

prob03_2__gedaempftes_Pendel

Sage Notebook zur Aufgabe

3.2 Das gedämpfte mathematische Pendel

der Vorlesung

Theoretische Physik 1. Mechanik
Uni Leipzig, Wintersemester 2018/19
Autor: Jürgen Vollmer (2018)
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.

```
baseName = 'XXX--bitte editieren--XXX/2018W_Mechanik/Uebungen  
/Sage/prob03_2__gedaempftes_Pendel__'
```

Pakete laden für Plotten und Numerik

```
import scipy; from scipy import integrate  
import numpy as np
```

Differentialgleichung

Parameter:

- Dämpfung γ
- Frequenz ist in Zeitskala absorbiert

```
gamma = var('gamma')
```

Definition als Funktion von t

```
t = var('t')  
def dX_dt(X, t=0) :  
    return [ X[1], -gamma * X[1] - sin( X[0] ) ]
```

Trajektorien plotten

Dämpfung festlegen

```
gamma = 0.15
```

Trajektorien berechnen

```
t = srange(0, 30, 0.05 )
V = VectorSpace(RR, 2)

# Vektor mit Anfangsbedingungen
ICs = [ V([-pi, 1e-6]), V([-pi, 0.9]), V([-pi, 1.813]), V([-pi, 1.814]),
        V([-pi, 2.3]), V([-pi, 2.909]), V([-pi, 2.91]), V([-pi, 3.6]) ]

Trajektorien = []
for IC in ICs :
    Trajektorien.append( integrate.odeint( dX_dt, IC, t ) )
```

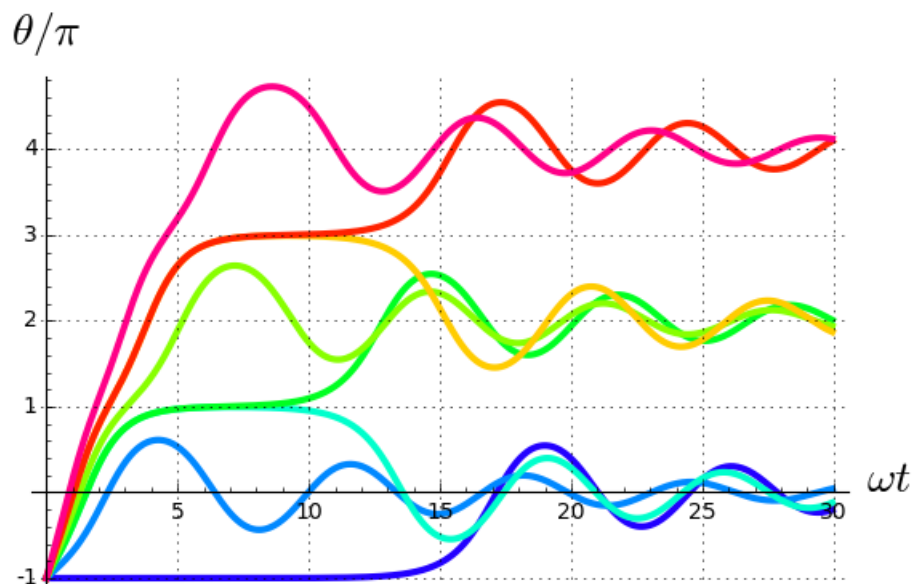
...und plotten

```
p = plot([])
p.axes_labels( [r'$\omega t$', r'$\theta$'] )

n=0
for X in Trajektorien :
    theta, dtheta = X.T
    n += 1
    p += line( zip( t, theta/pi ), color=hue(.8-float(n)/(1.8*5)), thickness=3
)

p.axes_labels( [r'$\omega t$', r'$\theta/\pi$'] )
p.axes_labels_size( 2 )
p.show( gridlines=True, figsize=[6,4] )

# p.save_image(baseName+'Trajektorien_015.svg', figsize=[6,4])
```



Phasenraumplot mit Vektorfeld

Definition des Vektorfeldes

```
def g(x,y) :  
    v = vector( dX_dt([x,y]) )  
    return v / v.norm()
```

$$\gamma = 0.15$$

```
gamma = 0.15
```

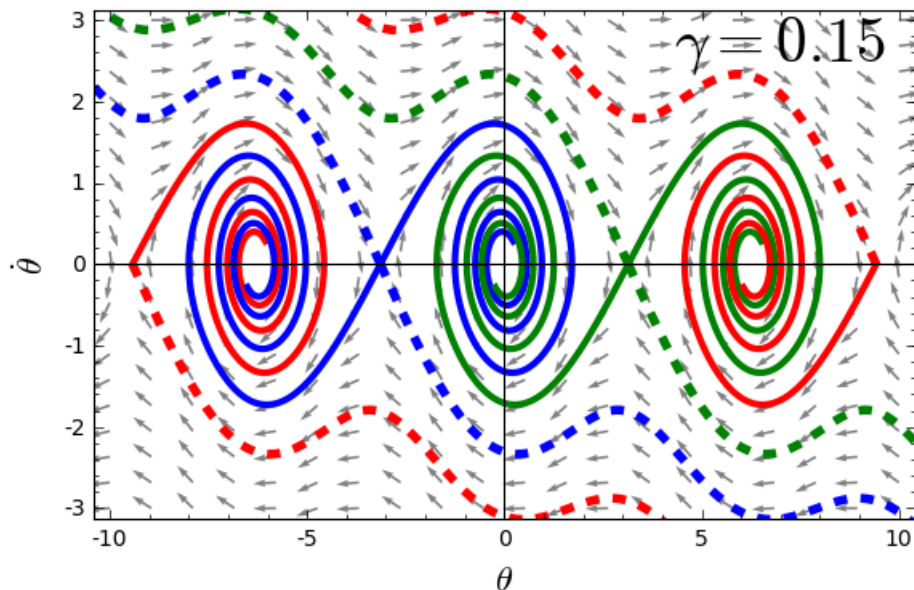
Trajektorien für Oszillationen und stabile Mannigfaltigkeiten der instabilen Fixpunkte

- IC: Anfangsbedingungen ("Initial Conditions")
- Mi: instabile Mannigfaltigkeiten der instabilen Fixpunkte
- Ms: stabile Mannigfaltigkeiten der instabilen Fixpunkte

```
V = VectorSpace(RR, 2)  
  
# instabile Mannigfaltigkeit  
ti = srange(0, 30, 0.05 )  
IC = V([-3*pi+1e-3 , 1e-3])  
Mi,dMi = integrate.odeint( dX_dt, IC, ti ).T  
  
# stabile Mannigfaltigkeit  
ts = srange(0, -20, -0.05 )  
IC = V([3*pi-1e-5 , 1e-5])  
Ms,dMs = integrate.odeint( dX_dt, IC, ts ).T
```

Graphik erstellen

```
x,y = var('x', 'y')  
q = plot_vector_field( g(x,y), (x,-10, 10), (y, -3, 3), color='gray' )  
  
q += line( zip( 0*pi+Mi, dMi), color='red' , thickness=3 )  
q += line( zip( 2*pi+Mi, dMi), color='blue' , thickness=3 )  
q += line( zip( 4*pi+Mi, dMi), color='green', thickness=3 )  
q += line( zip(-4*pi-Mi, -dMi), color='blue' , thickness=3 )  
q += line( zip(-2*pi-Mi, -dMi), color='green', thickness=3 )  
q += line( zip(      -Mi, -dMi), color='red' , thickness=3 )  
  
q += line( zip( Ms,      dMs), color='red',   thickness=4, linestyle='--' )  
q += line( zip( Ms-2*pi, dMs), color='green', thickness=4, linestyle='--' )  
q += line( zip( Ms-4*pi, dMs), color='blue',  thickness=4, linestyle='--' )  
q += line( zip(-Ms,      -dMs), color='red',   thickness=4, linestyle='--' )  
q += line( zip(-Ms+2*pi, -dMs), color='blue',  thickness=4, linestyle='--' )  
q += line( zip(-Ms+4*pi, -dMs), color='green', thickness=4, linestyle='--' )  
  
q += text( r'$\gamma=0.15$', (7,2.7), fontsize=28, color='black' )  
q.axes_labels([ r'$\theta$', r'$\dot{\theta}$' ] )  
  
q.show(xmin=-10, xmax=10, ymin=-3, ymax=3, figsize=[6,4])  
  
# q.save_image(baseName+'Phasenraum_015.svg', figsize=[6,4])
```



$$\gamma = 2$$

```
gamma = 2
```

Trajektorien für Oszillationen und stabile Mannigfaltigkeiten der instabilen Fixpunkte

- IC: Anfangsbedingungen ("Initial Conditions")
- Mi: instabile Mannigfaltigkeiten der instabilen Fixpunkte
- Ms: stabile Mannigfaltigkeiten der instabilen Fixpunkte

```
# instabile Mannigfaltigkeit von  $(-\pi, 0)$ 
ti = srange(0, 50, 0.05 )
IC = V([-3*pi+1e-3, 1e-3])
Mi,dMi = integrate.odeint( dX_dt, IC, ti ).T

# stabile Mannigfaltigkeit  $(3 \pi, 0)$ 
ts = srange(0, -20, -0.05 )
IC = V([3*pi-1e-5, 1e-5])
Ms,dMs = integrate.odeint( dX_dt, IC, ts ).T

# stabile Mannigfaltigkeit  $(2 \pi, 0)$ 
ts = srange(0, -20, -0.05 )
IC = V([2*pi+1e-5, 1e-5])
MsN,dMsN = integrate.odeint( dX_dt, IC, ts ).T
```

Graphik erstellen

```
x,y = var('x', 'y')
q = plot_vector_field( g(x,y), (x,-10, 10), (y, -1, 1), color='gray' )

q += line( zip( 0*pi+Mi, dMi), color='red' , thickness=3 )
q += line( zip( 2*pi+Mi, dMi), color='blue' , thickness=3 )
q += line( zip( 4*pi+Mi, dMi), color='green' , thickness=3 )
q += line( zip(-4*pi-Mi, -dMi), color='blue' , thickness=3 )
q += line( zip(-2*pi-Mi, -dMi), color='green' , thickness=3 )
q += line( zip( -Mi, -dMi), color='red' , thickness=3 )
```

```

q += line( zip( Ms,      dMs), color='red',   thickness=4, linestyle='--' )
q += line( zip( Ms-2*pi, dMs), color='green', thickness=4, linestyle='--' )
q += line( zip( Ms-4*pi, dMs), color='blue',  thickness=4, linestyle='--' )
q += line( zip(-Ms,     -dMs), color='red',   thickness=4, linestyle='--' )
q += line( zip(-Ms+2*pi, -dMs), color='blue', thickness=4, linestyle='--' )
q += line( zip(-Ms+4*pi, -dMs), color='green', thickness=4, linestyle='--' )

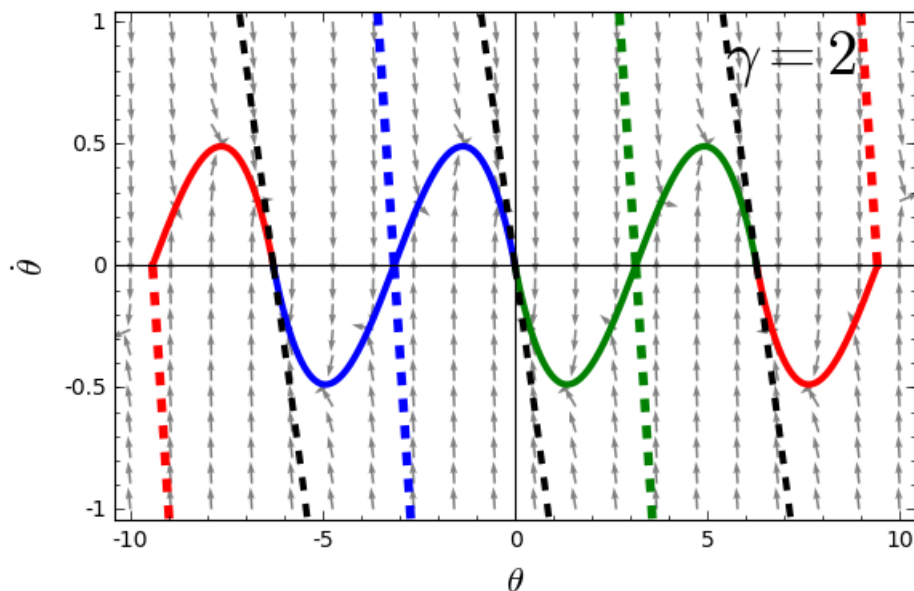
q += line( zip( MsN,      dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip( MsN-2*pi, dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip( MsN-4*pi, dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip(-MsN,     -dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip(-MsN+2*pi, -dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip(-MsN+4*pi, -dMsN), color='black', thickness=3, linestyle='--' )

q += text( r'$\gamma=2$', (7.2,0.85), fontsize=28, color='black' )
q.axes_labels([ r'$\theta$', r'$\dot{\theta}$' ] )

q.show(xmin=-10, xmax=10, ymin=-1, ymax=1, figsize=[6,4])

# q.save_image(baseName+'Phasenraum_2.svg', figsize=[6,4])

```



$\gamma = 3$

```
gamma = 3
```

Trajektorien für Oszillationen und stabile Mannigfaltigkeiten der instabilen Fixpunkte

- IC: Anfangsbedingungen ("Initial Conditions")
- Mi: instabile Mannigfaltigkeiten der instabilen Fixpunkte
- Ms: stabile Mannigfaltigkeiten der instabilen Fixpunkte

```

# instabile Mannigfaltigkeit von  $(-\pi, 0)$ 
ti = srange(0, 50, 0.05 )
IC = V([-3*pi+1e-3, 1e-3])
Mi,dMi = integrate.odeint( dX_dt, IC, ti ).T

# stabile Mannigfaltigkeit  $(3 \pi, 0)$ 

```

```

ts = srange(0, -20, -0.05 )
IC = V([3*pi-1e-5 , 1e-5])
Ms,dMs = integrate.odeint( dX_dt, IC, ts ).T

# stabile Mannigfaltigkeit  $(2 \pi, 0)$ 
ts = srange(0, -20, -0.05 )
IC = V([2*pi+1e-5 , 1e-5])
MsN,dMsN = integrate.odeint( dX_dt, IC, ts ).T

```

Graphik erstellen

```

x,y = var('x', 'y')
q = plot_vector_field( g(x,y), (x,-10, 10), (y, -1, 1), color='gray' )

q += line( zip( 0*pi+Mi, dMi), color='red' , thickness=3 )
q += line( zip( 2*pi+Mi, dMi), color='blue' , thickness=3 )
q += line( zip( 4*pi+Mi, dMi), color='green', thickness=3 )
q += line( zip(-4*pi-Mi, -dMi), color='blue' , thickness=3 )
q += line( zip(-2*pi-Mi, -dMi), color='green', thickness=3 )
q += line( zip(      -Mi, -dMi), color='red' , thickness=3 )

q += line( zip( Ms,      dMs), color='red',   thickness=4, linestyle='--' )
q += line( zip( Ms-2*pi, dMs), color='green', thickness=4, linestyle='--' )
q += line( zip( Ms-4*pi, dMs), color='blue',  thickness=4, linestyle='--' )
q += line( zip(-Ms,     -dMs), color='red',   thickness=4, linestyle='--' )
q += line( zip(-Ms+2*pi, -dMs), color='blue', thickness=4, linestyle='--' )
q += line( zip(-Ms+4*pi, -dMs), color='green', thickness=4, linestyle='--' )

q += line( zip( MsN,      dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip( MsN-2*pi, dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip( MsN-4*pi, dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip(-MsN,     -dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip(-MsN+2*pi, -dMsN), color='black', thickness=3, linestyle='--' )
q += line( zip(-MsN+4*pi, -dMsN), color='black', thickness=3, linestyle='--' )

q += text( r'\gamma=3$', (7.5,0.85), fontsize=28, color='black' )
q.axes_labels([ r'\theta$', r'\dot{\theta}$' ] )

q.show(xmin=-10, xmax=10, ymin=-1, ymax=1, figsize=[6,4])

# q.save_image(baseName+'Phasenraum_3.svg', figsize=[6,4])

```

