D-ITET, D-MATL — Numerische Methoden — FS 2018

Dr. R. Käppeli
P. Bansal

Lösung 10

1. *Zu Einfaches adaptives Heun-Verfahren*

a) Die Implementierung finden Sie im kommentierten `adaptHeunSimple.m`.

b) Zunächst müssen wir die Van der Pol-Gleichung umschreiben in ein System erster Ordnung:

$$\begin{aligned}\dot{y}_0(t) &= y_1(t)\\ \dot{y}_1(t) &= 8(1 - y_0(t)^2)y_1(t) - y_0(t).\end{aligned}$$

Die Anfangswerte sind dann

$$y_0(0) = 2 \quad , \quad y_1(0) = 0.$$

In Abb. 1 werden die erhaltene Näherungslösung $y(t)$ (links) und die Schrittweite h (rechts) gezeigt (erstellt mit `vanDerPol.m`). Wir beobachten, dass wenn

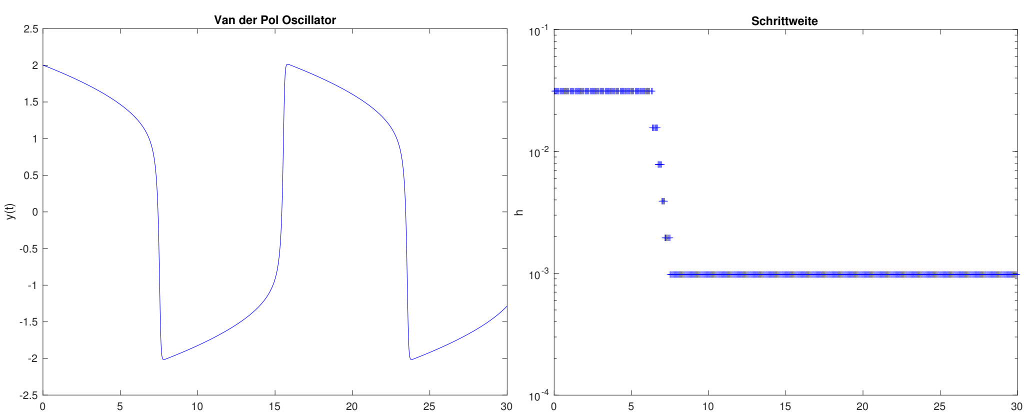

Abbildung 1: Lösung $y(t)$ links und Schrittweitte h rechts.

die Lösung anfängt stärker zu variieren (bei Zeit ~ 7) reduziert der Algorithmus die Schrittweite sukzessiv. Anschliessend bleibt die Schrittweite konstant.

c) Der in der Aufgabe beschriebene weist (mindestens) zwei offensichtliche Schwächen auf:

1. Wenn die Schrittweite einmal verkleinert wurde, z.B. wenn die Lösung stark variiert, wird sie nicht mehr erhöht, z.B. wenn die Lösung weniger variiert.
2. Es könnte passieren, dass der Algorithmus die Schrittweite halbiert ohne jemals das Toleranz-Kriterium zu erreichen, z.B. wenn die Toleranzen sehr klein gewählt sind.

2. *Zu Einfaches adaptives Runge-Kutta-Fehlberg Verfahren*

a) Siehe `RKF45Simple.m`.

b) Siehe `KonvTestRK.m`. Wir sehen, dass die experimentale Ordnungen für die RK4 und RK5 Verfahren 3.94 und 4.97 sind. Deshalb produzieren beide Verfahren erwartete Konvergenzresultate.

c) Siehe `vanDerPol.m`. In Abb. 2 werden die erhaltene Näherungslösung $y(t)$ (links) und die Schrittweite h (rechts) gezeigt (erstellt mit `vanDerPol.m`). Die

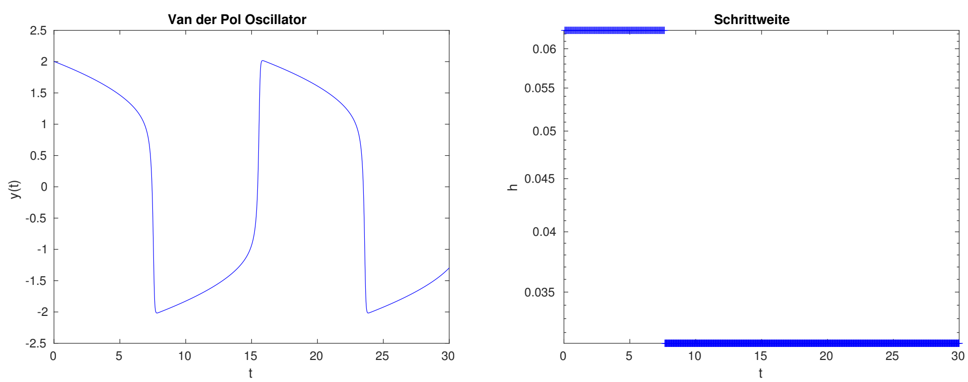

Abbildung 2: Lösung $y(t)$ links und Schrittweite h rechts.

Bemerkungen hier sind die gleiche als bei der Aufgabe **1.b)**.

3. *Adaptive Schrittweitensteuerung*

a) Siehe `RKF45.m` und `adaptHeun.m`.

b) In Abb. 3 werden die erhaltene Schrittweite h für die Heun (links) und RKF45 (rechts) Verfahren gezeigt (erstellt mit `vanDerPol.m`). Wir beobachten, dass wenn die Lösung anfängt stärker zu variieren (bei Zeit $\sim$ 7) reduziert der Algorithmus die Schrittweite sukzessiv. Der Unterschied zwischen diesem Algorithmus und der einfacheren Versionen aus letzter Serie besteht darin, dass wenn die Lösung wieder weniger stark variiert wird die Schrittweite wieder vergrössert. Die erhaltene Approximationen sind in diesen Falle nicht schlechter, d.h.

Siehe nächstes Blatt!

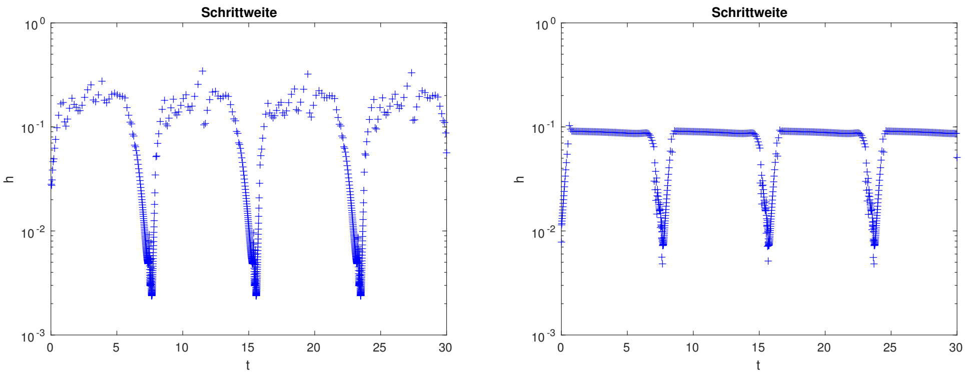

Abbildung 3: Schrittweitte h für die Heun (links) und RKF45 (rechts) Verfahren.

sie erfüllen unser lokales Toleranz-Kriterium, aber wir benutzen viel weniger Funktionsauswertungen. Deshalb können wir ähnliche Ergebnisse mit weninger Aufwand erhalten, d.h. wir sind effizienter!