lektion13

January 29, 2018

Table of Contents


```
In [1]: from IPython.display import display
        import numpy as np
        import matplotlib.pyplot as plt
        from sympy import *
        init_printing()
        %matplotlib inline
        x,t,n = symbols('x t n')
        y = Function('y')
        u = Function('u')
```

1 Lektion 13

1.1 Pendelgleichung

$$\ddot{y}(t) = -\sin(y(t))$$

aequivalent dazu

$$\dot{y}_1(t) = y_2(t) \tag{1}$$
$$\dot{y}_2(t) = -\sin(y_1(t)) \tag{2}$$

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

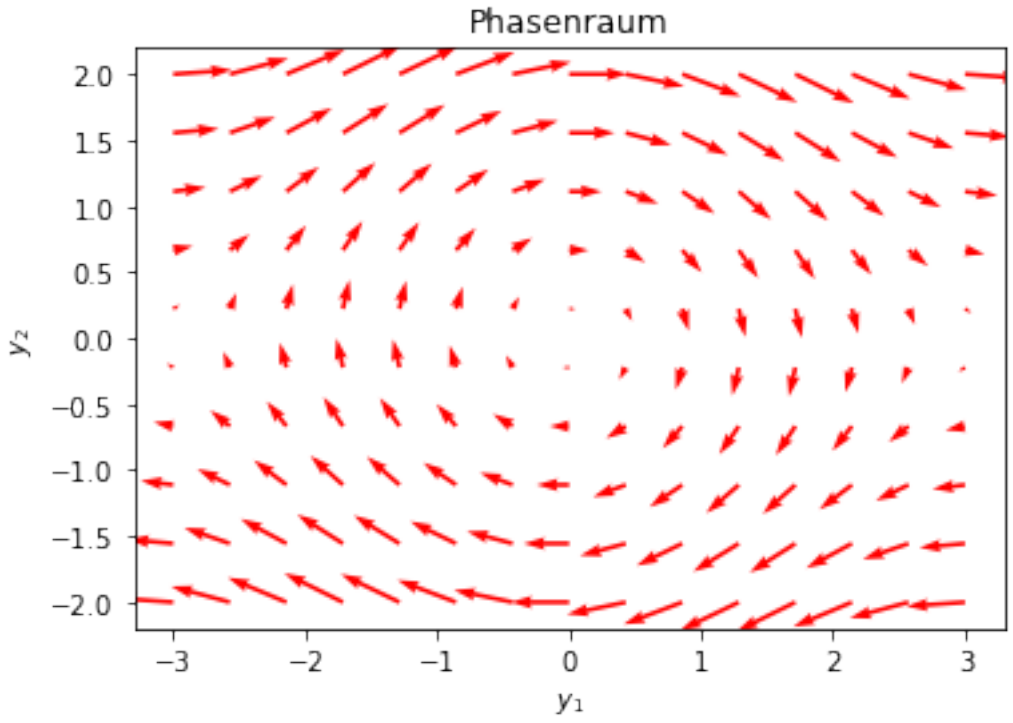
Out[2]:

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

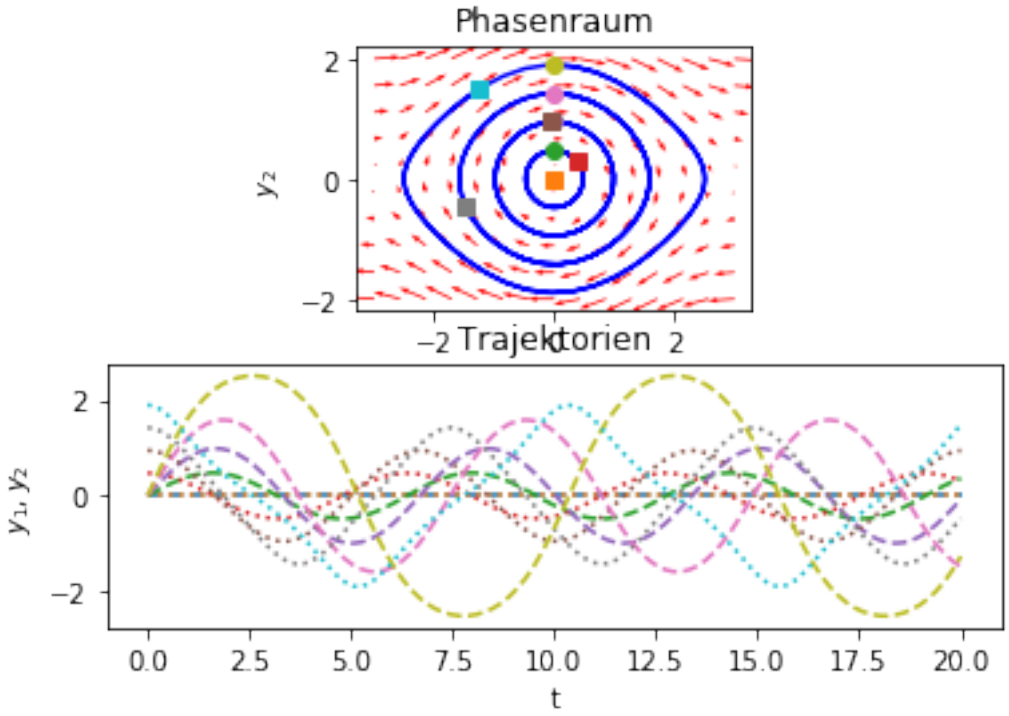
```
In [3]: ##dsolve(dgl)
```

1.1.1 Phasenraum und Trajektorien

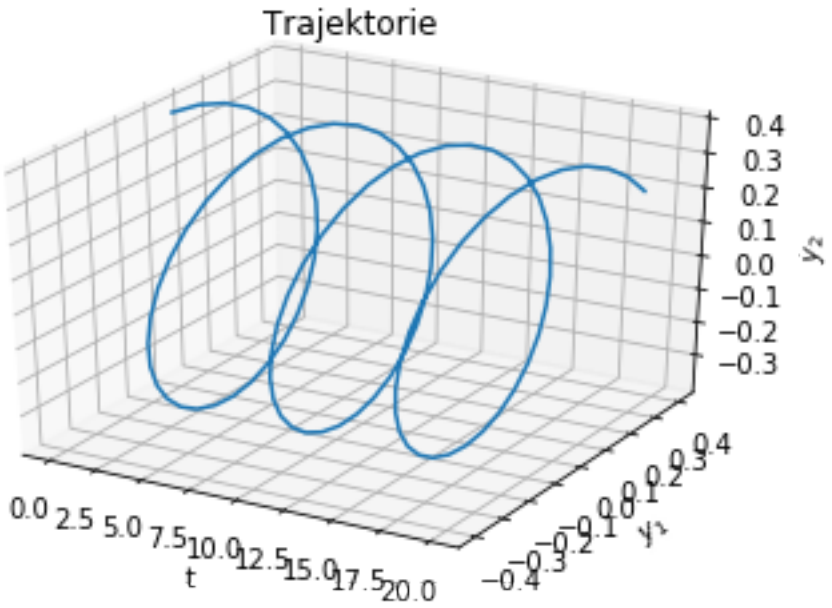
```
In [4]: def f(y,t):
            y1 = y[0]
            y2 = y[1]
            return y2,-np.sin(y1)

In [5]: y1 = np.linspace(-3.0,3.0,15)
        y2 = np.linspace(-2.0,2.0,10)
        Y1,Y2 = np.meshgrid(y1,y2)
        ts=0

In [6]: U,V = np.zeros_like(Y1) , np.zeros_like(Y1)

In [7]: U,V = f([Y1,Y2],ts)

In [8]: fig = plt.figure()
        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('$y_1$');
        ax.set_ylabel('$y_2$');
```



In [9]:
```
from scipy.integrate import odeint

fig = plt.figure()

ax1 = fig.add_subplot(211)
ax1.quiver(Y1,Y2,U,V,angles='xy',color='r')
ax1.set_aspect('equal');
ax1.set_xlabel('$y_1$');
ax1.set_ylabel('$y_2$');
ax1.set_title('Phasenraum')

ax2 = fig.add_subplot(212)
ax2.set_title('Trajektorien')

tvec = np.linspace(0,20,100)
for y20 in np.linspace(0,1.9,5):
    y0 = [0.0,y20]
    ys = odeint(f,y0,tvec)
    ax1.plot(ys[:,0],ys[:,1],'b-') # Loesung im Phasenraum
    ax1.plot(ys[0,0],ys[0,1],'o')  # Startwert
    ax1.plot(ys[-1,0],ys[-1,1],'s') # Wert zum Endzeitpunkt

    ax2.plot(tvec,ys[:,0],'--')
    ax2.plot(tvec,ys[:,1],':')
    ax2.set_xlabel('t')
    ax2.set_ylabel('$y_1, y_2$')
```



```
In [10]: from mpl_toolkits.mplot3d import Axes3D

         y0 = [0.0,0.4]
         ys = odeint(f,y0,tvec)

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




```
In [11]: ax.view_init(0, 0)
         plt.show()

In [12]: ax.view_init(0, 90)
         plt.show()

In [13]: ax.view_init(90,-90)
         plt.show()
```

falls der Winkel klein ist, ist

$$\sin(y) \approx y$$

und wir erhalten als Approximation

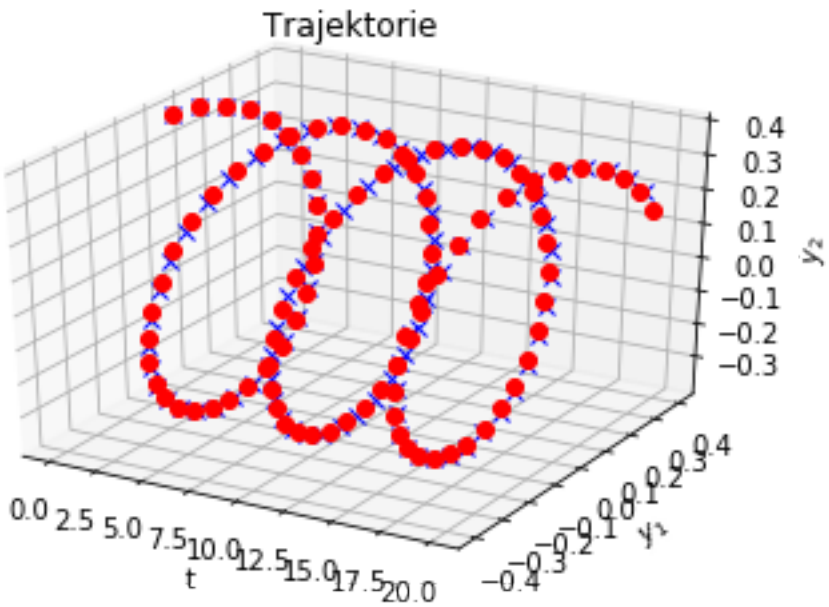
$$\dot{u}_1(t) = u_2(t) \tag{3}$$
$$\dot{u}_2(t) = -u_1(t)) \tag{4}$$

```
In [14]: dgl = Eq(u(t).diff(t,2),-u(t))
         sol = dsolve(dgl)
         yu = sol.rhs.subs(solve([(sol.rhs).subs(t,0)-y0[0],(sol.rhs.diff(t)).subs(t,0)-y0[1]]))
         yu
```

Out[14]:

$$0.4 \sin(t)$$

```
In [15]: fig = plt.figure()
         ax = fig.add_subplot(111,projection='3d')
         ax.plot3D(tvec,ys[:,0],ys[:,1],'bx')
         ax.plot3D(tvec,lambdify(t,yu)(tvec),lambdify(t,yu.diff(t))(tvec),'ro')
         ax.set_xlabel('t')
         ax.set_ylabel('$y_1$')
         ax.set_zlabel('$y_2$')
         ax.set_title('Trajektorie')
         plt.show()
```



1.2 Bessel Differentialgleichung

```
In [16]: dgl = Eq(y(x).diff(x,2), -1/x * y(x).diff(x)+ 1/x**2*y(x) + 4*y(x))
         dgl
```

Out[16]:

$$\frac{d^2}{dx^2}y(x) = 4y(x) - \frac{1}{x}\frac{d}{dx}y(x) + \frac{1}{x^2}y(x)$$

In [17]:
```
##dsolve(dgl) # funktioniert nicht
```

In [18]:
```
a = symbols('a:18')
N = len(a)
```

In [19]:
```
ys = sum([a[j]*x**j for j in range(N)])
ys
```

Out[19]:

$$a_0 + a_1 x + a_{10}x^{10} + a_{11}x^{11} + a_{12}x^{12} + a_{13}x^{13} + a_{14}x^{14} + a_{15}x^{15} + a_{16}x^{16} + a_{17}x^{17} + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a$$

In [20]:
```
gl = dgl.subs(y(x),ys).doit()
gl
```

Out[20]:

$$2\left(45a_{10}x^8 + 55a_{11}x^9 + 66a_{12}x^{10} + 78a_{13}x^{11} + 91a_{14}x^{12} + 105a_{15}x^{13} + 120a_{16}x^{14} + 136a_{17}x^{15} + a_2 + 3a_3 x + 6a_4 x^2 +$$

In [21]:
```
gl1 = (gl.lhs - gl.rhs).expand()
gl1
```

Out[21]:

$$-4a_0 - \frac{a_0}{x^2} - 4a_1 x - 4a_{10}x^{10} + 99a_{10}x^8 - 4a_{11}x^{11} + 120a_{11}x^9 - 4a_{12}x^{12} + 143a_{12}x^{10} - 4a_{13}x^{13} + 168a_{13}x^{11} - 4a_{14}x^{14} + 1$$

In [22]:
```
#gl1.as_poly(t).all_coeffs()
# klappt nicht
```

In [23]:
```
gl1.coeff(x**(-2))
```

Out[23]:

$$-a_0$$

In [24]:
```
gl1.coeff(x**(-1))
```

Out[24]:

$$0$$

In [25]:
```
gl1.coeff(x,-1)
```

$$6$$

Out[25]:

$$0$$

In [26]:
```
gl1.coeff(x,1) # Konstanter Term
```

Out[26]:

$$-4a_1 + 8a_3$$

In [27]:
```
gls = []
for j in range(N+1):
    glg = Eq(gl1.coeff(x,j-2),0)
    if glg != True:
        gls.append(glg)
gls
```

Out[27]:

$$[-a_0 = 0, \quad -4a_0 + 3a_2 = 0, \quad -4a_1 + 8a_3 = 0, \quad -4a_2 + 15a_4 = 0, \quad -4a_3 + 24a_5 = 0, \quad -4a_4 + 35a_6 = 0, \quad -4a$$

In [28]:
```
solve(gls[:-1])
```

Out[28]:

$$\left\{a_0 : 0, \quad a_1 : 2880a_9, \quad a_{10} : 0, \quad a_{11} : \frac{a_9}{30}, \quad a_{12} : 0, \quad a_{13} : \frac{a_9}{1260}, \quad a_{14} : 0, \quad a_{15} : \frac{a_9}{70560}, \quad a_{16} : 0, \quad a_{17} : \frac{a_9}{5080320},\right.$$

In [29]:
```
var = list(a).copy()
del var[1]
var
```

Out[29]:

$$[a_0, \quad a_2, \quad a_3, \quad a_4, \quad a_5, \quad a_6, \quad a_7, \quad a_8, \quad a_9, \quad a_{10}, \quad a_{11}, \quad a_{12}, \quad a_{13}, \quad a_{14}, \quad a_{15}, \quad a_{16}, \quad a_{17}]$$

In [30]:
```
Lsg = solve(gls[:-1],var)
Lsg
```

Out[30]:

$$\left\{a_0 : 0, \quad a_{10} : 0, \quad a_{11} : \frac{a_1}{86400}, \quad a_{12} : 0, \quad a_{13} : \frac{a_1}{3628800}, \quad a_{14} : 0, \quad a_{15} : \frac{a_1}{203212800}, \quad a_{16} : 0, \quad a_{17} : \frac{a_1}{1463132160}\right.$$

In [31]:
```
Lsg[a[1]] = a[1]
Lsg
```

Out[31]:

$$\left\{a_0 : 0, \quad a_1 : a_1, \quad a_{10} : 0, \quad a_{11} : \frac{a_1}{86400}, \quad a_{12} : 0, \quad a_{13} : \frac{a_1}{3628800}, \quad a_{14} : 0, \quad a_{15} : \frac{a_1}{203212800}, \quad a_{16} : 0, \quad a_{17} : \frac{a_1}{146}\right.$$

In [32]:
```
q = [Lsg[a[2*j+1]]/Lsg[a[2*j+3]] for j in range(int(N/2)-2)]
q
```

Out[32]:

$$[2, \quad 6, \quad 12, \quad 20, \quad 30, \quad 42, \quad 56]$$

In [33]:
```
q = [Lsg[a[2*j+1]]/Lsg[a[2*j+3]]/(j+2) for j in range(int(N/2)-2)]
q
```

Out[33]:

$$[1, \quad 2, \quad 3, \quad 4, \quad 5, \quad 6, \quad 7]$$

Also ist

$$\frac{a_{2j+1}}{a_{2j+3}} = (j+1)(j+2)$$

Test

In [34]:
```
q = [Lsg[a[2*j+1]]/Lsg[a[2*j+3]] - (j+1)*(j+2) for j in range(int(N/2)-2)]
q
```

Out[34]:

$$[0, \quad 0, \quad 0, \quad 0, \quad 0, \quad 0, \quad 0]$$

und damit

$$a_{2n+1} = \prod_{j=0}^{n-1} \frac{1}{(j+1)(j+2)} a_1 = \frac{a_1}{(n)!(n+1)!}$$

In [35]:
```
for j in range(int(N/2)-2):
    display((Lsg[a[2*j+1]], a[1]/factorial(j)/factorial(j+1)))
```

$$(a_1, \quad a_1)$$

$$\left(\frac{a_1}{2}, \quad \frac{a_1}{2}\right)$$

$$\left(\frac{a_1}{12}, \quad \frac{a_1}{12}\right)$$

$$\left(\frac{a_1}{144}, \quad \frac{a_1}{144}\right)$$

$$\left(\frac{a_1}{2880}, \quad \frac{a_1}{2880}\right)$$

$$\left(\frac{a_1}{86400}, \quad \frac{a_1}{86400}\right)$$

$$\left(\frac{a_1}{3628800}, \quad \frac{a_1}{3628800}\right)$$

In [36]:
```
yR = Sum(x**(2*n+1)/factorial(n)/factorial(n+1),(n,0,oo))
yR
```

Out[36]:

$$\sum_{n=0}^{\infty} \frac{x^{2n+1}}{n!\,(n+1)!}$$

In [37]:
```
u = yR.doit()
u
```

Out[37]:

$$I_1\left(2x\right)$$

In [38]:
```
srepr(u)
```

Out[38]:
```
"besseli(Integer(1), Mul(Integer(2), Symbol('x')))"
```

In [39]:
```
?besseli
```

In [40]:
```
tmp  = dgl.subs(y(x),u).doit()
tmp
```

Out[40]:

$$3I_1\left(2x\right) + I_3\left(2x\right) = 4I_1\left(2x\right) - \frac{1}{x}\left(I_0\left(2x\right) + I_2\left(2x\right)\right) + \frac{I_1\left(2x\right)}{x^2}$$

In [41]:
```
simplify(tmp.lhs - tmp.rhs)
```

Out[41]:

$$0$$

1.3 Bernoulli DGL

In [42]:
```
dgl = Eq(y(x).diff(x),sqrt(x*y(x)))
dgl
```

Out[42]:

$$\frac{d}{dx}y(x) = \sqrt{xy(x)}$$

```
In [43]: sol = dsolve(dgl)
         sol
```

Out[43]:

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

```
In [44]: C1 = sol[0].atoms(Symbol).difference(dgl.atoms(Symbol)).pop()
```

```
In [45]: sol[0].rhs.subs(C1,1)
```

Out[45]:

$$\frac{x^3}{9} - \frac{\sqrt{x^3}}{3} + \frac{1}{4}$$

```
In [46]: y1  = sol[0].rhs
         y2  = sol[1].rhs
```

```
In [47]: gl1 = Eq(y1.subs(x,1),Rational(1,50)) #Startwert
         gl1
```

Out[47]:

$$\frac{C_1^2}{4} - \frac{C_1}{3} + \frac{1}{9} = \frac{1}{50}$$

```
In [48]: K1 = solve(gl1)
         K1
```

Out[48]:

$$\left[-\frac{\sqrt{2}}{5} + \frac{2}{3}, \quad \frac{\sqrt{2}}{5} + \frac{2}{3}\right]$$

```
In [49]: gl2 = Eq(y2.subs(x,1),Rational(1,50))
         gl2
```

Out[49]:

$$\frac{C_1^2}{4} + \frac{C_1}{3} + \frac{1}{9} = \frac{1}{50}$$

```
In [50]: K2 = solve(gl2)
         K2
```

Out[50]:

$$\left[-\frac{2}{3} - \frac{\sqrt{2}}{5}, \quad -\frac{2}{3} + \frac{\sqrt{2}}{5}\right]$$

In [51]:
```
phi1 = y1.subs(C1,K1[0])
phi1.expand()
```

Out[51]:

$$\frac{x^3}{9} - \frac{2\sqrt{x^3}}{9} + \frac{\sqrt{2}\sqrt{x^3}}{15} - \frac{\sqrt{2}}{15} + \frac{59}{450}$$

In [52]:
```
phi2 = y1.subs(C1,K1[1])
phi2.expand()
```

Out[52]:

$$\frac{x^3}{9} - \frac{2\sqrt{x^3}}{9} - \frac{\sqrt{2}\sqrt{x^3}}{15} + \frac{\sqrt{2}}{15} + \frac{59}{450}$$

In [53]:
```
phi3 = y2.subs(C1,K2[0])
phi3.expand()
```

Out[53]:

$$\frac{x^3}{9} - \frac{2\sqrt{x^3}}{9} - \frac{\sqrt{2}\sqrt{x^3}}{15} + \frac{\sqrt{2}}{15} + \frac{59}{450}$$

In [54]:
```
(phi2-phi3).simplify()
```

Out[54]:

$$0$$

In [55]:
```
phi4 = y2.subs(C1,K2[1])
phi4.expand()
```

Out[55]:

$$\frac{x^3}{9} - \frac{2\sqrt{x^3}}{9} + \frac{\sqrt{2}\sqrt{x^3}}{15} - \frac{\sqrt{2}}{15} + \frac{59}{450}$$

In [56]:
```
(phi1-phi4).simplify()
```

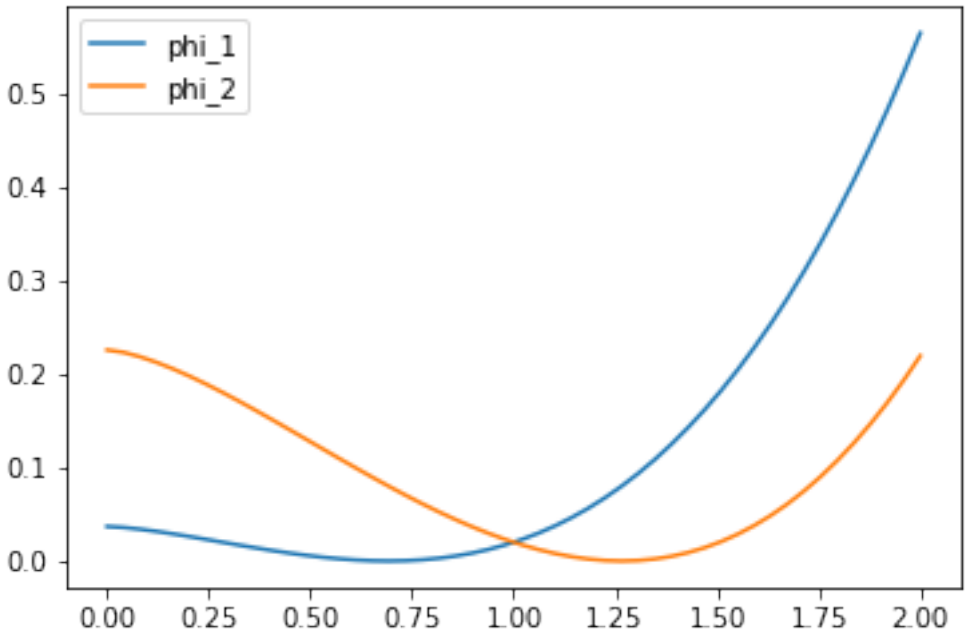
Out[56]:

$$0$$

In [57]:
```
xn = np.linspace(0,2)
phi1n = lambdify(x,phi1)
phi2n = lambdify(x,phi2)
```

In [58]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(xn,phi1n(xn),label='phi_1')
ax.plot(xn,phi2n(xn),label='phi_2')
ax.legend(loc='upper left')
```

Out[58]: <matplotlib.legend.Legend at 0x7ff8b11bb470>

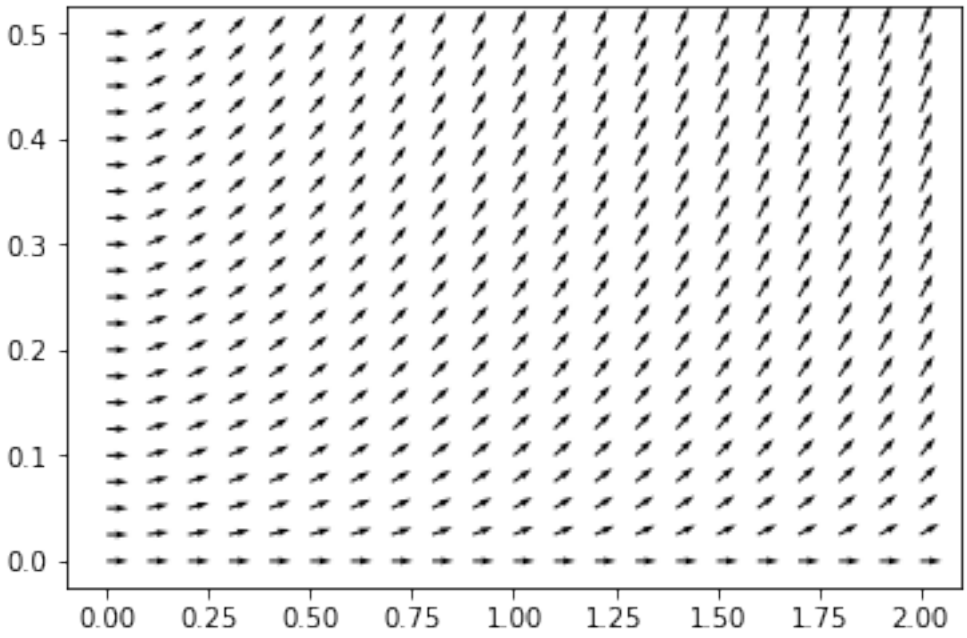


In [59]:
```
vn = lambdify( (x,y(x)), dgl.rhs)
```

In [60]:
```
xg = np.linspace(0,2,21)
yg = np.linspace(0,0.5,21)
X,Y = np.meshgrid(xg,yg)
U = np.ones_like(X)
V = vn(X,Y).astype(float)
```

In [61]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.quiver(X,Y,U,V,angles='xy')
```

Out[61]: <matplotlib.quiver.Quiver at 0x7ff8b121e390>




```
In [62]: ax.plot(xn,phi1n(xn),label='phi_1')
         ax.plot(xn,phi2n(xn),label='phi_2')
         ax.legend(loc='upper left')
```

```
Out[62]: <matplotlib.legend.Legend at 0x7ff8b119a518>
```

Welche ist die “richtige” ?