Aufgabenblatt 2

Table of Contents

Numerik 2, Wintersemester 2017, Hans W Borchers

PDF Version: MathWorks

Aufgabe 2.1 (Einfache DGL)

Loese die folgende Differenzialgleichung symbolisch und numerisch im Intervall [0, pi] mit der Anfangswertbedingung y(0) = 1.

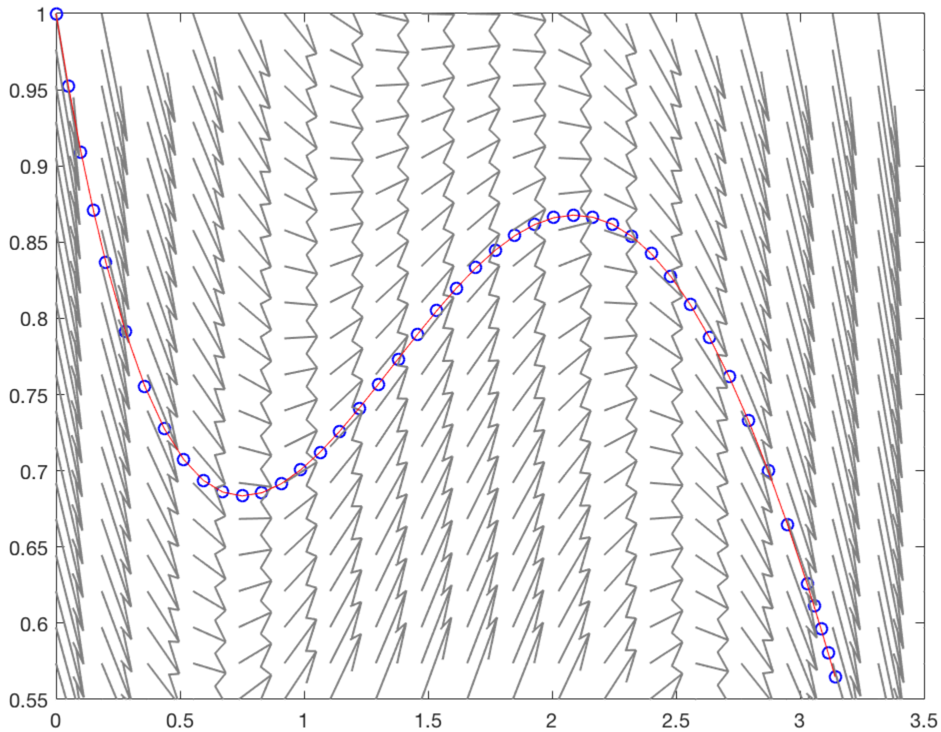
```
y' + y = sin(x)
```

Zeichne in den von *ode45* erstellten Plot die symbolische Loesung als rote Kurve ein. Und zeichne zusaetzlich das Vektorfeld ein. Wie gross ist der maximale Fehler in den von *ode45* berechneten Punkten der Kurve?

```
% DGL Funktion mit Test
f21 = @(x, y) sin(x) - y;
f21(0, 1);

% Aufruf des Solvers
[x, y] = ode45(f21, [0, pi], 1.0);

% Plot Loesung+Vektorfeldes
plot(x, y, 'b--o')
hold on
richtungsfeld(f21, [0, 3.5], [0.55, 1.0], 20)

% Symbolische Loesung
pretty(dsolve('Dy = sin(x) - y', 'y(0) = 1', 'x'))
% (3*exp(-x))/2 - (2^(1/2)*cos(x + pi/4))/2

% Plotten der symbolischen Loesung
ys = (3*exp(-x))/2 - (2^(1/2)*cos(x + pi/4))/2;
plot(x, ys, 'r')
hold off

% Abstand von num. und theor. Loesung
maxDist = max(abs(y - ys));
display(maxDist)
```

```
                  /     pi \
      sqrt(2) cos| x + -- |
```

```
3 exp(-x)              \      4 /
--------- - --------------------
    2                  2

maxDist =
     3.256682105989839e-06
```



Aufgabe 2.2 (Traktrix)

Eine Traktrix (oder Schleppkurve) ist die Kurve, entlang der sich ein Objekt unter dem Einfluss der Reibung bewegt, dass entlang einer Linie an einem Seil, einer Kette gezogen wird.

Ein schwerer Stein/Findling befindet sich in an der Stelle (0,10) (in [m]), ein Traktor in (1,0). Der Stein wird vom dem Traktor durch ein Seil der Laenge L = sqrt(101) befestigt, dann faehrt der Traktor auf der Strasse (x-Achse) etwa 60 m weit. Wie sieht die Kurve aus, entlang der der Stein gezogen wird.

Die Tangente der gesuchten Kurve zeigt immer in Richtung des Traktors, die Laenge der Tangente bis zur x-Achse betraegt immer L. Daraus folgt, dass die Kurve die folgende Differenzialgleichung erfiellt:

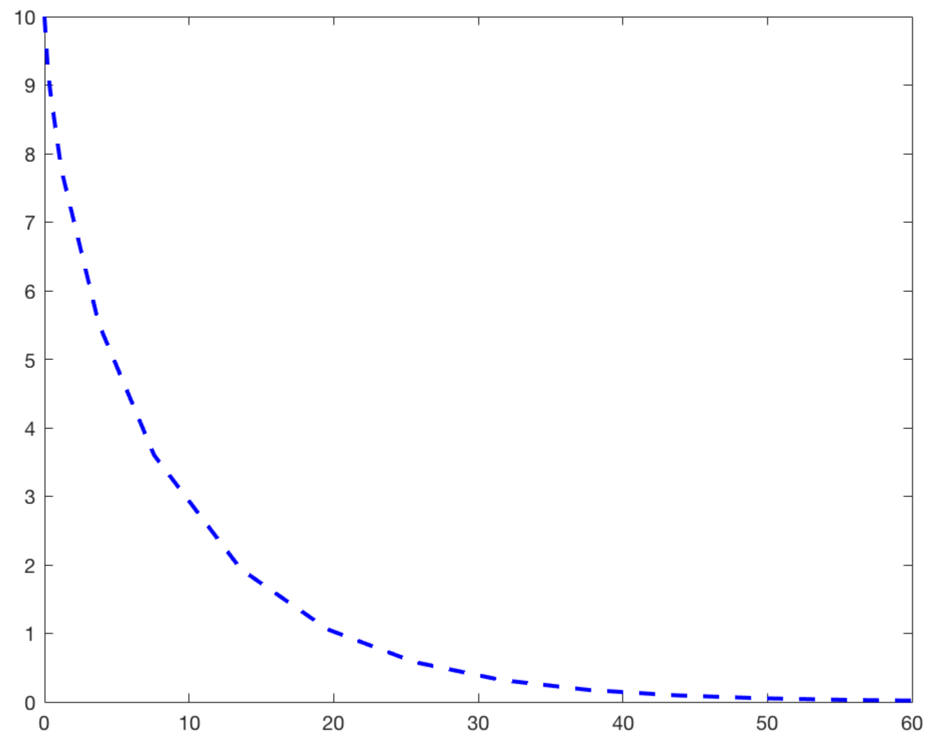
y' = -y(x) / sqrt(L^2 - y(x)^2)

Zeichne den Weg, den der Stein gezogen wird, als gestrichelte blaue Linie mit einer Linienbreite von 2. Wie nahe ist das Objekt an der Strasse, nachdem es 60 Meter geschleppt wurde?

```
% DGL Funktion mit Test
L = sqrt(101);
f22 = @(x, y) - y / sqrt(L^2 - y^2);
```

```
f22(1, 10);

% Plotten der Loesung
sol = ode45(f22, [0, 60], 10);
plot(sol.x, sol.y, 'b--', 'LineWidth', 2)

a = deval(sol, 60);
fprintf('Letzter Abstand zur x-Achse: %6.4f [m]\n', a)
```

```
Letzter Abstand zur x-Achse: 0.0189 [m]
```



Zusatz : Symbolische Loesung der Traktrix

Eine explizite Darstellung von y(x) ist nicht moeglich!

```
syms L
dsolve('Dy = -y / sqrt(L^2 - y^2)', 'y(0) = L', 'x')
```

```
Warning: Explicit solution could not be found; implicit solution
 returned.
ans =
solve((L^2 - y^2)^(1/2) - L*atanh((L^2 - y^2)^(1/2)/L) == -x, y)
```

Aufgabe 2.3 (Lotka-Volterra)

In einem geschlossenen Habitat gibt es Populationen von Beutetieren x und Raeubern y, die sich ueber die Zeit entwickeln nach folgendem Modell:

```
dx/dt =  a x(t) - b x(t) y(t)
dy/dt = -c y(t) + d x(t) y(t)
```

mit der Konstanten a = 2.0,b = 0.01,c = 1.0,d = 0.001

(1) Berechne des Modell als DGL mit den Anfangsbedingungen x(0) = 300, y(0) = 150 .

(2) Erstellen Sie ein Phasendiagramm dieses Models, d.h. tragen Sie in einem Plot die Anzahl der Beutetiere gegen die Anzahl der Raeuber auf.

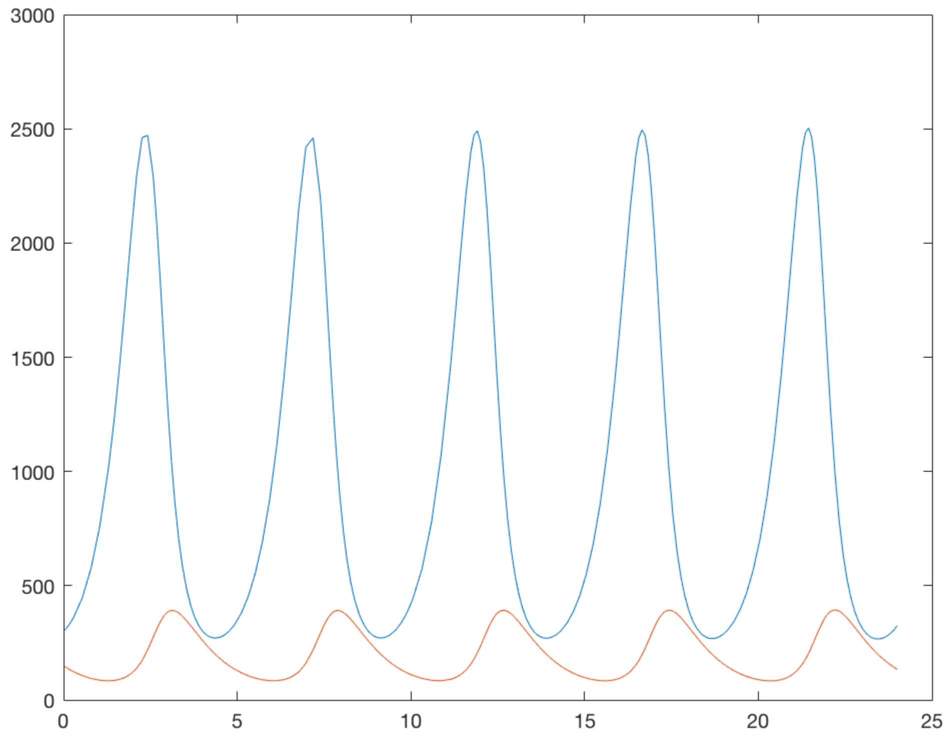
(3) Wenn sich in dem Habitat maximal x0 = 2000 Beutetiere (von Pflanzen) ernaehren koennen, aendert sich die erste Gleichung zu

```
dx/dt = a (1 - x(t)/2000) x(t) - b x(t) y(t)
```

Loesen Sie das System der DGLen fuer diesen Fall.

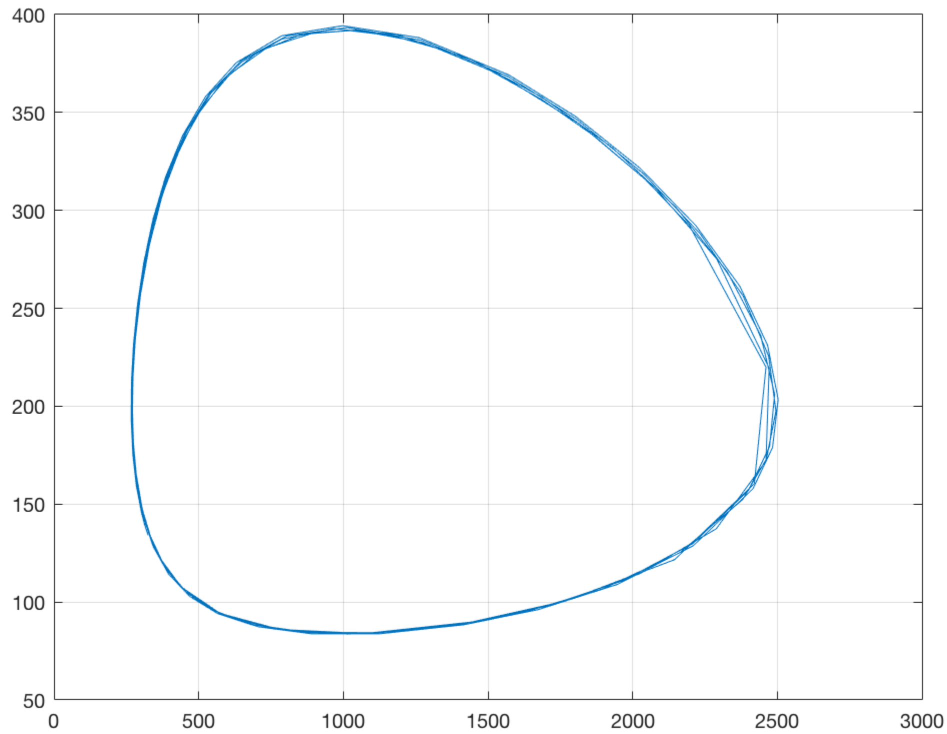
(1) Berechnung des Models

```
% Beschreibe das Habitat als H = [x; y]

% Aufstellen der DGL Funktion
a = 2.0; b = 0.01; c = 1.0;d = 0.001;
f23 = @(t, H) [ a*H(1) - b*H(1)*H(2)
               -c*H(2) + d*H(1)*H(2)];

% Loesung des DGL Systems
[t, H] = ode45(f23, [0, 24], [300; 150]);

% Plausibilitaet
plot(t, H(:, 1), t, H(:,2))
```



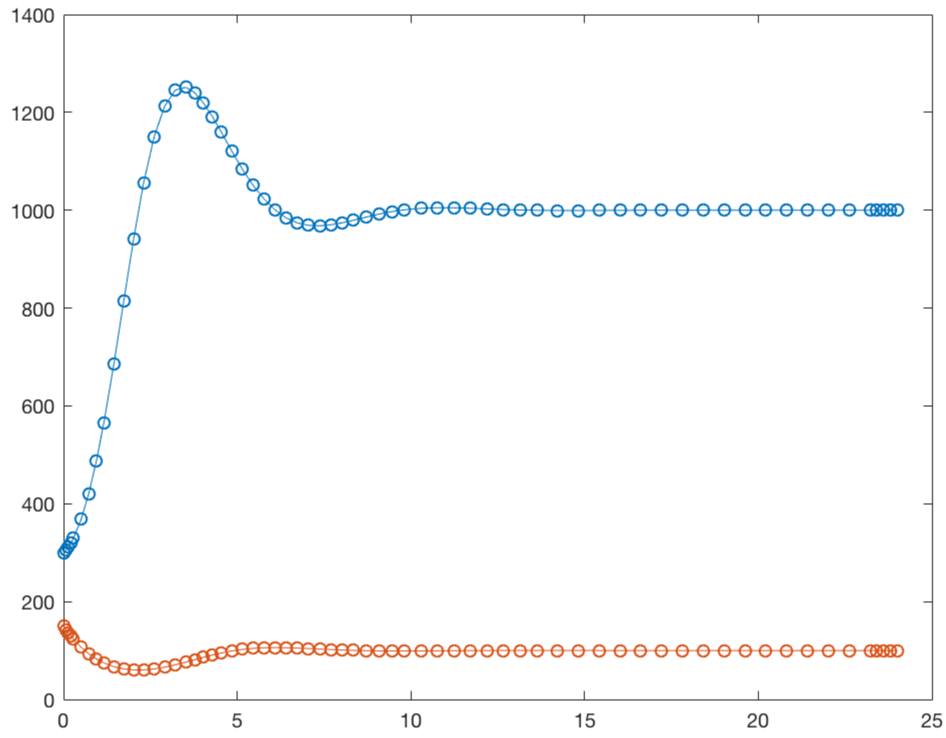
(2) Phasendiagramm des Modells

```
% Phasen-/Zustandsdiagramm
figure()
plot(H(:, 1), H(:,2))
grid()
```



(3) Habitat mit Begrenzung

```
% DGL Funktion mit Begrenzung
B = 2000;
f23b = @(t, H) [ a*(1 - H(1)/B)*H(1) - b*H(1)*H(2);
                -c*H(2) + d*H(1)*H(2)];

ode45(f23b, [0, 24], [300; 150]);
```



Die Anzahle der Raeuber und Beutetiere wird offenbar konstant.

Aufgabe 2.4 (Radioaktiver Zerfall)

Eine Probe enthalte 10 g eines radioaktiven Materials A, welches mit einer Halbwertszeit von 10 Stunden in ein anderes radioaktives Element B zerfaellt. B hat eine Halbwertszeit von zweieinhalb Tagen. Zum Zeitpunkt 0 gab es in der Probe nur das Elementes A.

Ist H die Halbwertszeit eines radioaktiven Elementes A, dann gilt: `A'(t) = -log(2) / H A(t)`. (Warum?)

Damit gilt es folgendes System von DGLen zu loesen:

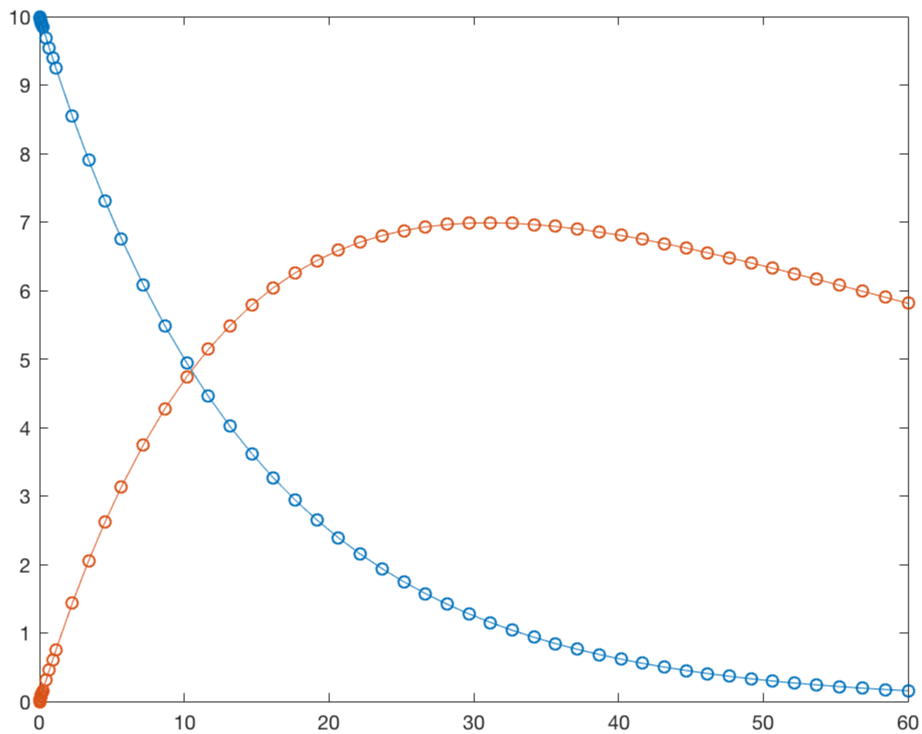
```
a = 10; b = 60
dA/dt = -log(2)/a A(t)
dB/dt = -log(2)/b B(t) + log(2)/a A(t)
```

1. Wann ist die Menge des Stoffes B in der Probe am groessten?

2. Zum Zeitpunkt einer Messung gibt es zehn Mal so viele B-Atome in der Probe wie A-Atome. Wie viel Zeit ist seit Beginn des Experimentes vergangen?

```
% Aufstellen des DGL Systems mit r = [A, B]
a = 10; b = 60;
f24 = @(t, r) [-log(2)/a*r(1)
               -log(2)/b*r(2) + log(2)/a*r(1)];
```

```
% Verlauf der Stoffmengen A (blau) und B (rot)
ode45(f24, [0, 60],[10; 0]);
```



(1) Wann ist die Stoffmenge B am groessten?

```
sol = ode45(f24, [0, 60],[10; 0]);

% Define B(t) als Funktion
Bfun = @(t) deval(sol, t, 2);

b_max = fminbnd(@(t) -deval(sol, t, 2), 0, 60)
```

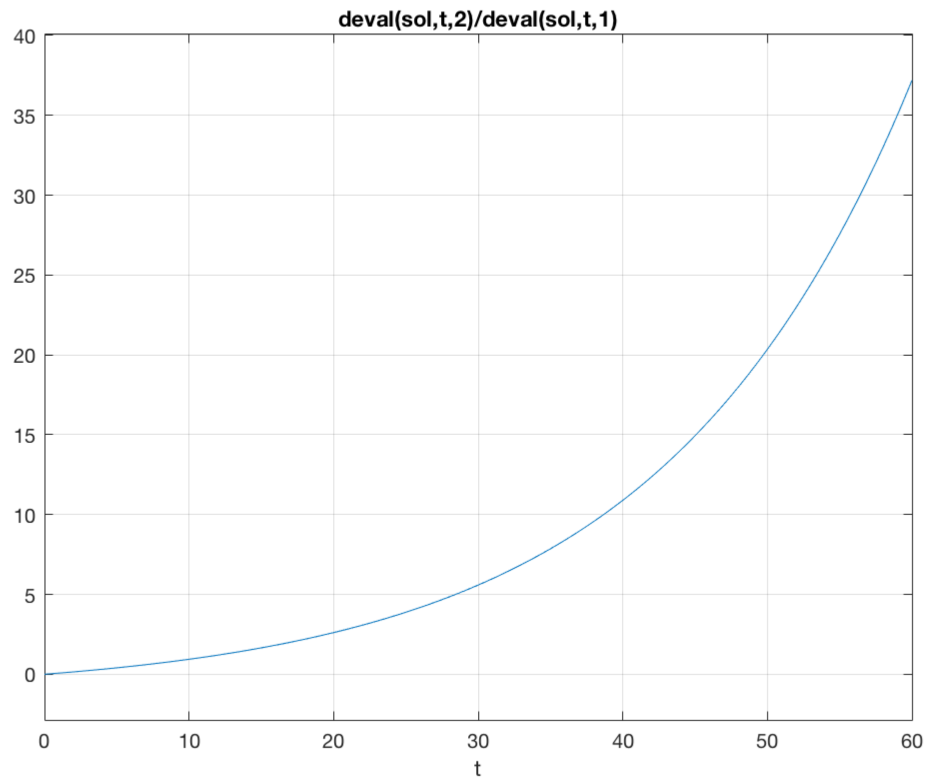
```
b_max =
  31.021143085108850
```

Bestimme das Verhaeltnis B/A

```
% B / A
BAfun = @(t) deval(sol, t, 2) ./ deval(sol, t, 1);

% Plotte diese Funktion auf [0, 60]
ezplot(BAfun, [0, 60]); grid()

% Bestimme t mit BAfun(t) = 10
BA_10 = fzero(@(t) BAfun(t) - 10, [0, 60])
fprintf('Zeitpunkt B/A = 10: %5.2f Stunden\n', BA_10)
```

```
BA_10 =
```

```
  38.668764134501721
Zeitpunkt B/A = 10: 38.67 Stunden
```



Aufgabe 2.5 (Chemischer Reaktor)

In einem Reaktor befinden sich drei chemische Substanzen mit den Anfangskonzentrationen 0.6, 0.2 und 0.2. Der Chemie-Ingenieur sagt Ihnen, dass die Konzentrationen der Stoffe in Gegenwart der anderen Stoffe sich nach folgenden Regeln ver?ndern:

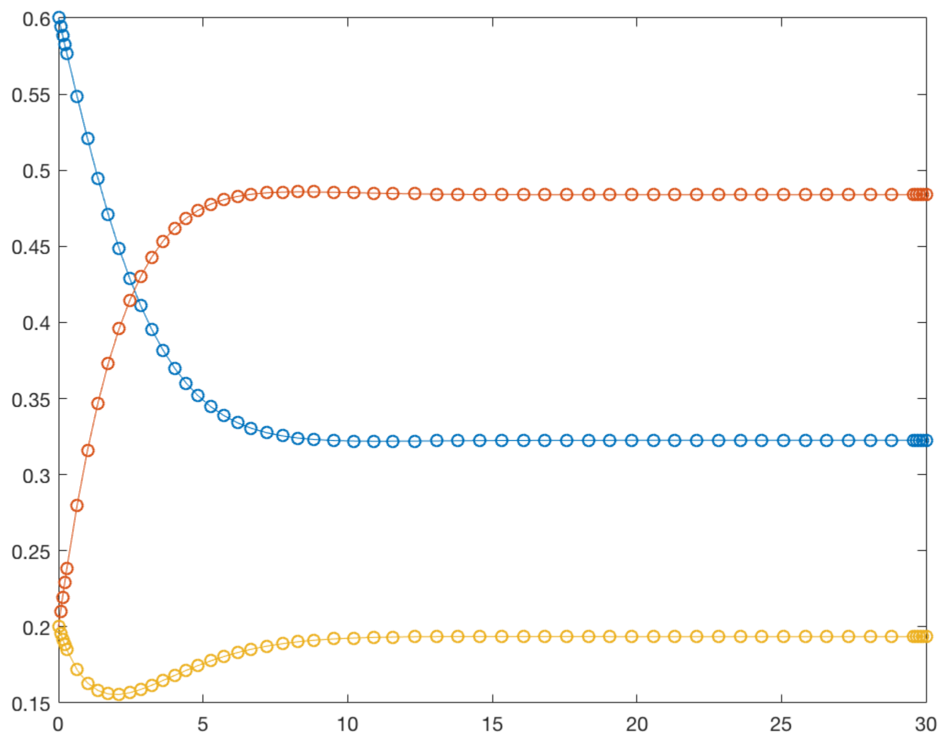
```
c1' = -k1 c1 + k3 c3
c2' =  k1 c1 - k2 c2
c3' =  k2 c2 - k3 c3
```

Die Konstanten gibt er an zu k1 = 0.3, k2 = 0.2, und k3 = 0.5. Bestimmen Sie die Konzentrationen der drei Substanzen nach einer geeigneten Reaktionszeit.

```
% Konstanten
k1 = 0.3; k2 = 0.2; k3 = 0.5;

% DGL Funktion mit c = [c1, c2, c3]
f25 = @(t, c) [ -k1*c(1) + k3*c(3)
                 k1*c(1) - k2*c(2)
                 k2*c(2) - k3*c(3) ];

% ode45 Solver
ode45(f25, [0, 30], [0.6, 0.2, 0.2])
```

```
%%=
% Nach 20 Sekunden/Minuten(?) sind die Konzentrationen relativ stabil.

% Bestimme Anteile mit deval
sol = ode45(f25, [0, 30], [0.6, 0.2, 0.2]);

fprintf('Anteil der Konzentrationen C1, C2, C3:\n')
disp(deval(sol, 20))
```

```
Anteil der Konzentrationen C1, C2, C3:
   0.322574344763979
   0.483868296628804
   0.193557358607217
```



Aufgabe 2.6 (Fallender Baseball)

Ein Baseball falle aus einer Hoehe von 20 Metern auf den Boden. Entsprechend den Fallgesetzen gilt f?r die Hoehe y ?ber dem Boden

```
y'' = c y'^2 - g
```

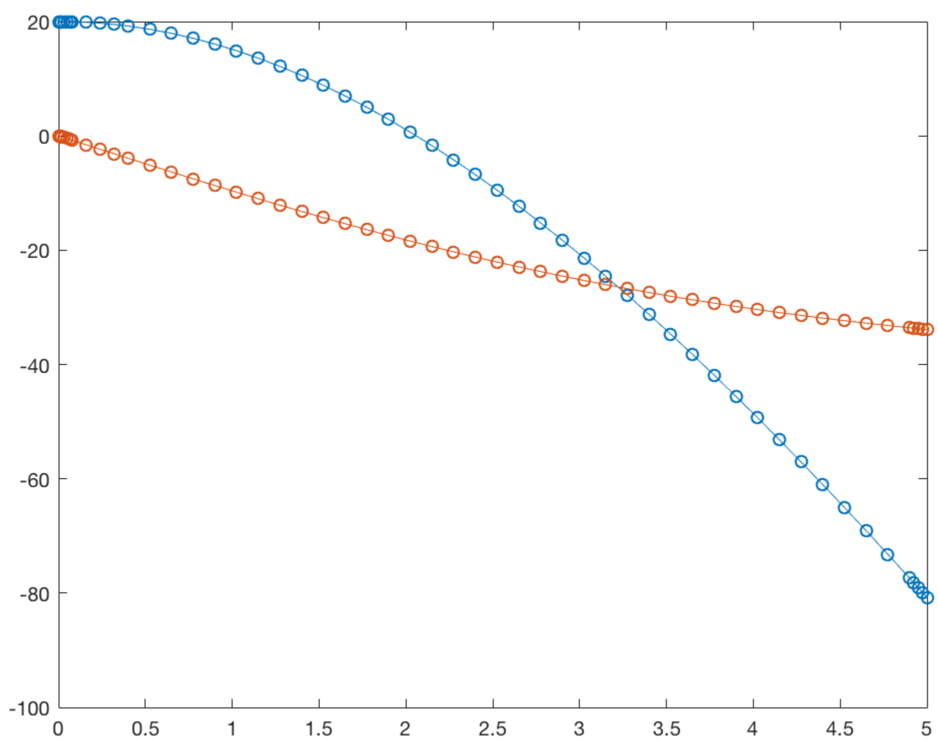
mit der Erdbeschleunigung g = 9.81 und einem fuer einen Baseball typischen Luftwiderstandsbeiwert c = 0.006 .

Definiere zwei neue Funktionen y1(t) = y(t) und y2(t) = y'(t) und schreibe die obige Gleichung als ein System von DGLen:

```
y1' = y2
y2' = c y2^2 - g
```

- Loese dieses Differenzialgleichungssystem. Nach wieviel Sekunden genau trifft der Baseball am Boden auf und mit welcher Geschwindigkeit?
- Nach wieviel Sekunden wuerde er auftreffen, wenn der Luftwiderstand nicht da waere. (Theoretisch zu loesen, nicht numerisch.)

```
% Konstanten
g = 9.81; % [m/s^2]
c = 0.006;

% DGL Funktion fuer y = [y1, y2]
f26 = @(t, y) [y(2)
               c*y(2)^2 - g];

% Plotten
ode45(f26, [0, 5], [20; 0])
```

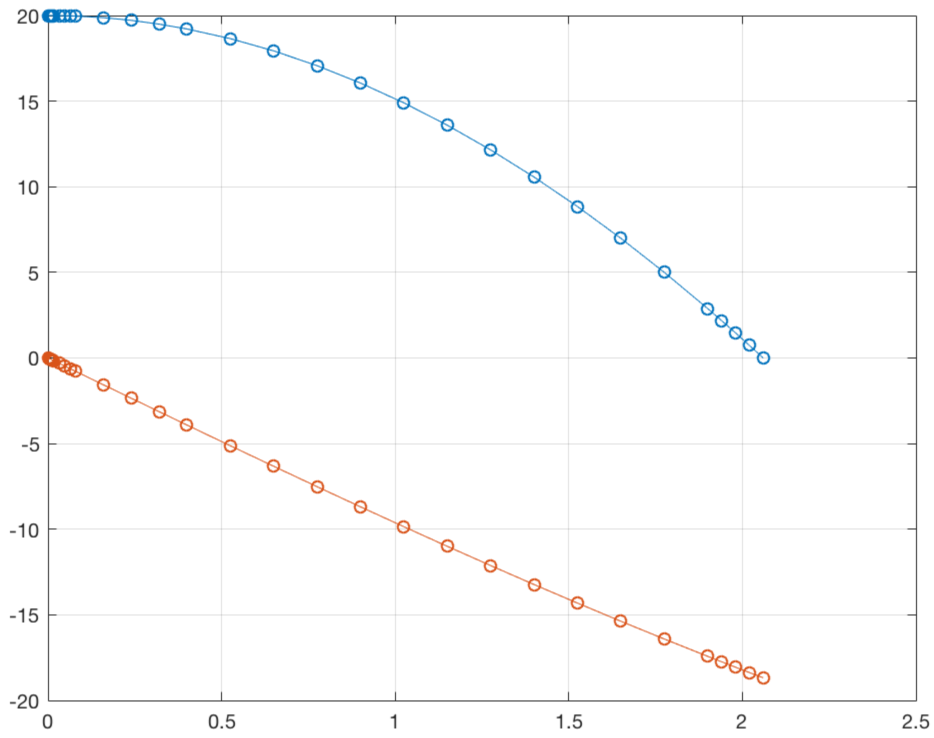


Definiere ein 'Event', wenn der Ball am Boden auftrifft, in einer Datei `ev_ball.m`

```
function [fun, act, dir] = ev_ball(t, y)
fun = y(1);  % Nullstelle y = 0
act = 1;     % Stop
dir = 0;     % -1 fallend, +1 steigend
end
```

```
% |ode45| mit Option
opts = odeset('Events', @ev_ball);
ode45(f26, [0, 5], [20; 0], opts)
grid()

%
[t, y] = ode45(f26, [0, 5], [20; 0], opts);
fprintf('Fallzeit: %5.2f sec\n', t(end))
```

```
Fallzeit:  2.06 sec
```



Die Fallzeit *ohne* Luftwiderstand ergibt sich aus der Gleichung $t = \frac{1}{2}gt^2$. Also: $t = \sqrt{\frac{2s}{g}}$

```
s = sqrt(2*20/9.82);
fprintf('Fallzeit ohne Luftwiderstand: %5.2f sec\n', s)
```

```
Fallzeit ohne Luftwiderstand:  2.02 sec
```

Published with MATLAB® R2016a