lektion13

January 20, 2022

Table of Contents

1 Lektion 13

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
from sympy import *
init_printing()
%matplotlib inline
```

1.1 Differentialgleichungen (Teil 2)

$$y'(t) = f(y,t), \quad y(t_0) = y_0$$

1.1.1 Logistische Gleichung (Wiederholung)

einfaches Modell zur Beschreibung von Wachstum in einem Habitat mit beschränkten Resourcen

$$y'(t) = (1 - y(t))y(t)$$

[2]:
```
y = Function('y')
t, t0, tau = symbols('t t_0 tau', real = True)
yy = symbols('yy')
dgl = Eq(y(t).diff(t), (1-y(t))*y(t))
dgl
```

[2]:
$$\frac{d}{dt}y(t) = (1 - y(t))\, y(t)$$

[3]:
```
sol = dsolve(dgl, y(t))
sol
```

[3]:
$$y(t) = \frac{1}{C_1 e^{-t} + 1}$$

[4]:
```
c1 = S('C1')
c1
```

[4]:
$$C_1$$

[5]:
```
C1 = solve(sol, c1)
C1
```

[5]:
$$\left[-e^{t} + \frac{e^{t}}{y(t)}\right]$$

[6]:
```
C1 = solve(sol, c1)[0].subs(t, t0)
C1
```

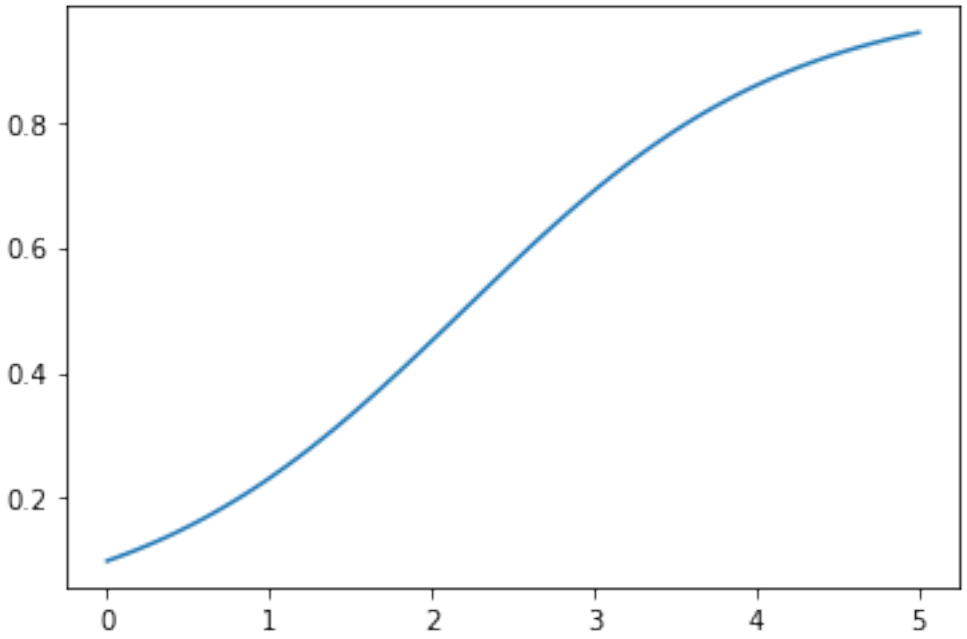
[6]:
$$-e^{t_0} + \frac{e^{t_0}}{y(t_0)}$$

[7]:
```
ys = sol.subs(c1, C1)
ys
```

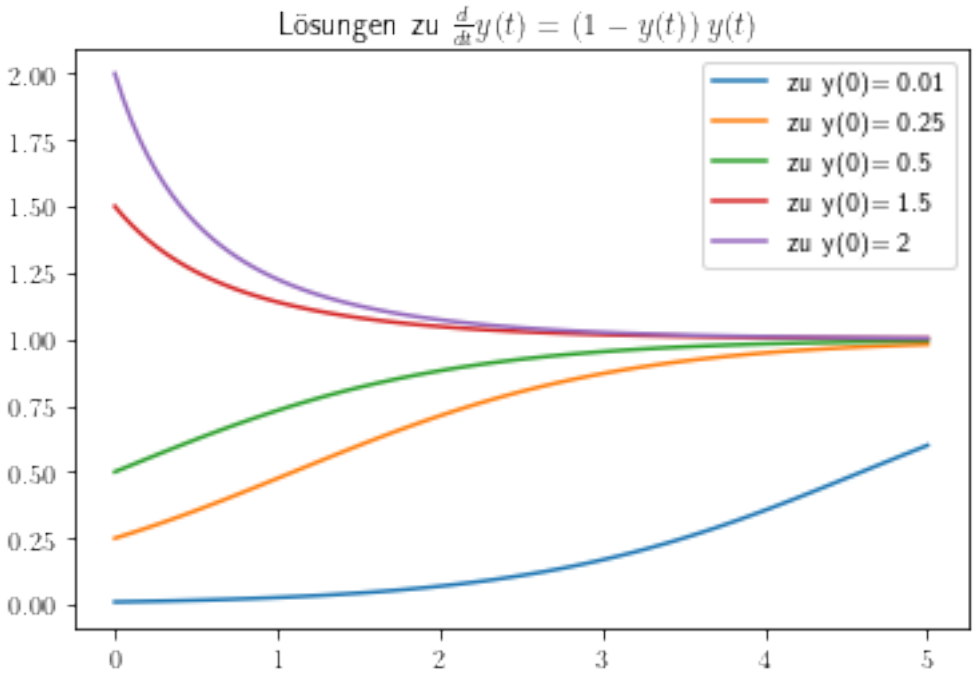
[7]:
$$y(t) = \frac{1}{\left(-e^{t_0} + \frac{e^{t_0}}{y(t_0)}\right) e^{-t} + 1}$$

[8]:
```
fig = plt.figure(1)
fig.clf()
ax = fig.gca()
tn = np.linspace(0, 5, 101)
yn = ys.rhs.subs(t0, 0).subs(y(0), .1) # Anfangsbedingung y(0)= 0.1
yn = lambdify(t, yn)
ax.plot(tn, yn(tn))
```

[8]:
```
[<matplotlib.lines.Line2D at 0x7f47ece8c850>]
```



[9]:
```
plt.rcParams['text.usetex'] = True
fig = plt.figure(2)
ax = fig.gca()
tn = np.linspace(0, 5, 101)
for y0 in [.01, .25, .5, 1.5, 2]: # mehrere Anfangsbedingungen
    yn = ys.rhs.subs(t0, 0).subs(y(0), y0)
    yn = lambdify(t, yn)
    ax.plot(tn, yn(tn), label="zu y(0)= {}".format(y0))
ax.legend()
ax.set_title(r"Lösungen zu ${}$".format(latex(dgl)));
```



[10]:
```
fy = dgl.rhs.subs(y(t), yy)
fy
```

[10]:
$$yy\,(1 - yy)$$

[11]:
```
fn = lambdify(yy, fy)
tg = np.linspace(0, 5, 10)
yg = np.linspace(0.1, 1.9, 20)
TT, YY = np.meshgrid(tg, yg)
ax.quiver(TT, YY, np.ones_like(YY), fn(YY), angles='xy');
```

1.1.2 Systeme von DGLen

Linearer Fall – Matrixexponentialfunktion Für eine Matrix $n \times n$ A suchen wir eine Lösung $u : [t_0, T] \to \mathbb{R}^n$ von

$$u'(t) = Au(t), \quad u(0) = u_0$$

[12]:
```
t = symbols('t', real=True)
```

[13]:
```
A = Matrix(3, 3, [2, 1, 0,\
                  0, 2, 1,\
                  0, 0, 2])
A
```

[13]:

$$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

[14]:
```
exp(t*A)
```

[14]:
$$\begin{bmatrix} e^{2t} & te^{2t} & \frac{t^2e^{2t}}{2} \\ 0 & e^{2t} & te^{2t} \\ 0 & 0 & e^{2t} \end{bmatrix}$$

[15]:
```
u0 = Matrix(3, 1, [1, 2, 3]) # Anfangswert
u0
```

[15]:
$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

[16]:
```
u = exp(t*A)*u0
u
```

[16]:
$$\begin{bmatrix} \frac{3t^2e^{2t}}{2} + 2te^{2t} + e^{2t} \\ 3te^{2t} + 2e^{2t} \\ 3e^{2t} \end{bmatrix}$$

[17]:
```
u.diff(t) - A*u
```

[17]:
$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

1.1.3 Zweite Ordnung DGL

Harmonischer Oszillator

$$y''(t) = y(t)$$

[18]:
```
dgl = Eq(y(t).diff(t, 2), -y(t))
dgl
```

[18]:
$$\frac{d^2}{dt^2}y(t) = -y(t)$$

[19]:
```
dsolve(dgl, y(t))
```

[19]:
$$y(t) = C_1 \sin(t) + C_2 \cos(t)$$

äquivalentes System 1. Ordnung

$$\begin{aligned} u_1'(t) &= u_2(t) \\ u_2'(t) &= -u_1(t) \end{aligned}$$

in Matrixschreibweise

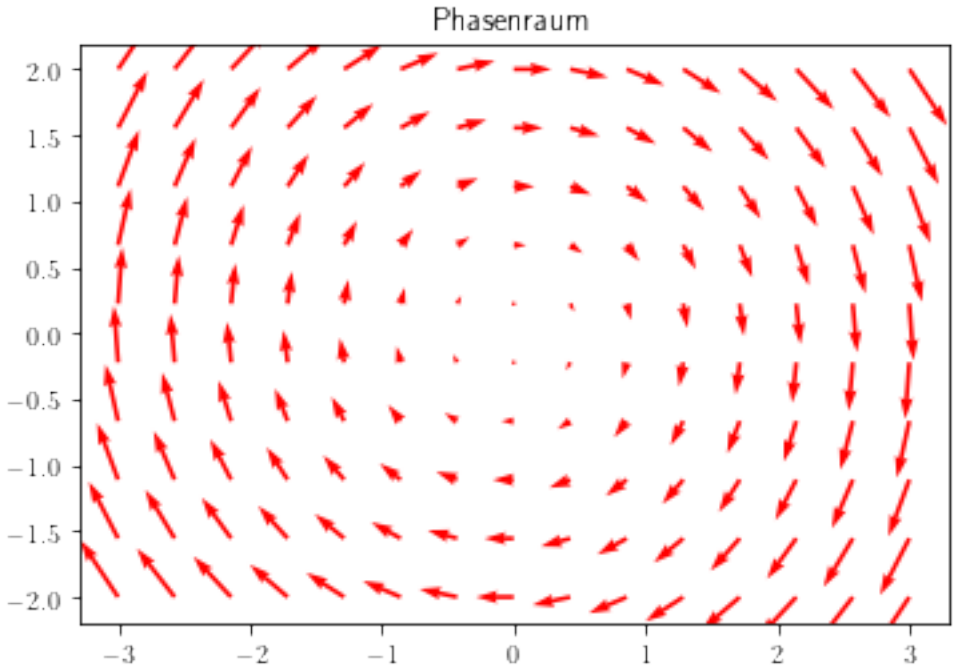
$$u'(t) = Au(t) := f(u(t))$$

[20]:
```
def f(u):
    return u[1], -u[0]
```

[21]:
```
u1 = np.linspace(-3.0, 3.0, 15)
u2 = np.linspace(-2.0, 2.0, 10)
U1, U2 = np.meshgrid(u1, u2)
```

[22]:
```
U, V = f([U1, U2])
```

[23]:
```
fig = plt.figure(3)
ax = fig.add_subplot(111)
ax.set_title('Phasenraum')
ax.quiver(U1, U2, U, V, angles='xy', color='r')
ax.set_aspect('equal');
```



Oszillator mit Dämpfung und Quelle

$$y''(t) = -2y'(t) - y(t) - 3\cos(t)$$

[24]:
```
dgl2 = Eq(y(t).diff(t, 2) , -2*y(t).diff(t) - y(t) - 3*cos(t))
dgl2
```

[24]:
$$\frac{d^2}{dt^2}y(t) = -y(t) - 3\cos(t) - 2\frac{d}{dt}y(t)$$

[25]:
```
sol = dsolve(dgl2 ,y(t))
sol.rhs
```

[25]:
$$(C_1 + C_2 t)\, e^{-t} - \frac{3\sin(t)}{2}$$

Anfangswerte

$$y(0) = 4,\ y'(0) = 0$$

[26]:
```
eqn = [Eq(sol.rhs.subs(t, 0), 4),
       Eq(sol.rhs.diff(t).subs(t, 0), 0)]
eqn
```

[26]:
$$\left[C_1 = 4,\ -C_1 + C_2 - \frac{3}{2} = 0\right]$$

[27]:
```
Constants = solve(eqn)
Constants
```

[27]:
$$\left\{C_1 : 4,\ C_2 : \frac{11}{2}\right\}$$

[28]:
```
sol = sol.subs(Constants)
sol
```

[28]:
$$y(t) = \left(\frac{11t}{2} + 4\right) e^{-t} - \frac{3\sin(t)}{2}$$

[29]:
```
aw = {y(0):4, y(t).diff(t).subs(t, 0): 0}
aw
```

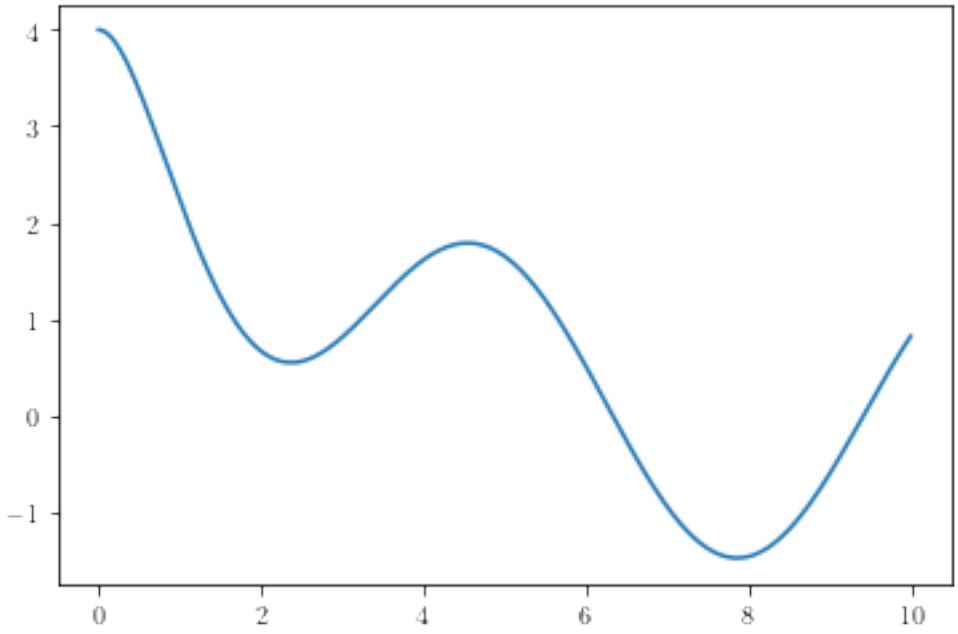
[29]:
$$\left\{y(0) : 4,\ \left.\frac{d}{dt}y(t)\right|_{t=0} : 0\right\}$$

[30]:
```
sol2 = dsolve(dgl2, y(t), ics=aw)
sol2
```

[30]:
$$y(t) = \left(\frac{11t}{2} + 4\right) e^{-t} - \frac{3\sin(t)}{2}$$

[31]:
```
tn = np.linspace(0, 10, 500)
yn = lambdify(t, sol.rhs)
fig = plt.figure(4)
ax = fig.gca()
ax.plot(tn, yn(tn))
```

[31]:
```
[<matplotlib.lines.Line2D at 0x7f47ec91d940>]
```



$$y''(t) = -2y'(t) - y(t) - 3\cos(t)$$

ist äquivalent zum System erster Ordnung

$$u_1'(t) = u_2(t) \tag{1}$$
$$u_2'(t) = -2u_2(t) - u_1(t) - 3\cos(t) \tag{2}$$

In Vektor-Schreibweise

$$u'(t) = \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & -2 \end{pmatrix} u(t) - \begin{pmatrix} 0 \\ 3\cos(t) \end{pmatrix}}_{f(u)}$$

[32]:
```
A = Matrix(2,2,[0, 1, -1, -2])
A
```

[32]:
$$\begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}$$

[33]:
```
u0 = Matrix(2, 1, [4, 0])
u0
```

[33]:
$$\begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

[34]:
```
g = lambda t: Matrix(2, 1, [0, -3*cos(t)])
g(t)
```

[34]:
$$\begin{bmatrix} 0 \\ -3\cos(t) \end{bmatrix}$$

[35]:
```
MexpA = lambda t: simplify((t*A).exp())
MexpA(t)
```

[35]:
$$\begin{bmatrix} (t+1)\,e^{-t} & te^{-t} \\ -te^{-t} & (1-t)\,e^{-t} \end{bmatrix}$$

[36]:
```
integrand = simplify(MexpA(t-tau)*g(tau))
integrand
```

[36]:
$$\begin{bmatrix} -3\,(t-\tau)\,e^{-t+\tau}\cos(\tau) \\ -3\,(-t+\tau+1)\,e^{-t+\tau}\cos(\tau) \end{bmatrix}$$

[37]:
```
u = MexpA(t)*u0 + integrate(integrand, (tau, 0, t))
u
```

[37]:
$$\begin{bmatrix} \frac{3te^{-t}}{2} + 4\,(t+1)\,e^{-t} - \frac{3\sin(t)}{2} \\ -\frac{11te^{-t}}{2} - \frac{3\cos(t)}{2} + \frac{3e^{-t}}{2} \end{bmatrix}$$

[38]:
```
simplify(u[0] - sol2.rhs)
```

[38]:
$$0$$

1.2 Pendelgleichung (physikalisches Pendel)

$$\alpha''(t) = -\sin(\alpha(t))$$

äquivalent zu System 1. Ordnung

$$y_1(t) := \alpha(t),\ y_2(t) := \frac{d}{dt}\alpha(t)$$

$$\begin{aligned} y_1'(t) &= y_2(t) \\ y_2'(t) &= -\sin(y_1(t)) \end{aligned}$$

y_1: Winkel

y_2: Winkelgeschwindigkeit

[39]:
```
dgl = Eq(y(t).diff(t,2), -sin(y(t)))
dgl
```

[39]:

$$\frac{d^2}{dt^2}y(t) = -\sin(y(t))$$

[40]:
```
#dsolve(dgl)
```

1.2.1 Phasenraum und Trajektorien

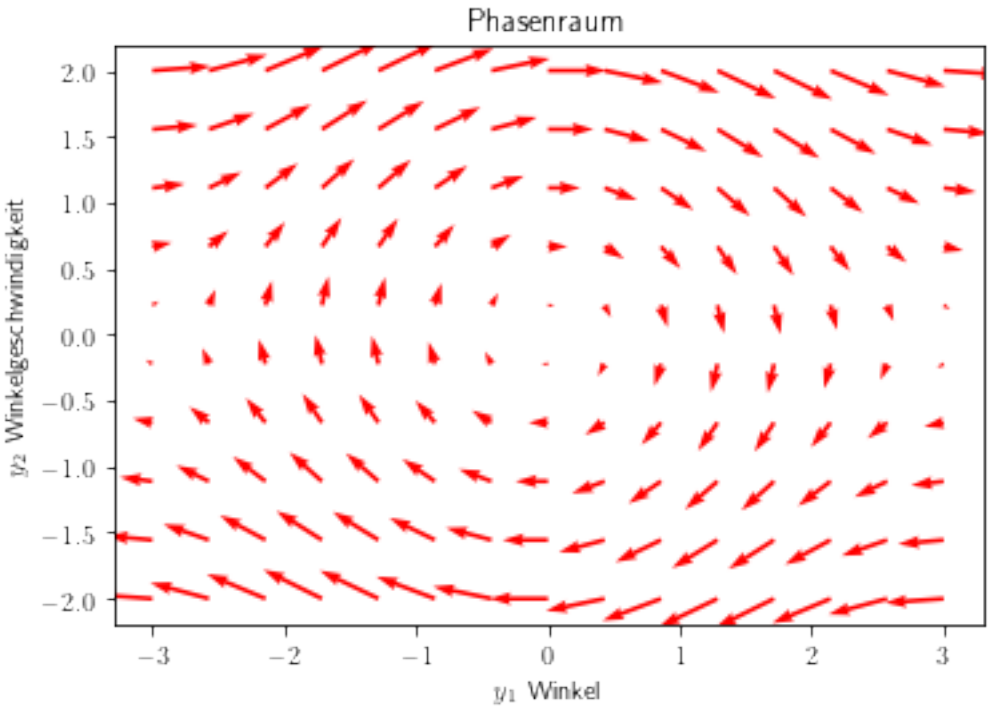
vgl. Ana II Kurzskript Kap. 20 http://www.math.uni-duesseldorf.de/~internet/ana2_19/vorlesung.pdf

[41]:
```
# Achtung die Reihenfolge von y und t ist hier im Vergleich zu obiger Ana II␣
↪Vorlesung vertauscht
def f(y, t):
    y1 = y[0]
    y2 = y[1]
    return y2, -np.sin(y1)
```

[42]:
```
y1 = np.linspace(-3.0, 3.0, 15)
y2 = np.linspace(-2.0, 2.0, 10)
Y1, Y2 = np.meshgrid(y1, y2)
t0 = 0
```

[43]:
```
U, V = f([Y1, Y2], t0)
```

[44]:
```
fig = plt.figure(5)
ax = fig.add_subplot(111)
ax.set_title('Phasenraum')
ax.quiver(Y1, Y2, U, V, angles='xy', color='r')
ax.set_aspect('equal');
ax.set_xlabel('$y_1$ Winkel');
ax.set_ylabel('$y_2$ Winkelgeschwindigkeit' );
```

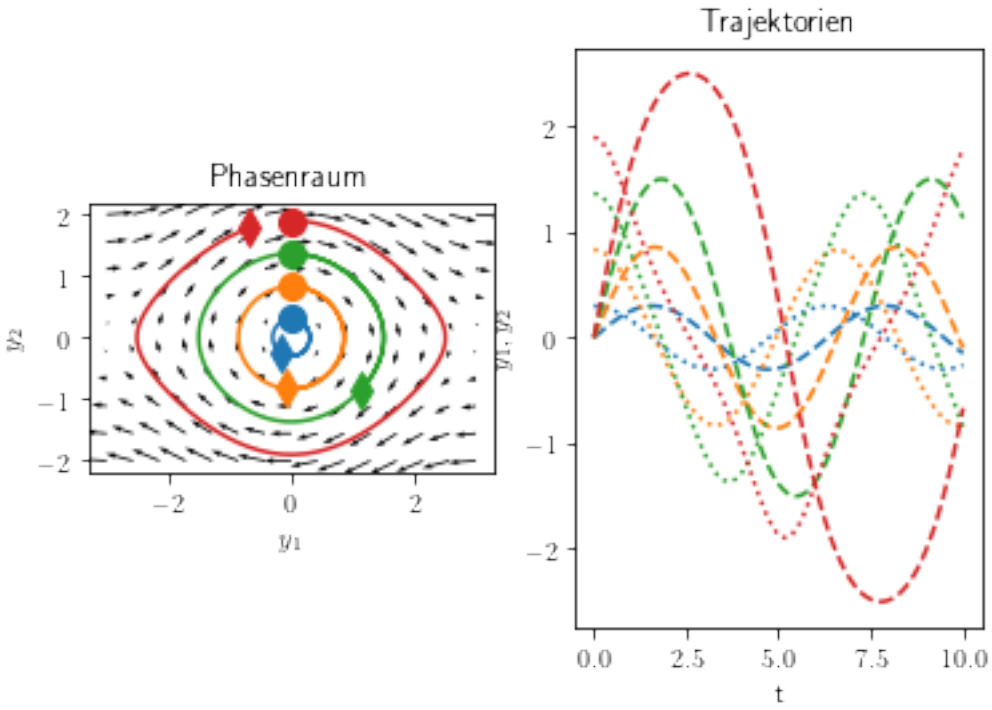


[45]:
```
from scipy.integrate import odeint  # numerische Integration (numerisches␣
 ↪Lösungsverfahren für DGL)
tn = np.linspace(0,1,10)
y0 = [0.0, 1] # Anfangswert
yn = odeint(f, y0, tn) #berechnet Näherungslösung y für jeden Wert in tn
yn
```

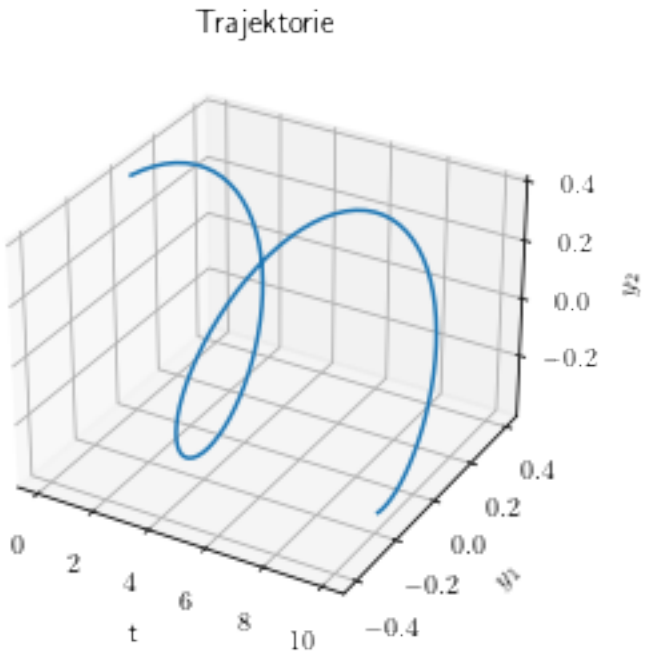
[45]:
```
array([[0.        , 1.        ],
       [0.11088278, 0.99383983],
       [0.22040221, 0.97550971],
       [0.32722793, 0.94544907],
       [0.43009301, 0.90435276],
       [0.52781962, 0.85312046],
       [0.61933847, 0.79279585],
       [0.70370131, 0.72450326],
       [0.78008653, 0.64938844],
       [0.84779864, 0.56856905]])
```

[46]:
```
fig = plt.figure(6)

ax1 = fig.add_subplot(121)
ax1.quiver(Y1, Y2, U, V, angles='xy', color='k')
ax1.set_aspect('equal');
ax1.set_xlabel('$y_1$');
```

```
ax1.set_ylabel('$y_2$');
ax1.set_title('Phasenraum')

ax2 = fig.add_subplot(122)
ax2.set_title('Trajektorien')

tn = np.linspace(0, 10, 100)
for y20 in np.linspace(0.3, 1.9, 4):
    y0 = [0.0, y20]

    yn = odeint(f, y0, tn)

    line, = ax1.plot(yn[:,0], yn[:,1], '-') # Phasenporträt (Lösung im␣
 ↪Phasenraum)
    ax1.plot(yn[0,0], yn[0,1], 'o', color=line.get_color(), markersize=10)  #␣
 ↪Startwert
    ax1.plot(yn[-1,0], yn[-1,1], 'd', color=line.get_color(), markersize=10) #␣
 ↪Wert zum Endzeitpunkt

    ax2.plot(tn, yn[:,0], '--', color=line.get_color()) # y_1
    ax2.plot(tn, yn[:,1], ':', color=line.get_color())  # y_2
    ax2.set_xlabel('t')
    ax2.set_ylabel('$y_1, y_2$')
```



[47]:
```
from mpl_toolkits.mplot3d import Axes3D

y0 = [0.0,0.4]
yn = odeint(f,y0,tn)

fig = plt.figure(7)
ax = fig.add_subplot(111,projection='3d')
ax.plot3D(tn,yn[:,0],yn[:,1])
ax.set_xlabel('t')
ax.set_ylabel('$y_1$')
ax.set_zlabel('$y_2$')
ax.set_title('Trajektorie')
plt.show()
```



[48]:
```
ax.view_init(0, 0)
plt.show()
```

[49]:
```
ax.view_init(0, 90)
plt.show()
```

[50]:
```
ax.view_init(90,-90)
plt.show()
```

falls der Winkel klein ist, ist

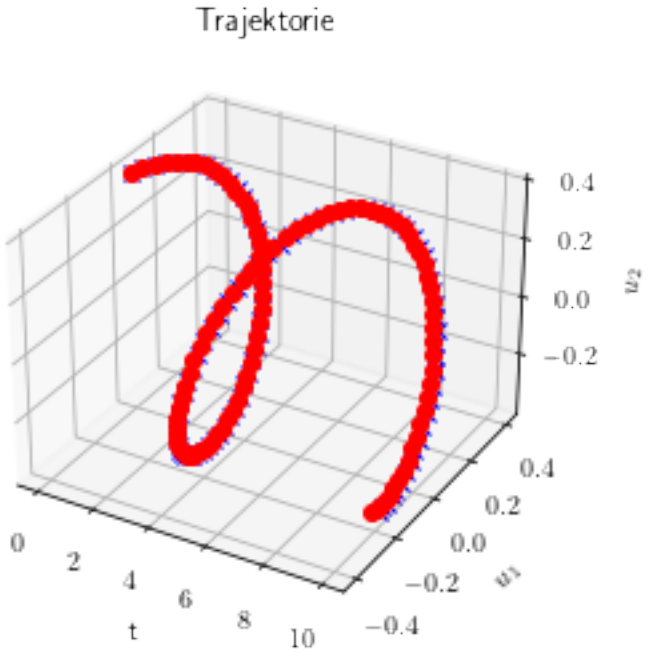
$$\sin(\alpha) \approx \alpha$$

und wir erhalten als Approximation den harmonischen Oszillator

$$\dot{u}_1(t) = u_2(t) \tag{3}$$
$$\dot{u}_2(t) = -u_1(t)) \tag{4}$$

[51]:
```
dgl3 = Eq(y(t).diff(t,2), -y(t))
awe = {y(0):y0[0], y(t).diff(t).subs(t, 0):y0[1]}
awe
sol3 = dsolve(dgl3, y(t), ics=awe).rhs
```

[52]:
```
fig = plt.figure(8)
ax = fig.add_subplot(111, projection='3d')
ax.plot3D(tn,yn[:,0], yn[:,1],'bx')
ax.plot3D(tn,lambdify(t, sol3)(tn), lambdify(t, sol3.diff(t))(tn), 'ro')
ax.set_xlabel('t')
ax.set_ylabel('$u_1$')
ax.set_zlabel('$u_2$')
ax.set_title('Trajektorie')
plt.show()
```



1.2.2 Gekoppeltes Pendel (Kleinwinkelnäherung)

$$y''(t) = w(t) - y(t) + \cos(2t) \tag{5}$$
$$w''(t) = y(t) - w(t) \tag{6}$$

Äquivalent zu einem System erster Ordnung

mit

$$x_0 = y, x_1 = y', x_2 = w, x_3 = w'$$

ergibt

$$x_0'(t) = x_1(t) \tag{7}$$
$$x_1'(t) = x_2(t) - x_0(t) + \cos(2t) \tag{8}$$
$$x_2'(t) = x_3(t) \tag{9}$$
$$x_3'(t) = x_0(t) - x_2(t) \tag{10}$$

Anfangswert

$$y(0) = 1, y'(0) = 0, w(0) = 1, w'(0) = 0$$

werden zu

$$x_0(0) = 1, x_1(0) = 0, x_2(0) = 1, x_3(0) = 0$$

[53]:
```
y = Function('y')
w = Function('w')
t, tau = symbols('t tau', real=True)
```

[54]:
```
dgl = [Eq( y(t).diff(t,2), w(t)-y(t)+cos(2*t)), \
       Eq( w(t).diff(t,2), y(t)-w(t))]
dgl
```

[54]:
$$\left[\frac{d^2}{dt^2}y(t) = w(t) - y(t) + \cos(2t),\ \frac{d^2}{dt^2}w(t) = -w(t) + y(t)\right]$$

[55]:
```
sol = dsolve(dgl, [y(t), w(t)])
sol
```

[55]:
$$\left[y(t) = C_1 + C_2 t - \frac{\sqrt{2}C_3 \sin\left(\sqrt{2}t\right)}{2} - \frac{\sqrt{2}C_4 \cos\left(\sqrt{2}t\right)}{2} - \frac{\sqrt{2}\sin(2t)\sin\left(\sqrt{2}t\right)\cos\left(\sqrt{2}t\right)}{4} - \frac{\sqrt{2}\sin\left(\sqrt{2}t\right)\int\left(\right.\right.$$

[56]:
```
sol[0].rhs
```

[56]:
$$C_1 + C_2 t - \frac{\sqrt{2}C_3 \sin\left(\sqrt{2}t\right)}{2} - \frac{\sqrt{2}C_4 \cos\left(\sqrt{2}t\right)}{2} - \frac{\sqrt{2}\sin(2t)\sin\left(\sqrt{2}t\right)\cos\left(\sqrt{2}t\right)}{4} - \frac{\sqrt{2}\sin\left(\sqrt{2}t\right)\int\left(\frac{\sin^2\left(\sqrt{2}t\right)\cos(2t)}{\cos\left(\sqrt{2}t\right)} - \frac{\cos(2t)}{\cos\left(\sqrt{2}t\right)}\right)dt}{4} - \frac{\cos(2t)\cos^2\left(\sqrt{2}t\right)}{4} - \frac{\cos(2t)}{8}$$

[57]:
```
simplify(sol[0].rhs)
```

[57]:

$$C_1 + C_2 t - \frac{\sqrt{2}C_3 \sin\left(\sqrt{2}t\right)}{2} - \frac{\sqrt{2}C_4 \cos\left(\sqrt{2}t\right)}{2} + \frac{\sqrt{2}\left(\sin\left(2t\right)\cos\left(\sqrt{2}t\right) - \frac{\sqrt{2}\sin\left(\sqrt{2}t\right)\cos\left(2t\right)}{2}\right)\sin\left(\sqrt{2}t\right)}{4} - \frac{\sqrt{2}\left(\cos\left(2t\left(1-\sqrt{2}\right)\right) - \cos\left(2t\left(1+\sqrt{2}\right)\right)\right)}{16} - \frac{\cos\left(2t\right)\cos^2\left(\sqrt{2}t\right)}{4} - \frac{\cos\left(2t\right)}{8}$$

[58]:
```
eqn = [Eq(simplify(sol[0].rhs).subs(t, 0), 1),
       Eq(simplify(sol[0].rhs).diff(t).subs(t, 0), 0),
       Eq(simplify(sol[1].rhs).subs(t, 0), 1),
       Eq(simplify(sol[1].rhs).diff(t).subs(t, 0), 0)]
eqn
```

[58]:
$$\left[C_1 - \frac{\sqrt{2}C_4}{2} - \frac{3}{8} = 1,\ C_2 - C_3 = 0,\ C_1 + \frac{\sqrt{2}C_4}{2} + \frac{1}{8} = 1,\ C_2 + C_3 = 0\right]$$

[59]:
```
cs = solve(eqn)
cs
```

[59]:
$$\left\{C_1 : \frac{9}{8},\ C_2 : 0,\ C_3 : 0,\ C_4 : -\frac{\sqrt{2}}{4}\right\}$$

[60]:
```
ysol = sol[0].rhs.simplify().subs(cs)
ysol
```

[60]:
$$\frac{\sqrt{2}\left(\sin\left(2t\right)\cos\left(\sqrt{2}t\right) - \frac{\sqrt{2}\sin\left(\sqrt{2}t\right)\cos\left(2t\right)}{2}\right)\sin\left(\sqrt{2}t\right)}{4} - \frac{\sqrt{2}\left(\cos\left(2t\left(1-\sqrt{2}\right)\right) - \cos\left(2t\left(1+\sqrt{2}\right)\right)\right)}{16} - \frac{\cos\left(2t\right)\cos^2\left(\sqrt{2}t\right)}{4} - \frac{\cos\left(2t\right)}{8} + \frac{\cos\left(\sqrt{2}t\right)}{4} + \frac{9}{8}$$

Alternative Lösung

[61]:
```
A = Matrix(4, 4, [0, 1, 0, 0,\
                  -1, 0, 1, 0,\
                  0, 0, 0, 1,\
                  1, 0, -1, 0])
A
```

[61]:
$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix}$$

[62]:
```
MexpA = lambda t: simplify((t*A).exp())
MexpA(t)
```

[62]:

$$\begin{bmatrix} \frac{\cos\left(\sqrt{2}t\right)}{2}+\frac{1}{2} & \frac{t}{2}+\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{4} & \frac{1}{2}-\frac{\cos\left(\sqrt{2}t\right)}{2} & \frac{t}{2}-\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{4} \\ -\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{2} & \frac{\cos\left(\sqrt{2}t\right)}{2}+\frac{1}{2} & \frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{2} & \frac{1}{2}-\frac{\cos\left(\sqrt{2}t\right)}{2} \\ \frac{1}{2}-\frac{\cos\left(\sqrt{2}t\right)}{2} & \frac{t}{2}-\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{4} & \frac{\cos\left(\sqrt{2}t\right)}{2}+\frac{1}{2} & \frac{t}{2}+\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{4} \\ \frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{2} & \frac{1}{2}-\frac{\cos\left(\sqrt{2}t\right)}{2} & -\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{2} & \frac{\cos\left(\sqrt{2}t\right)}{2}+\frac{1}{2} \end{bmatrix}$$

Wir nutzen die Variation der Konstantenformel

[63]:
```
u0 = Matrix(4, 1, [1, 0, 1, 0])
f = lambda t:  Matrix(4, 1, [0, cos(2*t), 0, 0])
u0, f(tau)
```

[63]:
$$\left(\begin{bmatrix}1\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\\cos\left(2\tau\right)\\0\\0\end{bmatrix}\right)$$

[64]:
```
integrand = simplify(MexpA(t-tau)*f(tau))
integrand
```

[64]:
$$\begin{bmatrix} \frac{\left(2t-2\tau+\sqrt{2}\sin\left(\sqrt{2}(t-\tau)\right)\right)\cos\left(2\tau\right)}{4} \\ \frac{\left(\cos\left(\sqrt{2}(t-\tau)\right)+1\right)\cos\left(2\tau\right)}{2} \\ \frac{\left(2t-2\tau-\sqrt{2}\sin\left(\sqrt{2}(t-\tau)\right)\right)\cos\left(2\tau\right)}{4} \\ \frac{\left(1-\cos\left(\sqrt{2}(t-\tau)\right)\right)\cos\left(2\tau\right)}{2} \end{bmatrix}$$

[65]:
```
u = MexpA(t)*u0 + integrate(integrand, (tau, 0, t))
```

[66]:
```
u
```

[66]:
$$\begin{bmatrix} \frac{\sin^2\left(t\right)\sin^2\left(\sqrt{2}t\right)}{2}+\frac{\sin^2\left(t\right)\cos^2\left(\sqrt{2}t\right)}{2}-\frac{\sin^2\left(\sqrt{2}t\right)}{4}-\frac{\cos\left(2t\right)}{8}-\frac{\cos^2\left(\sqrt{2}t\right)}{4}+\frac{\cos\left(\sqrt{2}t\right)}{4}+\frac{9}{8} \\ \sin\left(t\right)\sin^2\left(\sqrt{2}t\right)\cos\left(t\right)+\sin\left(t\right)\cos\left(t\right)\cos^2\left(\sqrt{2}t\right)+\frac{\sin\left(2t\right)}{4}-\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{4} \\ -\frac{\sin^2\left(t\right)\sin^2\left(\sqrt{2}t\right)}{2}-\frac{\sin^2\left(t\right)\cos^2\left(\sqrt{2}t\right)}{2}+\frac{\sin^2\left(\sqrt{2}t\right)}{4}-\frac{\cos\left(2t\right)}{8}+\frac{\cos^2\left(\sqrt{2}t\right)}{4}-\frac{\cos\left(\sqrt{2}t\right)}{4}+\frac{9}{8} \\ -\sin\left(t\right)\sin^2\left(\sqrt{2}t\right)\cos\left(t\right)-\sin\left(t\right)\cos\left(t\right)\cos^2\left(\sqrt{2}t\right)+\frac{\sin\left(2t\right)}{4}+\frac{\sqrt{2}\sin\left(\sqrt{2}t\right)}{4} \end{bmatrix}$$

[67]:
```
simplify(u.diff(t) - A*u - f(t)) , u.subs(t,0)-u0 # Test
```

[67]:
$$\left(\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\end{bmatrix}\right)$$

[68]:
```
simplify(u[0]-ysol)
```

[68]:
$$0$$

1.3 Bernoulli DGL

[69]:
```
x = symbols('x')
dgl = Eq(y(x).diff(x), sqrt(x*y(x)))
dgl
```

[69]:
$$\frac{d}{dx}y(x) = \sqrt{xy(x)}$$

[70]:
```
sol = dsolve(dgl)
sol
```

[70]:
$$\left[y(x) = \frac{C_1^2}{4} - \frac{C_1\sqrt{x^3}}{3} + \frac{x^3}{9},\ y(x) = \frac{C_1^2}{4} + \frac{C_1\sqrt{x^3}}{3} + \frac{x^3}{9}\right]$$

[71]:
```
C1 = S('C1')
```

[72]:
```
sol[0].rhs.subs(C1,1)
```

[72]:
$$\frac{x^3}{9} - \frac{\sqrt{x^3}}{3} + \frac{1}{4}$$

Welche ist die “richtige” ?