Übung 1 *Tschebyscheff Polynome*

Tschebyscheff Polynome der ersten Art sind folgendermaßen definiert:

$$T_n = \cos(n \arccos t), \qquad t \in [-1,1] \qquad n = 0,1,2,...$$

a) Leite die Rekursionsformel

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t),\ n = 1,2,...$$

her und zeige damit, dass $T_n(t)$ ein Polynom n-ten Grades ist und für $n \geq 1$ den führenden Koeffizienten 2^{n-1} besitzt.

b) $T_n(t)$ besitzt auf dem Intervall $[-1,1]$ insgesamt n Nulstellen $\tau_1^{(n)},...,\tau_n^{(n)}$. Wie lauten diese? Berechne auch die sich in dem abgeschlossenen Intervall befindlichen Extrema $\sigma_k^{(n)}$ und ihre Werte $T_n(\sigma_k^{(n)})$.

c) Zeige, dass $T_n(t)/2^{n-1}$ die Maximumsnorm

$$\|p_n\|_\infty = \max_{t\in[-1,1]} |p_n(t)|$$

unter allen normierten Polynomen

$$p_n(t) = t^n + \sum_{i=0}^{n-1} a_i t^i$$

vom Grad $n \geq 1$ minimiert.

Hinweis: Führe den Beweis indirekt und berechne hierzu für ein Polynom P_n, welches die Behauptung widerlegen würde, die Anzahl der Nullstellen des Differenzpolynoms $P_n(x) - T_n(x)/2^{n-1}$.

d) Seien $t_0,...,t_n \in [-1,1]$ die Stützwerte einer Interpolationsaufgabe. Sei $\omega(t) := \prod_{i=0}^{n}(t-t_i)$. Wähle die Stützwerte so, dass $\|\omega(t)\|_\infty$ minimal wird und berechne dieses Minimum. Was bedeutet dieses Resultat für den Interpolationsfehler?

(6 Punkte)

Übung 2 *Diskrete Fouriertransformation*

Betrachte eine quadratintegrierbare 2π-periodische Funktion $f(x) = f(x+2\pi)\ \forall x \in \mathbb{R}$ auf dem Intervall $[-\pi,\pi)$. Wie aus der Vorlesung bekannt, sind die Koeffizienten der Fourierreihe von $f(x)$ gegeben durch

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(kx)dx, \qquad k = 0,1,2,...$$

und

$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(kx)dx, \qquad k = 1,2,...$$

Unterteile das Intervall $[-\pi,\pi)$ in $n+1$ gleich große Teile, so dass die Stützstellen durch

$$x_j = -\pi + \frac{2\pi j}{n+1}$$

gegeben sind. Die Peridiozität schreibt vor, dass

$$f(x_{n+1}) = f(x_0)$$

sein muss. Wende die *zusammengesetzte Trapezregel* für diese Unterteilung des Intervalls an auf die Berechnung der Fourierkoeffizienten. Als Resultat sollten genau die Koeffizienten der diskreten Fourieranalyse herauskommen.

(4 Punkte)

Übung 3 *Diskrete Fouriertransformation - praktische Übung*

Lade von der Webseite der Vorlesung die Datei `realDFT.cc` herunter und implementiere die fehlenden Funktionen!
Für zwei Vektoren x und y der Dimension $n+1$ generiert die C++-Funktion[1]

```
template<typename VECTOR>
void generateData( VECTOR &x, VECTOR &y )
```

einen Datensatz mit den Stützwerten $y_0, ..., y_n$ zu den Stützstellen $x_0, ..., x_n$.

a) Implementiere die Funktion

```
template<typename VECTOR>
void realDFT( const VECTOR& x, const VECTOR& y, VECTOR& a, VECTOR& b )
```

Sie soll aus den Datenvektoren x und y die Koeffizienten a_k und b_k des trigonometrischen Polynoms $t_n(x)$ der diskreten Fourieranalyse ermitteln. *Achtung: Die Dimensionen von a und b müssen angepasst werden!*

b) Implementiere die Funktion

```
template<typename VECTOR>
REAL evaluateRealDFT( const REAL x, const VECTOR& a, const VECTOR& b )
```

Sie soll das trigonometrische Polynom $t_n(x)$ an der Stelle x evaluieren. Sie wird von der Funktion

```
template<typename VECTOR>
void gnuplotPeriodicFunction( const VECTOR& a,
                              const VECTOR& b,
                              const UINT nInt,
                              const UINT nRes,
                              std::string filename
                              )
```

benötigt, welche die Daten für `gnuplot` liefert: Das Polynom $t_n(x)$ soll über n_{Int} Perioden und mit einer Gesamtauflösung von n_{Res} Punkten in eine Datei `filename` geschrieben werden.

Mit dem `gnuplot`-Skript `plotDFT.gnp` kann man sich die verschiedenen Ausgaben aus der Funktion `testcase()` so anzeigen lassen:

```
gnuplot> load 'plotDFT.gnp'
```

Natürlich können auch weitere neue Testfälle ausprobiert und untersucht werden!

(5 Punkte)

[1] *Hinter dem Makro* `REAL` *verbirgt sich* `double` *und hinter dem Template-Parameter* `VECTOR` *verbirgt sich natürlich* `hdnum::Vector<REAL>`.