D-ITET, D-MATL
Dr. R. Käppeli
B. Fitzpatrick, D. Devaud

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:

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

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 Schrittweitte h (rechts) gezeigt (erstellt mit `vanDerPol.m`). Wir beobachten, dass wenn

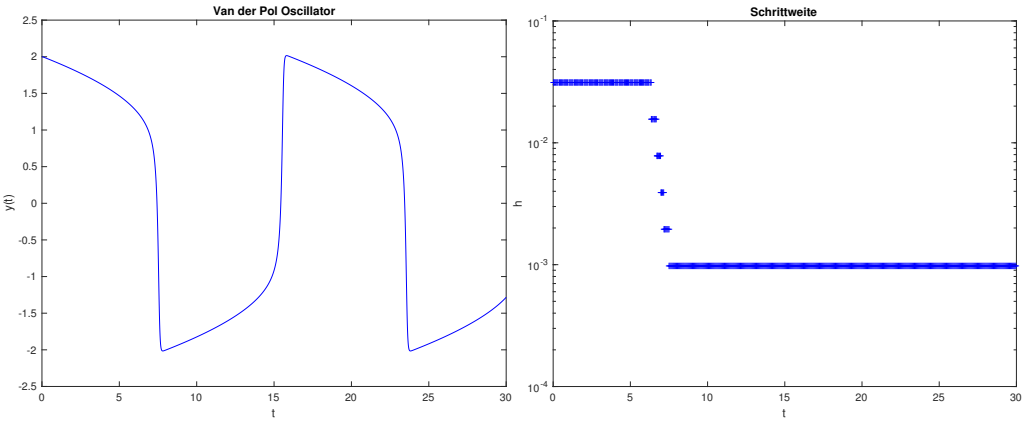

Abbildung 1 – Lösung $y(t)$ links und Schrittweitte h rechts.

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:

Bitte wenden!

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 `RKF45.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

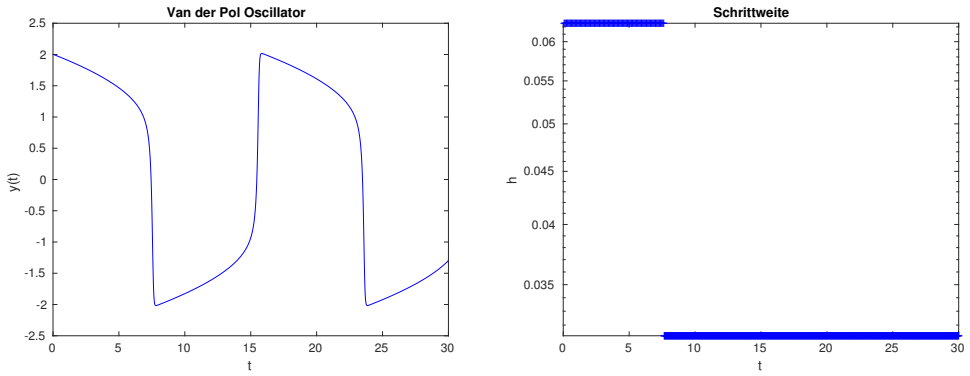

Abbildung 2 – Lösung $y(t)$ links und Schrittweite h rechts.

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

3. *Explizites und Implizites Euler-Verfahren*

a) Für das Problem

$$\dot{y}(t) = -\lambda y(t), \qquad y(t_0) = y_0,$$

ist einen Schritt des expliziten Euler-Verfahrens gegeben durch

$$y_1 = y_0 + hf(t_0, y_0) = y_0 + h(-\lambda y_0) = (1 - h\lambda)y_0.$$

Für das implizite Euler-Verfahren erhalten wir

$$y_1 = y_0 + hf(t_1, y_1) = y_0 + h(-\lambda y_1) = y_0 - h\lambda y_1 \Rightarrow y_1 = \frac{y_0}{1 + h\lambda}.$$

Siehe nächstes Blatt!

Wir bemerken, dass für das implizite Euler-Verfahren eine (lineare) Gleichung gelöst werden muss.

b) Für das Problem

$$\dot{y}(t) = -t(y(t))^2, \qquad y(t_0) = y_0 > 0,$$

ist einen Schritt des expliziten Euler-Verfahrens gegeben durch

$$y_1 = y_0 + hf(t_0, y_0) = y_0 + h(-t_0 y_0^2) = y_0(1 - ht_0 y_0).$$

Für das implizite Euler-Verfahren erhalten wir

$$\begin{aligned} y_1 &= y_0 + hf(t_1, y_1) = y_0 + h(-t_1 y_1^2) = y_0 - ht_1 y_1^2 \\ &\Leftrightarrow ht_1 y_1^2 + y_1 - y_0 = 0 \\ &\Leftrightarrow y_1 = \frac{-1 \pm \sqrt{1 + 4ht_1 y_0}}{2ht_1}. \end{aligned}$$

Wir bemerken, dass für das implizite Eulerverfahren ist die Lösung nicht eindeutig.

c) Siehe `expEulerlinear.m`, `implEulerlinear.m` und `KonvTestEuler.m`. Wir bemerken, dass für grosse Schrittweiten das explizite Euler-Verfahren "instabil" scheint (die Lösung oszilliert und wächst). Hingegen liefert das implizite Verfahren für jede Schrittweite eine stets abnehmende Lösung (was qualitativ mit der exakten Lösung dieses AWP übereinstimmt).