

Sage Notebook zur Aufgabe 1.5 Freier Fall mit Reibung

der Vorlesung

Theoretische Mechanik und Mathematische Methoden

Uni Leipzig, Sommersemester 2019

Autor: Jürgen Vollmer (2019)

Lizenz: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

see: <https://creativecommons.org/licenses/by-sa/4.0/deed.en>

Sage ist ein OpenSource Projekt, das viele Methoden der computerbasierten Mathematik und Computer-Algebra in einem Python-basierten Notebook anbietet.

Dokumentation und Informationen zur **Installation** findet man auf

<https://sagemath.org>

Eine hervorragende Einführung in das Arbeiten mit Sage bietet das Buch

Paul Zimmermann, u.a.: "Computational Mathematics with SageMath"

<http://sagebook.gforge.inria.fr/english.html>

Allgemeine Definitionen, Variablen, Konstanten

Pfad und Stammname für Abbildungen

Bitte den Pfad editiert und die Kommentarzeichen vor den "save_image()"-Befehlen entfernen, um die erstellten Dateien zu speichern.

```
In [1]: # in order to print the picture:  
# uncomment this line and insert a proper path instead of <PATH>  
# baseName = '<PATH>/prob01_5__freier_Fall__'
```

Pakete laden für Plotten und Numerik

```
In [2]: import scipy; from scipy import integrate  
import numpy as np
```

Differentialgleichung als Funktion von t

```
In [3]: t = var('t')  
def dW_dt(w, t=0) :  
    return -1 - w
```

Trajektorien plotten

2 Trajektorien berechnen

```
In [4]: t_array = srange(0, 4, 0.1 )

w_ini = -2
Solution1 = integrate.odeint( dW_dt, w_ini, t_array )
W1 = Solution1.T

w_ini = 1
Solution2 = integrate.odeint( dW_dt, w_ini, t_array )
W2 = Solution2.T
```

Trajektorien plotten und Theoriekurven darüberlegen

```
In [6]: p = line( zip(np.array(t_array), W1[0]), color='red', thickness=2 )
p += plot( -1+(1-2)*exp(-x), x, -.4, 4.5, color='green', thickness=5, linestyle=':' )
p += text( r'$w(\tau_0) = -2$', (3,1.2), fontsize=16, horizontal_alignment='left', color='red' )

p += line( zip(np.array(t_array), W2[0]), color='blue', thickness=3 )
p += plot( -1+(1+1)*exp(-x), x, -.4, 4.5, color='cyan', thickness=5, linestyle=':' )
p += text( r'$w(\tau_0) = 1$', (3,0.6), fontsize=16, horizontal_alignment='left', color='blue' )

p.axes_labels( [r'$\tau - \tau_0$', r'$w$'] )
p.show( gridlines=True, figsize=[6,4] )

p.save_image(baseName+'Trajektorien.svg', figsize=[6,4])
```

