12.2 Gauß-Quadratur

Erinnerung: Mit der Newton-Cotes Quadratur

$$I_n[f] = \sum_{i=0}^{n} g_i f(x_i) \approx I[f] = \int_a^b f(x)\,dx$$

werden Polynome vom Grad n exakt integriert.

Dabei sind die Knoten x_i, $0 \leq i \leq n$, äquidistant auf $[a, b]$ verteilt.

Grundidee der Gauß-Quadratur: Variiere die Knoten $x_0, \ldots, x_n$.

Grundidee der Gauß-Quadratur.

Ziel:

Variiere Knoten, um Polynome möglichst hohen Grades exakt zu integrieren.

Genauer:

Approximiere für eine feste positive Gewichtsfunktion $w : (a, b) \to (0, \infty)$ Integrale der Form

$$I[f] = \int_a^b f(x)w(x)\,dx$$

durch Quadratur der Form

$$I[f] \approx \sum_{i=0}^{n} f(x_i)w_i$$

mit einer speziellen Wahl von Stützstellen x_i und positiven Gewichten w_i.

Ergebnis: Gaußsche Quadraturformeln mit $(n+1)$ Knoten integrieren Polynome vom Grad $2n+1$ exakt. □

Beispiel: Gauß-Tschebyscheff-Quadratur.

- **Integrationsintervall:** $I = [-1, 1]$
- **Gewichtsfunktion:** $w(x) = 1/\sqrt{1-x^2}$.
- **Knoten:** Nullstellen

$$x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right) \qquad \text{für } 0 \le i \le n$$

  des $(n+1)$-ten Tschebyscheff-Polynoms

$$T_{n+1}(x) = \cos((n+1)\arccos(x)) \in \mathcal{P}_{n+1} \qquad \text{für } x \in [-1, 1].$$

- **Konstante Gewichte:** $w_i \equiv \pi/(n+1)$.
- Gauß-Tschebyscheff Quadratur:

$$I_n[f] = \frac{\pi}{n+1} \sum_{i=0}^{n} f(x_i) \approx I_w[f] = \int_{-1}^{1} f(x) w(x)\, dx.$$

Eigenschaften der Tschebyscheff-Polynome.

Satz: *Die Tschebyscheff-Polynome* $T_0, \dots, T_n$ *bilden eine orthogonale Basis des Polynomraums* $\mathcal{P}_n$ *bezüglich des gewichteten Skalarprodukts*

$$(f, g)_w := \int_{-1}^{1} f(x)g(x)w(x)\,dx.$$

Genauer gilt:

$$(T_k, T_j)_w = \begin{cases} \pi & \textit{für } k = j = 0 \\ \pi/2 & \textit{für } k = j > 0 \\ 0 & \textit{für } j \neq k \end{cases}$$

Beweis: Übung (mit Substitution $t = \cos(x)$) □

Satz: *Für die Tschebyscheff-Polynome gilt die Rekursionsformel*

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x) \qquad \textit{für } k \geq 1,$$

wobei $T_0 \equiv 1$ *und* $T_1(x) = x.$ □

Legendre-Polynome.

Satz: *Für die Gewichtsfunktion* $w \equiv 1$ *auf dem Intervall* $I = [-1, 1]$ *sind die* Legendre-Polynome

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}(x^2 - 1)^n \in \mathcal{P}_n$$

Orthogonalpolynome. Genauer gilt:

$$(L_n, L_m) = \begin{cases} \frac{2}{2n+1} & \text{für } n = m \geq 0 \\ 0 & \text{für } n \neq m \end{cases}$$

■

Beweis: Übung (per Induktion). □

Satz: *Für die Legendre-Polynome gilt die Rekursionsformel*

$$L_{n+1}(x) = \frac{2n+1}{n+1} x L_n(x) - \frac{n}{n+1} L_{n-1} \qquad \text{für } n \geq 1,$$

wobei $L_0 \equiv 1$ *und* $L_1(x) = x$. □

Weitere Eigenschaften der Legendre-Polynome.

Die ersten Legendre-Polynome sind gegeben durch

$$\begin{aligned} L_0(x) &\equiv 1 \\ L_1(x) &= x \\ L_2(x) &= (3x^2 - 1)/2 \\ L_3(x) &= (5x^3 - 3x)/2 \\ L_4(x) &= (35x^4 - 30x^2 + 3)/8 \end{aligned}$$

Deren jeweilige Nullstellen sind gegeben durch

$$\begin{aligned} L_1 &: \quad x_0 = 0 \\ L_2 &: \quad x_{0/1} = \pm\sqrt{1/3} \\ L_3 &: \quad x_0 = 0, x_{1/2} = \pm\sqrt{3/5} \\ L_4 &: \quad x_{0/1/2/3} = \pm\sqrt{\frac{3}{7} \pm \frac{1}{7}\sqrt{\frac{24}{5}}} \end{aligned}$$

Zur Konstruktion der Gauß-Legendre-Quadratur.

- **Integrationsintervall:** $I = [-1, 1]$
- **Gewichtsfunktion:** $w(x) \equiv 1$.
- **Knoten:** $n + 1$ Nullstellen $x_0, \ldots, x_n$ des Legendre-Polynoms $L_{n+1} \in \mathcal{P}_{n+1}$.
- **Gewichte:** Mit festen Knoten $x_0, \ldots, x_n$ zu berechnen aus

$$w_i = \int_{-1}^{1} \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \, dx > 0.$$

- Gauß-Legendre Quadratur:

$$I_n[f] = \sum_{i=0}^{n} w_i f(x_i) \approx I[f] = \int_{-1}^{1} f(x) \, dx.$$

Weitere Spezialfälle der Gauß-Quadratur.

Name	Intervall	Gewicht
Gauß-Legendre	$[-1, 1]$	$w \equiv 1$
Gauß-Tschebyscheff	$[-1, 1]$	$w(x) = 1/\sqrt{1 - x^2}$
Gauß-Jacobi	$[-1, 1]$	$w(x) = (1 - x)(1 + x)$
Gauß-Laguerre	$[0, \infty)$	$w(x) = e^{-x}$
Gauß-Hermite	$(-\infty, \infty)$	$w(x) = e^{-x^2}$

Zur Konstruktion von Gauß-Quadraturformeln.

- Konstruiere zu festem Intervall $[a, b]$ und Gewichtsfunktion w eine Folge

$$p_0, p_1, \ldots, p_n, p_{n+1}$$

von Orthogonalpolynomen, wobei $p_k \in \mathcal{P}_k$ und $(p_k, p_j)_w = \delta_{jk}$.

- Verwende Nullstellen $x_0, x_1, \ldots, x_n$ von p_{n+1} als Knoten.

- Berechne (positive) Gewichte

$$w_i = \int_{-1}^{1} \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \, dx \qquad \text{für } 0 \leq i \leq n.$$

- **Ergebnis:** Gauß-Quadraturformel

$$I_n[f] = \sum_{i=0}^{n} w_i f(x_i) \approx I[f] = \int_a^b f(x) w(x) \, dx$$

mit $I_n[f] = I_n[p]$ für alle $p \in \mathcal{P}_{2n+1}$.

13 Die Schnelle Fourier-Transformation

Ziel: Effiziente Berechnung der Diskreten Fourier-Transformation (DFT)

$$\hat{z}(m) = \sum_{n=0}^{N-1} z(n)\omega_N^{nm} \qquad \text{für } 0 \le m \le N-1,$$

wobei

$$\omega_N := e^{-2\pi i/N}.$$

- **Methode** (COOLEY & TUKEY, 1965)**:**

Schnelle Fourier-Transformation, "Fast Fourier-Transformation" (FFT).

- **INPUT:** Vektor

$$z = (z(0), z(1), \ldots, z(N-1))^T \in \mathbb{C}^N$$

- **OUTPUT:** $\hat{z}$, die Diskrete Fourier-Transformation (DFT) von z:

$$\hat{z} = (\hat{z}(0), \hat{z}(1), \ldots, \hat{z}(N-1))^T \in \mathbb{C}^N$$

Grundidee der schnellen Fourier-Transformation.

Wichtige Beobachtung: Es gilt $\omega_{2N}^2 = \omega_N$.

Divide and Conquer: Für $N = 2^k$ und $0 \leq m \leq N-1$ gilt

$$
\begin{aligned}
\hat{z}(m) &= \sum_{n=0}^{N-1} z(n)\omega_N^{mn} \\
&= \sum_{n \text{ gerade}} z(n)\omega_N^{mn} + \sum_{n \text{ ungerade}} z(n)\omega_N^{mn} \\
&= \sum_{n=0}^{N/2-1} z(2n)\omega_N^{2mn} + \sum_{n=0}^{N/2-1} z(2n+1)\omega_N^{(2n+1)m} \\
&= \sum_{n=0}^{N/2-1} z(2n)\omega_N^{2mn} + \omega_N^m \sum_{n=0}^{N/2-1} z(2n+1)\omega_N^{2mn}.
\end{aligned}
$$

Reduktionsschritt.

Sei $M = N/2$. Dann gilt für $m = 0, 1, \ldots, N-1$

$$\begin{aligned}
\hat{z}(m) &= \sum_{n=0}^{M-1} z(2n)\omega_N^{2mn} + \omega_N^m \sum_{n=0}^{M-1} z(2n+1)\omega_N^{2mn} \\
&= \sum_{n=0}^{M-1} u(n)\omega_{N/2}^{mn} + \omega_N^m \sum_{n=0}^{M-1} v(n)\omega_{N/2}^{mn} \\
&= \sum_{n=0}^{M-1} u(n)\omega_M^{mn} + \omega_N^m \sum_{n=0}^{M-1} v(n)\omega_M^{mn},
\end{aligned}$$

wobei $u(n) := z(2n)$ und $v(n) := z(2n+1)$ für $n = 0, 1, \ldots, M-1$.

Es gilt:

$$\begin{aligned}
\hat{z}(m) &= \hat{u}(m) + \omega_N^m \hat{v}(m) && \text{für } m = 0, 1, \ldots, M-1. \\
\hat{z}(m) &= \hat{u}(\ell) - \omega_N^\ell \hat{v}(\ell) && \text{für } m = M, M+1, \ldots, N-1,\ m = \ell + M.
\end{aligned}$$

Komplexität der schnellen Fourier-Transformation.

Fazit: Die Diskrete Fourier-Transformation von $z \in \mathbb{C}^N$ schreibt sich als Summe zweier Diskreter Fourier-Transformationen der Länge $N/2$.

Satz: *Die schnelle Fourier-Transformation von $z \in \mathbb{C}^N$ kann für $N = 2^k$ in $\mathcal{O}(N \log(N))$ Schritten berechnet werden.*

Beweisskizze:

- Zerlege FFT von z der Länge N in zwei FFTs der Länge $N/2$.
- Per Induktion: Zerlege FFT der Länge $N/2^j$ in zwei FFTs der Länge $N/2^{j+1}$.
- Es gilt $N = 2^k$, d.h. $k = \log_2(N)$.
- Daher bleiben nach $j = k$ Schritten nur noch N FFTs der Länge Eins übrig.
- Nun gilt $\hat{z}(0) = z(0)$ für $z \in \mathbb{C}^1$, d.h. konstante Kosten $\mathcal{O}(1)$ für $N = 1$.
- Man bekommt $\hat{z} \in \mathbb{C}^N$ mit dieser Rekursion nach $N \log_2(N)$ Schritten. $\square$

Schnelle Fourier-Transformation mit Matlab.

- Berechne Fourier-Transformation $w = \hat{z} \in \mathbb{C}^N$ aus $z \in \mathbb{C}^N$ mit Matlab.

```
w = fft(z);
```

- Berechne inverse FFT (IFFT) $z \in \mathbb{C}^N$ aus $w = \hat{z} \in \mathbb{C}^N$ mit

```
z = ifft(w);
```

Grundlage der IFFT: Die Inversionsformel

$$z(n) = \frac{1}{N} \sum_{m=0}^{N-1} \hat{z}(m) e^{2\pi i m n / N} \qquad \text{für } 0 \leq n \leq N-1.$$

$\square$