Aufgabenblatt 9

Abgabe am Montag, dem 23.06.25 vor der Vorlesung.
Die Programmieraufgabe kann bis zum Mittwoch, dem 30.06.25 vor der Vorlesung auf der Moodle-Seite abgegeben werden.

Der Fachschaftsrat Mathematik lädt herzlich zum Sommerfest am 26. Juni ein! Die Feier findet von 15-22 Uhr am Richard-Wagner-Hain statt und lockt mit einem spannenden Volleyballturnier, coolen DIY-Aktionen wie Henna-Tattoos und Seedballs, sowie guter Musik, Snacks und Sommerlaune pur.

Aufgabe 1 (Exaktheitsgrad von Quadraturregeln)
Zeigen Sie, dass die Mittelpunktsregel Exaktheitsgrad 1 hat, die Trapezregel Exaktheitsgrad 1 und die Simpsonregel Exaktheitsgrad 3. Zeigen Sie, dass die drei Regeln nicht exakt von einem höheren Grad sind.

Aufgabe 2 (Orthogonalpolynome) (a) Es sei $(a,b) \subseteq \mathbb{R}$, wobei $a = \infty$ und $b = \infty$ zugelassen sind. Es sei $\omega : (a,b) \to \mathbb{R}$ mit $\omega > 0$ und $\int_a^b \omega \, dx < \infty$. Definiere

$$L^2_\omega(a,b) := \left\{ f \in L^1(\Omega) \mid \int_a^b |f|^2 \omega \, dx < \infty \right\}.$$

Zeigen Sie, dass durch

$$(f,g)_\omega := \int_a^b f \, g \, \omega \, dx$$

ein reelles Skalarprodukt auf dem Raum $L^2_\omega(a,b)$ definiert wird.

(b) Zeigen Sie, dass die sogenannten Legendre-Polynome $p_n : (-1,1) \to \mathbb{R}$ gemäß

$$p_n(x) := \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

erfüllen, dass $p_n \in P_n(-1,1)$ und $(p_n, p_k)_\omega = 0$ für $k \neq n$ und $\omega = 1$.

Aufgabe 3 (Tschebyscheff-Polynome sind Orthogonalpolynome)
Zeigen Sie, dass die Tschebyscheff-Polynome $T_n : (-1,1) \to \mathbb{R}$ orthogonal sind bezüglich des in Aufgabe 2(a) definierten Skalarproduktes mit $\omega = (1 - x^2)^{-1/2}$.

Aufgabe 4 (Programmieraufgabe, Abgabe bis 30.06.2025)
Machen Sie sich mit dem julia-Programm `solve_ODE_2.jl` vertraut, welches die Lösung $u(x,t)$ der nichtlinearen Differentialgleichung

$$u_t - (|u'|^{p-2}u')' = 0$$

mit $1 < p < \infty$ in $[0,1] \times [0,0.2]$ approximiert, wobei u_t die Ableitung in Zeitrichtung (t) von u und u' die Ableitung im Ort (x) von u ist.

Diskretisiert man die Differentialgleichung im Ort erhält man

$$U_t = F(U), \qquad U \in \mathbb{R}^N.$$

Die Funktion F ist bereits im Programm als Funktion `nonlinear_F` gegeben. Diskretisiert man nun die Zeitableitung erhält man für gegebene Anfangsdaten U_0 und $\Delta t = \frac{0.2}{K}$ für ein $K \in \mathbb{N}$ als explizites Verfahren

$$\frac{U^{k+1} - U^k}{\Delta t} = F(U^k) \qquad \text{für alle } k = 0, \dots, K-1$$

und als implizites Verfahren

$$\frac{U^{k+1} - U^k}{\Delta t} = F(U^{k+1}) \qquad \text{für alle } k = 0, \dots, K-1,$$

wobei $U^k \in \mathbb{R}^N$ die Approximation von u nach dem k-ten Zeitschritt darstellt. Man kann das implizite Verfahren in jedem Zeitschritt umschreiben zu

$$U^{k+1} - \Delta t F(U^{k+1}) - U^k = 0.$$

(a) Vervollständigen Sie die Methode `Newton_iteration` im Programm `solve_ODE_2.jl` so, dass sie für eine gegebene Funktion G mit Ableitung DG eine Nullstelle von G durch das Newton-Verfahren approximiert. Beachten Sie dabei auch die Kommentare in `Newton_iteration` zur Ein- und Ausgabe der Funktion.

(b) Testen Sie das Programm für die Werte $p \in \{1.3, 1.8, 2, 5, 10\}$ und geben Sie jeweils die Anzahl an Iterationen an, die im Newton-Verfahren in einem Zeitschritt maximal ausgeführt werden, d.h. die Zahl

$$\max_{k=1,\dots,K-1} \texttt{nIter\_k},$$

wobei `nIter_k` die Anzahl der Iterationen im k-ten Schritt bezeichne.

(c) Setzen Sie den Startwert des Newton-Verfahrens in jedem Schritt auf 0 und führen Sie Teil (b) erneut durch. Was fällt Ihnen dabei auf?

Die Abgabe des Codes erfolgt auf der Moodle-Seite. Für die Aufgabenteile (b) und (c) geben Sie die Ergebnisse in einem .txt File an.