D-ITET, D-MATL
Dr. R. Käppeli
R. Hohl

Serie 5

1. In dieser Aufgabe wollen wir eine adaptive Quadratur Methode zur Berechnung des bestimmten Integrals

$$I[f] = \int_a^b f(x)dx$$

entwickeln und implementieren basierend auf der Trapezregel. Wie in der Vorlesung diskutiert, benötigt man dafür einen Fehler-Schätzer. Hierzu vergleichen wir das Resultat der Trapezregel

$$Q_1[f] = \frac{b-a}{2}\left(f(a) + f(b)\right)$$

mit dem Resultat der zusammengesetzen Trapezregel

$$Q_1^2[f] = \frac{b-a}{4}\left(f(a) + 2f\left(\frac{a+b}{2}\right) + f(b)\right)$$

(mit zwei Teil-Intervallen).

a) Bestimmen Sie den Fehler-Schätzer $E^2[f] = |Q_1^2[f] - I[f]|$.

Hinweis : Beispiel (14) und (15) in der Vorlesung.

b) Implementieren Sie die adaptive Quadratur Methode in der MatlabFunktion `adapttrapez_simple_Template.m`.

Hinweis : Verwenden Sie den in der Vorlesung gezeigten Pseudo-MatlabCode.

c) Der in der Vorlesung gezeigte Pseudo-MatlabCode ist sehr simple und besitzt einige Schwächen. Geben Sie zwei offensichtliche Schwächen an und versuchen Sie diese zu beheben.

Hinweis : Die `adaptsimp.m` Funktion von Serie 4, Aufgabe 4 könnte hilfreich sein.

Bitte wenden!

2. *Explizites Euler Verfahren*

a) Wir betrachten die logistische Differentialgleichung

$$\dot{y}(t) = (a - by(t))\, y(t). \tag{1}$$

Plotten Sie das Richtungsfeld von Gl. (1) mit der MATLAB Funktion `richtungs_feld.m` für $t \in [0, 5]$ und $y \in [0, 1]$. Wählen Sie $a = 1$, $b = 2$ und verwenden Sie jeweils 20 Punkte in t und y Richtung.
Hinweis: Arbeiten Sie im MATLAB-Template `logistischeDGL_richtungsfeld.m`.

b) Ergänzen Sie die MATLAB-Funktion

```
[t,y] = expEuler(f,t0,T,y0,N),
```

die die Lösung eines allgemeinen Anfangswertproblem

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0$$

zum Endzeitpunkt $T > t_0$ mit N Schritten des expliziten Euler Verfahrens approximiert.

c) Lösen Sie mit Ihrer `expEuler` Funktion aus **b)** die Gl. (1) für die Anfangswerte $y(0) = 0.2$ und $y(0) = 1$. Plotten Sie die beiden Lösungskurven in das Richtungsfeld von **a)**. Verwenden Sie jeweils $N = 100$ Euler Schritte.

d) Nun wollen wir untersuchen wie gut das Euler Verfahren funktionniert. Hierzu betrachten wir das (einfachste!) Anfangswertproblem

$$\begin{aligned} \dot{y}(t) &= y(t) \\ y(0) &= 1. \end{aligned} \tag{2}$$

Dieses hat bekanntlich die exakte Lösung

$$y(t) = e^t.$$

Lösen Sie (2) mit dem expliziten Euler Verfahren bis zur Zeit $T = 2$ mit $N = 2^i$ ($i = 0, 1, ..., 10$) Schritten und berechnen Sie den absoluten Fehler zur Endzeit

$$E_N = |y_N - y(T)|.$$

Plotten Sie den absoluten Fehler E_N als Funktion des Zeitschritts $h = (T - t_0)/N$ in einem `loglog`-Plot. Mit Ihrem Plot bestimmen Sie p, die sog. Ordnung des Verfahrens, graphisch die abhängig des Fehlers als Funktion von h, d.h. $E_N = O(h^p)$.
Hinweis: Arbeiten Sie im MATLAB-Template `expEulerConv.m`.

Siehe nächstes Blatt!

3. In einfachen Fällen kann man mittels des sog. Picard-Lindelöfschen Iterationsverfahrens die Lösungen einer Differentialgleichung explizit berechnen. Wir betrachten das AWP

$$\begin{aligned}\dot{\mathbf{y}}(t) &= \mathbf{A}\mathbf{y}(t),\\ \mathbf{y}(0) &= \mathbf{c},\end{aligned}$$

wobei $\mathbf{y}, \mathbf{c} \in \mathbb{R}^n$ und $\mathbf{A} \in \mathbb{R}^{n\times n}$.
In diesem Fall ist ein Schritt dieses Verfahrens gegeben durch

$$\mathbf{y}_{k+1}(t) = \mathbf{c} + \int_0^t \mathbf{A}\mathbf{y}_k(\tau)d\tau,$$

zusammen mit $\mathbf{y}_0(t) = \mathbf{c}$. Berechnen Sie sie die ersten Iterationen dieses Verfahrens und folgern Sie daraus die exakte Lösung des AWPs.

Hinweis: Folgende Reihe könnte von nutzen sein:

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}t)^k}{k!} = \mathbf{I} + \mathbf{A}t + \frac{(\mathbf{A}t)^2}{2} + \frac{(\mathbf{A}t)^3}{6} + \dots$$

4. *Umformen von Differentialgleichungen und Anfangswertproblemen*

a) Formen Sie folgendes Anfangswertproblem vierter Ordnung

$$y^{(4)}(t) = \cos\left(\ddot{y}(t)\right) + \dot{y}(t)e^{-5t}$$

mit Anfangswerten

$$y(t_0) = 0,\ \dot{y}(t_0) = 3,\ \ddot{y}(t_0) = -1,\ \dddot{y}(t_0) = 0$$

in ein Anfangswertproblem erster Ordnung um.

b) Eine gewöhnliche Differentialgleichung heisst autonom, falls die rechte Seite die form $\mathbf{f} = \mathbf{f}(\mathbf{y}(t))$ hat (anstatt $\mathbf{f} = \mathbf{f}(t, \mathbf{y}(t))$, d.h. $\mathbf{f}$ hängt nicht explizit von t ab). Eine gewöhnliche Differentialgleichung $\mathbf{y}(t) = \mathbf{f}(t, \mathbf{y}(t))$ kann man autonomisieren durch das Einführen einer neuen Variabel

$$\mathbf{z}(t) = \begin{pmatrix} \mathbf{y}(t) \\ t \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{z}}_1(t) \\ z_{n+1}(t) \end{pmatrix}$$

und der neuen rechten Seite

$$\mathbf{g}(\mathbf{z}(t)) = \begin{pmatrix} \mathbf{f}(t, \mathbf{y}(t)) \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{f}(z_{n+1}, \tilde{\mathbf{z}}_1(t)) \\ 1 \end{pmatrix}.$$

Autonomisieren Sie das Anfangswertproblem

$$\begin{aligned}\dot{y}(t) &= -y(t) + \cos(t)e^{-t},\\ y(0) &= 7.\end{aligned}$$

Bitte wenden!

c) Autonomisieren Sie das Anfangswertproblem aus **a)**.

Abgabe: Bis Freitag, den 27.03.2020.
Laden Sie Ihre Matlab-Programme und eingescannte, fotografierte oder direkt digitale schriftlichen Ergebnisse (Dateigrösse < 25MB) unter `sam-up.math.ethz.ch` hoch.