Serie 8

Zweidimensionale Spline-Interpolation

Aufgabe 8.1 (1 Punkt)**:** Lies die Beilage sorgfältig durch und beantworte folgende Fragen:

a) Wie geht man bei der Tensorproduktkonstruktion des zweidimensionalen Splines vor?

b) Berechne von Hand das Kronecker-Produkt $\mathbf{A} \otimes \mathbf{B}$ für

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

c) Führe die folgenden Zeilen Code in MATLAB aus:

```
M1 = reshape(1:16, 4, 4)
vec = M1(:)
M2 = reshape(vec, 4, 4)
```

Was passiert hier? Wo können ähnliche Befehle bei der Implementierung von der zweidimensionalen Spline-Interpolation benutzt werden?

Aufgabe 8.2 ($2 + 2 + 1 + 1$ Punkte)**:** In dieser Aufgabe wollen wir den Spline in zwei Dimensionen berechnen, der durch den eindimensionalen kubischen B-Spline induziert wird.

a) i) Schreibe eine Funktion `b = B_3(x)`, welche den eindimensionalen kubischen B-Spline $B_3(x)$ berechnet (siehe Beilage). Diese Funktion soll auch Vektoren und Matrizen als Eingabe akzeptieren können und **ohne Schleifen** auskommen. Hierbei ist u.a. der Befehl `find(1 < abs(x) & abs(x) < = 2)` hilfreich.

ii) Teste deine Funktion für $x = (-1, 0, 1)^\top$ (Antwort: $b = (\frac{1}{6}, \frac{2}{3}, \frac{1}{6})^\top$).

b) i) Schreibe eine Funktion `alpha = spline_coeff_2D(y)`, welche die Koeffizienten des zweidimensionalen Splines berechnet. `y` enthält hierbei die Knotenwerte.

ii) Um deine Funktion zu testen, lade die Datei `y.txt` von der Webseite herunter, welche die Knotenwerte einer Funktion auf $[0, 10]^2$ enthält. Berechne dafür die Koeffizienten (Zum Überprüfen: $\alpha_{1,1} = 0.0115$, $\alpha_{2,1} = 0.1101$).

c) i) Schreibe eine Funktion `s = spline_interpol_2D(x1, x2, alpha)`, welche den Spline $s(x_1, x_2)$ auswertet. Nutze dazu aus, dass deine Funktion `B_3` auch Vektoren und Matrizen als Eingabe akzeptiert.

ii) Teste deine Funktion, indem du den Spline aus Teil b), der an den ganzzahligen Knoten in $[0, 10]^2$ definiert ist, auswertest. Dabei ist der Befehl `meshgrid` hilfreich. Stimmen die Werte numerisch mit den Knotenwerten überein?

d) Benutze `meshgrid` und `surf` um die Knotenwerte `y` auf $[0, 10]^2$ (Gitterweite 1) zu zeichnen. Zeichne dann auch die B-Spline-Interpolierende auf $[0, 10]^2$, allerdings mit Gitterweite 0.1.

Aufgabe 8.3 (0.5 + 0.5 + 0.5 + 1.5 Punkte)**:** In dieser Aufgabe wollen wir die Runge-Funktion

$$f(x) = \frac{1}{1 + 25x^2}$$

auf dem Intervall $[-1, 1]$ auf drei verschiedene Arten approximieren. Die Auswertung soll auf dem Intervall $[-1, 1]$ mit der Schrittweite 0.01 geschehen.

a) Berechne das Interpolationspolynom p_1 in den äquidistanten Stützstellen

$$p_1(x_k) = f(x_k), \quad x_k = -1 + \frac{2k}{n}, \quad k = 0, 1, \ldots, n,$$

indem du deine Funktionen `newton_coeff` und `newton_interpol` aus Serie 4 benutzt.

b) Berechne das Interpolationspolynom p_2 in den Tschebyscheff-Stützstellen

$$p_2(x_k) = f(x_k), \quad x_k = \cos\left(\frac{2k + 1}{n + 1}\frac{\pi}{2}\right), \quad k = 0, 1, \ldots, n,$$

indem du deine Funktionen `newton_coeff` und `newton_interpol` aus Serie 4 benutzt.

c) Berechne den interpolierenden natürlichen kubischen Spline p_3 in den äquidistanten Knotenstellen

$$p_3(x_k) = f(x_k), \quad x_k = -1 + \frac{2k}{n}, \quad k = 0, 1, \ldots, n,$$

indem du deine Funktionen `spline_coeff` und `spline_interpol` aus Serie 7 benutzt.

d) Nun wollen wir die Interpolationspolynome und ihre Fehler visualisieren. Dazu betrachten wir das Intervall $[-1, 1]$ und nacheinander $n = 5$, $n = 10$ und $n = 25$.

   i) Zeichne für jedes n die Funktion f und die interpolierenden Funktionen in eine Abbildung.

   ii) Zeichne für jedes n die Fehlerfunktionen $(f - p_j)$, $j = 1, 2, 3$, in eine Abbildung.

   Benutze den Befehl `subplot`, um die 6 Abbildungen anzuzeigen, und füge jeweils eine Legende hinzu. Was beobachtest du?

Allgemeine Informationen befinden sich auf der Webseite.
Zuletzt editiert am 26. April 2024.