Numerische Mathematik 2, Übungen, WS15/16, Blatt 11

Bearbeitung: Hausaufgaben bis 7.1.2016, Programmieraufgabe bis 14.1.2016

Die folgenden Hausaufgaben haben mit *impliziten* Runge-Kutta Methoden zu tun, die für die numerische Lösung von Evolutionsgleichungen wie folgt definiert werden. Für eine gegebene Lipschitz-stetige Funktion $\boldsymbol{f} : \mathbb{R}^{d+1} \to \mathbb{R}^d$ soll das Anfangswertproblem

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad \boldsymbol{y}(0) = \boldsymbol{y}_0, \quad t \geq 0, \quad \boldsymbol{y} : \mathbb{R} \mapsto \mathbb{R}^d$$

gelöst werden. Für gegebene Parameter $A = \{a_{ij}\}_{i,j=1}^q$, $\boldsymbol{b} = \{b_i\}_{i=1}^q$, $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^q$, gegebene Zeiten $\{t_i = ih, i \in \mathbb{N}_0, h > 0\}$ und eine gegebene Approximation $\boldsymbol{y}_n \approx \boldsymbol{y}(t_n)$ sei $\boldsymbol{y}(t_{n+1})$ approximiert durch die Lösung $\boldsymbol{y}_{n+1}$ des folgenden Systems

$$\begin{aligned} \boldsymbol{y}_{n,i} &= \boldsymbol{y}_n + h \sum_{j=1}^{q} a_{ij} \boldsymbol{f}(t_n + h\tau_j, \boldsymbol{y}_{n,j}) \\ \boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + h \sum_{i=1}^{q} b_i \boldsymbol{f}(t_n + h\tau_i, \boldsymbol{y}_{n,i}) \end{aligned}$$

Die q Größen $\{\boldsymbol{y}_{n,i}\}_{i=1}^q$ sind die *Stufen*. Eine Runge-Kutta Methode ist *explizit* wenn gilt $a_{ij} = 0$, $j \geq i$, und sonst ist die Methode *implizit*. Die Methode wird mit einer Tabelle so dargestellt:

$$\begin{array}{c|c} \boldsymbol{\tau} & A \\ \hline & \boldsymbol{b} \end{array}$$

Zum Beispiel, die 2-stufige SDIRK (*Singly Diagonally Implicit Runge-Kutta*) Methode dritter Ordnung,

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \gamma = \tfrac{1}{2} + \tfrac{1}{6}\sqrt{3}$$

hat verschiedene Vorteile, die man unten sieht.

Hausaufgaben

1. Der lokale Abbruchsfehler $\boldsymbol{d}(t)$ der Runge-Kutta Methode ergibt sich durch Anwendung der Methode auf die exakte Lösung,

$$\begin{aligned} \boldsymbol{y}_i(t) &= \boldsymbol{y}(t) + h \sum_{j=1}^{q} a_{ij} \boldsymbol{f}(t + h\tau_j, \boldsymbol{y}_j(t)) \\ \boldsymbol{d}(t) &= \boldsymbol{y}(t+h) - \boldsymbol{y}(t) - h \sum_{i=1}^{q} b_i \boldsymbol{f}(t + h\tau_i, \boldsymbol{y}_i(t)) \end{aligned}$$

   Die Methode hat Ordnung p wenn $\boldsymbol{d}(t) = \mathcal{O}(h^{p+1})$. Um notwendige Bedingungen für eine gewisse Ordnung herzuleiten, wendet man die Methode auf das folgende bestimmte Anfangswertproblem an:

$$y'(t) = f(t, y(t)) = y(t) + t^{k-1}, \quad y(0) = 0$$

   deren Lösung gegeben ist durch

$$y(t) = \int_0^t e^{t-s} s^{k-1} ds.$$

(a) Zeigen Sie, es gilt

$$y(0) = y'(0) = \cdots = y^{(k-1)} = 0, \quad y^{(k+j)}(0) = (k-1)!, \quad j \geq 0$$

(b) Zeigen Sie, mit den Stufen

$$y_i(0) = h \sum_{j=1}^{q} a_{ij}(y_j(0) + (h\tau_j)^{k-1}), \quad Y(0) = (y_1(0), \ldots, y_q(0))^\top$$
$$(1 - hA)Y(0) = h^k A T^{k-1}\mathbf{1}, \quad Y(0) = h^k (I - hA)^{-1} A T^{k-1}\mathbf{1}$$
$$T = \operatorname{diag}(\boldsymbol{\tau}), \quad \mathbf{1} = (1, \ldots, 1)^\top$$

ist der lokale Abbruchsfehler $d(0)$ gegeben durch

$$d(0) = y(h) - y(0) - h \sum_{i=1}^{q} b_i [y_i(0) + (h\tau_i)^{k-1}]$$
$$= y(h) - h^k \boldsymbol{b}^\top \left[(I - hA)^{-1} hA + I\right] T^{k-1}\mathbf{1} = \sum_{l=k}^{\infty} \frac{(k-1)!}{l!} h^l - h^k \boldsymbol{b}^\top \left[\sum_{l=0}^{\infty} h^l A^l\right] T^{k-1}\mathbf{1}$$

wobei die letzte unendliche Summe für h ausreichend klein folgt.

(c) Zeigen Sie anhand der letzten Ergebnisse, eine Methode mit Ordnung p muss erfüllen,

$$\boldsymbol{b}^\top A^l T^{k-1}\mathbf{1} = \frac{(k-1)!}{(k+l)!}, \quad 1 \leq k + l \leq p.$$

(d) Zeigen Sie, die 2-stufige SDIRK Methode erfüllt diese Bedingungen für $p = 3$.

2. Eine Runge-Kutta Methode ist absolut stabil, wenn bei der Anwendung für die Evolutionsgleichung $y'(t) = \lambda y(t)$, $\Re\{\lambda\} < 0$, gilt $y_n \to 0$ für $n \to \infty$.

(a) Zeigen Sie, wenn eine Runge-Kutta Methode zur Lösung der Evolutionsgleichung $y'(t) = \lambda y(t)$, $\Re\{\lambda\} < 0$, angewendet wird, gilt $y_n = R(h\lambda) y_{n-1}$, $n \in \mathbb{N}$, wobei

$$R(z) = 1 + z\boldsymbol{b}^\top (I - zA)^{-1}\mathbf{1}$$

(b) Das Gebiet $G \subset \mathbb{C}$ der absoluten Stabilität einer Runge-Kutte Methode besteht aus den Werten $z = h\lambda$, für die die Methode absolut stabil ist. Wenn $(-\infty, 0) \subset G$ gilt, ist eine Methode A_0-stabil. Eine Methode ist A-stabil, wenn $G = \{z \in \mathbb{C} : \Re\{z\} < 0\}$ gilt.
Zeigen Sie für die SDIRK Methode, es gelten

$$R(z) = \frac{2 + (2 - 4\gamma)z + (1 - 4\gamma + 2\gamma^2)z^2}{2(\gamma z - 1)^2}, \quad |R(z)| < 1 \quad \text{für} \quad z < 0$$

und daher ist die SDIRK Methode A_0-stabil.

Bonus: Zeigen Sie, die SDIRK Methode ist tatsächlich A-stabil.

3. Man errinert sich an die Lipschitz-Stetigkeit der Funktion $\boldsymbol{f}$,

$$\|\boldsymbol{f}(t, \boldsymbol{y}_2) - \boldsymbol{f}(t, \boldsymbol{y}_1)\| \leq L\|\boldsymbol{y}_2 - \boldsymbol{y}_1\|, \quad \forall t \geq 0, \quad \forall \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^d.$$

Seien die Stufen $\boldsymbol{y}_{n,i}$ durch $\boldsymbol{y}_{n,i}^l$, $\boldsymbol{y}_{n,i}^0 = \boldsymbol{y}_n$, in der folgenden Iteration approximiert:

$$\boldsymbol{y}_{n,i}^l = \boldsymbol{y}_n + h \sum_{j=1}^{q} a_{ij} \boldsymbol{f}(t_n + h\tau_j, \boldsymbol{y}_{n,j}^{l-1}), \quad l = 1, 2, \ldots$$

(a) Zeigen Sie für $\boldsymbol{Y}_n^l = (\boldsymbol{y}_{n,1}^l; \ldots; \boldsymbol{y}_{n,q}^l) \in \mathbb{R}^{qd}$, es gilt

$$\|\boldsymbol{Y}_n^{l+1} - \boldsymbol{Y}_n^l\|_2 \leq hL\sqrt{q}\|A\|_1\|\boldsymbol{Y}_n^l - \boldsymbol{Y}_n^{l-1}\|_2$$

(b) Zeigen Sie für h ausreichend klein, $\{\boldsymbol{Y}_n^l\}_{l\geq 0}$ ist eine Cauchy-Folge in $\mathbb{R}^{qd}$, und daher werden die Stufen durch den eindeutigen Limes $\boldsymbol{Y}_n^\infty = (\boldsymbol{y}_{n,1}; \ldots; \boldsymbol{y}_{n,q})$ gegeben.

Programmieraufgabe

Alle Codes sollen an
`stephen.keeling@uni-graz.at`
mit Betreff
Num2 Programmieraufgabe
per Email bis zum 14.1.2015 geschickt werden.

Das Problem der anisotropischen Diffusion des 5. Blatts soll jetzt mit der SDIRK Methode gelöst werden:

$$U^{n,i} = U^n - dt\sum_{j=1}^{2} a_{ij}B_nU^{n,j}, \quad U^{n+1} = U^n - dt\sum_{i=1}^{2} b_iB_nU^{n,i}, \quad n = 0, \ldots, M-1$$

oder

$$\begin{aligned}(I + \gamma dtB_n)U^{n,1} &= U^n \\ (I + \gamma dtB_n)U^{n,2} &= U^n - dt(1-2\gamma)B_nU^{n,1}\end{aligned}$$
$$U^{n+1} = U^n - dtB_n(U^{n,1} + U^{n,2})/2, \quad n = 0, \ldots, M-1,$$

wobei die Notation B_n hier verwendet wird, um zu betonen dass die damalige Matrix B von U^n abhängt; sehen Sie `Xu` und `Yu` im Pseudo-Code am Ende des 5. Blatts. Zusätzlich wird die Notation $n = 0, \ldots, M-1$ hier zur Bequemlichkeit verwendet, aber nach wie vor gibt es M Zeit-Schritte, wie man im angesprochenen Pseudo-Code sieht.

Nun wird kein iteratives Verfahren zur Lösung der Stufen-Systeme verwendet. Die (absichtlich konstruierte) besondere Struktur der Matrix A wird ausgenutzt, in dem man (für jedes n) einmal für die beiden Stufen eine LU-Faktorisierung der Matrix $(I + \gamma dtB_n)$ durchführt.

Man erinnert sich an die Konstruktionen und die Parameter des 5. Blatts. Schreiben Sie einen Matlab Code der Form (`N`= N, `M`= M, `T`= T, `Ustar`= $u^\star$, `V`= $\boldsymbol{U}^0$, `U`= $\boldsymbol{U}^M$)

```
U = familienv(N,M,T,Ustar,V,gon)
```

Die Lösung $\boldsymbol{U}^M$ zur Zeit $t^M = T$ soll zum Schluss der Rechnungen ausgegeben werden. Mit `gon = true` soll das Ergebnis für jeden Zeit-Schritt grafisch dargestellt werden, wie man auf dem 5. Blatt sieht. Falls Sie noch weitere Parameter im Befehlsfenster ausgeben wollen, soll dies nur mit `gon=true` geschehen.

Achtung: Getested werden die Ergebnisse z.B. mit den folgenden Daten

```
N = 101; N1 = N+1; NN1 = N*N1; N1N1 = N1*N1; N4 = round(N/4);
M = 101; T = 1;
Ustar = [zeros(N4,1);ones(N1-2*N4,1);zeros(N4,1)];
Ustar = kron(Ustar,Ustar');
V = Ustar + 0.5*randn(N1,N1);
V = reshape(V,N1N1,1);
```

und mit einem Code, der zu dem am Ende des 5. Blatts ähnlich ist.