Serie 12 **Diskussion am 24.01.2020**

Aufgabe 1 (Exakte Lösung; praktische Aufgabe). Gegeben sei das Anfangswertproblem

$$y'(t) = \frac{11}{10} t^{1/10} y(t), \quad y(0) = 1$$

mit exakter Lösung $y(t) = \exp(t^{11/10})$.
(a) Implementieren Sie die Lösung Anfangswertproblem auf dem Intervall $[0, 1]$ mit Hilfe des expliziten Euler-Verfahrens und des RK4-Verfahrens auf uniformen Gittern der Form

$$t_j = t_0 + j\frac{1}{N} \quad \text{für ein } N \in \mathbb{N}.$$

Sie können die Implementierung des Euler-Verfahrens von der Vorlesungshomepage nutzen. Berechnen Sie Lösungen für eine Folge von Schrittzahlen $N = 2, 4, \ldots, 2^{12}$. Plotten Sie den exakten Fehler $e(N) = |y_N(1) - y(1)|$ im Zeitpunkt $t = 1$ in Abhängigkeit der jeweiligen Schrittzahl N in doppelt logarithmischer Skala. Nutzen Sie beispielsweise das Modul `matplotlib.loglog`. Betrachten sie vergleichend die Konvergenzraten für beide Verfahren, wenn statt der uniformen Schrittweiten ein graduiertes Gitter der folgenden Form verwendet wird

$$t_j = t_0 + \left(\frac{j}{N}\right)^{50/11}.$$

(b) Implementieren Sie ein beliebiges Mehrschrittverfahren Ihrer Wahl zur Lösungen des Anfangswertproblems mit einer unterschiedlichen Schrittzahl N. Berechnen Sie jeweils den Fehler $e(N)$ und plotten ihn in der Grafik aus Aufgabenteil (a).

(c) **Zusatz** Implementieren Sie die adaptive Schrittweitensteuerung mit Hilfe des eingebetteten RK4(3)-Verfahrens mit dem Butcher-Tableau

$$\begin{array}{c|ccccc}
0 & & & & & \\
\frac{1}{2} & \frac{1}{2} & & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & & \\
1 & 0 & 0 & 1 & & \\
1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & 0 \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{6}
\end{array}$$

Stellen Sie dabei sicher, dass die Schrittweite h stets zwischen gegebenen Schranken $h_{\min} \leq h \leq h_{\max}$ liegt. Erzeugen Sie für verschiedene Parameter h_0, $h_{\min}$, $h_{\max}$ sowie der Toleranz $\mathrm{tol} > 0$ Lösungen des Anfangswertproblems mit einer unterschiedlichen Schrittzahl N. Berechnen Sie jeweils den Fehler $e(N)$ und plotten ihn in der Grafik aus Aufgabenteil (a).

Aufgabe 2 (Satellitenbahn im Erde-Mond-System II; praktische Aufgabe). Lösen Sie das Anfangswertproblem aus Aufgabe 4.2 mit dem Verfahren aus Aufgabe 1.(b).
Zusatz Nutzen Sie Ihr RK4(3)-Verfahren aus Aufgabe 1 und lösen das Anfangswertproblem aus Aufgabe 4.2.
(a) Finden Sie Parameter, sodass die Rechnung mit adaptiver Schrittweitensteuerung eine Laufbahn liefert, die mit der Referenz-Lösung durch das explizite RK4-Verfahren mit feiner uniformer Schrittweite ($N = 6000$) übereinstimmt. Vergleichen Sie die Anzahl der verwendeten Schritte, die Rechenzeit (Python-Modul `time`) sowie den Rechenaufwand in Bezug auf Funktionsauswertungen.
(b) Vergleichen Sie die Ergebnisse Ihres RK4(3)-Verfahrens mit Lösungen mit Hilfe der Funktion `solve_ivp` aus dem Python Standard-Modul `scipy.integrate`.

Aufgabe 3 (Wärmeleitungsgleichung; praktische Aufgabe). Auf dem Einheitsquadrat $\Omega := (0,1)^2 \subset \mathbb{R}^2$ sei eine Wärmequelle $f : [0,T] \times \Omega \to \mathbb{R}$ (Heizung) gegeben sowie eine anfängliche Temperaturverteilung $u_0 : \Omega \to \mathbb{R}$. Die zeitliche und räumliche Änderung der Temperatur in einem Körper wird beschrieben durch folgende parabolische partielle Differentialgleichung beschrieben mit dem Laplace-Operator $\Delta_x u := \partial^2 u/\partial x_1^2 + \partial^2 u/\partial x_2^2$ bzgl. der x-Komponente in Ω. Gesucht ist $u : [0,T] \times \Omega \to \mathbb{R}$ mit

$$\frac{\partial}{\partial t} u - \Delta_x u = f \quad \text{in } [0,T] \times \Omega, \quad u = 0 \quad \text{auf } [0,T] \times \partial\Omega \quad \text{und} \quad u(0,\bullet) = u_0 \quad \text{in } \Omega.$$

Dabei beschreibt die zweite Gleichung, dass der Körper am Rand stets auf 0 °C heruntergekühlt wird. Wir nehmen hier an, dass $f \equiv 0$.
Zur Herleitung einer geeigneten Variationsformulierung muss die Differentialgleichung mit geeigneten Sobolev-Funktionen $v \in V \equiv H_0^1(\Omega)$ getestet werden. Eine formale Definition wird in der Funktionalanalysis gegeben. Zum Verständnis dieser Aufgabe soll lediglich gesagt werden, dass „1“ im oberen Index für die Existenz einfacher Ableitungen im schwachen Sinn steht und „0“ für die Nullranddaten auf $\partial\Omega$. Nach partieller Integration (in zwei Raumdimensionen) erhalten wir die folgende Variationsformulierung. Gesucht ist $u : [0,T] \to V$ mit

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega u(t)\, v \,\mathrm{d}x + \int_\Omega \nabla_x u(t) \cdot \nabla_x v \,\mathrm{d}x = 0 \quad \text{für alle } v \in V \text{ und für fast alle } t \in (0,T),$$

$$\int_\Omega u(0)\, v \,\mathrm{d}x = \int_\Omega u_0\, v \,\mathrm{d}x \quad \text{für alle } v \in V.$$

Gegeben sei eine reguläre Triangulierung $\mathcal{T}$ von Ω in abgeschlossene Dreiecke, d.h., $\mathcal{T}$ ist eine Überdeckung $\overline{\Omega} = \bigcup_{T \in \mathcal{T}} T$ und zwei nicht-identische Dreiecke in $\mathcal{T}$ sind entweder disjunkt oder sie teilen sich genau eine Kante oder genau einen Knoten. Es bezeichne $\mathcal{N}(\Omega) = \{z_1, \ldots, z_J\}$ die nummerierte Menge der Knoten von $\mathcal{T}$, die nicht auf dem Rand $\partial\Omega$ liegen. Die Funktionen u und v sollen mit Hilfe von stückweise affinen und global stetigen Finiten-Elemente-Funktionen $u_h(t) \in S_0^1(\mathcal{T}) := P_1(\mathcal{T}) \cap C(\overline{\Omega}) \cap V$, mit zeitabhängigen Koeffizienten diskretisiert werden. Dazu wähle eine Basis $(\varphi_j \,:\, j = 1, \ldots, J)$ mit $\varphi_j(z_k) = \delta_{jk}$ für $j, k = 1, \ldots, J$ und erhalte die Basisdarstellung

$$u_h(t) = \sum_{j=1}^{J} u_j(t), \varphi_j \in S_0^1(\mathcal{T}).$$

(a) Setzen Sie Basisdarstellungen von $u_h(t)$ und $v_h(t)$ in die Variationsformulierung ein und leiten folgendes lineares Anfangswertproblem für den Vektor $U(t) = (u_1(t), \ldots, u_J(t))^\top$ her.

$$M\,U'(t) + A\,U(t) = 0 \quad \text{für alle } t \in (0,T) \quad \text{und} \quad M\,U(0) = b.$$

Dabei ist $M \in \mathbb{R}^{J\times J}$ mit $M_{jk} = \int_\Omega \varphi_j\,\varphi_k \,\mathrm{d}x$ invertierbar und heißt Massematrix und $A \in \mathbb{R}^{J\times J}$ Steifigkeitsmatrix mit mit $A_{jk} = \int_\Omega \nabla_x\varphi_j \cdot \nabla_x\varphi_k \,\mathrm{d}x$ sowie die rechte Seite $b \in \mathbb{R}^J$ mit $b_j = \int_\Omega u_0\,\varphi_j \,\mathrm{d}x$. Zur Erzeugung der Matrizen und der rechten Seite können Sie das Python-Modul `Heat` von der Vorlesungshomepage verwenden.

(b) Setze $m := (0.5, 0.5)^\top$. Lösen Sie das Anfangswertproblem aus (a) mit dem expliziten und impliziten Euler-Verfahren für folgenden Anfangswert

$$u_0(x) := \exp\left(1 - \frac{1}{1 - 16\,|x-m|^2}\right) \quad \text{für } |x-m| \le \frac{1}{4} \quad \text{und} \quad u_0(x) := 0 \quad \text{sonst.}$$

(c) Plotten Sie den Anfangswert u_0 in Ω sowie die Lösung zum Zeitpunkt $T = 10^{-2}$. Sie können dafür die Funktion `plot_trisurf` aus dem `matplotlib` Modul verwenden. Die Daten der Geometrie erhalten Sie aus dem `Heat` Modul.

Zusatz: Erzeugen Sie mit Hilfe des `matplotlib.animation` Moduls ein Video vom zeitlichen Verlauf der Temperaturverteilung.

Literatur: Zur vertiefenden Lektüre sei die Bücherserie „Numerische Mathematik“ von Walter Zulehner empfohlen, für parabolische Gleichungen wie in dieser Aufgabe insbesondere Band 2 über instationäre Probleme, erschienen in Springer, Basel 2011. Beide Bände sind im HU-Netz verfügbar

`https://link.springer.com/book/10.1007/978-3-7643-8427-2`
`https://link.springer.com/book/10.1007/978-3-7643-8429-6`