# Serie 5 (TM)

Diskussion des Blattes: Fr., 28.4.2023

**1.** Sei $a(\cdot,\cdot)$ eine Bilinearform mit $\gamma\|v\|^2_{H^1(\Omega)} \le a(v,v)$ für alle $v \in H^1_0(\Omega)$. Das $\theta$-Schema ist gegeben durch

$$\frac{1}{k}(u_h^{n+1} - u_h^n, v)_{L^2(\Omega)} + a(u_h^{n+\theta}, v) = (f(t_{n+\theta}), v)_{L^2(\Omega)} \qquad \forall v \in V_h,$$

wobei $u_h^{n+\theta} = \theta u_h^{n+1} + (1-\theta)u_h^n$ und $t_{n+\theta} = \theta t_{n+1} + (1-\theta)t_n$. Zeigen Sie folgende Stabiliätsaussage für $\theta \in [1/2, 1]$:

$$\|u_h^n\|^2_{L^2(\Omega)} + \sum_{i=0}^{n-1} k\gamma\|u_h^{i+\theta}\|^2_{H^1(\Omega)} + (2\theta-1)\|u_h^{i+1} - u_h^i\|^2_{L^2(\Omega)} \le \|u_h^0\|^2_{L^2(\Omega)} + C\sum_{i=0}^{n-1} k\|f(t_{i+\theta})\|^2_{L^2(\Omega)}.$$

**2.** Das Eulerverfahren und das Crank-Nicolson-Verfahren für das AWP $y' = f(t,y)$, $y(0) = y_0$ sind definiert durch die Rekurrenzen

$$\begin{aligned}
\text{expl. Euler} \quad & y^{n+1} = y^n + kf(t_n, y^n) \\
\text{impl. Euler} \quad & y^{n+1} = y^n + kf(t_{n+1}, y^{n+1}) \\
\text{Crank-Nicolson} \quad & y^{n+1} = y^n + kf(t_{n+1/2}, \frac{1}{2}(y^{n+1} + y^n)), \qquad t_{n+1/2} = \frac{1}{2}(t_{n+1} + t_n)
\end{aligned}$$

**a)** Geben Sie eine explizite Formel für $y^n$ an, wenn das das skalare AWP

$$y' = \lambda y, \qquad y(0) = y_0 \tag{1}$$

betrachtet wird. Ihre Formel definiert für jedes der 3 Verfahren die *Stabilitätsfunktion* $R(z)$ durch die Beziehung

$$y^{n+1} = R(z)y^n, \qquad z = \lambda k.$$

Geben Sie $R$ an.

**b)** Die Stabilitätsfunktion $R(z)$ ist für $z \in \mathbb{C}$ definiert. Das *Stabilitätsgebiet* $S$ eines numerischen Verfahrens ist definiert als $S = \{z \in \mathbb{C} \,|\, |R(z)| \le 1\}$. Skizzieren Sie das Stabilitätsgebiet für die drei obigen Verfahren.

**c)** Sei $\lambda < 0$. Dann ist die exakte Lösung von (1) beschränkt für alle $t > 0$. Geben Sie die Schrittweite $k$ an, so daß dieses qualitative Verhalten auch vom numerischen Verfahren geerbt wird.

**3.** Für das AWP $y' = f(t,y)$ haben *explizite Einschrittverfahren* die Form $y^{n+1} = y^n + k\Phi(t_n, k, y_n)$ [1] Explizite Runge-Kutta-Verfahren mit $s$ *Stufen* sind von der Form

$$\begin{aligned}
\Phi(t,y,h) &:= \sum_{i=1}^{s} b_i k_i, \\
k_1 &:= f(t,y), \\
k_2 &:= f(t + c_2 h, y + h a_{21} k_1), \\
k_3 &:= f\left(t + c_3 h, y + h(a_{31}k_1 + a_{32}k_2)\right), \\
\vdots \quad & \vdots \quad \vdots \\
k_s &:= f\left(t + c_s h, y + h(a_{s1}k_1 + \cdots + a_{s\,s-1}k_{s-1})\right).
\end{aligned}$$

Die Zahlen $a_{ij}$, $c_i$, $b_i$, die ein Runge-Kutta-Verfahren festlegen, werden üblicherweise kompakt in einem Tableau wie folgt notiert:

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s\,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

[1] Bsp: expliziter Euler mit $\Phi(t_n, k, y^n, y^{n+1}) = f(t_n, y^n)$.

oder wesentlich kompakter als $\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^\top \end{array}$ mit $\mathbf{c} \in \mathbb{R}^s$, $\mathbf{b} \in \mathbb{R}^s$, $\mathbf{A}^{s\times s}$ wobei $\mathbf{A}$ eine untere Dreiecksmatrix ist.

**a)** Betrachten Sie die ODE $y' = \lambda y$. Zeigen Sie, daß die Stabilitätsfunktion $R = R(z)$ (definiert wie in Aufg. 2) für ein explizites Runge-Kutta-Verfahren von der obigen Form ein Polynom (vom Grad $s$) ist. (Hier ist wieder $z = \lambda k$).

**b)** Man nennt ein Runge-Kutta-Verfahren A-stabil, falls das Stabilitätsgebiet $S = \{z \in \mathbb{C} \,|\, |R(z)| \leq 1\}$ die Halbebene $\{z \in \mathbb{C} \,|\, \operatorname{Re} z < 0\}$ enthält. Können explizite Runge-Kutta-Verfahren A-stabil sein?

**4.** Die folgende Aufgabe müssen Sie nicht unbedingt mit MATLAB lösen. Sie können auch NgSolve verwenden und eine interessantere Geometrie verwenden. In diesem Fall soll als Fehlermaß der $L^2(\Omega)$ zu Endzeitpunkt zurückgegeben werden, wobei der Fehler der Einfachheit dadurch bestimmt wird, daß Sie die Lösung mit einer genaueren Lösung vergleichen, die Sie z.B. durch Halbieren der Ortsschrittweite und der Zeitschrittweite erhalten.

Erstellen Sie ein MATLAB-Programm mit der Signatur

$$[\mathtt{u}] = \mathtt{heat_eqn_1D_implicit_euler}(\mathtt{x}, \mathtt{k}, \mathtt{K}, \mathtt{u0}, \mathtt{f}),$$

das eine Approximation an die Lösung des Problems

$$u_t - u_{xx} = f, \qquad \text{auf } \Omega \times (0,T), \qquad u(x,t) = 0 \quad (x,t) \in \partial\Omega \times (0,T)$$

erzeugt. Dabei soll das implizite Eulerverfahren in der Zeit und klassische FEM im Ort verwendet werden. Hier ist x ein (sortierter) Vektor von Knoten, der das Ortsgitter beschreibt (d.h. $\Omega = (\mathtt{min}(x), \mathtt{max}(x))$. $k$ ist die Zeitschrittweite, $K$ ist die Anzahl Zeitschritte die durchgeführt werden soll. $u0$ ist ein Vektor (der Länge $\mathtt{length}(x) - 2$) mit Knotenwerten für den Startwert. $f$ ist ein *function handle* für eine Funktion $f : (x,t) \mapsto f(x,t)$, die die rechte Seite beschreibt. Die Ausgabe soll ein Vektor der Länge $\mathtt{length}(x) - 2$ sein, der aus den Knotenwerten der Approximation zum Zeitpunkt $Kk$ besteht. Testen Sie Ihr Programm mit der exakten Lösung $u(x,t) = e^{-t}x(1-x)$ zum Zeitpunkt $T = 1$ und $k = 0.1h$ (sie sollten $O(h)$ sehen, wenn Sie als Fehlermaß $\sqrt{h}\sqrt{\sum_i |u(x_i,T) - u_i^K|^2}$ nehmen; dieses Fehlermaß imitiert die $L^2(\Omega)$-Norm).

**5.** **a)** Falls Sie in Aufg. 4 NgSolve verwendet haben, dann modifizieren Sie Ihren Code entsprechend.

Modifizieren Sie Ihr MATLAB-Programm aus Aufg. 4 so, daß Sie das explizite Eulerverfahren realisieren, und plotten Sie wieder den Fehler gegen die Ortsschritteweite $h$. Für die Zeitschrittweite $k$ verwenden Sie zwei verschiedene Wahlen:

$$k = \frac{2.001}{\lambda_{max}}, \qquad k = \frac{1.999}{\lambda_{max}},$$

wobei $\lambda_{max}$ der größte Eigenwert des EWP

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{M}\mathbf{x}$$

ist (hier sind $\mathbf{M}$ und $\mathbf{A}$ die Masse- und Steifigkeitsmatrix). Verwenden Sie $h = 2^{-i}$, $i = 3, \ldots, 8$. Plotten Sie auch $\lambda_{max}$ gegen die Ortsschrittweite $h$.

**b)** Formulieren Sie das Verfahren, wenn Sie anstelle des expliziten Eulerverfahrens das klassische Runge-Kutta-Verfahren der Ordnung 4 (kurz: RK4) verwenden. RK4 ist gegeben durch das Tableau

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

Modifizieren Sie Ihr Programm, so daß Sie mir RK4 rechnen anstelle des expliziten Eulerverfahrens. Verwenden Sie die beiden Wahlen $k = 2/\lambda_{max}$ und $k = 3/\lambda_{max}$.