Programmieraufgaben

(1) Programmieren Sie die kubische Spline-Interpolation für eingespannte Splines für beliebiges n bei äquidistanter Verteilung der Stützstellen x_i. Verwenden Sie hierbei Hermite-Interpolation, lösen Sie das auftretende lineare Gleichungssystem zur Bestimmung der Ableitungen p_i an den Stellen x_i mit einer vorgefertigten Routine, z.B. `numpy.linalg.solve`. Testen Sie Ihr Programm an der Funktion

$$f\colon [-1,1] \to \mathbb{R} : x \mapsto \frac{1}{1+4x^2}$$

mit $f'(-1) = \frac{8}{25}$ und $f'(1) = -\frac{8}{25}$. Zeichnen Sie die Funktion, die Interpolationspunkte und den Spline in einen gemeinsamen Plot.

(2) Nehmen Sie die sechs Eckpunkte und die Mittelpunkte der sechs Seiten eines regelmäßigen Sechsecks als Datensatz:

$$(0,0)\,, \left(\tfrac{1}{2},0\right), (1,0)\,, \left(\tfrac{5}{4},\tfrac{\sqrt{3}}{4}\right), \left(\tfrac{3}{2},\tfrac{\sqrt{3}}{2}\right), \left(\tfrac{5}{4},\tfrac{3\sqrt{3}}{4}\right), \left(1,\sqrt{3}\right),$$
$$\left(\tfrac{1}{2},\sqrt{3}\right), \left(0,\sqrt{3}\right), \left(-\tfrac{1}{4},\tfrac{3\sqrt{3}}{4}\right), \left(-\tfrac{1}{2},\tfrac{\sqrt{3}}{2}\right), \left(-\tfrac{1}{4},\tfrac{\sqrt{3}}{4}\right), (0,0)\ .$$

Berechnen Sie die parametrischen Interpolationen bei

- Polynominterpolation mit äquidistanten Stützstellen $t_i = i$ (mit dividierten Differenzen und Horner-Schema oder vorgefertigte Routine verwenden);
- Polynominterpolation mit Chebyshev-Knoten (mit dividierten Differenzen und Horner-Schema oder vorgefertigte Routine verwenden);
- Spline-Interpolation (periodischer kubischer Spline) mit äquidistanten Stützstellen $t_i = i$. Verwenden Sie hierfür eine vorgefertigte Routine verwenden, z.B. `CubicSpline` aus der Bibliothek `scipy.interpolate`.

Zeichnen Sie jeweils das Sechseck, die Datenpunkte sowie die Interpolationskurve in einen gemeinsamen Plot.

Theorieaufgaben

(3) (Hermite-Interpolation). Zeigen Sie, dass für paarweise verschiedene reelle Zahlen $x_0, x_1, \ldots, x_r$ sowie nichtnegative ganze Zahlen $m_0, m_1, \ldots, m_r \in \mathbb{N}_0$ mit $\sum_{j=0}^{r} m_j = n+1$ und vorgegebenen Zahlen $f_j^{(\nu)} \in \mathbb{R}$ für $\nu = 0,1,\ldots,m_j-1$ und $j = 0,1,\ldots,r$ genau ein Polynom p vom Grad kleiner gleich n existiert mit den Eigenschaften

$$p^{(\nu)}(x_j) = f_j^{(\nu)} \quad \text{für } \nu = 0,1,\ldots,m_j-1, \quad j = 0,1,\ldots,r.$$

(4) Gegeben sind die Datenpunkte $(-1,0.2), (-0.5,0.5), (0,1), (0.5,0.5), (1,0.2)$ (Auswertung der Funktion aus Aufgabe (1) an 5 äquidistanten Punkten) sowie $p = f'(-1) = \frac{8}{25}$ und $q = f'(1) = -\frac{8}{25}$. Stellen Sie das zugehörige lineare Gleichungssystem zur Berechnung der Koeffizienten der interpolierenden eingespannten kubischen Splinefunktion auf. Lösen Sie das lineare Gleichungssystem mit einem CAS. Berechnen Sie nun mit einem CAS den Spline und werten Sie den Spline an der Stelle 0.1 aus.

(5) Betrachten Sie die Funktion $f\colon [0, 2\pi] \to \mathbb{R}$, $x \mapsto \sin(Lx)$ mit einer positiven Zahl L und äquidistanten Stützstellen $x_k = 2\pi k/N$ für $k = 0, 1, \ldots, N$. Geben Sie für die interpolierende kubische eingespannte Splinefunktion eine (von N und L abhängige) Abschätzung für den Interpolationsfehler an. Wie groß muss im Fall $L = 2$ der Wert von N gewählt werden, so dass ein Interpolationsfehler kleiner als 10^{-10} garantiert werden kann?