Serie 4

Abgabe: bis Fr., 11.4.08, 12 Uhr bei Frau Kovalj

4.1. (**schriftlich**) Auf $[0,1]$ betrachte man die rechte Seite $f(t,y) = (11/10)\,t^{1/10}$ und den Anfangswert $y_0 = 0$. Für eine fixe Graduierung $\beta > 0$ betrachten wir Gitter $\Delta_N^\beta = (t_0, \dots, t_N)$ mit $t_j = (j/n)^\beta$. Zur Diskretisierung verwenden wir das explizite Runge-Kutta-Verfahren der Ordnung 4 aus der Vorlesung. Welches Konvergenzverhalten $O(N^{-\alpha})$ erhält man für eine Folge uniformer Netze, d.h. für die Wahl $\beta = 1$? Wie muß man β wählen, damit das explizite Runge-Kutta-Verfahren der Ordnung 4 aus der Vorlesung mit Ordnung $O(N^{-4})$ konvergiert.

Hinweis. Die Problemstellung ist äquivalent zur numerischen Quadratur der Funktion $y(t) = (11/10)\,t^{1/10}$ über das Intervall $[0,1]$. Das RK4 ist gerade die Simpson-Regel, sodass die elementweisen Fehler für $[t_{j-1}, t_j]$ explizit berechnet werden können. Zur Abschätzung der Fehlerterme verwende man den Mittelwertsatz. Überlegen Sie sich, daß für $h_j = t_{j+1} - t_j$ gilt: $h_j \le CN^{-1} t_j^{1-1/\beta}$ ($j \ge 1$, $C > 0$ geeignet). Das Element $I_0 = [t_0, t_1] = [0, t_1]$ ist speziell. Dort ist $h_0 = t_1$. Überlegen Sie sich, daß auf I_0 der Quadraturfehler sich durch $Ch_0 \|y'\|_{L^1(I_0)}$ abschätzen läßt.

4.2. (**Programmieraufgabe 4.2**) Implementieren Sie das ein Einschrittverfahren mit Schrittweitensteuerung basierend auf dem klassischen RK4-Verfahren. Verwenden Sie die $(h, h/2)$-Strategie wie in Alg. 2.23 der Vorlesung. Verwenden Sie den Sicherheitsfaktor $\rho = 0.8$, Vergrößerungsparameter $\eta = 2$ und $h_{min} = \texttt{tol}$ sowie als Startschrittweite $h = \texttt{tol}^{1/5}$. Schreiben Sie eine `MATLAB`-Funktion

```
[t,y,Fevals,rejects]=rk4adaptive(f,y0,a,b,tol)
```

der als Eingabeparameter das Functionhandle der rechten Seite `f = f(t,y)`, der Anfangswert y_0, die Intervallgrenzen a und b des Zeitintervalls $[a,b]$ sowie die Toleranz `TOL` > 0 übergeben werden. Dabei darf die unbekannte Lösung vektorwertig sein, d.h. $y_0 \in \mathbb{R}^n$ ist ein Spaltenvektor. Die weiteren Rückgabewerte *Fevals* und *rejects* geben die Anzahl benötigter Funktionsauswertungen und abgelehnter Schritte an. Betrachten Sie die Van der Pol Gleichung

$$y'' = \mu((1-y^2)y' - y)$$

mit Parameter $\mu = 20$ und Startwert $(y(0), y'(0)) = (2, -2/3)$ auf dem Intervall $[0,4]$, um den Programmcode zu testen.

Hinweis: Transformieren Sie die Differentialgleichung 2. Ordnung in ein System 1. Ordnung.

4.3. Programmieren Sie für das Quadraturproblem aus Aufg. 1 das RK4-Verfahren auf den Gittern Δ_N^β aus Aufg. 1 für (i) $\beta = 1$ (d.h. uniforme Gitter), (ii) $\beta = 3$, (iii) $\beta = 4/1.1$, (iv) $\beta = 5$, sowie (vi) mittels des adaptiven Algorithmus aus Aufg. 2 (Toleranzen 10^{-n}, $n = 1, \dots, 10$). Plotten Sie (doppelt logarithmisch) für alle Fälle den Fehler gegen die Anzahl benötigter Funktionsauswertungen. Diskutieren das Verhalten der Verfahren anhand Ihrer Plots. Falls man die Gitter Δ_N^β aus Aufg. 1 verwendet, sollte man β lieber zu groß oder zu klein wählen? Ist ein adaptives Verfahren besser als die "richtige" Wahl von β?

4.4. Es soll das Verhalten des adaptiven Algorithmus `rk4adaptive` für Quadraturprobleme mit stückweise glatten f analysiert werden. Als Modell betrachten wir $f(t,y)$ von der Form

$$f(t,y) = \begin{cases} f_1(t) & t < t_* \\ f_2(t) & t \ge t_* \end{cases}$$

wobei f_1, f_2 glatt sind, aber f bei $t = t_*$ einen Sprung hat.

a) Verwenden Sie Ihren adaptiven Algoritmus `rk4adaptive` aus Aufg. 2 mit $h_{min} = 0$ für von Ihnen gewählte Funktionen f_1, f_2, wobei $t_* \in (t_0, T)$. Was beobachten Sie?

b) Versuchen Sie, die Beobachtung aus Teilaufg. a) zu erklären. Betrachten Sie hierzu vereinfachend den Fall, daß f_1 und f_2 konstante Funktionen sind. Geben Sie den Konsistenzfehler $\tau(t, y, h)$ an. Tun Sie nun so, als ob Ihr Algorithmus den Konsistenzfehler nicht schätzt, sondern exakt zur Verfügung hat.

c) Betrachten Sie nun den Fall $h_{min} > 0$ und f wie in Teilaufg. a). Welchen Fehler (in Abhängigkeit von h_{min} und `tol`) zum Endzeitpunkt T erwarten Sie für Ihren Algorithmus `rk4adaptive`? Wie soll man also h_{min} sinnvoll wählen?

4.5. Gegeben sei ein Einschrittverfahren mit der Inkrementfunktion

$$\Phi(t, y, h, f) = b_1 f(t, y) + b_2 f(t + c_2 h, y + a_{21} h f(t, y)).$$

Zeigen Sie, daß ein solches Verfahren nicht von Ordnung 3 sein kann.

4.6. (**Programmieraufgabe 4.6**)

a) In der Vorlesung wurde gezeigt, wie der Schrittweitenvorschlag zu wählen ist, wenn der adaptive Algorithmus auf dem *error per unit step*-Kriterium basiert. Geben Sie die Schrittweitenvorschläge an, die entstehen, wenn die Schrittweiten nach dem *error per step* Kriterium gewählt werden sollen. Betrachten Sie den Fall eines Verfahrens der Ordnung p.

b) Welche Genauigkeit zum Endzeitpunkt T erwarten Sie, wenn Sie ein auf dem *error per step*-Kriterium basierenden Algorithmus mit Toleranz `tol` verwenden?

c) Modifizieren Sie Ihren Löser `rk4adaptive` aus Aufg. 2, so daß er mit dem *error per step*-Kriterium arbeitet. Die Signatur dieses Lösers ist

```
[t,y,Fevals,rejects]=rk4adaptiveEpS(f,y0,a,b,tol)
```

d) Vergleichen Sie `rk4adaptive` und `rk4adaptiveEpS` für die Auswertung von $y(1)$, wobei die Lösung $y(t) = 1/(1 + 100t^2)$ das Anfangswertproblem $y' = -200ty^2$ mit $y(0) = 1$ löst. Vergleichen Sie hierzu insbesondere die Verfahren hinsichtlich "Fehler gegen Anzahl Funktionsauswertungen" und "Fehler gegen vorgegebene Toleranz".

4.7. Im Konvergenzsatz Satz 2.10 wurde die Lipschitzstetigkeit der Inkrementfunktion Φ benötigt. Zeigen Sie, daß diese Voraussetzung für explizite s-stufige RK-Verfahren erfüllt ist. Nehmen Sie hierzu an, daß $f \in C(\mathbb{R}^2)$ eine (globale) Lipschitzbedingung im 2. Argument mit Lipschitzkonstante $L > 0$ erfüllt. Zeigen Sie nun, daß dann die Inkrementfunktion Φ eine Lipschitzbedingung mit Konstante L_Φ erfüllt, wobei

$$L_\Phi \leq L \sum_{i=1}^{s} |b_i|(1 + \eta)^i$$

wobei

$$\eta = Lh \max_{i=1,\ldots,s} \sum_{j=0}^{i-1} |a_{ij}|.$$

Im Wesentlichen vererbt sich die Lipschitzkonstante von L auf L_Φ. *Hinweis:* Betrachten Sie die Differenz $k_i - \widehat{k}_i$, wobei k_i und $\widehat{k}_i$ die Stufen für Daten y, $\widehat{y}$ bezeichnen.