

Angewandte Numerik 2

Abgabetermin: Freitag, 04.12.2015, vor der Übung

Für dieses Übungsblatt gibt es 16 Theorie- und 21 Matlab-Punkte, sowie 9 Theorie- und 11 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.

Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 7) bei 59 Theoriepunkten und 66 Matlabpunkten.

Aufgabe 28 (*Programmieraufgabe: Konvergenzordnung verschiedener ESV*) (4M+3M+3T Punkte)

In dieser Aufgabe wollen wir den Diskretisierungsfehler der Verfahren aus Aufgabe 26 vom letzten Übungsblatt (Blatt 6) für verschiedene Schrittweiten untersuchen.

- Schreiben Sie ein Matlabskript `konvergenzESV`, das für das Anfangswertproblem aus Aufgabe 26 und für die Schrittweiten $0.5^4, \dots, 0.5^{14}$ den Diskretisierungsfehler des expliziten Euler-Verfahrens `eulerExplizit` berechnet und doppelt logarithmisch über den Schrittweiten plottet (Matlabbefehl `loglog`).
- Ergänzen Sie Ihr Matlabskript `konvergenzESV` und Ihr Schaubild um die Diskretisierungsfehler der anderen Verfahren aus Aufgabe 26 (`eulerImplizit`, `trapezMethode` und `heun`).
- Interpretieren Sie das Schaubild.

Aufgabe 29 („Bessere“ *Einschrittverfahren mit der Simpsonregel?!*) (3T+3M+2M+(2M+1T)+
+(2M+1T)+2T+1T+2T+2M+(3M+1T)+2T+(3M*+1T*)+(3M*+1T*)+4T*+5M*+3T* Punkte)

Der allgemeine Ansatz für Einschrittverfahren ist, in der Fredholmschen Integralgleichung

$$y(t_{j+1}) = y(t_j) + \int_{t_j}^{t_{j+1}} f(s, y(s)) ds$$

(Gleichung (3.11) im Skript) das Integral $\int_{t_j}^{t_{j+1}} f(s, y(s)) ds$ durch eine Quadraturformel

$$\hat{I} \approx \int_{t_j}^{t_{j+1}} f(s, y(s)) ds$$

zu ersetzen. In Aufgabe 26 haben wir das mit verschiedenen Quadraturformeln getan: Im expliziten Eulerverfahren haben wir die linksseitige Rechteckregel verwendet und im impliziten Eulerverfahren die rechtsseitige Rechteckregel. Mit der Trapezregel ($h_j := (t_{j+1} - t_j)$)

$$\hat{I} = h_j \frac{f_l + f_r}{2}$$

haben wir ein implizites Verfahren programmiert. Dabei steht f_l für den Wert von f am linken Rand des Intervalls $[t_j, t_{j+1}]$, also $f_l = f(t_j, y(t_j))$ und f_r für den Wert am rechten Rand also $f_r = f(t_{j+1}, y(t_{j+1}))$. Und um mit der Trapezregel ein explizites Verfahren zu erhalten, haben wir im Verfahren von Heun zweiter Ordnung den Wert f_r angenähert durch einen Schritt eines expliziten Eulerverfahrens:

$$f_r = f(t_{j+1}, y(t_{j+1})) \approx f(t_{j+1}, y(t_j) + h_j y'(t_j)) = f(t_{j+1}, y(t_j) + h_j f(t_j, y(t_j))).$$

Diese Idee wollen wir in der vorliegenden Aufgabe weiter entwickeln. Dabei verwenden wir statt der Trapezregel eine Quadraturformel mit einer höheren Ordnung, nämlich die Simpsonregel:

$$\hat{I} = h_j \frac{f_l + 4f_m + f_r}{6}.$$

f_m steht dabei für den Wert in der Mitte des Intervalls, also $f_m = f\left(t_j + \frac{h_j}{2}, y\left(t_j + \frac{h_j}{2}\right)\right)$. Um ein explizites Verfahren zu erhalten, müssen wir wie beim Verfahren von Heun zweiter Ordnung $y(t_j + \frac{h_j}{2})$ (in f_m) und $y(t_{j+1})$ (in f_r) ausgehend von $y(t_j)$ annähern. Zur Approximation von $y(t_j + \frac{h_j}{2})$ führen wir ausgehend von $y(t_j)$ einen Schritt eines expliziten Eulerverfahrens mit Schrittweite $\frac{h_j}{2}$ durch, also

$$y\left(t_j + \frac{h_j}{2}\right) \approx y(t_j) + \frac{h_j}{2} y'(t_j) = y(t_j) + \frac{h_j}{2} f(t_j, y(t_j)).$$

Damit erhalten wir

$$f_m = f\left(t_j + \frac{h_j}{2}, y\left(t_j + \frac{h_j}{2}\right)\right) \approx f\left(t_j + \frac{h_j}{2}, y(t_j) + \frac{h_j}{2} f(t_j, y(t_j))\right) =: \hat{f}_m.$$

\hat{f}_m ist also eine Näherung für f_m .

Für die Approximation von $y(t_{j+1})$ (in f_r) gibt es mehrere Möglichkeiten. Man kann $y(t_{j+1})$ in f_r wiederum entsprechend der Fredholmschen Integralgleichung durch Anwendung verschiedener Quadraturformeln annähern.

- Approximieren Sie $y(t_{j+1})$ (in f_r) durch einen Schritt eines expliziten Eulerverfahrens mit Schrittweite h_j , ausgehend von $y(t_j)$. Wie lautet dann die Approximation für $y(t_{j+1})$? Und wie die Approximation für f_r ? Welche Quadraturformel haben Sie hier (lokal zur Approximation von f_r) angewendet?
- Schreiben Sie analog zu Aufgabe 26 eine Matlabfunktion `yk = simpson(f, y0, tk)`, die für einen gegebenen Startwert $y^0 \in \mathbb{R}^n$ eine Lösung $y: \mathbb{R} \rightarrow \mathbb{R}^n$, $n \in \mathbb{N}$ der Anfangswertaufgabe

$$y' = f(t, y), \quad y(t_0) = y^0$$

mit der Simpsonregel berechnet. Die Parameter sind dabei wie in Aufgabe 26: **f** ist die Funktion f als *function handle*, **y0** ist der Startwert $y^0 \in \mathbb{R}^n$ und **tk** ist ein Gitter mit den diskreten Zeitpunkten t_k . Der Rückgabewert **yk** ist der Vektor der einzelnen Näherungswerte y^k . Approximieren Sie $y(t_{j+1})$ (in f_r) durch einen Schritt eines expliziten Eulerverfahrens mit Schrittweite h_j .

- Erweitern Sie Ihr Matlabskript `vergleichESV` aus Aufgabe 26. Binden Sie Ihre Matlabfunktion `yk = simpson(f, y0, tk)` analog zu Aufgabe 26 ein. Bestimmen Sie also zwei Näherungslösungen der obigen Anfangswertaufgabe mit den Schrittweiten $h = \frac{1}{20}$ und $h = \frac{1}{40}$. Zeichnen Sie diese Näherungslösungen und die exakte Lösung in ein neues Schaubild ein. Ergänzen Sie Ihr Schaubild mit den Gitterfehlern um die Gitterfehler der beiden neuen Lösungen.
- Erweitern Sie Ihr Matlabskript `vergleichESV`: Geben Sie zu den bisher verwendeten fünf Verfahren und den beiden Schrittweiten jeweils den Diskretisierungsfehler aus. Vergleichen Sie die Verfahren.
- Erweitern Sie auch Ihr Matlabskript `konvergenzESV` und Ihr Schaubild aus Aufgabe 28 um die Diskretisierungsfehler der Simpsonregel aus Aufgabenteil a) und b). Vergleichen Sie die Simpsonregel mit den anderen Verfahren.
- Zur Approximation von $y(t_{j+1})$ in f_r kann auch der bereits approximierter Wert von $y\left(t_j + \frac{h_j}{2}\right)$ herangezogen werden. So kann beginnend bei $y\left(t_j + \frac{h_j}{2}\right)$ ein Schritt eines expliziten Euler-Verfahrens mit Schrittweite $\frac{h_j}{2}$ durchgeführt werden. Wie lautet jetzt die Approximation für $y(t_{j+1})$, wie die für f_r ?
- Die Approximation für $y(t_{j+1})$ lässt sich also schreiben als

$$y(t_{j+1}) \approx y(t_j) + h_j(\beta_{31} f(t_j, y(t_j)) + \beta_{32} \hat{f}_m) = y(t_j) + h_j(\beta_{31} f_l + \beta_{32} \hat{f}_m),$$

wobei β_{31} und β_{32} Parameter sind, deren Bezeichnung in der nächsten Vorlesung deutlich werden wird. Wie lautet dann die Approximation für f_r ?

- h) Wie lauten die Parameter β_{31} und β_{32} , wenn $y(t_{j+1})$ gemäß Aufgabenteil f) approximiert wird? Wie lauten die Parameter, wenn $y(t_{j+1})$ nach Aufgabenteil a) approximiert wird?
- i) Erweitern Sie Ihre Matlabfunktion `yk = simpson(f, y0, tk)` zu einer Matlabfunktion `yk = simpson3(f, y0, tk, beta31, beta32)`, bei der die verwendete Approximation von f_r durch die Parameter `beta31` und `beta32` gesteuert werden kann.
- j) Ergänzen Sie Ihre Matlabskripte `vergleichESV` und `konvergenzESV` um obige Variante der Simpsonregel, bei der $y(t_{j+1})$ gemäß Aufgabenteil f) approximiert wird. Verwenden Sie Ihre Matlabfunktion `yk = simpson3(f, y0, tk, beta31, beta32)` aus dem vorigen Aufgabenteil mit den entsprechenden Parametern. Interpretieren Sie die Ergebnisse.
- k) Wie lauten die Parameter β_{31} und β_{32} , wenn Sie $y(t_{j+1})$ in f_r mit der Mittelpunkregel approximieren?
- l) Ergänzen Sie Ihre Matlabskripte `vergleichESV` und `konvergenzESV` auch um die Variante der Simpsonregel, bei der $y(t_{j+1})$ gemäß dem vorigen Aufgabenteil k) mit der Mittelpunkregel approximiert wird. Verwenden Sie Ihre Matlabfunktion `yk = simpson3(f, y0, tk, beta31, beta32)` mit den entsprechenden Parametern. Interpretieren Sie die Ergebnisse.
- m) Verwenden Sie nun die Parameterwerte $\beta_{31} = -1$ und $\beta_{32} = 2$ und erweitern Sie Ihre Matlabskripte `vergleichESV` und `konvergenzESV` auch um diese Variante der Simpsonregel. Was stellen Sie nun fest?
- n) Ermuntert durch den Erfolg im vorigen Aufgabenteil möchten Sie Ihren Algorithmus weiter verfeinern. Sie überlegen sich, dass der mit dem Faktor 4 gewichtete Term f_m in der Simpsonregel möglicherweise noch Verbesserungspotential birgt und entschliessen sich, Ihre Simpsonregel um eine Stufe zu erweitern:

$$\hat{I} = h_j \frac{\hat{f}_l + 2\hat{f}_{m1} + 2\hat{f}_{m2} + \hat{f}_r}{6}.$$

Jetzt können Sie f_m auf zwei verschiedene Arten approximieren. Nach langwierigen Versuchen finden Sie die folgenden Näherungen für f_l , f_{m1} , f_{m2} und f_r :

$$\begin{aligned}\hat{f}_l &= f(t_j, y(t_j)), \\ \hat{f}_{m1} &= f\left(t_j + \frac{h_j}{2}, y(t_j) + \frac{h_j}{2} f(t_j, y(t_j))\right), \\ &= f\left(t_j + \frac{h_j}{2}, y(t_j) + \frac{h_j}{2} \hat{f}_l\right), \\ \hat{f}_{m2} &= f\left(t_j + \frac{h_j}{2}, y(t_j) + \frac{h_j}{2} \hat{f}_{m1}\right), \\ \hat{f}_r &= f\left(t_j + h_j, y(t_j) + h_j \hat{f}_{m2}\right).\end{aligned}$$

Machen Sie sich klar, was diese Approximationen anschaulich bedeuten.

- o) Schreiben Sie eine Matlabfunktion `yk = simpson4(f, y0, tk)`, mit der Sie den Algorithmus aus dem vorigen Aufgabenteil realisieren. Binden Sie Ihre Matlabfunktion `yk = simpson4(f, y0, tk)` ebenfalls in Ihre Matlabskripte `vergleichESV` und `konvergenzESV` ein.
- p) Was stellen Sie fest? Welche Werte haben Sie für den Diskretisierungsfehler erhalten? Was können Sie an Ihrem Schaubild aus dem Matlabskript `konvergenzESV` erkennen? War Ihre Euphorie berechtigt?

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Matlab-Files in einer E-mail mit dem Betreff **Loesung-Blatt07** an angewandte.numerik@uni-ulm.de (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.