Projekt 8: Bernsteinpolynome

Für jedes $n \in \mathbb{N}_0$ sind die Bernsteinpolynome definiert als

$$B_j^n(x) = \binom{n}{j} x^j (1-x)^{n-j}, \qquad j = 0, \dots, n, \qquad\qquad B_j^n = 0 \qquad \text{für } j > n \text{ oder } j < 0$$

Bernsteinpolynome haben viele Anwendungen, z.B. werden Bézierkurven mittels Bernsteinpolynomen definiert. Bézierkurven werden zu Beschreibung von "Freiformflächen" verwenden. Eine konkrete Anwendung ist die Definition von Buchstaben in vielen Schriftarten. Ziel des Projektes ist es, einige wichtige Eigenschaften der Bernsteinpolynome zu zeigen.

1. Aus der Analysis ist bekannt, daß die Menge der Polynome dicht in $C([0,1])$ liegt. Sinn dieser Aufgabe ist ein *konstruktiver* Beweis dieser Aussage.

Sei $\mathcal{B}_n : C([0,1]) \to \mathbb{P}_n$ definiert durch $\mathcal{B}_n(f) = \sum_{i=0}^n f(\frac{i}{n}) B_i^n$

a) Seien die Monome $m_i \in \mathbb{P}_i$ gegeben durch $m_i(x) := x^i$. Zeigen Sie: $\mathcal{B}_n(m_0) = m_0$ und $\mathcal{B}_n(m_1) = m_1$ für $n \geq 1$. Zeigen Sie $(\mathcal{B}_n(m_2))(x) = x^2 + \frac{x(1-x)}{n}$ für $n \geq 2$.

b) Zeigen Sie: Für jedes $\delta > 0$, $\overline{x} \in [0,1]$, $n \geq 2$ gilt

$$\sum_{\substack{i \in \{0,\dots,n\} \\ \left|\frac{i}{n} - \overline{x}\right| \geq \delta}} B_i^n(\overline{x}) \leq \frac{\overline{x}(1-\overline{x})}{\delta^2 n}$$

Hinweis: Betrachten Sie $\mathcal{B}_n(f)$ für ein geeignetes $f \in \mathcal{P}_2$.

c) Zeigen Sie, daß für jedes $f \in C([0,1])$ die Folge $f_n := \mathcal{B}_n(f) \in \mathbb{P}_n$ gleichmäßig gegen f konvergiert, d.h. $\lim_{n\to\infty} \|f - f_n\|_{C([0,1])} = 0$.

2. Der Bernsteinapproximationsoperator $\mathcal{B}_n$ hat auch Positivitätseigenschaften:

1. $f \geq 0$ auf $[0,1]$ impliziert $\mathcal{B}_n f \geq 0$ auf $[0,1]$
2. $f' \geq 0$ auf $[0,1]$ impliziert $(\mathcal{B}_n f)' \geq 0$ auf $[0,1]$
3. $f'' \geq 0$ auf $[0,1]$ impliziert $(\mathcal{B}_n f)'' \geq 0$ auf $[0,1]$

Insbesondere bleiben also Monotonie und Konvexität erhalten. Zeigen Sie hierzu

$$\frac{d}{dx} B_j^n = n\left(B_{j-1}^{n-1} - B_j^{n-1}\right) \tag{1}$$

sowie für alle $0 \leq k \leq n$

$$\frac{d^k}{dx^k}\mathcal{B}_n(f) = \frac{n!}{(n-k)!}\sum_{j=0}^{n-k} \Delta_{1/n}^k f\left(\frac{j}{n}\right) B_j^{n-k}, \tag{2}$$

wobei die k-te *Vorwärtsdifferenz* $\Delta_n^k f$ einer Funktion f definiert ist durch

$$\begin{aligned} \Delta_h f(x) &:= f(x+h) - f(x) \qquad \text{für } x,\, x+h \in [0,1] \\ \Delta_h^k f &:= \Delta^{k-1}(\Delta f) \qquad \text{für } x,\, x+kh \in [0,1] \end{aligned}$$

3. Der Approximationsoperator $\mathcal{B}_n$ hat auch die Eigenschaft der *simultanen* Approximation: Für $f \in C^k([0,1])$ gilt

$$\lim_{n\to\infty} \|(f - \mathcal{B}_n f)^{(j)}\|_{C([0,1])} = 0, \qquad j = 0, 1, \ldots, k.$$

Zeigen Sie diese Aussage für den speziellen Fall $k = 1$.

4. Die vielen schönen Eigenschaften von $\mathcal{B}_n$ "erkauft" man sich mit dem Preis der *Saturation*, d.h. selbst für sehr glatte f konvergiert $\|f - \mathcal{B}_n f\|_{C([0,1])}$ nur von erster Ordnung. Um dies zu sehen, zeigen Sie folgende Aussage: Für $f \in C^2([0,1])$ gilt für jedes $\overline{x} \in [0,1]$

$$\lim_{n\to\infty} n(\mathcal{B}_n(f) - f)(\overline{x}) = \frac{\overline{x}(1-\overline{x})}{2} f''(\overline{x}).$$

Hinweise: Überlegen Sie sich die Existenz einer Konstanten $C > 0$, so daß

$$\sum_{j=0}^{n} \left(\frac{j}{n} - \overline{x}\right)^4 B_j^n(\overline{x}) \le C n^{-2}, \qquad n = 1, 2, \ldots,$$

Betrachten Sie für $\overline{x} \in [0,1]$ die Taylorentwicklung $T_2 f$ zweiter Ordnung von f um den Punkt $\overline{x}$, und überlegen Sie sich, was $(\mathcal{B}_n(T_2 f))(\overline{x})$ ist. Überlegen Sie sich eine Abschätzung für $|(\mathcal{B}_n(f - T_2 f))(\overline{x})|$; gehen Sie ähnlich vor wie in Aufgabe 1.

5. Zeigen Sie mittels eines Tensorproduktarguments, daß die Polynome auch dicht in $C([0,1]^2)$ liegen. Nutzen Sie hierzu den univariate Operator $\mathcal{B}_n$, indem Sie für bivariate Funktionen f definieren

$$(\mathcal{B}_n^x f)(x,y) = (\mathcal{B}_n f(\cdot, y))(x), \qquad (\mathcal{B}_n^y f)(x,y) = (\mathcal{B}_n f(x, \cdot))(y), \qquad \mathcal{B}_n^{2D} := \mathcal{B}_n^x \circ \mathcal{B}_n^y$$

Überlegen Sie sich weiterhin, welchen Wert

$$\sup_{0 \ne f \in C([0,1])} \frac{\|\mathcal{B}_n f\|_{C([0,1])}}{\|f\|_{C([0,1])}}$$

hat.

6. **a)** Die Auswertung von Bernsteinpolynomen geschieht typischerweise mittels Rekursionsformeln [1]. Zeigen Sie:

$$B_i^n(x) = (1-x) B_i^{n-1}(x) + x B_{i-1}^{n-1}(x) \tag{3}$$

b) Schreiben Sie einen MATLAB-Code, der $\mathcal{B}_n f$ berechnet und plottet. Bestimmen Sie (mittels eines hinreichend feinen Gitters) den Fehler $\|f - \mathcal{B}_n\|_{C([0,1])}$ und plotten Sie (verwenden Sie `loglog`) diesen Fehler über n. Betrachten Sie hierzu die Funktionen

$$f_1(x) = |x - 0.5|, \qquad f_2(x) = e^x$$

Vergleichen Sie ihre Ergebnisse mit denen, die Sie bei Interpolation in den (auf $[0,1]$ skalierten) Tschebyscheffknoten erhalten (verwenden Sie Newtonsche dividierte Differenzen und das Hornerschema zur schnellen Auswertung der Tschebyscheffinterpolanten). Überprüfen Sie visuell für den Fall $n = 5$, ob die Tschebyscheffinterpolante auch die Konvexität von f_1 erhält.

---

[1] Im Fall von Bézierkurven spricht man dann vom Algorithmus von *de Casteljau*.