sum_{j=0}^{m+n} c_j x^j,
\end{aligned}$$

where $c_j = \sum_{l=0}^{j} a_{j-l} b_l$. This gives

Theorem 5.47.

i) Multiplication *of two polynomials corresponds to a* convolution *of their* coefficients*: If*

$$p(x) = \sum_{k=0}^{n} a_k x^k, \quad q(x) = \sum_{l=0}^{m} b_l x^l,$$

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

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

$$p(x) + q(x) = \sum c_j x^j, \quad \text{where } c_j = a_j + b_j.$$

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

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

Thus there is a *one-to-one correspondence* between

$$\text{polynomials} \iff \text{finite sequences},$$

where the operations correspond as described above. This is used in all symbolic computer computations of polynomials.

Note: Two different conventions are in common use:

A) first coefficient is a_0 (= lowest order),

B) first coefficient is a_n (= highest order).

5.8 Formal Power Series and Analytic Functions

Next we extend "polynomials" so that they may contain *infinitely many* terms.

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

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

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

Example 5.49. $\sum_{k=0}^{\infty} \frac{x^k}{k!}$ converges for all x (and the sum is e^x).

Example 5.50. $\sum_{k=0}^{\infty} x^k$ converges for $|x| < 1$ (and the sum is $\frac{1}{1-x}$).

Example 5.51. $\sum_{k=0}^{\infty} k! x^k$ converges for *no* $x \neq 0$.

All of these are formal power series (and the first two are "ordinary" power series).

Calculus with FPS 5.52. *We borrow the calculus rules from the polynomials:*

i) We add *two FPS:s by adding the coefficients:*

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

ii) We multiply *a FPS* by a constant λ *by multiplying each coefficients by* λ*:*

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

iii) We multiply *two FPS:s with each other by taking the* convolution *of the coefficients:*

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

where $C = A * B$*.*

Notation 5.53. *We denote* $\tilde{A}(x) = \sum_{k=0}^{\infty} A(k)x^k$*.*

Conclusion 5.54. *There is a* one-to-one *correspondence between all* Formal Power Series *and all* one-sided sequences *(bounded or not). We denoted these by* $\mathbb{C}^{\mathbb{Z}_+}$ *on page 116.*

Comment 5.55. *In the sequence (="fortsättningen") we operate with FPS:s. These power series* often converge*, and then they define analytic functions, but this fact is not used anywhere in the proofs.*

5.9 Inversion of (Formal) Power Series

Problem 5.56. *Given a (formal) power series* $\tilde{A}(x) = \sum A(k)x^k$*, find the* inverse *formal power series* $\tilde{B}(x) = \sum B(k)x^k$.

Thus, we want to find $\tilde{B}(x)$ so that

$$\begin{aligned} &\tilde{A}(x)\tilde{B}(x) = 1, \quad \text{i.e.,} \\ &\left[\sum_{k=0}^{\infty} A(k)x^k\right]\left[\sum_{l=0}^{\infty} B(l)x^l\right] = 1 + 0x + 0x^2 + \dots \end{aligned}$$

Notation 5.57. $\delta_0 = \{1, 0, 0, \dots\}$ = *the sequence whose power series is* $\{1 + 0x + 0x^2 + \dots\}$. *This series converges, and the sum is* $\equiv 1$. *More generally:*

$$\begin{aligned} \delta_k &= \{0, 0, 0, \dots, 0, 1, 0, 0, \dots\} \\ &= 0 + 0x + 0x^2 + \dots + 0x^{k-1} + 1x^k + 0x^{k+1} + 0x^{k+2} + \dots \\ &= x^k \end{aligned}$$

Power Series	*Sequence*
δ_k	x^k

Solution. We know that $A*B = \delta_0$, or equivalently, $\tilde{A}(x)\tilde{B}(x) = 1$. Explicitly,

$$\begin{aligned} \tilde{A}(x)\tilde{B}(x) &= A(0)B(0) \quad (\text{times } x^0) & (5.3) \\ &\quad +[A(0)B(1) + A(1)B(0)]x & (5.4) \\ &\quad +[A(0)B(2) + A(1)B(1) + A(2)B(0)]x^2 & (5.5) \\ &\quad +\dots & (5.6) \end{aligned}$$

From this we can solve:

i) $A(0)B(0) = 1 \implies A(0) \neq 0$ and $B(0) = \frac{1}{A(0)}$.

ii) $A(0)B(1) + A(1)B(0) = 0 \implies B(1) = -\frac{A(1)B(0)}{A(0)}$ (always possible)

iii) $A(0)B(2) + A(1)B(1) + A(2)B(0) = 1 \implies B(2) = -\frac{1}{A(0)}[A(1)B(1) + A(2)B(0)]$, etc.

we get a theorem:

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

Recursive means:

i) Solve $B(0)$

ii) Solve $B(1)$ using $B(0)$

iii) Solve $B(2)$ using $B(1)$ and $B(0)$

iv) Solve $B(3)$ using $B(2)$, $B(1)$ and $B(0)$, etc.

This is *Hard Work*. For example

$$\begin{aligned}
\sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \\
\cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \\
\tan(x) &= \frac{\sin(x)}{\cos(x)} = \underbrace{\sin(x)\frac{1}{\cos(x)}}_{\text{convolution}} = ???
\end{aligned}$$

Hard Work means: Number of FLOPS is a *constant times* N^2. Better method: Use FFT.

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

$$B_{|2k} = (B_{|k} * (2\delta_0 - A * B_{|k}))_{|2k} \tag{5.7}$$

Proof. See Gripenberg.
Usage: First compute $B_{|1} = \{\frac{1}{A(0)}, 0, 0, 0, \dots\}$
Then $B_{|2} = \{B(0), B(1), 0, 0, 0, \dots\}$ (use (5.7))
Then $B_{|4} = \{B(0), B(1), B(2), B(3), 0, , 0, 0, \dots\}$
Then $B_{|8} = \{8 \text{ terms} \neq 0\}$ etc.
Use the method on page 117 for the convolutions. (Useful only if you need *lots* of coefficients).

5.10 Multidimensional FFT

Especially in image processing we also need the discrete Fourier transform in *several* dimensions. Let $d = \{1, 2, 3, \dots\}$ be the "*space dimension*". Put $\Pi_N^d = \{$ sequences $x(k_1, k_2, \dots, k_d)$ which are N-periodic in each variable separately$\}$.

Definition 5.60. The d-dimensional Fourier transform is obtained by transforming d successive (="efter varandra") "ordinary" Fourier transformations, one for each variable.

Lemma 5.61. *The d-dimensional Fourier transform is given by*

$$\hat{x}(m_1, m_2, \ldots, m_d) = \frac{1}{N^d} \sum_{k_1} \sum_{k_2} \cdots \sum_{k_d} e^{\frac{-2\pi i(k_1 m_1 + k_2 m_2 + \ldots + k_d m_d)}{N}} x(k_1, k_2, \ldots, k_d).$$

PROOF. Easy.

All 1-dimensional results generalize easy to the d-dimensional case.

Notation 5.62. *We call* $\underline{k} = (k_1, k_2, \ldots, k_d)$ *and* $\underline{m} = (m_1, m_2, \ldots, m_d)$ multi-indeces *(=pluralis av "multi-index"), and put*

$$\underline{k} \cdot \underline{m} = k_1 m_1 + k_2 m_2 + \ldots + k_d m_d$$

(=the "inner product" of $\underline{k}$ *and* $\underline{m}$*).*

Lemma 5.63.

$$\begin{aligned} \hat{x}(\underline{m}) &= \frac{1}{N^d} \sum_{\underline{k}} e^{-\frac{2\pi i \underline{m} \cdot \underline{k}}{N}} x(\underline{k}), \\ x(\underline{k}) &= \sum_{\underline{m}} e^{\frac{2\pi i \underline{m} \cdot \underline{k}}{N}} \hat{x}(\underline{m}). \end{aligned}$$

Definition 5.64.

$$\begin{aligned} (F \cdot G)(\underline{\mathrm{m}}) &= F(\underline{\mathrm{m}}) G(\underline{\mathrm{m}}) \\ (F * G)(\underline{\mathrm{m}}) &= \sum_{\underline{\mathrm{k}}} F(\underline{\mathrm{m}} - \underline{\mathrm{k}}) G(\underline{\mathrm{k}}), \end{aligned}$$

where all the components of $\underline{\mathrm{m}}$ and $\underline{\mathrm{k}}$ run over one period.

Theorem 5.65.

$$\begin{aligned} (F \cdot G)\hat{} &= \hat{F} * \hat{G}, \\ (F * G)\hat{} &= N^d \hat{F} \cdot \hat{G}. \end{aligned}$$

PROOF. Follows from Theorem 5.8.

In practice: Either use *one* multi-dimensional, or use d one-dimensional transforms (not much difference, multi-dimensional a little faster).