

6. Übungsblatt (Dienstag 23/6/2015)

Bitte bearbeiten Sie das Ihnen zugeordnete Beispiel. Bereiten Sie eine kurze Präsentation an der Tafel oder mit Beamer vor und geben Sie eine ausführliche schriftliche Ausarbeitung (3-4 Seiten) Ihres Beispiels spätestens bis 28/6/2015 ab. Numerische Beispiele (*) mit MATLAB zu lösen. Wenden Sie sich bei Unklarheiten an Stefan Wurm (stefan.wurm@tuwien.ac.at) oder an Caroline Geiersbach (caroline.geiersbach@tuwien.ac.at).

1. (a) Zeigen Sie, dass das explizite Eulerverfahren für das Anfangswertproblem $y'(t) = \lambda y$, $\lambda \in \mathbb{R}$, mit $y(t_0) = y_0 > 0$ die Näherungswerte $\eta_\nu = (1 + h\lambda)^\nu y_0$ mit $\nu \geq 0$ ergibt und dass das implizite Eulerverfahren für dieses Problem die Näherungswerte $\eta_\nu = \frac{1}{(1-h\lambda)^\nu} y_0$ mit $\nu \geq 0$ produziert.
(b) (*) Lösen Sie die Riccati – Gleichung $y'(t) = t^2 + y^2$, $y(0) = -4$ für $0 \leq t \leq 2$ in MATLAB mit dem expliziten Eulerverfahren. Verwenden Sie die MATLAB Routine `ode45` um eine Referenzlösung zu erzeugen und damit den Fehler zu berechnen.
2. (*) Lösen Sie das Lotka – Volterra System

$$x' = x(y - 1) \quad y' = y(1 - x)$$

für $0 \leq t \leq 14$ mit den Anfangswerten $x(0) = 2$ und $y(0) = 2$ in MATLAB mit dem 1) expliziten Eulerverfahren und 2) impliziten Eulerverfahren und vergleichen Sie die Ergebnisse. Verwenden Sie jeweils verschiedene Schrittweiten und stellen Sie die Lösungen in einem Phasenporträt (x - y -Ebene) dar.

3. (a) (*) Programmieren Sie das explizite Eulerverfahren in MATLAB. Verwenden Sie Ihre Implementation und lösen Sie das lineare Differentialgleichungssystem

$$x' = y \quad y' = -x$$

mit den Anfangswerten $x(0) = 1$ und $y(0) = 0$ auf dem Intervall $[0, b]$ mit $b = 2\pi$, 10π und 200π . Erklären Sie die Beobachtungen.

- (b) Gegeben sei das Anfangswertproblem

$$y' = t - t^3, \quad y(0) = 0.$$

Zur Schrittweite h sollen mit dem expliziten Eulerverfahren Näherungswerte η_ν für $y(t_\nu)$ mit $t_\nu = \nu h$ berechnet werden. Geben Sie η_ν und $y(t_\nu)$ explizit an und zeigen Sie, dass an jeder Stelle t der lokale Fehler für $h = t/n \rightarrow 0$ gegen Null konvergiert.

4. (*) Bestimmen Sie die numerische Lösung des Differentialgleichungssystem

$$x' = \frac{y}{x^2 + y^2} \quad y' = -\frac{x}{x^2 + y^2}$$

zu den Anfangswerten $x(0) = 1$, $y(0) = 0$ mit:

- (a) dem expliziten Eulerverfahren und
 - (b) dem klassischen 4-stufigen Runge - Kutta Verfahren 4.Ordnung. Vergleichen Sie Ihre Ergebnisse mit der analytischen Lösung $x(t) = \cos t$ und $y(t) = \sin t$.
5. (*) Lösen Sie die Van der Polsche Differentialgleichung

$$y'' - \lambda(1 - y^2)y' + y = 0 \quad y(0) = 2, y'(0) = 0$$

für $\lambda = 0$ und $\lambda = 12$ numerisch mit dem expliziten Eulerverfahren und dem klassischen Runge – Kutta Verfahren der Ordnung 4. Verwenden Sie jeweils die Schrittweite $h = 0.025$ und $h = 0.0025$ und geben Sie tabellarisch die Näherungswerte an den Gitterpunkten $t = 0.5, 1.0, 1.5, \dots, 15$ an.

Hinweis: Schreiben Sie das gegebene Anfangswertproblem als System 1. Ordnung.

6. (*) Betrachten Sie das Anfangswertproblem

$$y' = -50(y - \cos t) \quad y(0) = 0.997$$

auf dem Intervall $[0, 10]$ mit der exakten Lösung

$$y = \frac{2500}{2501} \cos t + \frac{50}{2501} \sin t - \frac{6503}{250100} e^{-50t} \approx \cos(t - 0.02) - 0.0026e^{-50t}.$$

Lösen Sie das Problem mit dem expliziten und dem impliziten Eulerverfahren mit Schrittweiten $h = 10/n$ für $n = 250, 248$ und 246 . Stellen Sie die Lösungen und Fehler auch grafisch dar.

7. (*) *Explizites / Implizites Eulerverfahren*

- (a) Das explizite Eulerverfahren ist durch $\eta_\nu = \eta_{\nu-1} + hf(t_{\nu-1}, \eta_{\nu-1})$ definiert. Implementieren Sie für den Spezialfall $f(t, y) = M(t)y + g(t)$ das explizite Euler-Verfahren mit konstanter Schrittweite $h = (b - a)/N$ zur Lösung des Anfangswertproblems $y(a) = y_0, \quad y' = f(t, y)$ in $[a, b]$. Dabei kann die gesuchte Lösung $y : [a, b] \rightarrow \mathbb{R}^n$ vektorwertig sein, $M(t)$ ist in diesem Fall eine $n \times n$ Matrix. Der MATLAB-Funktion

$$[t, y] = \text{euler}(a, b, N, y_0, M, g)$$

sollen neben den Daten a, b, N und y_0 auch die *function handles* der Funktionen $Mt=M(t)$ und $gt=g(t)$ übergeben werden. Die übergebenen Funktionen können dann beispielsweise mit $Mt=\text{feval}(M, t)$ aus der Funktion `euler` heraus aufgerufen werden. Als Rückgabeparameter liefere die Funktion `euler` den Stützstellenvektor $t \in \mathbb{R}^{N+1}$ und die Matrix der dazugehörigen Funktionswerte $y \in \mathbb{R}^{n \times (N+1)}$, d.h. die $(\nu + 1)$ -te Spalte von y entspricht der Approximation η_ν an der Stelle t_ν .

- (b) Das implizite Eulerverfahren ist durch $\eta_\nu = \eta_{\nu-1} + hf(t_\nu, \eta_\nu)$ definiert. Schreiben Sie unter denselben Voraussetzungen wie in (a) eine MATLAB-Funktion

$$[t, y] = \text{ieuler}(a, b, N, y_0, M, g)$$

die das implizite Eulerverfahren realisiert. Vergleichen Sie die Fehler $|y(1) - \eta_N(1)|$ für explizites und implizites Eulerverfahren im (doppelt logarithmischen) Plot über $N = 2^i, i = 1, \dots, 10$. Als Beispiel nehmen Sie das Modellproblem

$$y(0) = 1, \quad y' = \lambda y \text{ auf } [0, 1]$$

mit exakter Lösung $y(t) = \exp(\lambda t)$ für verschiedene konstante Werte von $\lambda \in \{\pm 1, \pm 10\}$. Was beobachten Sie?

8. (*) *Runge-Kutta Verfahren*

- (a) Implementieren Sie das (explizite) Runge-Kutta Verfahren der Ordnung 4 in einer MATLAB-Funktion

$$y = \text{rk4}(f, y_0, t)$$

der als Eingabeparameter das *function handle* der rechten Seite $f = f(t, y)$, der Anfangswert $y_0 = y_0$ sowie der Vektor der Stützstellen $t = (t_0, t_1, \dots, t_N)$ übergeben werde. Dabei darf die unbekannte Lösung auch vektorwertig sein, d.h. $y_0 \in \mathbb{R}^n$ ist ein Spaltenvektor. Verwenden Sie das Modellproblem

$$y(0) = 1, \quad y' = \lambda y \text{ auf } [0, 1]$$

mit exakter Lösung $y(t) = \exp(\lambda t)$ für verschiedene konstante Werte von $\lambda \in \{\pm 1, \pm 10\}$ und geben Sie jeweils die Konvergenzordnungen aus, also die Ordnung des globalen Fehlers.

- (b) Schreiben Sie ein MATLAB-Programm

$$[p] = \text{ordnungrk4}(nmax)$$

welches die Konvergenzordnung des RK4-Verfahrens für das Modellproblem $y' = y$ und $t_0 = 0, y_0 = 1, T = 1$ schätzt. Das RK4-Verfahren soll die Schrittweiten $h = 2^{-n}, n = 0, 1, \dots$ verwenden. Die Anzahl der Schritte N ist durch $\frac{T}{h}$ gegeben. Fitten Sie den tatsächlichen Fehler $\varepsilon(h) := |y(T) - y_N|$ an ein Gesetz der Form $\varepsilon(h) = Ch^p$, wobei Sie die Daten für $h = 2^{-n}, n \in \{n_{max} - 2, n_{max} - 1, n_{max}\}$ verwenden.

Hinweise: Das Gesetz ist äquivalent zu $\log \varepsilon(h) = \log C + p \log h$; `help polyfit`.

9. (*) Schreiben Sie ein MATLAB-Programm mit der Signatur

$$[iter, y] = \text{IRK}(f, fy, t_0, y_0, h, A, b, c, tol),$$

welches einen Schritt der Länge h eines impliziten Runge-Kutta-Verfahrens macht. Dabei sind \mathbf{b} , \mathbf{c} Spaltenvektoren und \mathbf{A} eine Matrix, die zusammen das Butcherschema des IRK beschreiben. Die Lösung der Gleichung für die Stufen soll mit dem Newtonverfahren bestimmt werden, wobei die Abbruchbedingung (der Einfachheit halber) ist, dass die 2-Norm der Differenz zweier aufeinanderfolgender Iterierten $\leq \text{tol}$ ist. \mathbf{f} und \mathbf{fy} sollen *function handles* sein für die skalaren Funktionen $(t, y) \mapsto f(t, y)$ und $(t, y) \mapsto f_y(t, y)$. Die Ausgabevariable \mathbf{iter} soll die Anzahl benötigter Newtonschritte sein. Testen Sie Ihr Programm mit dem 2-stufigen Runge-Kutta Verfahren gegeben durch

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Plotten Sie für $f(t, y) = \lambda y, y(t_0) = 1$, die Konvergenz, d.h. "Anzahl Schritte N gegen Fehler bei $T = 1$ " bei unformer Schrittweite. Für $\lambda = -25$ geben Sie auch die maximale Anzahl Newtonschritte an, die Sie in einem Schritt Ihres IRK benötigt haben.

10. (*) Gegeben sei das AWP

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} 2 \sin t \\ 2(\cos t - \sin t) \end{pmatrix}, \quad \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

mit exakter Lösung $y_1(t) = 2e^{-t} + \sin t, y_2(t) = 2e^{-t} + \cos t$. Es soll der Fehler $\epsilon(h) := \|y(T) - y_N\|_2$ für $T = 1$ als Funktion der Schrittweite h (verwenden Sie uniforme Gitter) doppelt logarithmisch dargestellt werden. Vergleichen Sie das RK4-Verfahren mit dem expliziten Eulerverfahren für die Fälle $h = 2^{-n}, n = 1, \dots, 20$ bei Verwendung von *doppelt genauer* Arithmetik und *einfacher Genauigkeit*. Erklären Sie Ihre Beobachtungen.

Hinweis: `help single` in MATLAB.

11. (*) Testen Sie die in MATLAB vorhandenen Routinen `ODE15s`, basierend auf BDF Verfahren und `ODE45`, basierend auf Runge-Kutta Verfahren anhand des folgenden Beispiels:

$$\begin{aligned} y_1' &= -y_2 - \lambda y_1(1 - y_1^2 - y_2^2) \\ y_2' &= y_1 - 3\lambda y_2(1 - y_1^2 - y_2^2) \end{aligned}$$

Wählen Sie mehrere geeignete Anfangswerte $y_1(0), y_2(0)$ und mit $y_1(0)^2 + y_2(0)^2 \neq 1$, setzen Sie für λ die Werte -10 und -100 ein. Plotten Sie das entsprechende Phasenporträt. Verwenden Sie für Informationen zu den genannten Routinen die bei MATLAB vorhandene Online Hilfe.

12. (a) Zeigen Sie, dass das RK4-Verfahren angewandt auf das Anfangswertproblem

$$y' = y, \quad y(0) = y_0$$

eine Rekurrenz der Form $y_1 = R(h)y_0$ mit

$$R(h) = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \frac{h^4}{4!}$$

besitzt.

- (b) Welche Quadraturformel wird durch das explizite Runge-Kutta-Verfahren der Ordnung 4 induziert? Welche Quadraturformel wird durch das modifizierte Eulerverfahren und das Verfahren von Heun, welche durch die folgenden Butcherschemata beschrieben werden:

$$\text{modifizierter Euler: } \begin{array}{c|c} 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \\ \hline & 0 \quad 1 \end{array}$$

$$\text{Heun: } \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

13. (*) Ein Schritt des BDF2 Verfahrens ist gegeben durch:

$$\eta_{\nu+2} - \frac{4}{3}\eta_{\nu+1} + \frac{1}{3}\eta_{\nu} = \frac{2}{3}hf(t_{\nu+2}, \eta_{\nu+2}).$$

Implementieren Sie für den Spezialfall $f(t, y) = M(t)y + g(t)$ dieses Verfahren mit konstanter Schrittweite $h = (b - a)/N$ zur Lösung des Anfangswertproblems $y(a) = y_0, y' = f(t, y)$ in $[a, b]$. Dabei kann die gesuchte Lösung $y : [a, b] \rightarrow \mathbb{R}^n$ vektorwertig sein, $M(t)$ ist in diesem Fall eine $n \times n$ Matrix. Der MATLAB-Funktion

$$[\mathbf{t}, \mathbf{y}] = \text{bdf2}(\mathbf{a}, \mathbf{b}, N, y_0, y_1, \mathbf{M}, \mathbf{g})$$

soll neben den Daten a, b, N und Startwerten y_0, y_1 auch die Function-Handles der Funktionen $\mathbf{M}\mathbf{t}=\mathbf{M}(\mathbf{t})$ und $\mathbf{g}\mathbf{t}=\mathbf{g}(\mathbf{t})$ übergeben werden. Dabei muss der zusätzlich benötigte Startwert mit einem geeigneten Einschrittverfahren berechnet werden. Die übergebenen Funktionen können dann beispielsweise mit $\mathbf{M}\mathbf{t}=\text{feval}(\mathbf{M}, \mathbf{t})$ aus der Funktion `bdf2` heraus aufgerufen werden. Als Rückgabeparameter liefere die Funktion `bdf2` den Vektor der Stützstellen $t \in \mathbb{R}^{N+1}$ und die Matrix der dazugehörigen Funktionswerte $y \in \mathbb{R}^{n \times (N+1)}$, d.h. die ν -te Spalte

von y entspricht der Approximation η_ν an der Stelle t_ν . Wählen Sie konkret $M(t) = \begin{pmatrix} -10t & 0 \\ -(10t+1)\cos t & 1 \end{pmatrix}$,

$g(t) = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}$, $[a, b] = [1, 4]$ und $y_0 = (1, 1)^T$ und stellen Sie die Ordnung des Verfahrens dar.