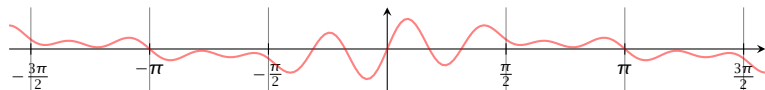
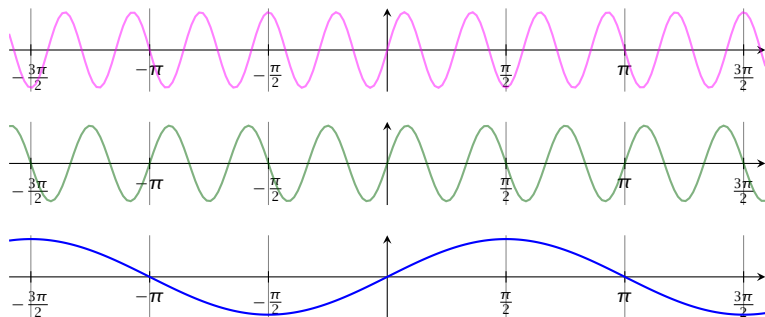


Fourieranalyse

Eingabe: Signal (periodisch)



Ausgabe: Zerlegung in Teilfrequenzen



Eingabe

- ▶ $x(t)$, periodisches Signal
- ▶ $x(t)$ ist Überlagerung von endlich vielen skalierten Sinus- und Kosinusfunktionen unterschiedlicher Frequenzen
 $\omega_s \geq 0$

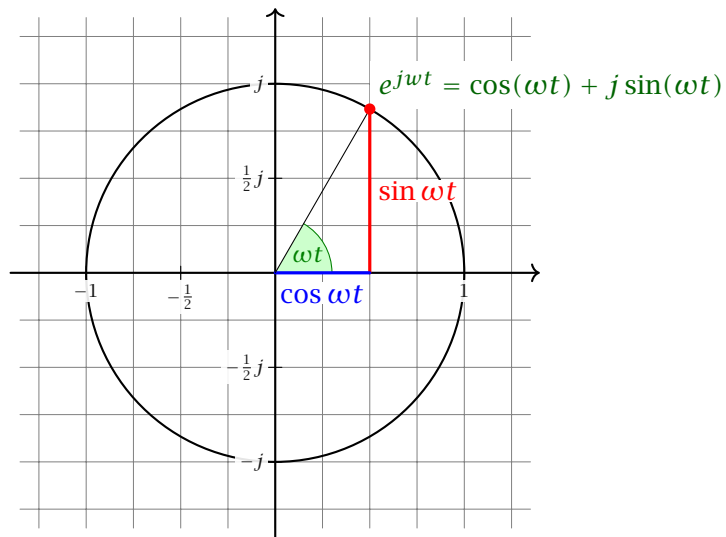
Ausgabe

- ▶ schreibe $x(t)$ als

$$x(t) = \sum_i \hat{x}[\omega_i] \cdot e^{j\omega_i t}$$

- ▶ **Beachte:** für jede Frequenz ω_s , die in $x(t)$ vorkommt gibt es in der obigen Formel zwei „Frequenzen“, $\omega_{i_1} = \omega_s$ und $\omega_{i_2} = -\omega_s$

Euler's Formula



Euler's Formula

Es gilt

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$$

Damit auch

$$\cos(\omega t) = \frac{1}{2} (e^{j\omega t} + e^{-j\omega t})$$

$$\sin(\omega t) = \frac{1}{2j} (e^{j\omega t} - e^{-j\omega t})$$

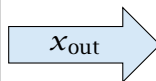
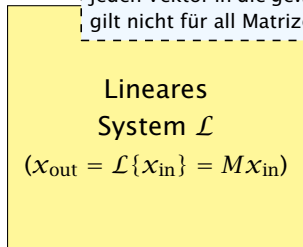
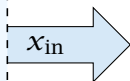
D.h. wenn mein Signal aus Überlagerung von Sinus- und Kosinusfunktionen besteht, kann ich es in die geforderte Form bringen.

Warum sollte ich das tun?

Komplexe Exponentialfunktionen sind Eigenfunktionen von linear zeitinvarianten Systemen.

Vektoren

Wenn wir Vektoren haben ($x_{\text{in}} \in \mathbb{R}^n$ und $x_{\text{out}} \in \mathbb{R}^n$) ist ein lineares System einfach eine Matrixmultiplikation mit einer $n \times n$ Matrix M .



Dieses Verfahren ist natürlich nur möglich wenn die Matrix M eine Basis von Eigenvektoren besitzt, so dass man jeden Vektor in die gewünschte Form bringen kann. Dies gilt nicht für all Matrizen.

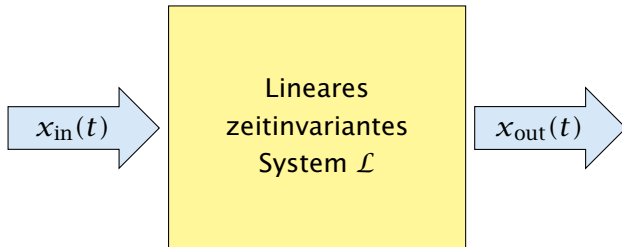
- ▶ linear: $\mathcal{L}\{\alpha x_{\text{in}}\} = \alpha \mathcal{L}\{x_{\text{in}}\}$; $\mathcal{L}\{x_{\text{in}} + y_{\text{in}}\} = \mathcal{L}\{x_{\text{in}}\} + \mathcal{L}\{y_{\text{in}}\}$
- ▶ zerlege x_{in} in **Eigenvektoren**, d.h., schreibe $x_{\text{in}} = \sum_i \alpha_i v_i$, wobei v_i Eigenvektoren von M sind.
- ▶ dann gilt:

$$x_{\text{out}} = \mathcal{L}\left\{\sum_i \alpha_i v_i\right\} = \sum_i \alpha_i \mathcal{L}\{v_i\} = \sum_i \alpha_i \lambda_i v_i$$

wobei λ_i Eigenwert zum Eigenvektor v_i .

Periodische Funktionen

$f(t)$ ist Eigenfunktion zum Eigenwert λ
falls $\mathcal{L}\{f(t)\} = \lambda f(t)$.



- ▶ zerlege $x_{\text{in}}(t)$ in **Eigenfunktionen**, d.h., schreibe $x_{\text{in}}(t) = \sum_i \alpha_i f_i(t)$, wobei f_i Eigenfunktionen von \mathcal{L} sind.
- ▶ dann gilt:

$$x_{\text{out}}(t) = \mathcal{L} \left\{ \sum_i \alpha_i f_i(t) \right\} = \sum_i \alpha_i \mathcal{L}\{f_i(t)\} = \sum_i \alpha_i \lambda_i f_i(t)$$

wobei λ_i Eigenwert zur Eigenfunktion $f_i(t)$ ist.

Das ist natürlich nur möglich, wenn wir die Funktion $x_{\text{in}}(t)$ als Summe von Eigenfunktionen von \mathcal{L} ausdrücken können.

Periodische Funktionen

Das heißt wenn wir periodische Signale haben benötigen wir nur eine Teilmenge der Eigenfunktionen. Insbesondere ist der Exponent rein imaginär.

Theorem

Komplexe Exponentialfunktionen ($f(t) = e^{st}$) sind Eigenfunktionen von linear zeitinvarianten Systemen.

Theorem

Jede periodische Funktion $x(t)$ mit Periode T (d.h., $x(t+T) = x(t)$ für alle t) läßt sich als

$$x(t) = \sum_{i=-\infty}^{\infty} \alpha_i e^{j \frac{2\pi i}{T} t}$$

schreiben. Im folgenden $w := 2\pi/T$ und $f_i(t) = e^{jwit}$.

Die periodischen Signale müssen hinreichend gutartig sein. Stetig und differenzierbar reicht z.B. aus.

Bestimmung der α_i — Vektoren

Wie schreibe ich einen Vektor x als $\sum_i \alpha_i v_i$?

Bestimme eine **Orthonormalbasis** von Eigenvektoren, v_1, \dots, v_n .

- ▶ Vektoren sind auf 1 normiert: $\|v_i\| = \langle v_i, v_i \rangle = 1$, wobei $\langle \cdot, \cdot \rangle$ das Standardskalarprodukt ist.
- ▶ Vektoren sind orthogonal, d.h. für $i \neq j$ gilt $\langle v_i, v_j \rangle = 0$.

dann gilt $\alpha_i = \langle v_i, x \rangle$, da

$$\langle v_i, x \rangle = \langle v_i, \sum_j \alpha_j v_j \rangle = \sum_j \alpha_j \langle v_i, v_j \rangle = \alpha_i$$

Das Standardskalarprodukt für einen Vektorraum über \mathbb{C} ist $\langle x, y \rangle = \sum_i \bar{x}_i y_i$.

Bestimmung der α_i — Funktionen

Wie schreibe ich ein Signal $x(t)$ als $\sum_i \alpha_i f_i(t)$?

Definiere ein Skalarprodukt, so dass $f_i(t)$'s orthonormal sind.

Auf der Menge der T -periodischen Funktionen definiert

$$\langle a(t), b(t) \rangle := \frac{1}{T} \int_0^T \overline{a(t)} b(t) dt$$

ein Skalarprodukt.

Bestimmung der α_i — Funktionen

Skalarprodukt

1. $\langle x(t), x(t) \rangle \geq 0$
2. $\langle x(t), x(t) \rangle = 0 \iff x(t) = 0$
3. $\langle x(t), y(t) \rangle = \overline{\langle y(t), x(t) \rangle}$
4. $\langle x(t), \alpha y(t) \rangle = \alpha \langle x(t), y(t) \rangle$
 $\langle x(t), y(t) + z(t) \rangle = \langle x(t), y(t) \rangle + \langle x(t), z(t) \rangle$

1., 3., und 4. folgend direkt aus Integraleigenschaften. Für 2. muß man den Funktionenraum geeignet einschränken.

Bestimmung der α_i — Funktionen

$$\begin{aligned}\langle f_i(t), f_\ell(t) \rangle &= \frac{1}{T} \int_0^T \overline{e^{j\omega i t}} e^{j\omega \ell t} dt \\ &= \frac{1}{T} \int_0^T e^{j\omega(\ell-i)t} dt\end{aligned}$$

$i = \ell$:

$$= \frac{1}{T} \int_0^T 1 dt = 1$$

$i \neq \ell$: ($\beta := j\omega(\ell - i) \neq 0$)

$$= \frac{1}{T} \int_0^T e^{\beta t} dt = \frac{1}{T} \left[\frac{1}{\beta} e^{\beta t} \right]_0^T = \frac{1}{T\beta} - \frac{1}{T\beta} = 0$$

da $e^{\beta T} = e^{j2\pi(\ell-i)} = 1$.

Bestimmung der α_i — Funktionen

Damit gilt $\alpha_i = \langle f_i(t), x(t) \rangle$, d.h.

$$\alpha_i = \frac{1}{T} \int_0^T x(t) e^{-j\omega_i t} dt$$

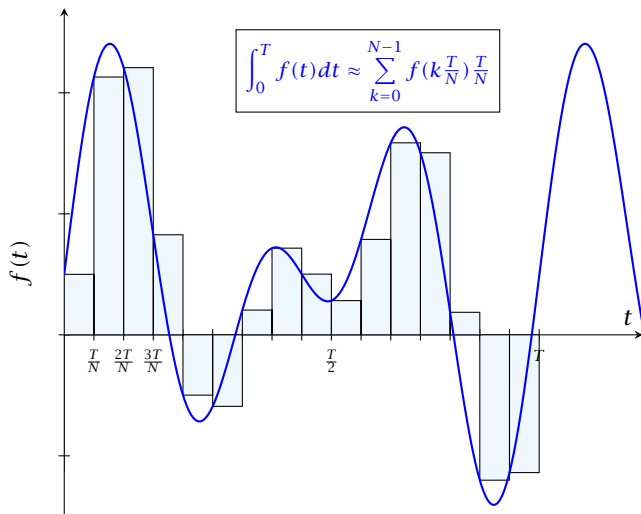
da

$$\langle f_i(t), x(t) \rangle = \langle f_i(t), \sum_{\ell} \alpha_{\ell} f_{\ell}(t) \rangle = \sum_{\ell} \alpha_{\ell} \langle f_i(t), f_{\ell}(t) \rangle = \alpha_i$$

leider muß man hier integrieren...

Wenn mein Signal $x(t)$ aus nur wenigen Frequenzen besteht, dann ist der eigentliche „Informationsinhalt“ des Signals sehr gering. Hier muß ich aber trotzdem das Signal über dem gesamten Intervall $[0, T]$ auswerten um α_i zu bestimmen. Das ist sehr ineffizient.

Approximation des Integrals



Bestimmung der α_i

Angenommen mein Signal ist

$$x(t) = \sum_{\ell} \alpha_{\ell} e^{j\omega_{\ell} t}$$

aber ich sehe nur das Gesamtsignal.

Bilde

$$\begin{aligned} \frac{1}{T} \sum_{k=0}^{N-1} x\left(\frac{T}{N}k\right) e^{-j\omega_i \frac{T}{N}k} &= \frac{1}{T} \sum_{k=0}^{N-1} \sum_{\ell} \alpha_{\ell} e^{j\omega_{\ell} \frac{T}{N}k} e^{-j\omega_i \frac{T}{N}k} \\ &= \frac{1}{T} \sum_{k=0}^{N-1} \sum_{\ell} \alpha_{\ell} e^{j\frac{2\pi}{T}\ell \frac{T}{N}k} e^{-j\frac{2\pi}{T}i \frac{T}{N}k} \\ &= \sum_i \alpha_i \frac{1}{N} \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}(\ell-i)k} \stackrel{?}{=} \alpha_i \end{aligned}$$

Wie finde ich den Koeffizienten für Frequenz ω_ℓ ?

Für $i \neq \ell \pmod N$ gilt

$$\begin{aligned}\frac{1}{N} \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}(\ell-i)k} &= \frac{1}{N} \sum_{k=0}^{N-1} q^k && \text{mit } q = e^{j\frac{2\pi}{N}(\ell-i)} \neq 1 \\ &= \frac{1}{N} \frac{q^N - 1}{q - 1} = 0 && \text{da } q^N = 1\end{aligned}$$

Für $i = \ell \pmod N$

$$\frac{1}{N} \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}(\ell-i)k} = \frac{1}{N} \sum_{k=0}^{N-1} 1 = 1$$

Sei $\frac{2\pi}{N}s$ größte Frequenz die vorkommt. Wir wählen $N \geq 2s + 1$.

Dann bekommt man α_i exakt!!!

Diskrete Fouriertransformation

Gegeben x_0, \dots, x_{N-1} mit $x_k = x\left(\frac{T}{N}k\right)$. Berechne

$$\frac{1}{N} \sum_{k=0}^{N-1} x_k e^{-j\frac{2\pi}{N}\ell k} \quad \text{für } \ell \in \{-s, \dots, s\}$$

Anstatt negativer Werte für ℓ können wir für negatives ℓ , ℓ durch $\ell' := N + \ell$ ersetzen.

Diskrete Fouriertransformation (DFT)

Gegeben x_0, \dots, x_{N-1} mit $x_k = x\left(\frac{T}{N}k\right)$. Berechne

$$\hat{x}_\ell = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{-j\frac{2\pi}{N}\ell k} \quad \text{für } \ell \in \{0, \dots, N-1\}$$

Wenn wir $N > 2s + 1$ gewählt haben berechnen wir eventuell mehr Werte als wir brauchen. Dies ist aber kein Problem.

N-te Einheitswurzeln

Definition: Einheitswurzel

$q \in \mathbb{C}$ ist **N-te Einheitswurzel**, gdw., $q^N = 1$.

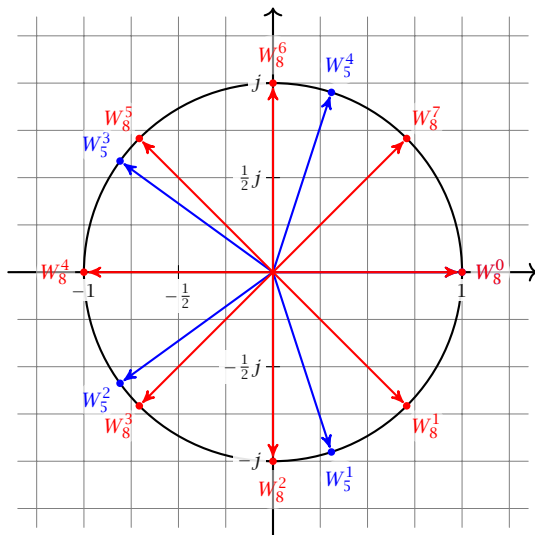
Definition: primitive Einheitswurzel

Einheitswurzel $q \in \mathbb{C}$ ist **primitiv** falls sich jede Einheitswurzel q' als $q' = q^i$ mit $i \in \mathbb{N}$ schreiben läßt.

Beispiel:

$W_N := e^{-j\frac{2\pi}{N}}$ ist primitive **N-te** Einheitswurzel.

N-te Einheitswurzeln



Diskrete Fouriertransformation

Diskrete Fouriertransformation (DFT)

Gegeben x_0, \dots, x_{N-1} mit $x_k = x\left(\frac{T}{N}k\right)$. Berechne

$$\hat{x}_\ell = \frac{1}{N} \sum_{k=0}^{N-1} x_k W_N^{\ell k} \quad \text{für } \ell \in \{0, \dots, N-1\}$$

Das heißt wir möchten im wesentlichen das durch Koeffizienten x_0, \dots, x_{N-1} definierte Polynom an den Stellen W_N^ℓ auswerten ($\ell \in \{0, \dots, N-1\}$).

Ein Polynom vom Grad $N-1$ hat die Form $\sum_{i=0}^{N-1} c_i X^i$. Hier sind die Koeffizienten $c_i = x_i$ und die Unbestimmte (Variable) ist X . Wir möchten dieses Polynom nun an den Stellen W_N^0, \dots, W_N^{N-1} auswerten. Danach müssen wir noch all Ergebnisse durch N dividieren.

Polynomauswertung

Gegeben Koeffizienten c_0, \dots, c_{N-1} , die ein Polynom

$$P[X] = \sum_{i=0}^{N-1} c_i X^i$$

definieren. Wir können $P[z]$ in Zeit $\mathcal{O}(N)$ berechnen.

Horner-Schema:

$$P[X] = c_0 + X(c_1 + X(c_2 + X(\dots X(c_{N-2} + X(c_{N-1})) \dots)))$$

```
1 evaluateHorner(C,z) // C is array of coefficients
2   res = C[N-1];
3   for i = N-2 to 0
4     res *= z;
5     res += C[i];
6   return res;
```

Diskrete Fouriertransformation

Damit können wir eine DFT in Zeit $\mathcal{O}(N^2)$ berechnen.

Geht das schneller?

JA: Polynomauswertung an N -ten Einheitswurzeln kann sehr effizient gemacht werden.

Im folgenden beschreiben wir ein Verfahren, das ein gegebenes Polynom P vom Grad $N - 1$ (gegeben durch Koeffizienten c_0, \dots, c_{N-1}) an den N -ten Einheitswurzeln, d.h. an den Stellen W_N^ℓ , $\ell \in \{0, \dots, N - 1\}$, auswertet, wobei W_N eine beliebige primitive N -te Einheitswurzel ist. Das heißt im folgenden gehen wir nicht davon aus, dass $W_N = e^{-j2\pi/N}$ ist. Um eine DFT zu erhalten muß man das folgende Verfahren zur Auswertung von Polynomen mit $W_N = e^{-j2\pi/N}$ anwenden und danach mit $1/N$ multiplizieren.

N-te Einheitswurzeln

Halbierungslemma

Sei $N \in \mathbb{N}$ gerade. Dann gilt W_N^2 ist $N/2$ -te primitive Einheitswurzel.

Beweis

- ▶ $(W_N^2)^{N/2} = W_N^N = 1$, d.h. W_N^2 ist $N/2$ -te Einheitswurzel.
- ▶ Wir wollen zeigen, dass W_N^2 primitive $N/2$ -te Einheitswurzel ist. Dafür zeigen wir, dass $(W_N^2)^\ell$ für $\ell \in \{0, \dots, N/2 - 1\}$ insgesamt $N/2$ verschiedene Werte ergibt.

Angenommen es existiert $\ell_1, \ell_2 \in \{0, \dots, N/2 - 1\}$ mit $(W_N^2)^{\ell_1} = (W_N^2)^{\ell_2}$ aber $\ell_1 \neq \ell_2$ (o.B.d.A. $\ell_1 < \ell_2$).

Dann folgt $W_N^{2(\ell_2 - \ell_1)} = 1$. Da $0 < 2(\ell_2 - \ell_1) \leq N - 1$ kann W_N damit keine primitive N -te Einheitswurzel sein (♣).

Fast Fourier Transform (FFT)

Aufgabe:

Werte Polynom P an Stellen W_N^0, \dots, W_N^{N-1} aus.

Annahme: N ist 2er-Potenz

definiere:

$$P_{\text{odd}} = c_1 + c_3X + c_5X^2 + \dots + c_{N-1}X^{\frac{N-2}{2}}$$

$$P_{\text{even}} = c_0 + c_2X + c_4X^2 + \dots + c_{N-2}X^{\frac{N-2}{2}}$$

dann gilt:

$$P[a] = P_{\text{even}}[a^2] + aP_{\text{odd}}[a^2]$$

Wir müssen z.B. P_{odd} für alle a^2 mit $a \in \{W_N^0, \dots, W_N^{N-1}\}$ berechnen, d.h. für Menge $\{(W_N^0)^2, \dots, (W_N^{N-1})^2\}$, das ist aber die Menge $\{W_{N/2}^0, \dots, W_{N/2}^{N/2-1}\}$ (Halbierungslemma).

Implementierung

```
1 Input: Polynom P of degree N-1, by coefficient array C
2 Output: R[l] contains P[W_N^l]
3
4 FFT(C,N,W_N) // C is array of coefficients
5   if (N == 1) return C;
6   C_odd = [C[1],C[3],...,C[N-1]];
7   C_even = [C[0],C[2],...,C[N-2]];
8   R_odd = FFT(C_odd,N/2,W_N^2);
9   R_even = FFT(C_even,N/2,W_N^2);
10  W = 1; // W = W_N^0
11  for l = 0 to N/2-1
12    R[l] = R_even[l] + W * R_odd[l];
13    R[l + N/2] = R_even[l] - W * R_odd[l];
14    W *= W_N; // W = W_N^l
15  return R;
```

Um die DFT zu erhalten müssen wir $W_N = e^{-j2\pi/N}$ setzen, und das Ergebnis durch N dividieren.

Korrektheit

Für $N = 1$ besteht das Polynom nur aus Koeffizient c_0 ,
d.h., bei der Auswertung ist nichts zu tun.

Nach Zeile 5/6 gilt

$$R_{\text{odd}}[\ell] = P_{\text{odd}}[W_{N/2}^{\ell}] = P_{\text{odd}}[W_N^{2\ell}] \quad \text{für } \ell = 0, \dots, N/2 - 1$$

$$R_{\text{even}}[\ell] = P_{\text{even}}[W_{N/2}^{\ell}] = P_{\text{even}}[W_N^{2\ell}] \quad \text{für } \ell = 0, \dots, N/2 - 1$$

Nach Zeile 9 gilt

$$R[\ell] = P_{\text{even}}[W_N^{2\ell}] + W_N^{\ell} \cdot P_{\text{odd}}[W_N^{2\ell}] = P[W_N^{\ell}]$$

Nach Zeile 10 gilt

$$\begin{aligned} R[\ell + N/2] &= P_{\text{even}}[W_N^{2\ell}] - W_N^{\ell} \cdot P_{\text{odd}}[W_N^{2\ell}] \\ &= P_{\text{even}}[(W_N^{\ell+N/2})^2] + W_N^{\ell+N/2} \cdot P_{\text{odd}}[(W_N^{\ell+N/2})^2] \\ &= P[W_N^{\ell+N/2}] \end{aligned}$$

Beachte, dass $W_N^{N/2} = -1$.

Fast Fourier Transform (FFT)

Laufzeit:

- ▶ Rekurrenzgleichung: $T(n) = 2T(n/2) + \mathcal{O}(n)$
- ▶ Mastertheorem ergibt: $T(n) = \mathcal{O}(n \log n)$

Diskrete Fouriertransformation

Hier verwenden wir wieder $W_N = e^{-j2\pi/N}$, d.h., W_N ist keine beliebige Einheitswurzel.

Eine DFT berechnet das folgende Matrix-Vektor Produkt.

$$\frac{1}{N} \underbrace{\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-2)} & \cdots & W_N^{(N-1)^2} \end{pmatrix}}_{V_N} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} \hat{x}_0 \\ \hat{x}_1 \\ \vdots \\ \hat{x}_{N-1} \end{pmatrix}$$

V_N heißt Vandermonde matrix. Die Inverse Matrix V_N^{-1} ist gegeben durch

$$V_N^{-1} = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-2)} & \cdots & W_N^{-(N-1)^2} \end{pmatrix}$$

Diskrete Fouriertransformation

Beweis:

Das Element in Zeile i und Spalte j in $V_N \cdot V_N^{-1}$ ist gegeben durch

$$\frac{1}{N} \sum_{k=0}^{N-1} (W_N^i)^k \cdot (W_N^{-j})^k = \frac{1}{N} \sum_{k=0}^{N-1} (W_N^{i-j})^k = \begin{cases} 1 & i = j \pmod N \\ 0 & i \neq j \pmod N \end{cases}$$

Polynominterpolation

Angenommen wir haben ein Polynom an Stellen W_N^0, \dots, W_N^{N-1} ausgewertet. Mit Hilfe von V_N^{-1} können wir die Koeffizienten des Polynoms herausfinden:

$$\underbrace{\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-2)} & \cdots & W_N^{-(N-1)^2} \end{pmatrix}}_{N \cdot V_N^{-1}} \begin{pmatrix} P[W_N^0] \\ P[W_N^1] \\ \vdots \\ P[W_N^{N-1}] \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{pmatrix}$$

Um das Matrix-Vektor-Produkt auszurechnen müssen wir den FFT-Algorithmus mit $-W_N$ benutzen; wobei $P[W_N^0], \dots, P[W_N^{N-1}]$ die Koeffizienten des auszuwertenden Polynoms sind.

Polynome

Man kann Polynome als Funktionen auffassen; z.B. $f : \mathbb{R} \rightarrow \mathbb{R}, f(x) \mapsto P[x]$. Allgemeiner ist es aber Polynome als algebraische Objekte aufzufassen. Auf diesen Objekten gibt es Operationen wie z.B. Auswertung, Multiplikation, Addition, oder Division.

Ein formaler Ausdruck der Form

$$P[X] = c_0 + c_1X + c_2X^2 + \dots + c_{n-1}X^{n-1}$$

mit $c_i \in \mathbb{R}$ und $c_{n-1} \neq 0$ heißt **Polynom** vom **Grad** $n - 1$. Die c_i heißen **Koeffizienten** des Polynoms.

- ▶ $\text{grad}(P)$ bezeichnet den Grad des Polynoms;

Operationen auf Polynomen

Seien $A[X], B[X]$ Polynome

- ▶ **Auswertung:** berechne für $x_0 \in \mathbb{R}$

$$A[x_0]$$

- ▶ **Addition:** berechne Polynom $C[X]$, so daß für alle $x \in \mathbb{R}$

$$C[x] = A[x] \cdot B[x]$$

- ▶ **Multiplikation:** berechne Polynom $C[X]$, so daß für alle $x \in \mathbb{R}$

$$C[x] = A[x] + B[x]$$

Repräsentation von Polynomen

Koeffizientendarstellung

Wir repräsentieren das Polynom durch die Koeffizienten

c_0, \dots, c_{n-1} .

Repräsentation von Polynomen

Theorem:

Ein Polynom vom Grad $n - 1$ ist durch Funktionswerte an n paarweise verschiedenen Stellen eindeutig definiert.

Stützstellendarstellung

Wir wählen n paarweise verschiedene Stützstellen x_0, \dots, x_{n-1} .

Wir repräsentieren das Polynom P durch die Funktionswerte $P[x_0], \dots, P[x_{n-1}]$.

Die Stützstellendarstellung ist nicht eindeutig. Jede Menge von verschiedenen Punkten x_0, \dots, x_{n-1} kann verwendet werden.

Auswertung von Polynomen

Koeffizientendarstellung

Mit dem Horner Schema können wir $P[z]$ in Zeit $\mathcal{O}(n)$ berechnen.

Stützstellendarstellung bzgl. x_0, \dots, x_{n-1}

nicht so einfach...

Addition von Polynomen

Koeffizientendarstellung

Seien a_0, \dots, a_{n-1} Koeffizienten von $A[X]$ und b_0, \dots, b_{n-1} Koeffizienten von $B[X]$.

Koeffizienten von $C[X] = A[X] + B[X]$ sind gegeben durch $c_i = a_i + b_i$, d.h. wir benötigen n Additionen.

Stützstellendarstellung bzgl. x_0, \dots, x_{n-1}

Sei $A[x_i] = a_i$ und $B[x_i] = b_i$. $C[x_i] = a_i + b_i$ (direkt aus der Definition). D.h. wir benötigen n Additionen.

Multiplikation von Polynomen

Stützstellendarstellung bzgl. x_0, \dots, x_{n-1}

Sei $A[x_i] = a_i$ und $B[x_i] = b_i$. $C[x_i] = a_i \cdot b_i$ (direkt aus der Definition). D.h. wir benötigen n Multiplikationen.

Koeffizientendarstellung

Seien a_0, \dots, a_{n-1} Koeffizienten von $A[X]$ und b_0, \dots, b_{n-1} Koeffizienten von $B[X]$.

Koeffizienten von $C[X] = A[X] \cdot B[X]$ sind gegeben durch

$$c_i = \sum_{k=0}^i a_k b_{i-k}$$

Für Element c_i benötigt man $i + 1$ Multiplikationen und i Additionen. Insgesamt Aufwand $\mathcal{O}(n^2)$.

Transformation eines Polynoms von **Stützstellendarstellung** in **Koeffizientendarstellung** heißt **Polynominterpolation**.

Interpolation

Stützstellen x_0, \dots, x_{n-1} beliebig. Sei $y_k = A[x_k]$.

Lagrange-Interpolationsformel:

$$A[X] = \sum_{k=0}^{n-1} y_k \frac{\prod_{j \neq k} (X - x_j)}{\prod_{j \neq k} (x_k - x_j)}$$

Aufwand: $\mathcal{O}(n^2)$ durch geschickte Reihenfolge der Operationen.

Es ist klar, dass die Formel ein Polynom vom Grad höchstens $n-1$ liefert. Außerdem sieht man leicht, dass $A[x_k] = y_k$ ist. Also ergibt die Formel das gesuchte Polynom.

Beobachtung

Wenn wir komplexe Einheitswurzeln als Stützstellen verwenden können wir in Zeit $\mathcal{O}(n \log n)$ zwischen Koeffizientendarstellung und Stützstellendarstellung wechseln!!!!!!!!!!!!

Wir können Polynome in Koeffizientendarstellung in Zeit $\mathcal{O}(n \log n)$ multiplizieren.

- ▶ berechne Stützstellendarstellung bzgl. Einheitswurzeln (Zeit $\mathcal{O}(n \log n)$ via FFT)
- ▶ berechne Produkt in Stützstellendarstellung: Zeit (n)
- ▶ berechne Koeffizientendarstellung aus Stützstellendarstellung (Zeit $\mathcal{O}(n \log n)$ via inverser FFT)