Dr. Vasile Gradinaru
Dr. Adrian Montgomery Ruf

Serie 6

Best before: Di. 07.04. / Mi. 08.04, in den Übungsgruppen
Koordinatoren: Adrian Montgomery Ruf, HG G 54.1, `adrian.ruf@sam.math.ethz.ch`
Webpage: http://metaphor.ethz.ch/x/2020/fs/401-1662-10L/#exercises

1. Kernaufgabe Runge–Kutta-Methoden I

Modellierung der Physik

Hinweis: Diese Kernaufgabe besitzt kein Template.

a) Schreiben Sie das explizite Euler-Verfahren, das implizite Euler-Verfahren und die implizite Mittelpunktsregel als Runge–Kutta-Verfahren; geben Sie die entsprechenden Butcher-Tabellen an und erklären Sie die Herleitung jeder diesen Methoden als Runge–Kutta-Verfahren im Sinne der Vorlesung.

b) Programmieren Sie die Methoden von (a) als Runge–Kutta-Verfahren. Verwenden Sie diese um jeweils eine numerische Approximation der Lösung der Gleichung

$$\begin{aligned} \dot{y} &= y(t) - 2\sin(t)\,, \qquad t \in [0,4]\,, \\ y(0) &= 1\,, \end{aligned}$$

mit $N = 100$ gleich grossen Zeitschritten zu berechnen und zu ploten. Verwenden Sie Ihren Code um die entsprechenden Konvergenzordungen dieser drei Methoden empirisch zu finden. Die exakte Lösung ist $y(t) = \cos(t) + \sin(t)$.

2. *Runge–Kutta-Methoden II*

Gegeben ist das klassische Runge–Kutta-Verfahren mit dem Butcher-Tableau:

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

a) Implementieren Sie dieses Runge–Kutta-Verfahren für allgemeine Systeme erster Ordnung.

Hinweis: Verwende das Template `klassRK.py`.

b) Verwenden Sie das Programm aus Aufgabenteil a), um das Anfangswertproblem:

$$\begin{cases} y''(t) + 2y'(t) + 101y(t) = 0, \\ y(0) = 1, \\ y'(0) = -1, \end{cases} \tag{1}$$

Bitte wenden!

mit exakter Lösung:

$$y(t) = e^{-t} \cdot \cos(10t) \tag{2}$$

approximativ innerhalb des Intervalls $t \in [0, 3]$ zu lösen. Überführen Sie dazu zunächst die Differentialgleichung in ein System erster Ordnung. Wählen Sie eine geeignete Schrittweite h und plotten Sie sowohl die Näherungs- als auch die exakte Lösung.

Hinweis: Schreiben Sie ein MainFile, das das Python Code `klassRK.py` benutzt.

c) Ermitteln Sie empirisch, d.h. mit numerischen Experimente und geeignete Konvergenzplots, die Konvergenzordnung dieses Verfahrens für das gegebene Problem.

3. *Airy-Gleichung*

Es soll die so genannte Airy-Gleichung

$$\ddot{u}(t) - t\, u(t) = 0$$

numerisch gelöst werden wobei folgende Anfangswerte zum Zeitpunkt $T_{start} = 0$ gegeben sind

$$u(0) = \frac{1}{3^{\frac{2}{3}}\Gamma(\frac{2}{3})} \approx 0.3550280539,$$
$$\dot{u}(0) = -\frac{1}{3^{\frac{1}{3}}\Gamma(\frac{1}{3})} \approx -0.2588194038$$

und rückwärts in der Zeit integriert wird bis zu $T_{end} = -40$. Dieses Anfangswertproblem definiert die Airy-Funktion $\mathrm{Ai}(t)$ welche in der Physik eine grosse Bedeutung hat.

1. Schreiben Sie die Gleichung um in ein System erster Ordnung für $y(t)$ und leiten Sie daraus die rechte Seite her. Implementieren Sie die rechte Seite in der Funktion `rhs` welche t und $y(t)$ als Argumente hat.

   Hinweis: Verwenden Sie das Template `airy.py`

2. Implementieren Sie das explizite und das implizite Euler-Verfahren, die explizite und die implizite Mittelpunktsregel. Die Argumente dieser Funktionen sind: der Anfangswert $y(0)$, Anfangszeit T_{start}, Endzeit T_{end} und die Anzahl Schritte N. Lösen Sie das Anfangswertproblem und plotten Sie die Lösung.

   Hinweis: Benutzen Sie die Funktion `fsolve` aus `scipy.optimize`.

3. Implementieren Sie die 4 entsprechenden Runge–Kutta-Methoden mit allgemeinem Butcher-Schema in der Funktion `RK`. Bekommen Sie dieselbe Ergebnisse wie beim Punkt 2?

4. Lösen Sie das gegebene Anfangswertproblem mit einer Runge–Kutta-3/8-Zeitintegration. Implementieren Sie dafür die Funktion `RK_38` und plotten Sie die Lösung.

   Hinweis: Das Butcher-Schema der Runge–Kutta-3/8-Regel lautet

$$\begin{array}{c|cccc} 0 & & & & \\ 1/3 & 1/3 & & & \\ 2/3 & -1/3 & 1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & 1/8 & 3/8 & 3/8 & 1/8 \end{array}$$

Siehe nächstes Blatt!

Das 3/8-Butcher-Schema in dem Programm ist repräsentiert wie eine Matrix

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 \\ 2/3 & -1/3 & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 \\ 0 & 1/8 & 3/8 & 3/8 & 1/8 \end{pmatrix}$$