Korrektur Aufgabe 8.1

Anstatt x^2-2x+3

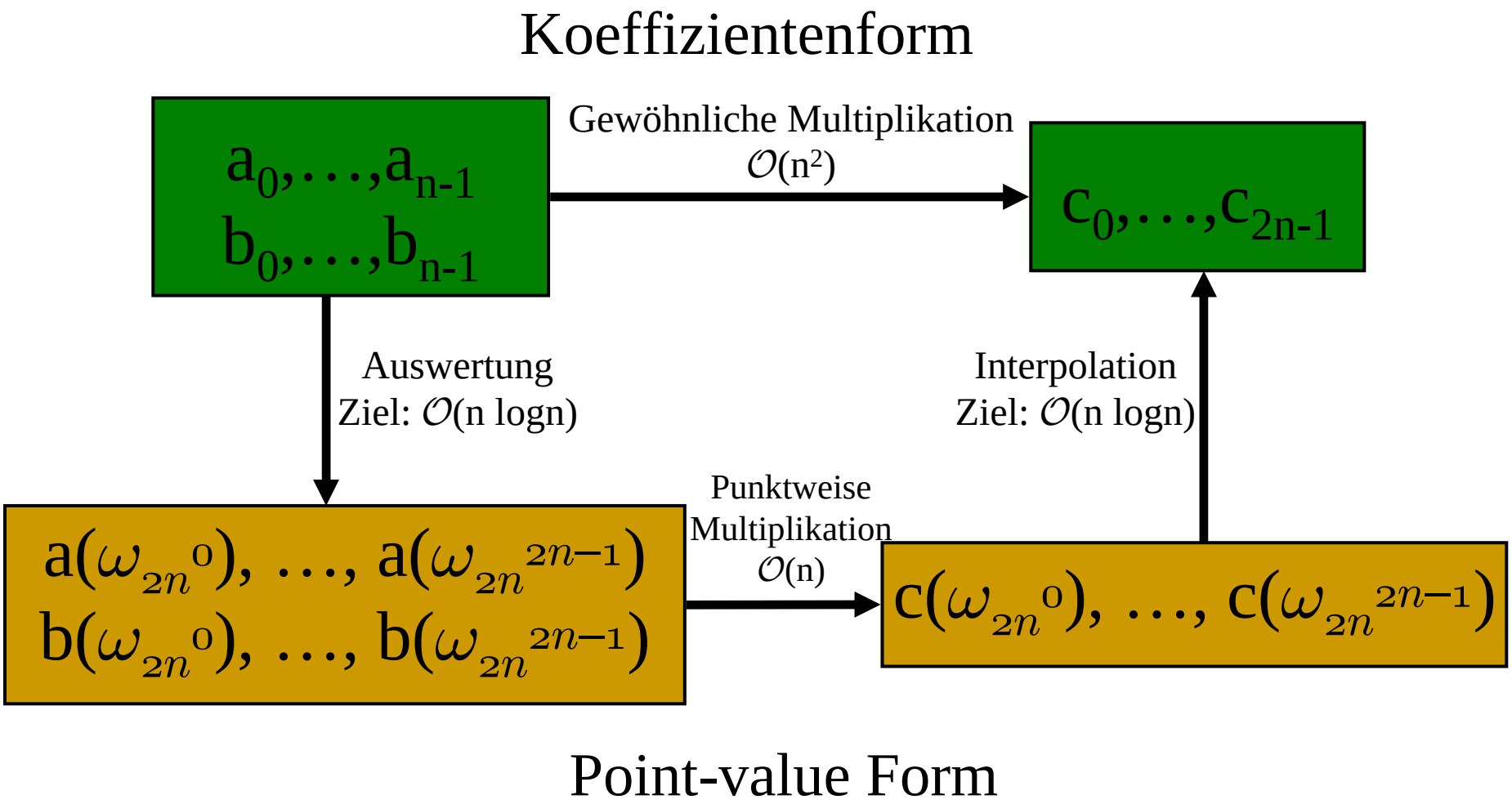
muss es heissen $x^2\mathbf{-4x}+3$.

Wiederholung

Arithmetik auf Polynomen mit Grad n

- Polynome in Koeffizientendarstellung
  - Auswerten: $\mathcal{O}(n)$ mit Horner-Schema
  - Addition/Subtraktion: $\mathcal{O}(n)$
  - Multiplikation/Division: $\mathcal{O}(n^2)$ mit Schulmethode
- Polynome in Point-value Form
  - Interpolation: $\mathcal{O}(n^2)$ mit Lagrange-Interpolation
  - Addition/Subtraktion: $\mathcal{O}(n)$
  - Multiplikation: $\mathcal{O}(n)$

Multiplizieren von Polynomen



n-te Einheitswurzeln

Def: Eine n-te Einheitswurzel ist eine komplexe Zahl $\omega \in \mathbb{C}$ mit der Eigenschaft $\omega^n = 1$

- Es gibt exakt n viele n-te Einheitswurzeln

  $e^{2\pi ik/n}$ für k=0,1,…,n-1
- Erinnerung: $e^{i\alpha} = \cos(\alpha) + i*\sin(\alpha)$
  - n-te Einheitswurzeln besitzen gleichen Abstand auf dem Einheitskreis in der komplexen Ebene
- $\omega_n = e^{2\pi i/n}$ heißt primitive n-te Einheitswurzel
  - Alle n-ten Einheitswurzeln sind Potenzen von ω_n

Gruppeneigenschaft

Satz: ($<\omega_n>$,*) ist eine multiplikative, zyklische Gruppe der Ordnung n.

- Neutrales Element: 1.
- Ordnung: $\omega_n^n = e^{2\pi i} = 1$.
- Abgeschlossenheit: $\omega_n^i * \omega_n^j = \omega_n^{i+j \bmod n}$
- Inverses zu ω_n^i ist ω_n^{n-i}.

Eliminations- und Halbierungslemma

Eliminationslemma: Für alle n,k $\in \mathbb{N}_0$ und d $\in \mathbb{N}$:

$$\omega_{dn}^{dk} = \omega_n^k$$

$\omega_{dn}^{dk} = (e^{2\pi i/dn})^{dk} = (e^{2\pi\, i/n})^k = \omega_n^k$

Korollar: Für jede gerade Zahl n $\in \mathbb{N}$: $\omega_n^{n/2} = \omega_2 = -1$

- $\omega_2 = e^{\pi i} = \cos(\pi) + i\sin(\pi) = (-1)$

Halbierungslemma: Sei n>0 gerade. Dann sind die Quadrate der n-ten Einheitsheitswurzeln gerade die n/2-ten Einheitswurzeln.

- Eliminationslemma: $(\omega_n^k)^2 = \omega_{n/2}^k$.
- Jede zwei n-te Einheitswurzeln haben dasselbe Quadrat:
  $(\omega_n^k)^2 = (-\omega_n^k)^2 = (\omega_n^{n/2} * \omega_n^k)^2 = (\omega_n^{k+n/2})^2$

Discrete Fourier Transform (DTF)

ObdA: grad(a) < n = 2^k

- a(x) sei als Koeffizientenvektor a=(a_0,…,a_{n-1}) gegeben
  - a_g = (a_0, a_2, a_4, …, a_{n-2})
  - a_u = (a_1, a_3, a_5, …, $a_{\mathbf{n-1}}$)
- Werten a(x) an den Stützstellen ω_n^k für k=0,…,n-1 aus
- Sei $y_k = a(\omega_n^k) = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$ für k=0,…,n-1

- y = (y_0,…,y_{n-1}) ist die **diskrete Fouriertransformierte** von a=(a_0,…,a_{n-1})
- Schreibweise: $y = DFT_n(a, \omega)$

Fast Fourier Transform (FFT)

Splitte Polynom a(x) in zwei Teile

1. $a_g(x) = a_0 + a_2x + a_4x^2 + \ldots + a_{n-2}x^{n/2-2}$ bzw. $a^g = (a_0, a_2, a_4, \ldots, a_{n-2})$
2. $a_u(x) = a_1 + a_3x + a_5x^2 + \ldots + a_{n-1}x^{n/2-1}$ bzw. $a^u = (a_1, a_3, a_5, \ldots, a_{n-2})$

Dann gilt:

$$\mathbf{a(x) = a_g(x^2) + x*a_u(x^2)}$$

Berechnen von $DFT_n(a)$: Auswerten von a(x) an $\omega_n^0,\ldots,\omega_n^{n-1}$

- Auswerten der Polynome a_g, a_u vom Grad < n/2 an den Stellen
  - $(\omega_n^0)^2 = \omega_{n/2}^0$, $(\omega_n^1)^2 = \omega_{n/2}^1$, …, $(\omega_n^{n/2-1})^2 = \omega_{n/2}^{n/2-1}$,
    $(\omega_n^{n/2})^2 = \omega_{n/2}^0$, $(\omega_n^{n/2+1})^2 = \omega_{n/2}^1$, …, $(\omega_n^{n-1})^2 = \omega_{n/2}^{n/2-1}$
  - D.h. es genügt a_g und a_u an n/2 vielen Stellen auszuwerten.
  - Berechnen von $DFT_{n/2}(a^g,\omega)$ und $DFT_{n/2}(a^u,\omega)$.
- Kombinieren von a(x) mittels a_g und a_u.
  - $y = DFT(a)$ mit $y_k = a(\omega_n^k) = a_g(\omega_{n/2}^k) + \omega_n^k * a_u(\omega_{n/2}^k) = a_g(\omega_n^{2k}) + \omega_n^k * a_u(\omega_n^{2k})$

Laufzeit: $T(n) = 2*T(n/2) + \mathcal{O}(n)$, $T(1) = \mathcal{O}(1)$.

$\Rightarrow$ **$T(n) = \mathcal{O}(n \log n)$** (Übungsaufgabe)

FFT-Algorithmus

Algorithmus FFT

EINGABE: $a=(a_0,\ldots,a_{n-1})$, $n=2^k$

1. If (n=1) return a
2. $\omega_n \leftarrow e^{2\pi i/n}$; $\omega \leftarrow 1$
3. $(y_1^g, y_2^g, \ldots, y_{n/2}^g) \leftarrow FFT(a_0, a_2, \ldots, a_{n-2}, n/2)$
4. $(y_1^u, y_2^u, \ldots, y_{n/2}^u) \leftarrow FFT(a_1, a_3, \ldots, a_{n-1}, n/2)$
5. for k = 0 to n/2-1
    1. $y_k \leftarrow y_k^g + \omega\, y_k^u$
    2. $y_{k+(n/2)} \leftarrow y_k^g - \omega\, y_k^u$
    3. $\omega \leftarrow \omega^*\omega_n$

AUSGABE: $y=(y_0,\ldots,y_{n-1})$

Korrektheit

- Schritt 1:
  - $y_0 = a(\omega_1) = a_0 * \omega_1^0 = a_0$
- Schritt 2+5.2:
  - Update von $\omega = \omega_n^k$.
- Schritt 3+4:
  - $(y_1^g, y_2^g, \ldots, y_{n/2}^g) = DFT_{n/2}(a_g, \omega)$, d.h. $y_k^g = a_g(\omega_{n/2}^k) = a_g(\omega_n^{2k})$
  - $(y_1^u, y_2^u, \ldots, y_{n/2}^u) = DFT_{n/2}(a_u, \omega)$, d.h. $y_k^u = a_u(\omega_{n/2}^k) = a_u(\omega_n^{2k})$
- Schritt 5.1:
  - $y_k = y_k^g + \omega_n^k * y_k^u = a_g(\omega_n^{2k}) + \omega_n^k * a_u(\omega_n^{2k}) = a(\omega_n^k)$ für k=0,…,n/2-1
- Schritt 5.2:
  - $y_{k+n/2} = y_k^g - \omega_n^k y_k^u = y_k^g + \omega_n^{k+n/2} y_k^u$
    $= a_g(\omega_n^{2k}) + \omega_n^{k+n/2} a_u(\omega_n^{2k})$
    $= a_g(\omega_n^{2k+n}) + \omega_n^{k+n/2} a_u(\omega_n^{2k+n})$
    $= a(\omega_n^{k+n/2})$ für k+n/2=n/2,…,n-1

Summationslemma

Lemma: Für alle n,i $\in \mathbb{N}$ mit n $\nmid$ i

$$\sum_{k=0}^{n-1} (\omega_n^i)^k = 0.$$

$$\begin{aligned}
\sum_{k=0}^{n-1} (\omega_n^i)^k &= \frac{(\omega_n^i)^n - 1}{\omega_n^i - 1} \\
&= \frac{(\omega_n^n)^i - 1}{\omega_n^i - 1} \\
&= \frac{1^i - 1}{\omega_n^i - 1} = 0.
\end{aligned}$$

Inverse Diskrete Fourier Transformation

Interpolation: Point-value $\rightarrow$ Koeffizientenform
Darstellung der DFT als Matrix-Vektor Produkt:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

- Vandermonde-Matrix V mit Einträgen $v_{ij} = \omega_n^{ij}$ für $0 \leq i,j \leq n-1$
- Man beachte: $V = V^T$
- Inverse Diskrete Fourier Transformation
  - $a = DFT_n^{-1}(y, \omega) = V^{-1}y$

Invertieren von V

Satz: Sei $V^{-1}=(v'_{i,j})$ für $1 \leq i,j \leq n$. Dann gilt $v'_{i,j} = 1/n * \omega_n^{-ij}$.

- Zeigen $V*V^{-1} = I_n$, die $(n \times n)$-Einheitsmatrix
- Betrachten (i,j)-Eintrag von $V*V^{-1}$
  - $\sum_{k=0}^{n-1} \omega_n^{ik} * 1/n*\omega_n^{-kj} = 1/n \sum_{k=0}^{n-1} \omega_n^{ki} * \omega_n^{-kj}$
    $= 1/n \sum_{k=0}^{n-1} \omega_n^{k(i-j)}$
  - Für i = j liefert die Summation 1.
  - Fall $i \neq j$:
    $|i-j| < n-1$, d.h. $n \nmid i-j$
    $\Rightarrow \sum_{k=0}^{n-1} (\omega_n^{i-j})^k = 0$ (Summationslemma)

Korollar: $a = DFT_n^{-1}(y, \omega) = 1/n*DFT_n(y, \omega^{-1})$. D.h. die inverse diskrete Fouriertransformation lässt sich in Zeit $\mathcal{O}$(n logn) berechnen.

Multiplizieren von Polynomen