

Übungen zur Klassischen Theoretischen Physik I WS 2022/23

PROF. DR. CARSTEN ROCKSTUHL

Blatt 1

MITCHELL WHITTAM, DAVID DAMS, BENEDIKT ZERULLA

Abgabe: 09.11.2022

1. Bahnkurven in 2D

6 Punkte

Wir betrachten einen Massenpunkt in einer Ebene, dessen Bewegung durch die folgenden Gleichungen beschrieben wird:

$$x(t) = x_0 \cos(\omega_x t), \quad y(t) = y_0 \cos(\omega_y t + \phi_y) \quad (1)$$

Die Bahnkurve eines solchen Massepunkts beschreibt eine sogenannte *Lissajous-Figur*.

- (a) [2 Punkte] Mit Jupyter Notebook, plotten Sie die Bahnkurve für $x_0 = y_0 = 1$.

In allen Plots sollte mindestens eine vollständige Periode $0 \leq t \leq \max(\frac{2\pi}{\omega_x}, \frac{2\pi}{\omega_y})$ zu sehen sein.

Die Lösungen für 1(a) sind in der Datei blatt01_muster.ipynb zu finden.

- (b) [4 Punkte] Die Geschwindigkeit des Massepunkts bestimmt man über die zeitliche Ableitung der Bahnkurve und erhält für $x_0 = y_0$, $\omega_x = \omega_y$ und $\phi_y = \pi/2$

$$\mathbf{v}(t) = \frac{d\mathbf{r}(t)}{dt} = \begin{pmatrix} -x_0\omega_x \sin(\omega_x t) \\ -y_0\omega_x \sin(\omega_y t + \phi_y) \end{pmatrix} = -x_0\omega_x \begin{pmatrix} \sin(\omega_x t) \\ \cos(\omega_x t) \end{pmatrix}.$$

Die Beschleunigung $\mathbf{a}(t)$ ist gegeben durch

$$\mathbf{a}(t) = \frac{d\mathbf{v}(t)}{dt} = \frac{d^2\mathbf{r}(t)}{dt^2} = \begin{pmatrix} -x_0\omega_x^2 \cos(\omega_x t) \\ -y_0\omega_y^2 \cos(\omega_y t + \phi_y) \end{pmatrix} = -x_0\omega_x^2 \begin{pmatrix} \cos(\omega_x t) \\ -\sin(\omega_x t) \end{pmatrix}.$$

Die Länge eines Vektors $\mathbf{b}$ ist gegeben durch seinen Betrag (die euklidische Norm¹)

$$b = |\mathbf{b}| = \sqrt{\sum_i b_i^2}.$$

Damit erhält man

$$r(t) = \sqrt{x_0^2 \cos^2(\omega_x t) + y_0^2 \cos^2(\omega_y t + \phi_y)} = \sqrt{x_0^2 [\cos^2(\omega_x t) + \sin^2(\omega_x t)]} = |x_0|.$$

Die Bahnkurve ist eine Kreisbahn mit konstantem Abstand $r = |x_0|$ vom Ursprung.

$$v(t) = \sqrt{x_0^2 \omega_x^2 \sin^2(\omega_x t) + y_0^2 \omega_y^2 \sin^2(\omega_y t + \phi_y)} = \sqrt{x_0^2 \omega_x^2 [\sin^2(\omega_x t) + \cos^2(\omega_x t)]} = |x_0 \omega_x|,$$

Der Massenpunkt bewegt sich mit konstanter Geschwindigkeit $v = r|\omega_x|$. Und auch der Betrag der Beschleunigung,

$$a(t) = \sqrt{x_0^2 \omega_x^4 \cos^2(\omega_x t) + y_0^2 \omega_y^4 \cos^2(\omega_y t + \phi_y)} = \sqrt{x_0^2 \omega_x^4 [\cos^2(\omega_x t) + \sin^2(\omega_x t)]} = |x_0| \omega_x^2,$$

ist zeitlich konstant.

¹auch als Standardnorm, oder 2-Norm bezeichnet.

2. Bahnkurve in 3D

5 Punkte

Gegeben ist die Bahnkurve

$$\mathbf{r}(t) = \begin{pmatrix} r \cos(\omega t) \\ r \sin(\omega t) \\ ht \end{pmatrix} \quad \text{mit} \quad r, \omega, h = \text{const.} \quad (2)$$

- (a) [4 Punkte] In drei Dimensionen berechnen sich Geschwindigkeit $\mathbf{v}(t)$ und Beschleunigung $\mathbf{a}(t)$ wieder durch die Ableitungen der Bahnkurve nach der Zeit

$$\mathbf{v}(t) = \frac{d\mathbf{r}(t)}{dt} = \begin{pmatrix} -r\omega \sin(\omega t) \\ r\omega \cos(\omega t) \\ h \end{pmatrix} \quad \text{und} \quad \mathbf{a}(t) = \frac{d^2\mathbf{r}(t)}{dt^2} = \frac{d\mathbf{v}(t)}{dt} = \begin{pmatrix} -r\omega^2 \cos(\omega t) \\ -r\omega^2 \sin(\omega t) \\ 0 \end{pmatrix} \quad (3)$$

Für die entsprechenden Längen erhält man damit

$$r(t) = \sqrt{r^2 + h^2 t^2}, \quad v(t) = \sqrt{r^2 \omega^2 + h^2} \quad \text{und} \quad a(t) = |r|\omega^2. \quad (4)$$

- (b) [1 Punkt] Die Bahnkurve ist eine Schraubenlinie.

Die Lösung für 2(b) ist in der Datei blatt01_muster.ipynb zu finden.

3. Kronecker-Delta und Levi-Civita-Pseudotensor

6 Punkte

In der Regel verwenden wir die Einsteinsche Summenkonvention bei der implizit über identische Indizes summiert wird, und lassen entsprechend das Summenzeichen weg. Zum besseren Verständnis werden wir bei der Lösung dieser Aufgabe die Summen weitgehend ausschreiben.

(a) [2 Punkte] Es gilt

$$\epsilon_{ijk}\epsilon_{ilm} \equiv \sum_i \epsilon_{ijk}\epsilon_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl}.$$

Zuerst überzeugen wir uns von der Gültigkeit dieser Relation in dem wir einige Beispiele betrachten:

$$\begin{aligned} \epsilon_{122}\epsilon_{122} + \epsilon_{222}\epsilon_{222} + \epsilon_{322}\epsilon_{322} &= 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 0 & \delta_{22}\delta_{22} - \delta_{22}\delta_{22} &= 1 - 1 = 0 \\ \epsilon_{123}\epsilon_{132} + \epsilon_{223}\epsilon_{232} + \epsilon_{323}\epsilon_{332} &= 1 \cdot (-1) + 0 \cdot 0 + 0 \cdot 0 = -1 & \delta_{23}\delta_{32} - \delta_{22}\delta_{33} &= 0 - 1 = -1 \\ \epsilon_{123}\epsilon_{123} + \epsilon_{223}\epsilon_{223} + \epsilon_{323}\epsilon_{323} &= 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 = 1 & \delta_{22}\delta_{33} - \delta_{23}\delta_{32} &= 1 - 0 = 1 \\ \epsilon_{113}\epsilon_{112} + \epsilon_{213}\epsilon_{212} + \epsilon_{313}\epsilon_{312} &= 0 \cdot 0 + (-1) \cdot 0 + 0 \cdot (-1) = 0 & \delta_{11}\delta_{32} - \delta_{11}\delta_{32} &= 0 - 0 = 0 \\ & \dots & & \end{aligned}$$

Zeigen Sie mit Hilfe dieser Relation, dass weiter gilt $\epsilon_{ijk}\epsilon_{ijl} = 2\delta_{kl}$ und $\epsilon_{ijk}\epsilon_{ijk} = 6$.

$$\begin{aligned} \epsilon_{ijk}\epsilon_{ijl} &\equiv \sum_{i,j} \epsilon_{ijk}\epsilon_{ijl} = \sum_{j=1}^3 \delta_{jj}\delta_{kl} - \delta_{jl}\delta_{kj} = \left(\delta_{kl} \sum_{j=1}^3 \delta_{jj} \right) - \left(\sum_{j=1}^3 \delta_{jl}\delta_{kj} \right) \\ &= \left(\delta_{kl} \sum_{j=1}^3 1 \right) - \delta_{kl} = \delta_{kl}(3 - 1) = 2\delta_{kl} \end{aligned}$$

$$\epsilon_{ijk}\epsilon_{ijk} \equiv \sum_{i,j,k} \epsilon_{ijk}\epsilon_{ijk} = \sum_{k=1}^3 \sum_{i,j} \epsilon_{ijk}\epsilon_{ijk} = \sum_{k=1}^3 2\delta_{kk} = \sum_{k=1}^3 2 = 6$$

(b) [1 Punkt] Zeigen Sie $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \epsilon_{ijk}a_i b_j c_k$.

$$\begin{aligned} \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) &= \sum_i a_i (\mathbf{b} \times \mathbf{c})_i = \sum_i a_i \left(\sum_{jk} \epsilon_{jki} b_j c_k \right) \\ &= \sum_{ijk} \epsilon_{jki} a_i b_j c_k = \sum_{ijk} \epsilon_{ijk} a_i b_j c_k = \epsilon_{ijk} a_i b_j c_k \end{aligned}$$

(c) [1 Punkt] Beweisen Sie die Identität $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$.

Die Identität folgt durch zyklisches Vertauschen der Indizes des Levi-Civita-Tensors $\epsilon_{ijk} = \epsilon_{jki} = \epsilon_{kij}$:

$$\begin{aligned} \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) &= \sum_{ijk} \epsilon_{ijk} a_i b_j c_k \\ \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) &= \sum_{ijk} \epsilon_{jki} b_j c_k a_i = \sum_{ijk} \epsilon_{ijk} a_i b_j c_k \\ \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b}) &= \sum_{ijk} \epsilon_{kij} c_k a_i b_j = \sum_{ijk} \epsilon_{ijk} a_i b_j c_k \end{aligned}$$

(d) [2 Punkte] Beweisen Sie die “bac-cab”-Formel $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$.

$$\begin{aligned} [\mathbf{a} \times (\mathbf{b} \times \mathbf{c})]_i &= \sum_{jk} \epsilon_{ijk} a_j (\mathbf{b} \times \mathbf{c})_k = \sum_{jklm} \epsilon_{ijk} \epsilon_{klm} a_j b_l c_m \\ &= \sum_{jklm} \epsilon_{kij} \epsilon_{klm} a_j b_l c_m = \sum_{jlm} (\delta_{il}\delta_{jm} - \delta_{im}\delta_{jl}) a_j b_l c_m \\ &= \sum_j a_j b_i c_j - a_j b_j c_i = b_i(\mathbf{a} \cdot \mathbf{c}) - c_i(\mathbf{a} \cdot \mathbf{b}) \end{aligned}$$

Jupyter_Muster01

May 10, 2023

```
[1]: # was hier passiert, ist nicht so wichtig, sondern soll nur eine drehbare ↵  
      ↪ 3D-Ansicht ermöglichen  
      %matplotlib widget  
      import matplotlib.pyplot as plt  
      import numpy as np
```

```
[2]: ## es ist oft eine gute idee, teile eines programms, die man immer wieder ↵  
      ↪ benutzt, in funktionen auszulagern  
      def curve3D( t, r, w, h ):  
          """A 3D trajectory for a helix  
  
          Arguments:  
          t (array): parameter for the trajectory  
  
          Returns:  
          A list of three elements, the (x,y,z)-points of the trajectory  
          """  
          return [r*np.cos(w * t), r*np.sin(w * t), h*t]  
  
      def curve2D( t, omega_x, omega_y, phi_y ):  
          """A 2D trajectory for a Lissajous figure  
  
          Arguments:  
          t (array): parameter for the trajectory  
  
          Returns:  
          A list of two elements, the (x,y)-points of the trajectory  
          """  
          return [np.cos(omega_x * t), np.cos(omega_y * t + phi_y)]  
  
      def plotter_2d( xdata, ydata ):  
          fig = plt.figure()  
          ax = plt.axes()  
          ax.plot( xdata, ydata, 'o' )  
          ax.annotate( 'start', (xdata[0], ydata[0]))  
          ax.annotate( 'end', (xdata[-1], ydata[-1]))  
          ax.set_xlabel('X')
```

```

ax.set_ylabel('Y')
plt.show()

def plotter_3d( xdata, ydata, zdata ):
    fig = plt.figure()
    ax = plt.axes(projection='3d')
    ax.plot( xdata, ydata, zdata, 'o' )
    ax.text( xdata[0], ydata[0], zdata[0], 'start', 'x' )
    ax.text( xdata[-1], ydata[-1], zdata[-1], 'end', 'x' )
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    plt.show()

```

```

[3]: ## "linspace" erzeugt ein array mit 100 einträgen von 0 bis 10 in jeweils
      ↳ gleichem abstand
      parameter_range = np.linspace(0, 10, 100)

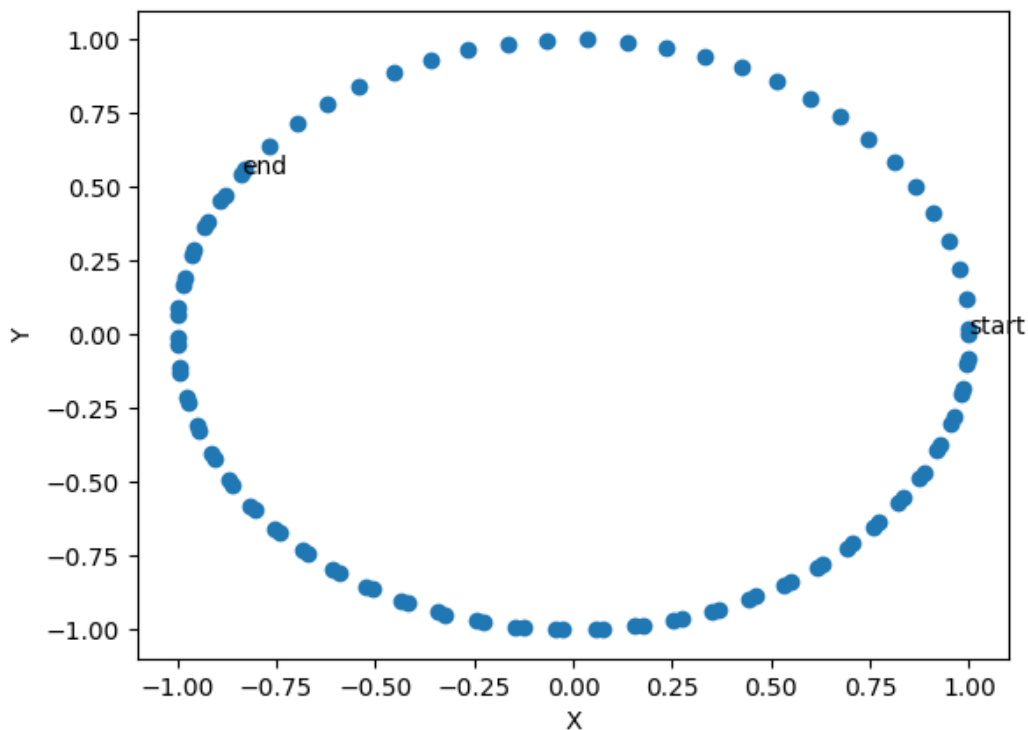
```

Lösung 1(a)(i)

```

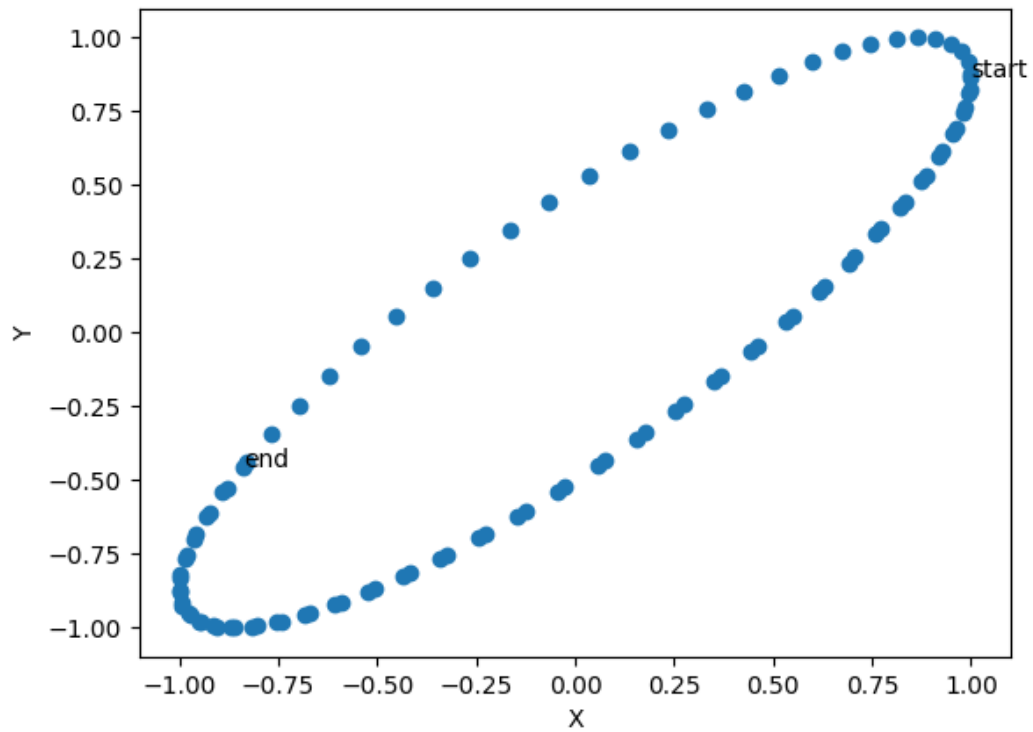
[5]: plotter_2d( *curve2D(parameter_range, 1, 1, np.pi/2) )

```



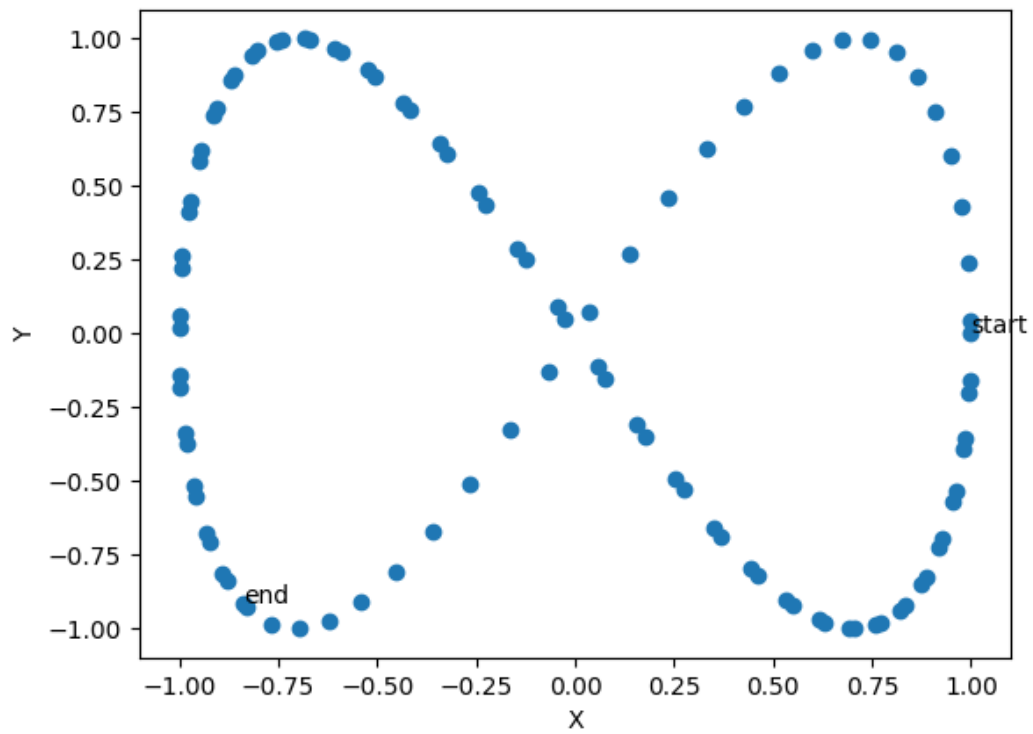
Lösung 1(a)(ii)

```
[6]: plotter_2d( *curve2D(parameter_range, 1, 1, np.pi/6) )
```



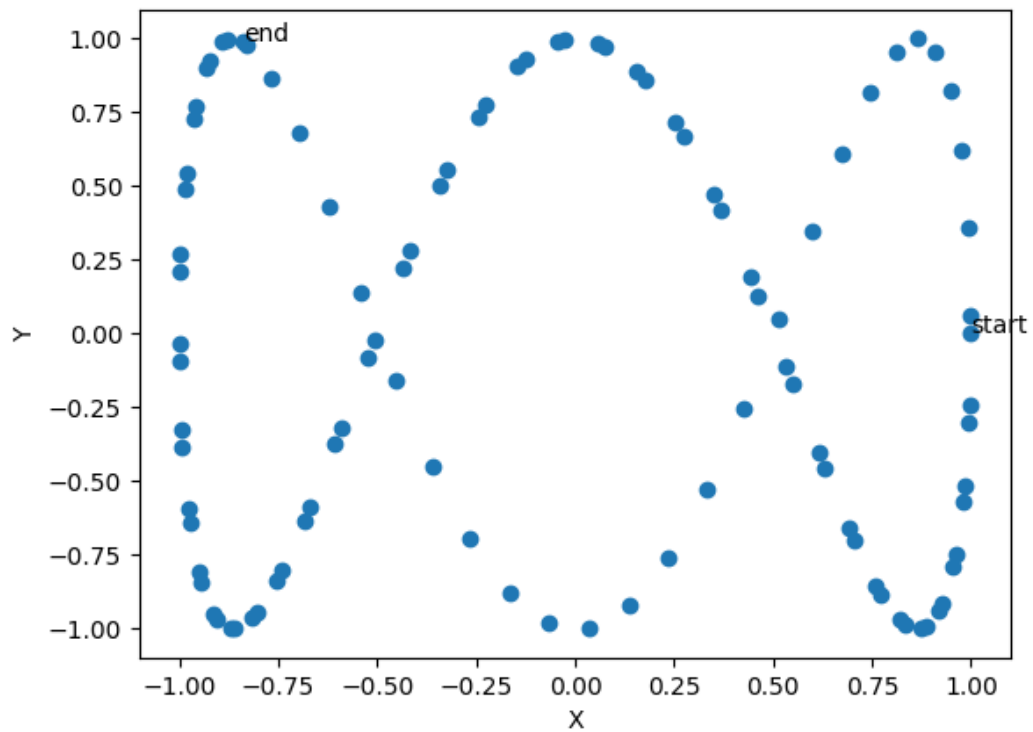
Lösung 1(a)(iii)

```
[7]: plotter_2d( *curve2D(parameter_range, 1, 2, np.pi/2) )
```



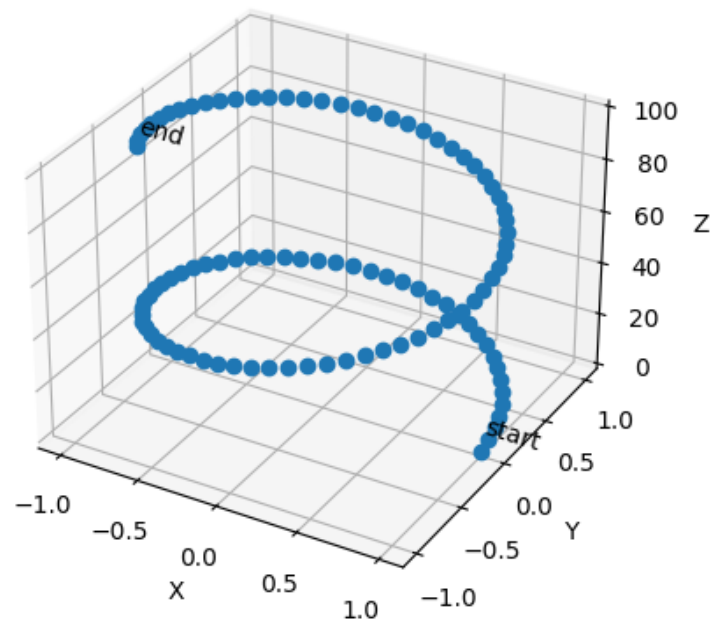
Lösung 1(a)(iv)

```
[8]: plotter_2d( *curve2D(parameter_range, 1, 3, np.pi/2) )
```



Lösung 2(b)

```
[5]: ## curve3D gibt eine liste mit drei elementen zurück, die wir separat
      ↪ "auspacken" können
      plotter_3d( *curve3D(parameter_range, 1, 1, 10) )
```



[]: