

## Ü 8 Achte Übungseinheit

Inhalt der achten Übungseinheit:

- Optionen in MATLABs `ode45` setzen
- Stabilität, implizite Verfahren
- Systeme von Differentialgleichungen 1. Ordnung

### Ü 8.1 Gewöhnliche Differentialgleichungen erster Ordnung (Fortsetzung von Ü 7.2)

#### Aufgabe 69: Setzen der Fehlerschranken

MATLABs `ode45`-Befehl löst mit den Standard-Einstellungen viele unkomplizierte Aufgaben zufriedenstellend. Manchmal reichen aber die voreingestellten Optionen nicht aus. Sie können umfangreiche Anpassungen durchführen. Diese Aufgabe zeigt Ihnen, wie Sie die Rechengenauigkeit beeinflussen können.

Gegeben sei für  $y = y(x)$  das Anfangswertproblem

$$y' = f(x, y)$$

mit

$$f(x, y) = -200xy^2, \quad y(-3) = \frac{1}{901}.$$

Berechnen Sie unter Verwendung des MATLAB-Solvers `ode45` die Lösung für  $x = -2, -1, 0$ .

Fehlerschranken: Verwenden Sie die Befehle

```
options=odeset('reltol',1.e-8,'abstol',1.e-8);  
sol=ode45(@g,xspan,y0,options)
```

um die vorgegebenen Fehlerschranken zu ändern. Versuchen Sie auch Fehlerschranken  $10^{-10}, 10^{-12}, 10^{-14}$ . Berechnen Sie jeweils den Wert  $y(0)$ .

Für dieses Problem lässt sich die exakte Lösung leicht angeben:

$$y_{\text{exakt}}(x) = \frac{1}{100x^2 + 1}$$

Vergleichen Sie für  $x = 0$  die numerisch berechneten Werte  $y(0)$  mit der exakten Lösung  $y_{\text{exakt}}(0) = 1$ . Berechnen Sie die Fehler (relativer und absoluter Fehler sind hier gleich!). Werden die gewünschten Fehlergrenzen erreicht?

#### Ü 8.1.1 Stabilität eines Verfahrens; implizite Verfahren

Rechenverfahren sollten den Verlauf der Lösung, wenn schon nicht genau, so doch wenigstens *qualitativ* richtig bestimmen.

Konkret: wenn die exakte Lösung eine exponentiell abklingende Funktion ist, dann soll auch die numerische Lösung abklingen. Leider berechnen bei zu großer Schrittweite die bisher vorgestellten *expliziten* Verfahren exponentiell *anwachsende* Lösungen – ziemlich genau das Gegenteil von dem, was passieren sollte.

Ein Verfahren heißt *stabil* bei Schrittweite  $h$ , wenn für das Modellproblem

$$y' = -y \quad y(0) = 1$$

die numerische Lösung mit wachsendem  $x$  nach Null konvergiert.

Sie lernen nun *implizite* Verfahren kennen; diese machen bei jedem einzelnen Rechenschritt ähnlich große Fehler wie explizite Verfahren. Aber bei expliziten Verfahren schaukeln sich Fehler von einem Schritt zum nächsten exponentiell auf; bei impliziten Verfahren klingen die Fehler in Summe ab und zerstören nicht den qualitativen Verlauf der Lösung.

### Aufgabe 70: Stabilität expliziter Einschrittverfahren

Modifizieren Sie das Musterprogramm `GDGdemo.m` und testen Sie die bisher behandelten Einschritt-Verfahren (Euler explizit, Euler modifiziert, Heun sowie das dreistufigen Verfahren aus Aufgabe 68) für verschiedene Schrittweiten  $h$  beim Anfangswertproblem für  $y(x)$

$$y' = -y \\ y(0) = 1$$

Rechnen Sie bis  $x = 10$ , und finden Sie für jedes Verfahren  $h$ -Werte, für die das Verfahren gerade noch abklingende, beziehungsweise bereits leicht anschwellende Lösungen berechnet – das sind also  $h$ -Werte gerade noch innerhalb, beziehungsweise knapp außerhalb der Stabilitätsgrenze.

Vergleichen Sie dazu die Abbildungen auf der vorletzten Vorlesungsfolie, 9. Vorlesung!

### Aufgabe 71: Implizites Eulerverfahren

Hier ist die Verfahrensfunktion gegeben durch

$$F(x, y, h) = f(x + h, y(x + h))$$

und der neue Näherungswert daher

$$y(x + h) = y(x) + hf(x + h, y(x + h))$$

Anders als beim expliziten Verfahren lässt sich der gesuchte Wert nicht direkt finden: Sie können  $y(x + h)$  auf der linken Seite nur berechnen, wenn Sie es schon kennen, weil Sie es auf der rechten Seite der Gleichung einsetzen müssen. Da beißt sich die Katze in den Schwanz. . .

Mathematisch gesprochen handelt es sich um eine *implizite Gleichung* für  $y(x + h)$ . Sie lässt sich bei unserem Testproblem mit der einfachen Funktion  $f(x, y) = -y$  leicht auflösen.

Orientieren sie sich am Musterprogramm `GDGdemo.m` und implementieren Sie das implizite Euler-Verfahren für das Testproblem

$$y' = -y \\ y(0) = 1$$

Untersuchen Sie die Stabilität dieses Verfahrens. Gibt es hier Schrittweiten  $h$ , für die die numerische Lösung anschwillt?

## Ü 8.2 Systeme von Differentialgleichungen erster Ordnung

### Ü 8.2.1 Beispiel: SIR-Modell zur Ausbreitung einer Epidemie

#### Hintergrundinformation

Das SIR-Modell (*susceptible-infected-removed model*) beschreibt in sehr vereinfachter Form die Ausbreitung von ansteckenden Krankheiten mit Immunitätsbildung.

Das Modell teilt eine Gesamtpopulation in drei Gruppen, S, I und R. Die erste Gruppe, S wie *susceptible* (empfindlich), enthält die anfälligen Individuen: sie sind nicht infiziert, können sich aber jederzeit anstecken. Gruppe I sind die Infizierten. In der dritten Gruppe befinden sich alle, die sich selbst und niemand anderen mehr infizieren können. Schön wär's, wenn wir R wie *recovered* sagen könnten, R steht aber für *removed*, weil das Modell Genesene und Verstorbene in dieser Gruppe zusammenfasst.

Die Urheber dieses Modells, der Biochemiker William Ogilvy Kermack und der Militärarzt Anderson Gray McKendrick, konnten trotz der Einfachheit des Modells gut die Daten einer Pestepidemie in Bombay 1905/06 modellieren. Nur etwas über 100 Jahre später, 2014, publizierten Harko und Koautoren eine exakte analytische Lösung dieses Differentialgleichungssystems. In aller Regel werden diese Gleichungen aber numerisch gelöst, und das werden wir hier tun.

In verfeinerter und weiterentwickelter Form haben solche Modellrechnungen in den letzten Jahren unser Alltagsleben mehr als uns lieb sein konnte beeinflusst. Sie können leicht und reichlich vertiefende Informationen finden. Im Moment reicht uns, dass das SIR-Modell als System von drei Differentialgleichungen 1. Ordnung formuliert ist:

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta IS}{N} \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

Die Funktionen  $S(t)$ ,  $I(t)$  und  $R(t)$  beschreiben die Größe der jeweiligen Gruppen in Abhängigkeit von der Zeit  $t$ .

Die Gesamtgröße der Population  $N = S(t) + I(t) + R(t)$  bleibt darin konstant. Die allgemeine demografische Entwicklung, also Geburten und nicht durch die Epidemie bedingte Todesfälle, wird nicht modelliert. Das ließe sich aber mit wenigen Zusatz-Termen berücksichtigen.

Der Parameter  $\beta$  gibt an, wie viele enge Kontakte ein Individuum pro Tag durchschnittlich hat (wobei nur ein Kontakt zwischen einem S- und einem I-Individuum zu einer Ansteckung führt; Kontakte vom Typ S-S, S-R, I-I und I-R bleiben folgenlos).

Der Parameter  $\gamma$  lässt sich einfacher in Form seines Reziprokwerts interpretieren als

$$\frac{1}{\gamma} \quad \dots \text{Typische Zeitdauer der Infektion.}$$

Maßgeblich für die Dynamik der Epidemie ist die Basisreproduktionszahl

$$R_0 = \frac{\beta}{\gamma} .$$

Dieser Wert gibt an, wie viele weitere Personen eine infizierte Person in der Anfangsphase der Epidemie, (wenn in der allgemeinen Population noch kaum jemand immun ist), im Durchschnitt ansteckt<sup>36</sup>

## Numerische Lösung in MATLAB

Es gibt dazu ein fertiges Musterprogramm, `SIRmodel.m`, aber die wesentlichen Programmcode-Zeilen sind so kurz, dass wir sie hier dem Differentialgleichungssystem gegenüberstellen:

Gleichungssystem

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta IS}{N} \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I \end{aligned}$$

Implementierung

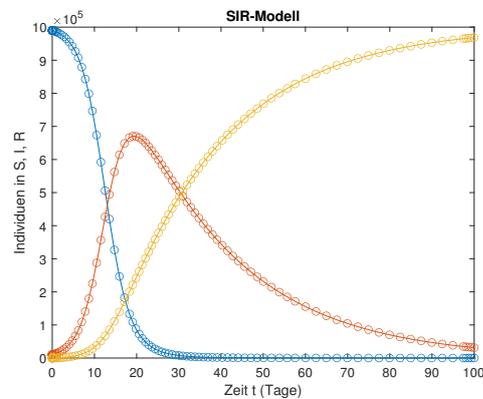
```
f = @(t,y) [
    - beta*y(2)*y(1)/N;
    beta*y(2)*y(1)/N - gamma*y(2);
    gamma*y(2) ];
```

$$0 \leq t \leq 100$$

$$S(0) = 999, \quad I(0) = 1, \quad R(0) = 0.$$

`ode45(f, [0 100], [999, 1, 0])`

Der Aufruf von `ode45` in seiner einfachsten Form, mit Parametern  $\beta = 0,4$ ;  $\gamma = 0,04$ ;  $N = 10\,000\,000$  liefert die nebenstehende Abbildung.



## Aufgabe 72: Aufgabe zum SIR-Modell

Wenden Sie ein SIR-Modell mit plausiblen Parametern auf die Situation in Österreich im Sommer und Herbst 2021 an. Setzen Sie die Start-Zeit  $t = 0$  mit 17. Mai fest; damals war  $I(0) \approx 10\,000$ ,  $S(0) \approx 6\,400\,000$  (8 Millionen, minus 1 Million Geimpfte und 600 000 Genesene),  $R(0) = N - S(0) - I(0)$ . Die Wahl  $\gamma = 1/14$  entspricht einer durchschnittlicher Dauer der infektiösen Phase von 14 Tagen. Wählen Sie für  $\beta$  einige verschiedene Werte, so dass die

<sup>36</sup>Die aus den Medien bekannte *effektive Reproduktionszahl*  $R_{\text{eff}}$  liegt unter  $R_0$ , im einfachsten Fall ist  $R_{\text{eff}} = R_0 S/N$ . Sie berücksichtigt bereits bestehende teilweise Immunität und gesenkte Übertragungswahrscheinlichkeit (durch Lockdown, Masken,...)

effektive Reproduktionszahl  $R_{\text{eff}} = \beta/\gamma \cdot S/N$  zwischen  $\approx 0,8$  (das war der Wert im Mai 2021) und 1,4 liegt (diesen Wert erreichte  $R_{\text{eff}}$  im Jänner 2022).

Zeichnen Sie für verschiedene  $R_{\text{eff}}$ -Szenarien den Verlauf der  $I$ - und  $R$ -Größen für die nächsten paar Monate (ohne  $S$ , weil es sich sonst mit dem Maßstab in der Abbildung nicht ausgeht: die  $S$ -Gruppe ist so groß, dass die  $I$ - und  $R$ -Anteile dann kaum sichtbar sind.)

Sie können dieses einfache SIR-Modell mit dem tatsächlichen Verlauf vergleichen: Im Mai und Juni 2021 blieb  $R_{\text{eff}} < 1$ , dem entsprechend sank  $I$  in diesem Zeitraum beständig. Von Juli bis November war  $R_{\text{eff}} \approx 1,2$ . In diesem Zeitraum baute sich eine neue Welle auf, die Mitte November mit  $I \approx 150\,000$  ihren Höhepunkt erreichte und nur durch einen Lockdown gestoppt werden konnte.

### Ü 8.2.2 Selbst programmierte Einschrittverfahren

Die Bewegungsgleichungen für Weg und Geschwindigkeit und ein einfaches Kraftgesetz lauten:

Zeitableitung des Weges = Geschwindigkeit  
Zeitableitung der Geschwindigkeit = Beschleunigung  
Beschleunigung ist proportional der Kraft  
Rücktreibende Kraft proportional zum Weg

Wenn die zeitabhängigen Funktionen  $y_1(t)$  und  $y_2(t)$  jeweils den Weg und die Geschwindigkeit bezeichnen, dann lautet die mathematische Formulierung der obigen Beziehungen im einfachsten Fall

$$\begin{aligned}\dot{y}_1(t) &= y_2(t) \\ \dot{y}_2(t) &= -y_1(t)\end{aligned}$$

Es handelt sich hier um ein *System von zwei Differentialgleichungen erster Ordnung*. Die gute Nachricht: Sowohl die selbst programmierten Einschrittverfahren nach dem Muster des Programmes `GDGdemo.m` als auch die MATLAB-Löser funktionieren – mit wenigen und offensichtlichen Änderungen – auch für Systeme.

**Änderung 1** Sie formulieren die rechte Seite des Systems als vektorwertige Funktion. Konkret setzen Sie in `GDGdemo.m`

```
function ystrich = f(x,y)
    ystrich = [ y(2)
               -y(1)];
end
```

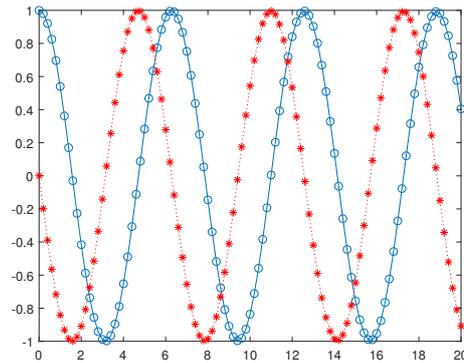
**Änderung 2** Sie brauchen, weil es sich um *zwei* Diffgleichungen erster Ordnung handelt, auch zwei Anfangsbedingungen; konkret Anfangsposition  $y_1(0) = 1$  und Anfangsgeschwindigkeit  $y_2(0) = 0$ . Setzen Sie dazu in `GDGdemo.m`

```
% Setze Anfangswerte, x-Intervall und Schrittweite
x = 0;
y = [1;0]
% Endwert von x und Schrittweite
x_end = 20;
h = 0.2;
```

**Änderung 3** Das Ergebnis  $y$  ist eine zweizeilige Matrix; erste Zeile enthält die berechnete Näherungslösung für den Weg  $y_1(t)$ , die zweite Zeile die entsprechenden Näherungen an die Geschwindigkeit  $y_2(t)$ . Sie stellen beide Funktionen in einem Plot dar, zum Beispiel so:

```
plot(xLsg, yLsg(1,:), '-o', xLsg, yLsg(2,:), '-or' );
```

So sollte Ihr Plot aussehen, wenn `GDGdemo.m` mit dem dreistufigen Verfahren aus Aufgabe 68 rechnet (die anderen Verfahren liefern ähnliche Plots):



### MATLAB-Löser

Die Funktion `ystrich` können Sie von oben übernehmen. Anfangswerte setzen und Löser aufrufen erfolgt in einfachster Form als

```
y0=[1; 0];
xspan=[0, 20];
ode45(@f,xspan,y0)
```

`ode45` erzeugt ein ganz ähnlich aussehendes Bild wie oben.

Wenn Sie die anderen möglichen Varianten des `ode45`-Befehls verwenden,

```
[X, Y] = ode45(@f,xspan,y0);
```

oder

```
sol=ode45(@g,xspan,y0);
x = linspace(0,20);
y = deval(sol,x)
```

können Sie Form und Darstellung des Ergebnisses gezielt steuern.

### Aufgabe 73: Populationsdynamik

Ein mathematisches Modell, das Entwicklung des Bestandes zweier Arten („Räuber“ und „Beute“) in vereinfachter Weise darstellt, wird durch das System nichtlinearer Differentialgleichungen

$$\begin{aligned}\dot{y}_1(t) &= k_1 y_1(t) - k_2 y_1(t) y_2(t) \\ \dot{y}_2(t) &= k_3 y_1(t) y_2(t) - k_4 y_2(t)\end{aligned}$$

ausgedrückt. Dabei stellen  $y_1(t)$  und  $y_2(t)$  den zeitlichen Bestand der Beutetiere beziehungsweise der Räuber dar.

Lösen Sie dieses System für  $0 \leq t \leq 4$  unter der Annahme, daß der Anfangsbestand an Beutetieren 1000, der der Räuber 200 und die Konstanten  $k_1 = 3$ ,  $k_2 = 0,002$ ,  $k_3 = 0,0006$  und  $k_4 = 0,5$  betragen. Methode: `ode45`. Um die Ergebnisse zu veranschaulichen, stellen Sie die Lösung, also die Anzahl der beiden Populationen versus Zeit, graphisch dar.

Wiederholen Sie die Rechnung für einen Anfangsbestand von 2000 Beutetieren und 1500 Räubern.

### Aufgabe 74: Schmetterlingseffekt

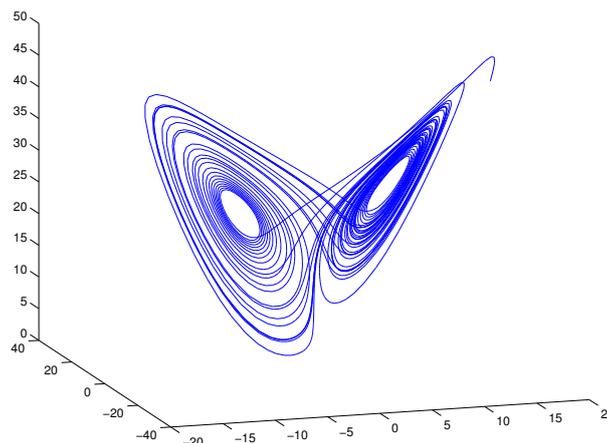
Der Meteorologe Edward N. Lorenz formulierte 1963 ein System von drei gekoppelten, nichtlinearen gewöhnlichen Differentialgleichungen zur Modellierung der Erdatmosphäre zum Zweck einer Langzeitvorhersage. Die Position eines Luftteilchens im dreidimensionalen Raum in Abhängigkeit von der Zeit wird durch die Koordinaten  $x(t)$ ,  $y(t)$ ,  $z(t)$  bestimmt. In Lorenz' Modell gehorchen sie den Bewegungsgleichungen

$$\begin{aligned}dx/dt &= a(y - x) \\ dy/dt &= x(b - z) - y \\ dz/dt &= xy - cz\end{aligned}$$

Die numerische Lösung des Systems zeigt bei bestimmten Parameterwerten chaotisches Verhalten. Die typische Parametereinstellung mit chaotischer Lösung lautet:

$$a = 10, \quad b = 28, \quad c = \frac{8}{3}$$

Lösen Sie dieses System für  $0 \leq t \leq 50$ , Anfangsbedingungen  $x = 20$ ,  $y = 20$ ,  $z = 40$ . Stellen Sie die Lösung als Kurve in Parameterform in einem  $xyz$ -Koordinatensystem (Befehl `plot3`) dar.



Die Lösung ist für ihre typische Schmetterlings-Form bekannt.