

## 3.5 Schnelle Fouriertransformation (FFT, DFT)

### 3.5.1 Grundlagen

Ein Polynom  $P = \sum_i a_i x^i \in \mathbb{C}[x]$  vom Grad  $\leq n$  ist eindeutig durch seine Koeffizienten  $a_i$  bestimmt, d.h. man hat eine Bijektion

$$\{\text{Polynome} \in \mathbb{C}[x] \text{ vom Grad} \leq n\} \rightarrow \mathbb{C}^{n+1}$$

$$P_{\vec{a}} = \sum_{i=0}^n a_i x^i \mapsto \vec{a} = (a_0, \dots, a_n).$$

Problem:  $P_{\vec{a}} \cdot P_{\vec{b}} = P_{\vec{c}}$  mit  $\vec{c} = (c_0, \dots, c_{2n})$ ,  $c_k = \sum_i a_{k-i} b_i$ , und die naive Berechnung von  $\vec{c}$  benötigt  $\Theta(n^2)$  Operationen.

**Bemerkung:**  $\vec{c} = \vec{a} * \vec{b}$  mit  $c_k = \sum_i a_{k-i} b_i$  ist die **Faltung** von  $\vec{a}$  und  $\vec{b}$ .

Es gibt noch eine weitere eindeutige Darstellung eines Polynoms.

### Lemma 144

Seien  $P = \sum_{i=0}^n a_i x^i$  und  $Q = \sum_{j=0}^n b_j x^j$  Polynome ( $\in \mathbb{C}[x]$ ) vom Grad  $\leq n$  und seien  $\omega_0, \dots, \omega_n \in \mathbb{C}$  paarweise verschiedene Elemente. Dann gilt:

$$P = Q \iff P(\omega_i) = Q(\omega_i) \quad \text{für alle } i = 0, \dots, n.$$

### Beweis:

„ $\Rightarrow$ “: Klar.

„ $\Leftarrow$ “: Es gelte  $P(\omega_i) = Q(\omega_i)$  für  $i = 0, \dots, n$ . Dann ist jedes  $\omega_i$  eine Nullstelle des Polynoms  $P - Q$ . Da  $\text{grad}(P - Q) \leq n$  gilt, folgt  $P - Q = 0$  aus Satz 139.  $\square$

Man kann leicht zeigen, dass es zu jedem Tupel  $(b_0, \dots, b_n) \in \mathbb{C}^{n+1}$  (genau) ein Polynom  $f \in \mathbb{C}[x]$  vom Grad  $\leq n$  gibt, mit  $f(\omega_i) = b_i$  für  $i = 0, \dots, n$  (z.B. das **Newton'sche Interpolationspolynom**, benannt nach **Sir Isaac Newton** (1643–1727)).

Somit erhalten wir eine weitere Bijektion:

$$\begin{aligned} \{\text{Polynome } \in \mathbb{C}[x] \text{ vom Grad } \leq n\} &\rightarrow \mathbb{C}^{n+1} \\ P &\mapsto (P(\omega_0), \dots, P(\omega_n)) \end{aligned}$$

Vorteil:

$$\begin{aligned} P \times Q \mapsto (P(\omega_0)Q(\omega_0), \dots, P(\omega_n)Q(\omega_n)) = \\ (P(\omega_0), \dots, P(\omega_n)) \cdot (Q(\omega_0), \dots, Q(\omega_n)). \end{aligned}$$

Multiplikation benötigt nur  $O(n)$  Operationen. „ $\cdot$ “ auf der rechten Seite bezeichnet hier das komponentenweise (**Hadamard**) Vektorprodukt (**Jacques S. Hadamard** (1865–1963)).

Problem: Bijektion i.a. zu komplex.

### Definition 145

Ein  $\omega \in \mathbb{C}$  heißt **primitive  $n$ -te Einheitswurzel**, wenn  $\omega^k \neq 1$  für alle  $k = 1, \dots, n-1$  und  $\omega^n = 1$  gilt, d.h.  $\text{ord}(\omega) = n$  in  $\mathbb{C}^* = \mathbb{C} \setminus 0$ .

**Bemerkung:** Es ist  $\omega = e^{2i\pi/n}$  eine primitive  $n$ -te Einheitswurzel.

### Definition 146

Sei  $\omega \in \mathbb{C}$  eine primitive  $n$ -te Einheitswurzel,  $n \in \mathbb{N}$ . Die Abbildung

$$\mathcal{F}_{n,\omega} : \mathbb{C}^n \rightarrow \mathbb{C}^n, \\ \vec{a} = (a_0, \dots, a_{n-1}) \mapsto (P_{\vec{a}}(1), P_{\vec{a}}(\omega), \dots, P_{\vec{a}}(\omega^{n-1}))$$

heißt **diskrete Fouriertransformation**; wir schreiben auch kurz  $\mathcal{F}$  für  $\mathcal{F}_{n,\omega}$ .

Die Fouriertransformation ist nach **Jean Baptiste Joseph Fourier** (1768–1830) benannt.

Bemerkung:  $\mathcal{F}$  ist nach Lemma 144 und anschließender Bemerkung eine Bijektion.

### Lemma 147

Seien  $\vec{a}, \vec{b} \in \mathbb{C}^n$  so, dass auch  $\vec{a} * \vec{b} \in \mathbb{C}^n$ . Dann gilt

$$\mathcal{F}(\vec{a} * \vec{b}) = \mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b}).$$

Beweis:

Es gilt

$$\begin{aligned}\mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b}) &= (P_{\vec{a}}(1)P_{\vec{b}}(1), P_{\vec{a}}(\omega)P_{\vec{b}}(\omega), \dots, P_{\vec{a}}(\omega^{n-1})P_{\vec{b}}(\omega^{n-1})) \\ &= (P_{\vec{c}}(1), P_{\vec{c}}(\omega), \dots, P_{\vec{c}}(\omega^{n-1})) \\ &= \mathcal{F}(\vec{c}), \quad \text{mit } \vec{c} = \vec{a} * \vec{b}.\end{aligned}$$



Idee: Berechne  $\vec{a} * \vec{b}$  vermöge  $\mathcal{F}^{-1}(\mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b}))$ . Die komponentenweise Multiplikation  $\mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b})$  benötigt nur  $O(n)$  Operationen.

Jedoch:  $\mathcal{F}$  ist eine lineare Abbildung  $\mathcal{F}(\vec{a}) = \Omega \cdot \vec{a}$ , mit  $\Omega = (\omega^{kl})_{0 \leq l, k \leq n-1}$ . Die Matrixmultiplikation benötigt aber  $\Omega(n^2)$  Operationen (also keine offensichtliche Verbesserung im Vergleich zur klassischen Polynom-Multiplikation)!

Ausweg: "Divide and Conquer" !!!

### 3.5.2 Berechnung der diskreten Fouriertransformation (FFT)

Sei  $n = 2^k$  eine 2er-Potenz. Zerlege  $\vec{a} = (a_0, \dots, a_{n-1})$  in einen

geraden Anteil  $\vec{a}_g = (a_0, a_2, \dots, a_{n-2})$  und einen  
ungeraden Anteil  $\vec{a}_u = (a_1, a_3, \dots, a_{n-1})$

Dann gilt:

$$P_{\vec{a}}(x) = P_{\vec{a}_g}(x^2) + xP_{\vec{a}_u}(x^2).$$

#### Beispiel 148

Sei  $\vec{a} = (1, 2, 4, 8)$ , also  $P_{\vec{a}}(x) = 1 + 2x + 4x^2 + 8x^3$ . Damit ist  $\vec{a}_g = (1, 4)$  und  $\vec{a}_u = (2, 8)$ , also

$$\begin{aligned} P_{\vec{a}_g}(x^2) + xP_{\vec{a}_u}(x^2) &= 1 \cdot (x^2)^0 + 4 \cdot (x^2)^1 + x \cdot (2 \cdot (x^2)^0 + 8 \cdot (x^2)^1) \\ &= 1 + 2 \cdot x + 4 \cdot x^2 + 8 \cdot x^3 \end{aligned}$$

## Lemma 149

Ist  $\mathcal{F}_{\frac{n}{2}, \omega^2}(\vec{a}_g) = (c_0, \dots, c_{\frac{n}{2}-1})$  und  $\mathcal{F}_{\frac{n}{2}, \omega^2}(\vec{a}_u) = (d_0, \dots, d_{\frac{n}{2}-1})$ , so gilt  $\mathcal{F}_{n, \omega}(\vec{a}) = (e_0, \dots, e_{n-1})$  mit

$$\begin{aligned}e_i &= P_{\vec{a}}(\omega^i) \\ &= P_{\vec{a}_g}(\omega^{2i}) + \omega^i P_{\vec{a}_u}(\omega^{2i}) \\ &= c_i + \omega^i d_i \\ e_{\frac{n}{2}+i} &= P_{\vec{a}}(\omega^{\frac{n}{2}+i}) \\ &= P_{\vec{a}_g}(\omega^{2(\frac{n}{2}+i)}) + \omega^{\frac{n}{2}+i} P_{\vec{a}_u}(\omega^{2(\frac{n}{2}+i)}) \\ &= c_i + \omega^{\frac{n}{2}+i} d_i\end{aligned}$$

für  $i = 0, \dots, \frac{n}{2} - 1$ .

Bem.:  $\omega^2$  ist primitive  $\frac{n}{2}$ -te Einheitswurzel. Natürlich ist  $\omega^{2\frac{n}{2}} = 1$ .



Dies liefert folgenden **Divide-and-Conquer**-Algorithmus:

DFT( $\vec{a}, \omega$ )

Eingabe:  $\vec{a} = (a_0, \dots, a_{n-1})$ ,  $n = 2^k$ ,  $\omega$

Ausgabe:  $\mathcal{F}_{n,\omega}(\vec{a}) = (e_0, \dots, e_{n-1})$

if  $n = 1$  then  $e_0 := a_0$

else

$\vec{a}_g := (a_0, a_2, \dots, a_{n-2})$

$\vec{a}_u := (a_1, a_3, \dots, a_{n-1})$

$(c_0, \dots, c_{\frac{n}{2}-1}) := \text{DFT}(\vec{a}_g, \omega^2)$

$(d_0, \dots, d_{\frac{n}{2}-1}) := \text{DFT}(\vec{a}_u, \omega^2)$

for  $i = 0$  to  $\frac{n}{2} - 1$  do

$e_i := c_i + \omega^i d_i$

$e_{\frac{n}{2}+i} := c_i + \omega^{\frac{n}{2}+i} d_i$

endfor

endif

return $(e_0, \dots, e_{n-1})$

## Satz 150

Der Algorithmus DFT berechnet  $\mathcal{F}_{n,\omega}(\vec{a})$  auf Eingabe  $n = 2^k$ ,  $\vec{a}$ ,  $\omega$  in  $T(n) = O(n \log n)$  Operationen.

### Beweis:

Aus dem Algorithmus erhält man folgende Rekursion

$$T(n) = 2T(n/2) + cn$$

mit einer Konstante  $c > 0$  und  $T(1) = 1$ . Mit  $n = 2^k$  folgt

$$\begin{aligned} T(2^k) &= 2T(2^{k-1}) + cn = 2(2T(2^{k-2}) + cn/2) + cn \\ &= \dots = 2^\ell T(2^{k-\ell}) + \ell cn \end{aligned}$$

Speziell für  $\ell = k$  gilt  $T(2^k) = kc2^k + 2^k T(1)$ , und wir erhalten  $T(2^k) = O(2^k k) = O(n \log n)$ . □

### 3.5.3 Berechnung der inversen diskreten Fouriertransformation

#### Satz 151

Es gilt

$$\mathcal{F}_{n,\omega}^{-1} = \frac{1}{n} \mathcal{F}_{n,\omega^{-1}}.$$

**Bemerkung:**  $\omega^{-1}$  ist ebenso eine primitive  $n$ -te Einheitswurzel.

Zum Beweis von Satz 151 benötigen wir folgendes Lemma:

#### Lemma 152

Ist  $\omega$  eine primitive  $n$ -te Einheitswurzel, so gilt

$$\sum_{j=0}^{n-1} \omega^{kj} = 0$$

für alle  $k = 1, \dots, n - 1$ .

### Beweis:

Für jedes  $a \in \mathbb{C}$ ,  $a \neq 1$ , gilt  $\sum_{j=0}^{n-1} a^j = \frac{a^n - 1}{a - 1}$ . Speziell für  $a = \omega^k$  ist  $a^n = \omega^{kn} = 1$ , ( $k = 1, \dots, n - 1$ ). □

Nun zum Beweis von Satz 151.

### Beweis:

Sei  $\vec{e} = \mathcal{F}_{n,\omega}(\vec{a}) = (e_0, \dots, e_{n-1})$ . Wir zeigen, dass gilt:

$$\frac{1}{n} \mathcal{F}_{n,\omega^{-1}}(\vec{e}) = \vec{a}$$

$$\begin{aligned} P_{\vec{e}}(\omega^{-k}) &= \sum_{j=0}^{n-1} e_j \omega^{-kj} = \sum_{j=0}^{n-1} P_{\vec{a}}(\omega^j) \omega^{-kj} \\ &= \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} a_i \omega^{ij} \omega^{-kj} = \sum_{i=0}^{n-1} a_i \sum_{j=0}^{n-1} \omega^{(i-k)j} = n a_k, \end{aligned}$$

denn nach Lemma 152 ist  $\sum_{j=0}^{n-1} \omega^{(i-k)j} = 0$ , falls  $i \neq k$ .

Im Fall  $i = k$  gilt  $\sum_{j=0}^{n-1} \omega^{(i-k)j} = n$ . □