Abteilung für
Angewandte Mathematik

SoSe 2021
Prof. Dr. Sören Bartels, Jakob Keck

Praktische Übungen zur Vorlesung: Numerik für Differentialgleichungen

Blatt 3 – 18.5.2021

Abgabe: 7.6.2021

Homepage zur Vorlesung:

`https://aam.uni-freiburg.de/agba/lehre/ss21/ndgln`

Projekt 1. Wir betrachten das SIR-Modell

$$\begin{aligned} S' &= -\beta S I, \\ I' &= \beta S I - \gamma I, \\ R' &= \gamma I \end{aligned}$$

mit $\beta = \frac{1}{2}$, $\gamma = \frac{1}{4}$ sowie $S(0) = 0.95$, $I(0) = 0.05$ und $R(0) = 0$ im Zeitintervall $[0, T]$ mit $T = 50$.

(i) Schreiben Sie ein Programm, das die Lösung des SIR-Modells mithilfe des expliziten Euler-Verfahrens approximiert. Testen Sie verschiedene Schrittweiten und beobachten Sie das qualitative Verhalten der Approximationslösungen. Für welche Schrittweiten ergeben sich sinnvolle Resultate?

(ii) Schreiben Sie ein Programm, das die Lösung des SIR-Modells mithilfe des impliziten Euler-Verfahrens approximiert. Verwenden Sie dabei zur Berechnung der numerischen Lösung y_{k+1} im $(k+1)$-ten Zeitschritt die Fixpunktiteration

$$\begin{aligned} y_{k+1}^0 &= y_k, \\ y_{k+1}^{n+1} &= y_k + \tau f(t_{k+1}, y_{k+1}^n) \end{aligned}$$

mit einem geeigneten Abbruchkriterium. Warum ist die Fixpunktiteration nötig? Wie ändert sich das qualitative Verhalten der numerischen Lösungen im Vergleich zu (i)?

Projekt 2. Das Matlab-Programm `runge_kutta_expl.m`, das Sie von der Vorlesungshomepage herunterladen können, realisiert ein explizites Runge–Kutta-Verfahren zur Lösung einer skalaren Differenzialgleichung $y' = f(t, y)$, $y(0) = y_0$.

(i) Dokumentieren Sie jede Zeile der Unterfunktion Phi. Welches Verfahren wird hier realisiert?

(ii) Die exakte Lösung für den Fall $f(t, y) = -2y + 5\cos(t)$ und $y_0 = 2$ ist durch $y(t) = 2\cos(t) + \sin(t)$ gegeben. Bestimmen Sie für die Schrittweiten $\tau = 2^{-s}$, $s = 0, 1, \ldots, 7$, den Approximationsfehler $|y(T) - y_K|$ mit $T = t_K = 10$.

(iii) Modifizieren Sie das Programm, um das explizite Euler-Verfahren, das Euler–Collatz-Verfahren, das klassische Runge–Kutta-Verfahren und die 3/8-Regel zu realisieren.

(iv) Bestimmen Sie für alle Verfahren die Approximationsfehler $|y(T) - y_K|$ zum Zeitpunkt $T = 10$ mit den Schrittweiten $\tau = 2^{-s}$, $s = 0, 1, \ldots, 7$. Stellen Sie diese vergleichend als Polygonzüge in einer Grafik mit logarithmischer y-Achsenskalierung dar, was in MATLAB mit dem Kommando `semilogy` realisiert werden kann.