TU Braunschweig **WS 2003/2004**
Fachbereich 1 — Mathematik und Informatik
M. Bollhöfer
Beispiele erstellt von Tobias Damm und Heike Faßbender

Differentialgleichungen mit MATLAB
„Numerik gewöhnlicher Differentialgleichungen“

http://www.tu-braunschweig.de/numerik/

---

1 Beispiel: Exponentielles Wachstum

Wir betrachten die Differentialgleichung

$$\dot{x} = x, \qquad x(t_0) = x_0 \,. \tag{1}$$

Mit `ode23` können wir sie numerisch lösen. Die Funktion `ode23` benötigt eine Datei, in der eine Funktion mit den Argumenten `t` und `x` definiert wird, die die Ableitung von (1) berechnet. Also wird die Datei `ExpWachs.m` mit folgendem Inhalt angelegt:

```
function dx = ExpWachs(t,x)
  dx = x;
end;
```

Als weitere Argumente benötigt `ode23` das Lösungsintervall $[t_0, t_{\max}]$ und den Anfangswert x_0. Wir berechnen verschiedene Lösungen, um ein Phasenportrait des Systems zu zeichnen.

```
>>  [t1,x1] = ode23('ExpWachs',[0,5],0.1);
>>  [t2,x2] = ode23('ExpWachs',[0,5],-0.1);
>>  [t3,x3] = ode23('ExpWachs',[0,7],0.01);
>>  [t4,x4] = ode23('ExpWachs',[0,7],-0.01);
>>  [t5,x5] = ode23('ExpWachs',[2,7],0.1);
>>  [t6,x6] = ode23('ExpWachs',[2,7],-0.1);
```

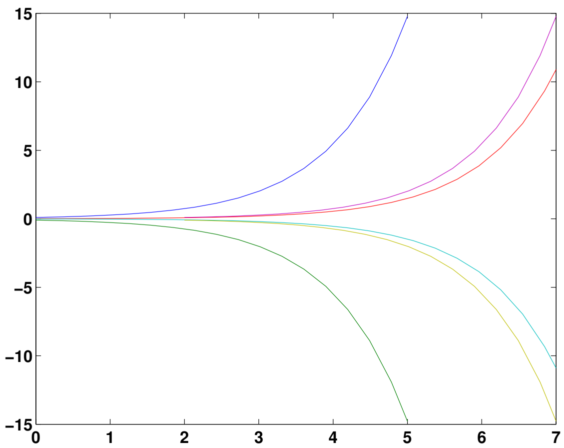
Alles wird auf einem Bild dargestellt (links):

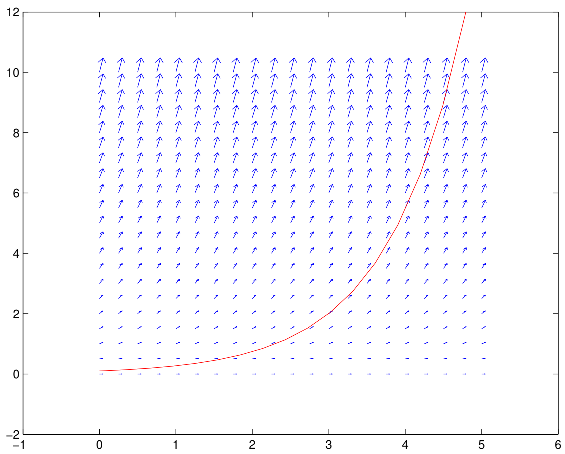
```
>>  plot(t1,x1,t2,x2,t3,x3,t4,x4,t5,x5,t6,x6)
```

Jetzt wird das Richtungsfeld berechnet und mit `quiver` dargestellt (rechts).

```
>>  x = 0:0.5:10;
```

```
>>  t = 0:0.25:5;
>>  for i=1:length(x)
for j = 1:length(t)
DX(i,j) = ExpWachs(t(j),x(i));
DT(i,j) = 1;
end
end
>>  quiver(t,x,DT,DX);
```





2 Beispiel: Unstetige Ableitung

Wir betrachten die Gleichung:

$$\dot{x} = \begin{cases} -1, & x > 0 \\ 0, & x = 0 \\ 1, & x < 0 \end{cases} \tag{2}$$

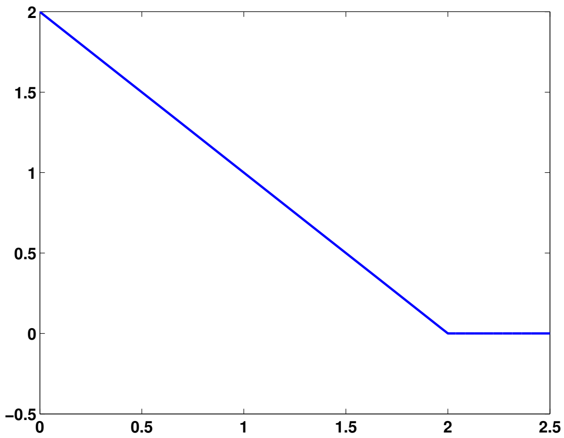
und beobachten ein Problem bei der numerische Lösung.
Wir legen folgende Datei `Sgn.m` an

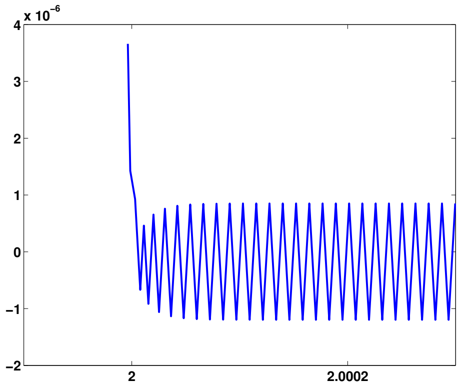
```
function dx = Sgn(t,x)
  if x < 0
    dx = 1;
  elseif x == 0
    dx = 0;
  else
    dx = -1;
  end
```

und berechnen mit `ode23` eine Lösung:

```
>>  [t1,x1] = ode23('Sgn',[0,2.5],2);
```
(wieso dauert das so lange ??)

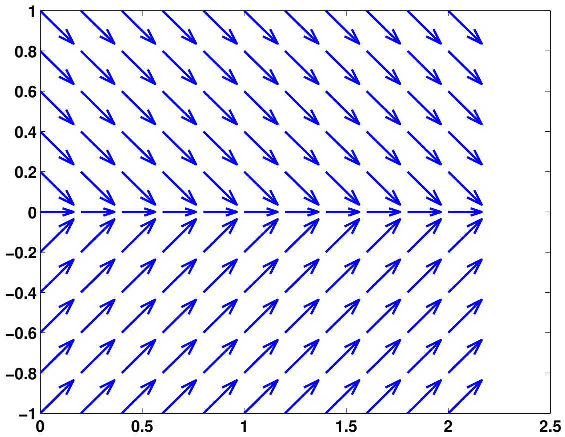
```
>>  plot(t1,x1)
```





Obwohl die korrekte Lösung ab dem Zeitpunkt $t = 0$ stationär ist, also immer den Wert 0 hat, konvergiert die berechnete Lösung nicht, sondern vibriert um 0. Dieses Verhalten ist durch die Ungenauigkeit des numerischen Verfahrens zu erklären. Die Vergrößerung (rechts) macht es deutlich. Diese Schwingungen sind sehr aufwendig zu berechnen.
Wie im vorherigen Beispiel wird das Richtungfeld gezeichnet:

```
>>  x = -1:0.2:1;
>>  t = 0:0.2:2;
>>  for i=1:length(x)
for j = 1:length(t)
DX(i,j) = Sgn(t(j),x(i));
DT(i,j) = 1;
end;
end;
>>  quiver(t,x,DT,DX);
>>  hold on
>>  plot(t1,x1);
```



3 Beispiel: Der lineare Oszillator

Wir finden numerische Lösungen der linearen Oszillatorgleichung, und benutzen MATLAB, um das Verhalten des Systems zu verstehen.

3.1 Mathematische Vorarbeit

Die Differentialgleichung für einen linearen Oszillator ist

(3) $$\ddot{x} + R\dot{x} + Fx = 0,$$

wobei R die Reibungskonstante, F die Federkonstante und $x \in \mathbb{R}$ der Ort ist. Weil man in MATLAB nur Gleichungen erster Ordnung der Form

$$\dot{\mathbf{x}} = f(\mathbf{x})$$

mit $\mathbf{x} \in \mathbb{R}^n$ lösen kann, schreiben wir (3) um:
Es seien $x_1 := x,\ x_2 := \dot{x}$. Dann ist (3) äquivalent zu dem System

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -Rx_2 - Fx_1 , \end{aligned}$$

das sich vektoriell folgendermaßen schreiben läßt:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -F & -R \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} . \tag{4}$$

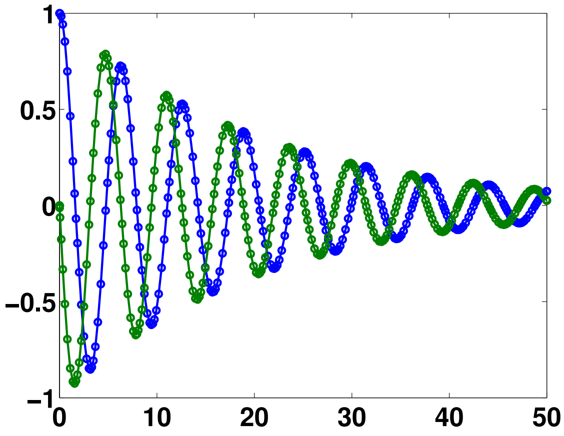
3.2 Numerische Lösung mit MATLAB

Zur Definition der Gleichung schreiben wir die Datei `LinOsc.m`:

```
function  dx = LinOsc(t,x)
   R = 0.1;
   F = 1;
   dx = [ 0 1 ; -F -R ] * x;
```

Mittels `ode23` berechnen wir die Lösung zur Anfangsbedingung $x(0) = 1,\ \dot{x}(0) = 0$ auf dem Zeitintervall $[0, 50]$.

```
>>   ode23('LinOsc',[0,50],[1;0]);
```



Wollen wir nur den zeitlichen Verlauf der Ortskoordinate x_1 darstellen, schreiben wir

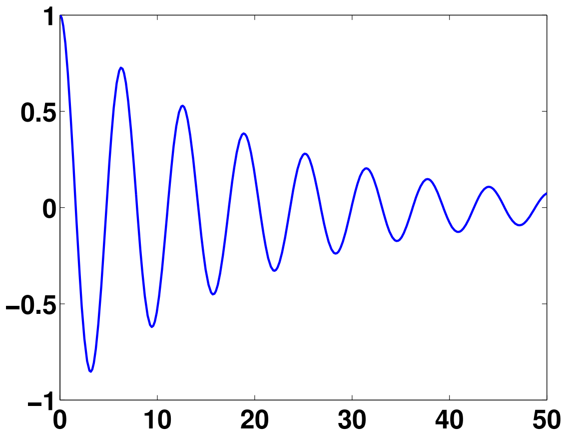
```
>>   [t,x] = ode23('LinOsc',[0,50],[1;0]);
>>   plot(t,x);
>>   plot(t,x(:,1));
```

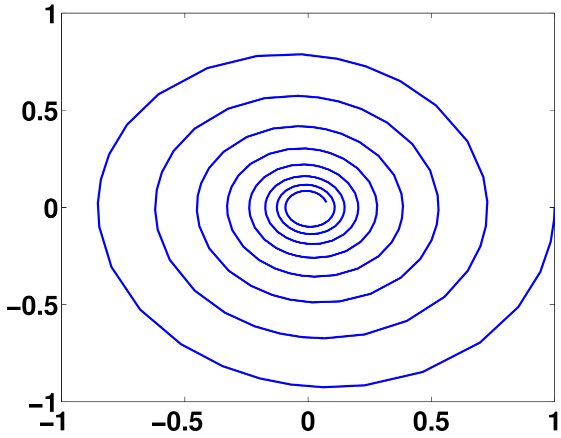
(links)

Die Lösungstrajektorie im Phasenraum $\mathbb{R}^2$ erhalten wir dann entsprechend mit:

```
>>   plot(x(:,1),x(:,2))
```

(rechts)





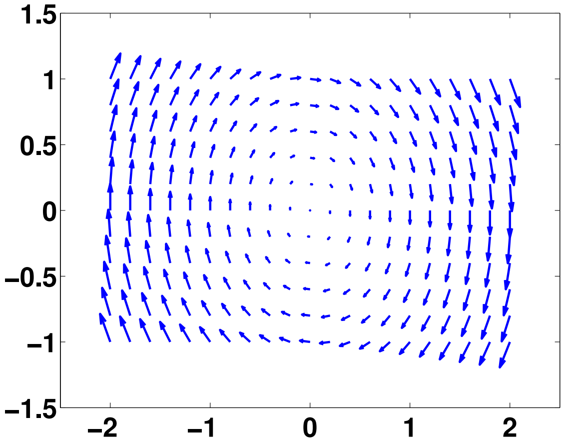
3.3 Phasenportrait für den linearen Oszillator

Um das Vektorfeld des Systems darzustellen, definieren wir ein Gitter und bestimmen in jedem Punkt des Gitters den zugehörigen Feldvektor.

```
>>  x1=-2:0.2:2;
>>  x2=-1:0.2:1;
>>  for i = 1:length(x1)
for j = 1:length(x2)
dx = LinOsc(0,[x1(i);x2(j)]);
DX1(i,j) = dx(1);
DX2(i,j) = dx(2);
end
end
```

Mit dem Befehl `quiver` erhalten wir das nebenstehende Bild:

```
>>  quiver(x1,x2,DX1',DX2')
```



4 Beispiel: Die Pendelgleichung

Die Bewegung eines Pendels kann durch die Differentialgleichung

$$\ddot{\theta} + R\dot{\theta} + K\sin(\theta) = 0 \tag{5}$$

beschrieben werden; hierbei sind θ der Winkel der Auslenkung aus der Vertikalen, R der Reibungskoeffizient und K eine Konstante, die von der Länge des Pendels und der Schwerkraft abhängt.
Wie im vorherigen Beispiel wird (5) als 2-dimensionales System umgeschrieben und durch eine MATLAB-Funktion `Pendel.m` dargestellt:

$$\begin{aligned} x_1 &:= \theta \\ x_2 &:= \dot{\theta} \\ \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -Rx_2 + \sin(x_1) \end{aligned}$$

```
function dx = Pendel(t,x)
   R = 0;     % keine Reibung
   K = 1;
   dx = [ x(2) ; -K*sin(x(1))-R*x(2) ];
```

Mit `ode23` berechnen wir die Lösungen zu folgenden Anfangsbedingungen und stellen alle in einem Bild dar: $(0.5, 0)$, $(1.0, 0)$, $(1.5, 0)$, $(2.0, 0)$, $(2.5, 0)$, $(3.0, 0)$, $(3.5, 0)$, $(4.0, 0)$, $(4.5, 0)$, $(2\pi, 0)$, $(-\pi, 0.01)$, $(3 * \pi, -0.01)$, $(-\pi, 1)$, $(-\pi, 2)$. Daneben wieder das Vektorfeld mit `quiver`: