Übung 1 Rundungsfehler

Bei der Durchführung einer expliziten (L-stetigen) Einschrittmethode mit Lösung $u(t)$ für $t \geq t_0$ und numerischer Approximation durch eine Gitterfunktion $(\tilde{y}_n)_{n\geq 0}$, wird wegen des unvermeidbaren Rundungsfehlers eine gestörte Rekursion

$$\tilde{y}_n = \tilde{y}_{n-1} + h_n F(h_n; t_{n-1}, \tilde{y}_{n-1}) + \epsilon_n, \quad n \geq 1,$$

gelöst. Die "lokalen" Fehler verhalten sich dabei wie $\|\epsilon_n\| \propto \text{eps}\|y_n\|$, wobei eps den maximalen relativen Rundungsfehler bezeichnet. O.B.d.A. sei $\tilde{y}_o = u(t_0)$ angenommen. Man beweise die Abschätzung

$$\|\tilde{y}_n - u(t_n)\| \leq K(t_n) \left(\max_{1\leq m\leq n} \|\tau_m\| + \text{eps} \max_{1\leq m\leq n} h_m^{-1}\|y_m\| \right).$$

4 Punkte

Übung 2 Runge-Kutta Verfahren dritter Ordnung

Man zeige, dass das Einschrittverfahren mit

$$F(h; t, x) = \sum_{r=1}^{3} b_r k_r(h; t, x)$$

und

$$\begin{array}{lll} k_1 := f(t, x), & k_2 := f(t + \frac{h}{2}, x + \frac{h}{2}k_1), & k_3 := f(t + h, x + h(-k_1 + 2k_2)) \\ b_1 = \frac{1}{6}, & b_2 = \frac{2}{3}, & b_3 = \frac{1}{6} \end{array}$$

die Konsistenzordnung drei besitzt. *3 Punkte*

Übung 3 Konvergenz-Analyse (Programmieraufgabe - Abgabe: 21.05.2010)

In dieser Aufgabe sollen die Konvergenzraten für zwei verschiedene Einschrittverfahren anhand der AWA

$$\begin{aligned} u'(t) &= -\lambda\, t\, u(t)^2, & (-\tau \leq t \leq \tau), \quad \tau, \lambda \in \mathbb{R}^+, \\ u(-\tau) &= u_{-\tau}, & u_{-\tau} \in \mathbb{R} \end{aligned}$$

verifiziert werden.

1. Implementieren Sie die gegebene AWA analog zu der Implementierung des exponentiellen Wachstumsgesetzes in der Datei *modelproblem.hh* (alle hier angegebenen Dateien sind im examples Verzeichnis der *hdnum* Bibliothek zu finden, welche auf der VL Homepage zur Verfügung gestellt wird).

2. Verifizieren Sie Ihre Implementierung. Schreiben Sie hierzu ein Programm, welches für die Parameterwahl

$$\lambda = 200, \quad \tau = 3, \quad u_0 = \frac{1}{901}$$

Approximationen der analytischen Lösung

$$u(t) = \frac{1}{1 + 100t^2}$$

unter Verwendung des expliziten Euler Verfahrens berechnet (siehe hierzu auch Datei *modelproblem.cc*). Eine Implementierung des Verfahrens finden Sie in der Datei *expliciteuler.hh*. Verifizieren Sie die Konvergenz 1. Ordnung dieses Verfahrens.

3. Implementieren Sie explizite Runge-Kutta Verfahren zweiter, dritter und vierter Ordnung analog zu der Implementierung in *expliciteuler.hh*. Verifizieren Sie die Konvergenz dieser Verfahren.

   Bonus: Die Implementierung eines allgemeinen Lösers für Runge-Kutta Verfahren, welcher beliebige (explizite) Butcher-Tableaus realisiert, ist mit nur geringem Mehraufwand möglich).

4. Die Aussage des Satzes von Aufgabe 1 impliziert im Falle konstanter Schrittweiten eine untere Schranke für die Schrittweite bei deren Unterschreiten der Approximationsfehler aufgrund des wachsenden Rundungsfehlers wieder zunimmt. Ersetzen Sie in ihren Programmen `double` durch `float` und bestimmen Sie diese Schranke näherungsweise.

6+2 Punkte

ÜBUNG 4 SCHRITTWEITENKONTROLLE (PROGRAMMIERAUFGABE - ABGABE: 21.05.2010)

In dieser Aufgabe sollen die Vorteile einer Schrittweitensteuerung untersucht werden. Sei $u(t) : \mathbb{R} \to \mathbb{R}$ die Lösung einer AWA und y^h_{n-1} eine numerische Approximation von $u(t_{n-1})$ zum Zeitpunkt t^{n-1}. Nun wird die nächste Approximation y^h_n durch ein Verfahren der Ordnung m und $\hat{y}^h_n$ durch ein Verfahren der Ordnung $m+1$ berechnet. Die resultierende Differenz

$$e_n = \|y^h_n - \hat{y}^h_n\|$$

wird als Maß für den lokalen Fehler interpretiert und soll der Beziehung $e_n < TOL$ genügen. Hierzu wird die neue Schrittweite stets gemäß

$$h_{n(+1)} := h_n \left(\frac{TOL}{\|\hat{y}^h_n - y^h_n\|} \right)^{1/(m+1)}$$

bestimmt, bzw. wenn notwendig solange rekursiv berechnet, bis $e_n < TOL$ erfüllt ist. Eine heuristische Wahl für TOL ist durch

$$TOL = \epsilon \, \frac{|y^h_n|}{h_n}$$

gegeben, wobei zunächst $\epsilon \approx$ eps gewählt werden kann.

1. Schreiben Sie ein Programm, welches die AWA aus Aufgabe 3.2 mit der oben beschriebenen Schrittweitensteuerung löst. Spielen Sie ein bisschen mit dem Wert für ϵ und beschreiben sie das resultierende Verhalten der Lösung qualitativ.

2. Vergleichen Sie das Verfahren mit Rechnungen konstanter Schrittweite. Berechnen Sie hierzu $\max_n |y^h_n - u(t_n)|$ für $10^4, 10^5$ und 10^6 konstante Schritte und vergleichen Sie die Ergebnisse mit den vorherigen Rechnungen variabler Schrittweite.

3. *Bonus*: Ein anderer Ansatz zur Schrittweitensteuerung sieht so aus: Für ein gegebenes h_n seien y^{2h}_n durch einen Schritt der Länge $2h_n$ und y^h_n durch zwei Schritte der Länge h_n (eines Einschrittverfahrens) aus y^h_{n-1} berechnet worden. Dabei soll der Zeitschritt h_n nun der Beziehung

   $$\frac{1}{2} h_n \leq h_{opt}$$

   genügen, wobei h_{opt} gemäß

   $$h_{opt} := \left(\frac{(2h_n)^{m+1}(1 - 2^{-m})}{|y^{2h}_n - y^h_n|} \, TOL \right)^{1/m}$$

   berechnet wird und m durch die Konsistenzordnung des Einschrittverfahrens gegeben ist. Abhängig davon, ob dies erfüllt wurde oder nicht, wird entweder die rekursive Beziehung $h_n := h_{opt}(h_n)$ iteriert (also der Zeitschritt wiederholt) oder $h_{n+1} := 2h_n$ gesetzt und mit dem nächsten Schritt fortgefahren.

   Implementieren Sie ein solches Verfahren und vergleichen Sie die beiden Methoden zur Schrittweitensteuerung unter Verwendung der expliziten Euler Methode und wenigstens einem anderen Verfahren höherer Ordnung.

6+2 Punkte