# Einführung in die Grundlagen der Numerik

Winter semester 2019/2020
Prof. Dr. Marc Alexander Schweitzer
Denis Düsseldorf

universitätbonn

## Übungsblatt 2.

Abgabe: 29.10.2019 vor der Vorlesung

**Aufgabe 5.** (Tschebyscheff Polynome)

Die Tschebyscheff-Polynome erster Art werden für $n \in \mathbb{N}$ definiert als

$$T_n : [-1,1] \to \mathbb{R}, \quad T_n(t) := \cos(n \cdot \arccos t)$$

Zeigen Sie die folgenden Eigenschaften der Tschebyscheff-Polynome:

a) Es gilt

$$T_n(\cos\theta) = \cos(n\theta), \quad \forall \theta \in [0,\pi]$$

Ausserdem gilt $\max_{t\in[-1,1]} = 1$.

b) Die Tschebyscheff-Polynome erfüllen die Rekursionsformel

$$T_{n+1}(t) = 2t \cdot T_n(t) - T_{n-1}(t), \quad \forall t \in [-1,1]$$

mit Startwerten $T_0(t) = 1$ und $T_1(t) = t$.

c) $T_n$ hat $(n+1)$ Extrema $s_k^{(n)}$ auf $[-1,1]$. Diese liegen bei

$$s_k^{(n)} := \cos\left(\frac{k\pi}{n}\right).$$

Ausserdem gilt

$$T_n(s_k^{(n)}) = (-1)^k, \quad k = 0,\dots,n.$$

d) $T_n$ hat $n$ einfache Nullstellen $t_k^{(n)}$ auf $[-1,1]$. Diese Nullstellen sind bei

$$t_k^{(n)} := \cos\left[\frac{(2k-1)\pi}{2n}\right], \quad k = 1,\dots,n.$$

e) Zeigen Sie, dass die Tschebyscheff-Polynome im Skalarprodukt

$$\langle f,g\rangle :== \int_{-1}^{1} f(x)g(x)\omega(x)\,dx, \qquad \text{mit } \omega(x) := \frac{1}{\sqrt{1-x^2}}$$

orthogonal sind. Explizit gilt

$$\langle T_n, T_m\rangle = \begin{cases} \pi, & n = m = 0 \\ \frac{\pi}{2}, & n = m > 0 \\ 0, & n \neq m. \end{cases}$$

(8 Punkte )

**Aufgabe 6.** (Clenshaw-Algorithmus)

> Mit dem Clenshaw-Algorithmus ist es möglich, Linearkombinationen von beliebigen Funktionen die einer Dreiterm-Rekursion genügen auszuwerten. Beispielsweise funktioniert der Algorithmus zur Auswertung von Orthogonalpolynomen (Legendre, Tschebyscheff, Laguerre, Hermite, Jacobi, ...).
>
> Der Clenshaw-Algorithmus ist interessant für die Numerik, da - wie die Aufgabe zeigen wird - die Auswertung einer Dreiterm-Rekursion auf etwa $n$ Multiplikationen und $2n$ Additionen reduziert wird.

Es sei im Folgenden $p_n \in \mathcal{P}^n[-1,1]$ dargestellt in der Basis der Tschebyscheff Polynome $T_i$, d.h.

$$p_n = \sum_{i=0}^{n} \alpha_i T_i.$$

Es sei $x \in [-1,1]$ und rekursiv

$$\beta_n := \alpha_n, \quad \beta_{n-1} := \alpha_{n-1} + 2x\beta_n, \quad \beta_i := \alpha_i + 2x\beta_{i+1} - \beta_{i+2}$$

für $i = n-2, \ldots, 1$. Zeigen Sie, dass mit diesen Koeffizienten die Beziehung

$$p_n(x) = \alpha_0 + x\beta_1 - \beta_2$$

erfüllt ist.

(4 Punkte )

**Programmieraufgabe 1.** (Clenshaw-Algorithmus)

a) Implementieren Sie den Clenshaw Algorithmus (siehe Aufgabe 6) zur Auswertung von

$$p_n(x) = \sum_{i=0}^{n} \alpha_i T_i(x)$$

mit `Python` / `NumPy`. Die Routine soll eine Liste von Punkten als Input entgegen nehmen, und eine Liste von Funktionswerten zurück geben. Benutzen Sie die Folgende Signatur für Ihre Funktion.

```
def clenshaw(alphas, xs):
    # Input:
    #    alphas    Liste / np.array mit den Koeffizienten der
                   Entwicklung in der Tschebyscheff Basis
    #    xs        Liste / np.array mit den x-Werten an denen
                   ausgewertet werden soll
    #
    # Output:
    #    ys        Liste mit den y-Werten

    ## Hier Ihr Code
```

b) Schreiben Sie eine Routine, die einen Plot der ersten $n$ Tschebyscheff Polynome $T_i$, $i = 1, \ldots, n$ unter Benutzung von Clenshaw-Auswertungen erstellt. Achten Sie auch einen passenden Titel und eine Legende. Speichern Sie den Plot für $n = 6$ (d.h. $T_0, \ldots, T_5$) als PDF Datei.

c) Implementieren Sie eine **iterative** Funktion zur Auswertung einer beliebigen Dreiterm-Rekursion

$$F_{n+1} = \beta F_n + \gamma F_{n-1}$$

Die Funktionen $F_0$ und $F_1$ sollen als lambda-functions übergeben werden. Die Koeffizienten $\beta, \gamma$ sind Konstanten, die ebenfalls als Parameter beim Aufruf übergeben werden sollen.

```
def recursion_3term_iterative(F0,F1,beta,gamma,n,xs):
    # Input:
    #    F0        lambda function, Startwert 0
    #    F1        lambda function, Startwert 1
    #    beta      Parameter aus Rekursion
    #    gamma     Zweiter Parameter aus Rekursion
    #    n         Ordnung der Rekursion
    #    xs        Punkte an denen ausgewertet werden soll
    #
    # Output:
    #    ys        Liste der Werte Fn(x) fuer x in xs


    ## Hier Ihr Code
```

Implementieren Sie ausgehend hiervon eine Routine zur Berechnung von

$$p_n(x) = \sum_{i=0}^{n} \alpha_i T_i(x)$$

in der Form

```python
def pn_tschebyscheff_iterative(alphas, xs):
    # Input:
    #    alphas   Koeffizienten aus der Entwicklung
    #    xs       Punkte an denen ausgewetet werden soll
    #
    # Output:
    #    ys       Funktionswerte an den Punkten x aus xs

    ## Hier Ihr Code
```

d) Implementieren Sie eine Routine zur Messung des absoluten Fehlers

$$|p_n^{clenshaw}(x) - p_n^{3termIterative}(x)|$$

zwischen der Clenshaw-Auswertung und der iterativen 3-Term-Rekursionsauswertung einer Funktion $p_n = \sum_{i=0}^{n} \alpha_i T_i$. Plotten Sie diesen absoluten Fehler für 100 Stützstellen im Interval $[0, 1]$, d.h. für alle Stützstellen `x = np.linspace(0, 10, 100)`, für die Funktion

$$p_6(x) = \frac{2}{3}T_2(x) + \frac{4}{14}T_4(x) + \frac{23}{96}T_6(x).$$

Speichern Sie den Plot als PDF.

(5 Punkte )

Abgabe per Mail an die Tutoren.