UNIWERSYTET PEDAGOGICZNY

im. Komisji Edukacji Narodowej w Krakowie

INSTYTUT INFORMATYKI

Kierunek: INFORMATYKA

specjalność: multimedia i technologie internetowe

Paweł Puźniak

Podstawy Modelowania i Symulacji

Projekty Zaliczeniowe

Kraków 2022

Link do projektu na GitHub: https://github.com/ppuzniak/PMIS

Stałe użyte w opracowaniu:

- C = 0,742
- delta = 0,074
- m1 = 14
- m2 = 5
- k1=k2 = 14,5

Lista zadań:

1. Obwód RC z wymuszeniem sinusoidalnym
2. Oscylator harmoniczny z tłumieniem bez wymuszenia
3. Oscylator harmoniczny tłumieniem i wymuszeniem harmonicznym
4. Oscylator harmoniczny podwójny bez tłumienia i bez wymuszenia
5. Wahadło matematyczne
6. Modele rozwoju epidemii
7. Automaty komórkowe

Zadanie 1 - Obwód RC z wymuszeniem sinusoidalnym.

1.1 Zapisz poprawnie równanie na ładunek w obwodzie. Zaklasyfikuj typ równania.

$$R \cdot i(t) + \frac{1}{C} \cdot q(t) = U(t) \tag{0.1}$$

$$\frac{dq}{dt} = i(t) \tag{0.2}$$

$$\frac{dq}{dt} + \frac{1}{RC} \cdot q(t) = \frac{U(t)}{R} \tag{0.3}$$

$$U(t) = sin(\omega \cdot t) \tag{0.4}$$

$$\frac{dq}{dt} + \frac{1}{RC} \cdot q(t) = \frac{sin(\omega \cdot t)}{R} \tag{0.5}$$

$$\frac{dq}{dt} + \alpha \cdot q(t) = sin(\omega \cdot t) \tag{0.6}$$

$$q(t) = A \cdot e^{-\alpha t} + \frac{\alpha sin(\omega t) - \omega cos(\omega t)}{\alpha^2 + \omega^2} \tag{0.7}$$

$$q(t) = \frac{\alpha sin(\omega t) - \omega cos(\omega t)}{\alpha^2 + \omega^2} \tag{0.8}$$

Typ równania: nieliniowe jednorodne.

1.2 Po jakim czasie w obwodzie bez wymuszenia napięcie spadnie do 0.1 napięcia początkowego?

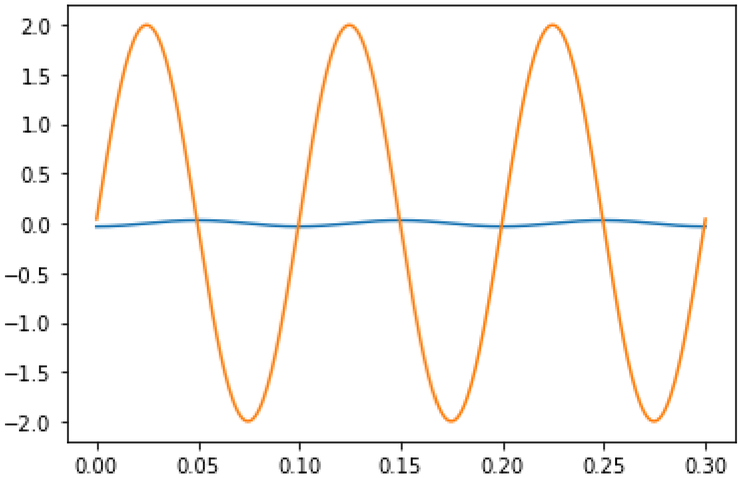
$$\frac{1}{10} = e^{-\alpha * t}$$
$$\alpha = \frac{-1}{RC}$$
$$R = 1$$
$$C = 0.742$$
$$\alpha = \frac{-1}{0.742}$$
$$\frac{1}{10} = e^{-\frac{1}{0.742} * t}$$
$$\frac{1}{10} = 2.718282^{-1.3477088t}$$
$$\log(\frac{1}{10}) = \log(2.718282^{-1.3477088t})$$
$$-1.3477088t * \log(2.718282) = \log(\frac{1}{10})$$
$$-1.34770881t = \frac{\log(\frac{1}{10})}{\log(2.718282)}$$
$$-1.34770881t = -2.302585$$
$$\frac{-1.34770881t}{-1.34770881} = \frac{-2.302585}{-1.34770881}$$

Przybliżona odpowiedź:

t = 1.708518

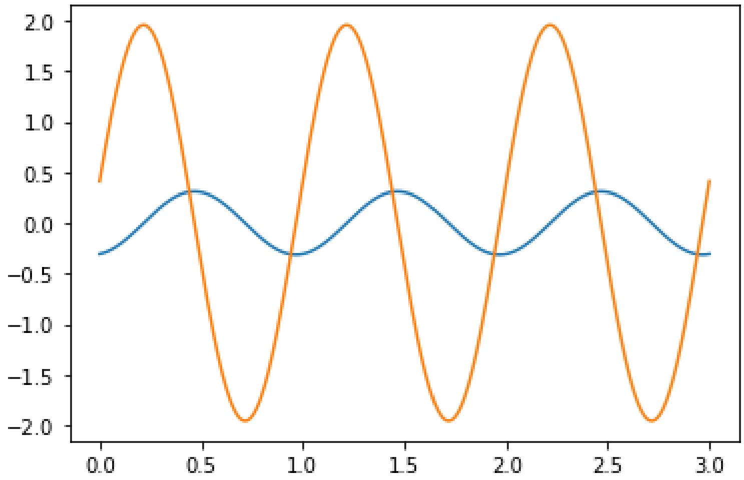
1.3 Jakie jest maksymalne napięcie na oporniku i kondensatorze po długim czasie (brak odpowiedzi swobodnej) w przypadku wymuszenia źródłem napięciowym sinusoidalnym o amplitudzie 2V i częstościach odpowiednio 0.1,1,10 Hz.

```
import matplotlib.pyplot as plt
import numpy as np
R = 1;
C = 0.742;
a = 1/(R*C)
A = 2

w1=(2*np.pi)/0.1
w2=(2*np.pi)/1
w3=(2*np.pi)/10

# 0.1 Hz
t=np.linspace(0, 3*2*np.pi/w1, 300)
s1=A*(a*np.sin(w1*t)-w1*np.cos(w1*t))/(w1**2+a**2)
sd1=A*(a*w1*np.cos(w1*t)+w1**2*np.sin(w1*t))/(w1**2+a**2)
fig,ax=plt.subplots()
ax.plot(t,s1,t,sd1)
plt.show()
max(s1),max(sd1)
```



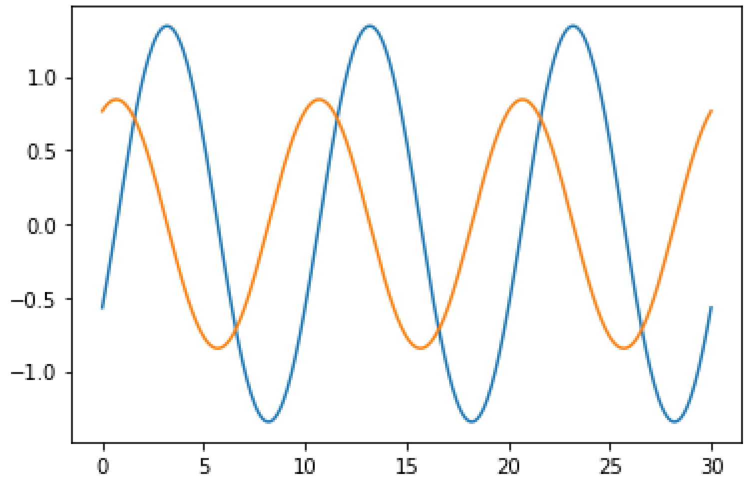

```
(0.031822053685531544, 1.9995077603675164)
```

```
import matplotlib.pyplot as plt
import numpy as np
R = 1;
C = 0.742;
a = 1/(R*C)
A = 2

w1=(2*np.pi)/0.1
w2=(2*np.pi)/1
w3=(2*np.pi)/10

# 1.0 Hz
t=np.linspace(0, 3*2*np.pi/w2, 300)
s1=A*(a*np.sin(w2*t)-w2*np.cos(w2*t))/(w2**2+a**2)
sd1=A*(a*w2*np.cos(w2*t)+w2**2*np.sin(w2*t))/(w2**2+a**2)
fig,ax=plt.subplots()
ax.plot(t,s1,t,sd1)
plt.show()
max(s1),max(sd1)
```




```
(0.3112172379385259, 1.9554809972663623)
```

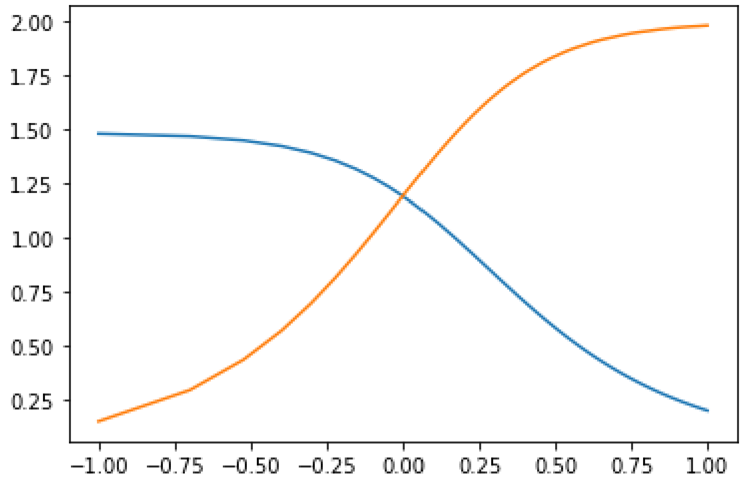
```python
import matplotlib.pyplot as plt
import numpy as np
R = 1;
C = 0.742;
a = 1/(R*C)
A = 2

w1=(2*np.pi)/0.1
w2=(2*np.pi)/1
w3=(2*np.pi)/10

# 10.0 Hz
t=np.linspace(0, 3*2*np.pi/w3, 300)
s1=A*(a*np.sin(w3*t)-w3*np.cos(w3*t))/(w3**2+a**2)
sd1=A*(a*w3*np.cos(w3*t)+w3**2*np.sin(w3*t))/(w3**2+a**2)
fig,ax=plt.subplots()
ax.plot(t,s1,t,sd1)
plt.show()
max(s1),max(sd1)
```




```
(1.3449385973437695, 0.8450838807916202)
```

1.4 Wykonaj wykres charakterystyki amplitudowo-częstościowej w przedziale częstości (0.1, 10) . Oś częstości przedstaw w skali logarytmicznej o podstawie 10.

```
logw = []
ss=[]
ssd=[]

ww=np.linspace(0.1,10,100)

for w in ww:
  t = np.linspace(0, 2 * np.pi / w, 100)
  s=A*(a*np.sin(w*t)-w*np.cos(w*t))/(w**2+a**2)
  sd=A*(a*w*np.cos(w*t)+w*w*np.sin(w*t))/(w**2+a**2)
  ss.append(max(s))
  ssd.append(max(sd))
  logw.append(np.log10(w))

fig,ax=plt.subplots()
ax.plot(logw,ss,logw,ssd)
plt.show()
```



Zadanie 2 - Oscylator harmoniczny z tłumieniem bez wymuszenia.

2.1 Zapisz poprawnie równanie dla masy m1 na sprężynie k1 z tłumieniem δ . Zaklasyfikuj typ równania.

$$\ddot{x} + 2\delta\dot{x} + \omega^2 x = 0$$
$$\omega^2 = \tfrac{k}{m}$$
$$\text{Po podstawieniu} = \ddot{x} + 2\delta\dot{x} + \tfrac{k}{m}x = 0$$

Typ równania:

jednorodne rzędu drugiego

2.2 Sprowadź równanie układu do układu równań stopnia pierwszego, zapisz równanie w postaci macierzowej.

$$\ddot{x} + 2\delta\dot{x} + \omega^2 x = 0$$
$$\begin{cases} \dot{x} = v \\ \dot{v} = -2\delta v - \omega^2 x \end{cases}$$
$$\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\delta \end{bmatrix}$$
$$\det \begin{bmatrix} -\lambda & 1 \\ \delta^2 & -2\delta - \lambda \end{bmatrix}$$

2.3 Wyznacz częstość własną drgań.

$$-\lambda(-2\lambda - \lambda) + \omega^2 = 0$$
$$2\delta\lambda + \lambda^2 + \omega^2 = 0$$
$$\lambda^2 + 2\delta\lambda + \omega^2 = 0$$
$$a = 1$$
$$b = 2\delta$$
$$\delta = 0,074$$
k=14.5
m=14
$$\frac{k}{m} = \omega^2$$
$$\Delta = 4\delta^2 - 4\omega^2$$
$$\Delta = 4 * 0,074^2 - 4\left(\frac{k}{m}\right)$$
$$\Delta = 4 * 0,074^2 - 4\left(\frac{14.5}{14}\right)$$
$$\Delta = -4.1209$$
$$\Delta = 4.1209i^2$$
$$\sqrt{\Delta} = 2.03i$$
$$\lambda_1 = -0,074 - 2.03i$$
$$\lambda_2 = -0,074 + 2.03i$$
Częstości własne drgań w przybliżeniu to:
-0,074 - 2.03i
-0,074 + 2.03i

2.4 Narysuj przebieg wychylenia w czasie dla dwóch wybranych warunków początkowych. Skomentuj interpretację tych warunków.

Warunki początkowe:

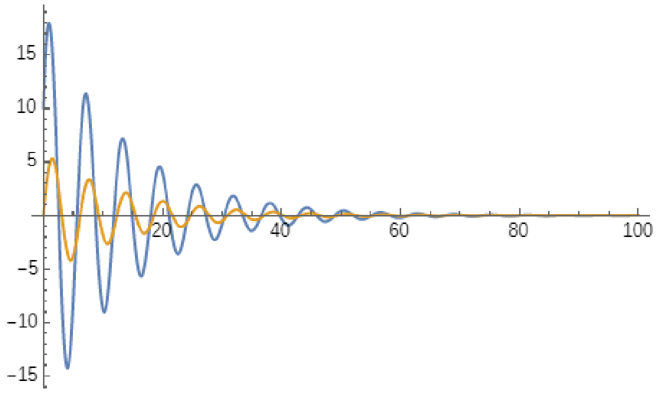
$$x[0] = 10$$
$$\dot{x}[10] = 16$$
$$\delta = 0.074$$
$$\omega = 1.0177$$

In[1]:=
```
ω = 1.0177
δ = 0.074
sol = DSolve[{x''[t] + 2 δ x'[t] + ω^2 x[t] == 0, x[0] == 10, x'[0] == 16}, x[t], {t, 0, 100}];
sol2 = DSolve[{x2''[t] + 2 δ x2'[t] + ω^2 x2[t] == 0, x2[0] == 0, x2'[0] == 6}, x2[t], {t, 0, 2
Plot[{x[t] /. sol, x2[t] /. sol2}, {t, 0, 100}, PlotRange → All]
```

Out[1]= 1.0177

Out[2]= 0.074

Out[5]=

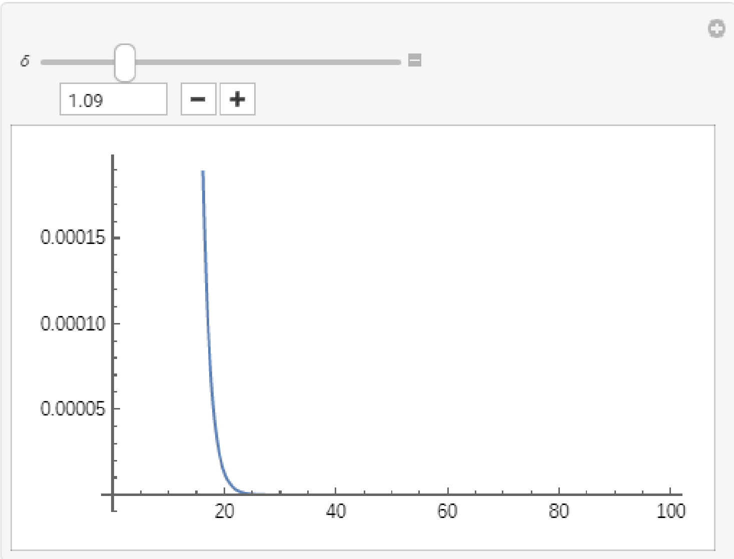


Choć na początku widoczna była duża różnica to po krótkim czasie następuje wyrównanie funkcji.

2.5 Wyznacz tłumienie krytyczne dla tego układu.

```
In[6]:= δ = 0.074
        w = 1.0177
        Manipulate[
         sol = DSolve[{x''[t] + 2 δ x'[t] + w^2 x[t] == 0, x[0] == 6, x'[0] == 3}, x[t], {t, 0, 50}];
         Plot[Evaluate[x[t] /. sol], {t, 0, 100}], {δ, 0, 5}]

Out[6]= 0.074

Out[7]= 1.0177
```

Out[8]=



Tłumienie krytyczne wynosi 1.01

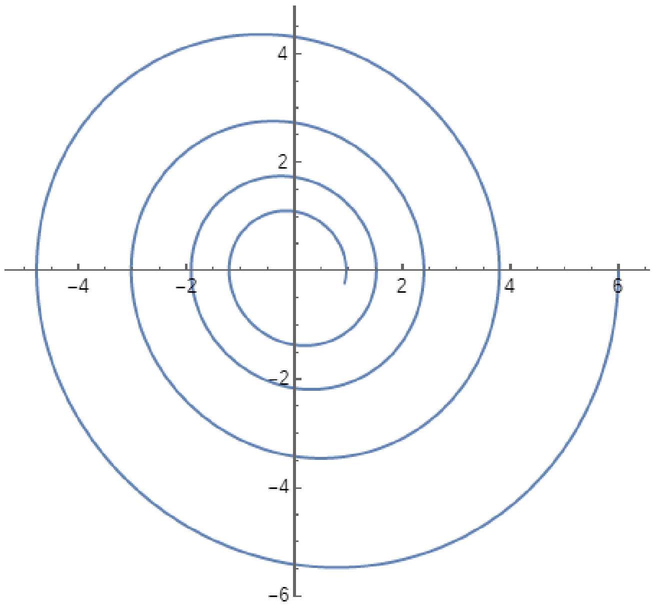
2.6 Narysuj ruch układu w przestrzenie fazowej (x,v).

In[9]:=
```
δ = 0.074
w = 1.0177
sol = NDSolve[{x''[t] + 2 δ x'[t] + w^2 x[t] == 0, x[0] == 6, x'[0] == 0}, {x[t], x'[t]}, {t,
ParametricPlot[Evaluate[{x[t], x'[t]} /. sol], {t, 0, 25}]
```

Out[9]= 0.074

Out[10]= 1.0177

Out[12]=



Zadanie 3 - Oscylator harmoniczny tłumieniem i wymuszeniem harmonicznym.

3.1 Zapisz poprawnie równanie dla masy m1 na sprężynie k1 z tłumieniem i wymuszeniem harmonicznym o częstości ω i amplitudzie 0.1m. Zaklasyfikuj typ równania.

$$\omega^2 = \tfrac{k}{m}$$

A=0.1

$$\ddot{x} + 2\delta\dot{x} + \omega^2 x = A\sin(\omega t)$$

$$\ddot{x} + 2\delta\dot{x} + \tfrac{k}{m}x = 0.1\sin(\sqrt{\tfrac{k}{m}}t)$$

Typ równania:

różniczkowe niejednorodne rzędu drugiego

3.2 Sprowadź równanie układu do układu równań stopnia pierwszego, zapisz równanie w postaci macierzowej.

$$\ddot{x} + 2\delta\dot{x} + \omega^2 x = A\sin(\omega t)$$

$$\begin{cases} \dot{x} = \dot{v} \\ v = A\sin(\omega t) - 2\delta\dot{v} - \omega^2 x \end{cases}$$

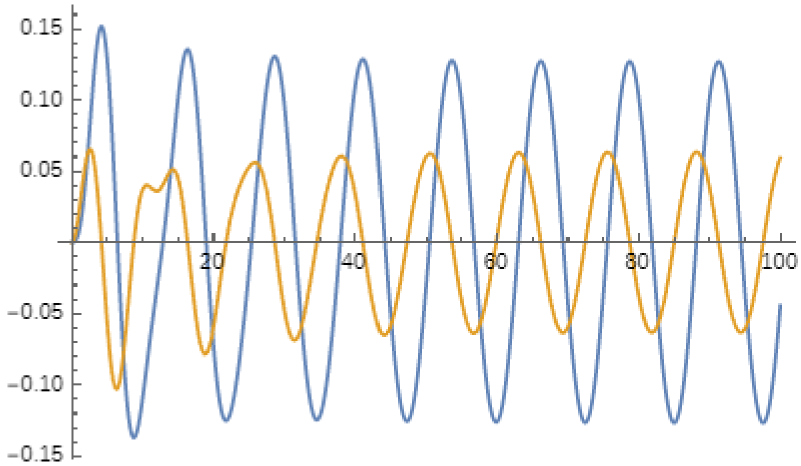
$$\begin{bmatrix} x' \\ v' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\delta \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ Asin(\omega t) \end{bmatrix}$$

3.3 Narysuj przebieg wychylenia w czasie dla zerowych warunków początkowych i wymuszenia z częstością 0.5ω, ω oraz 2ω, (ω-częstość drgań własnych). Skomentuj interpretację tych przebiegów. Na czym polega zjawisko rezonansu?

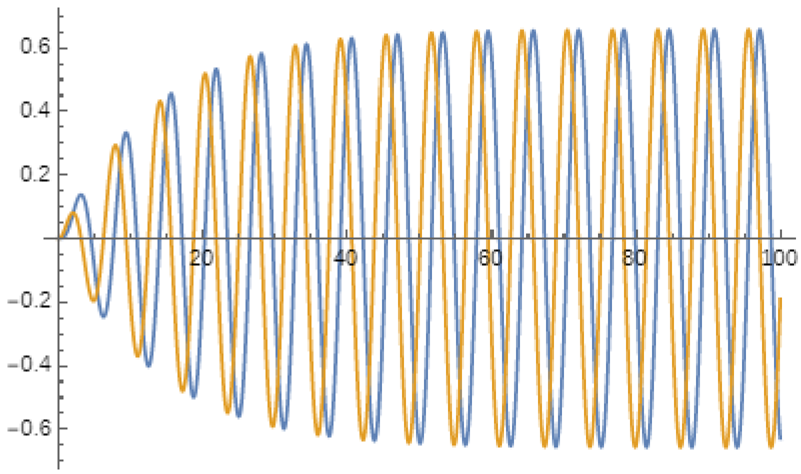
In[13]:=

```
δ = 0.074
w = 1.0177
tmax = 100
A = 0.1

sol = NDSolve[{x''[t] + 2 δ x'[t] + w^2 x[t] == A Sin[0.5 t], x[0] == 0, x'[0] == 0}, {x[t], x'[t]}, {t, 0, tmax}
Plot[Evaluate[{x[t], x'[t]} /. sol], {t, 0, tmax}]

sol2 = NDSolve[{x''[t] + 2 δ x'[t] + w^2 x[t] == A Sin[t], x[0] == 0, x'[0] == 0}, {x[t], x'[t]}, {t, 0, tmax}];
Plot[Evaluate[{x[t], x'[t]} /. sol2], {t, 0, tmax}]

sol3 = NDSolve[{x''[t] + 2 δ x'[t] + w^2 x[t] == A Sin[2 t], x[0] == 0, x'[0] == 0}, {x[t], x'[t]}, {t, 0, tmax}]
Plot[Evaluate[{x[t], x'[t]} /. sol3], {t, 0, tmax}]
```

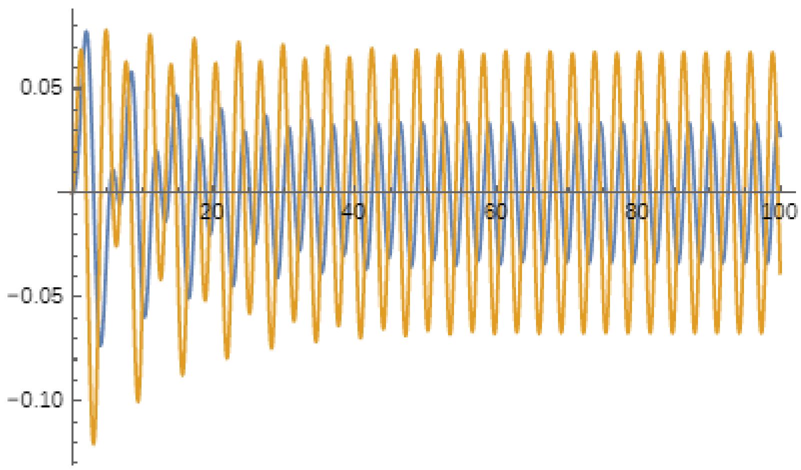
Out[18]=



Out[20]=



Out[22]=



Na podstawie przebiegów, możemy stwierdzić że od ok. 25 sekundach mamy do czynienia ze zjawiskiem rezonansu w każdym z przypadków.

Rezonans to zjawisko wzrastania amplitudy drgań układu drgającego, w którym częstotliwość siły wymuszającej te drgania jest podobna do częstości własnej układu, czyli częstości drgań swobodnych tego układu.

Zadanie 4 - Oscylator harmoniczny podwójny bez tłumienia i bez wymuszenia.

4.1 Zapisz poprawnie równanie dla masy m1 na sprężynie k1 i połączonej z nią masy m2 na sprężynie m2 (przykład z wykładu) . Zaklasyfikuj typ równania.

$$\begin{cases} m_1 * \ddot{x}_1 = -k1x1 = +k2(x2 - x1) \\ m_2 * \ddot{x}_2 = -k_2(x_2 - x_1) \end{cases}$$

$$\begin{cases} \ddot{x}_1 = \frac{1}{m_1}(-k_1x_1 + k_2(x_2 - x_1)) \\ \ddot{x}_2 = \frac{1}{m_2}(-k_2(x_2 - x_1)) \end{cases}$$

$$\begin{cases} \ddot{x}_1 = -\frac{k_1}{m_1}x_1 + \frac{k_2}{m_1}x_2 - \frac{k_2}{m_1}x_1 \\ \ddot{x}_2 = -\frac{k_2}{m_2}x_2 - \frac{k_2}{m_2}x_1 \end{cases}$$

Typ równania:

różniczkowe niejednorodne rzędu drugiego

4.2 Sprowadź równanie układu do układu równań stopnia pierwszego, zapisz równanie w postaci macierzowej.

$$x' = V \quad x'' = V'$$

$$\begin{cases} \dot{v}_1 = v_1 \\ \dot{v}_1 = \frac{-k_1}{m_1}x_1 + \frac{k_2}{m_1}\dot{x}_2 - \frac{k_2}{m1}x_1 \\ \dot{x}_2 = v_2 \\ \dot{v}_2 = -\frac{k_2}{m_2}x_2 + \frac{k_2}{m_2}x_1 \end{cases}$$

$$\begin{bmatrix} x_1 \\ \dot{v} \\ \ddot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1}{m_1} \cdot \frac{k_2}{m_1} & 0 & \frac{k_2}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & 0 & \frac{k_2}{m_2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix}$$

4.3 Wyznacz wartości własne macierzy układu. Jaki jest sens fizyczny tych wielkości?

$$\det(A - \lambda f) = 0$$

$$det = \begin{bmatrix} -\lambda & 1 & 0 & 0 \\ (-\frac{k_1}{m_1} - \frac{k_2}{m_1}) & -\lambda & \frac{k_2}{m_1} & 0 \\ 0 & 0 & -\lambda & 1 \\ \frac{k_2}{m_2} & 0 & \frac{k_2}{m_2} & -\lambda \end{bmatrix} = 0$$

$$\lambda_1 = 0. - 1.26849i$$

$$\lambda_2 = 0. + 1.26849i$$

$$\lambda_3 = 0. - 2.28619i$$

$$\lambda_4 = 0. + 2.28619i$$

Zadanie 6 - Modele rozwoju epidemi.

Kod realizujący model logistyczny Verhulsta oraz generujący wykres porównujący ten model z właściwymi danymi z Czech:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
import math
import scipy.optimize as optim
warnings.filterwarnings('ignore')

data = pd.read_csv('/content/sample_data/owid-covid-data.csv', sep=';')
data = data['total_cases']
data = data.reset_index(drop=False)
data.columns = ['Timestep', 'Total Cases']

def my_logistic(t, a, b, c):
  return c / (1 + a * np.exp(-b*t))
def my_logistic2(t):
  return c / (1 + a * np.exp(-b*t))

p0 = np.random.exponential(size=3)

bounds = (0, [100000., 3., 1000000000.])

x = np.array(data['Timestep']) + 1
y = np.array(data['Total Cases'])
(a,b,c),cov = optim.curve_fit(my_logistic, x, y, bounds=bounds, p0=p0)

plt.scatter(x, y)
plt.plot(x, my_logistic2(x), color="red")
plt.title('Logistic Model vs Real Observations of UKR COVID')
plt.legend(['Logistic Model', 'Real data'])
plt.xlabel('Time')
plt.ylabel('Infections')
```

Zadanie 7 - Automat komórkowy Snowflake.

```
size = 10
def neigh():
  nearest=[[0 for _ in range(size)] for _ in range(size)]
  for y in range(size):
    for x in range(size):
      nearest[y][x] = count(y,x)
  return nearest

def count(y, x):
  dy=[0,0,1,-1]
  dx=[-1,1,0,0]
  count = 0
  for k in range(len(dx)):
    kx=(x+dx[k]) % size
    ky=(y+dy[k]) % size
    count=count+tab[ky][kx]
  return count

def inicialize(y,x):
  tab=[[0 for _ in range(y)] for _ in range(x)]
  tab[4][4]=1
  tab[4][5]=1
  return tab

def print_tab(tab):
  for y in range(size):
    for x in range(size):
      print(tab[x][y], end=' ')
    print(' ')


tab=inicialize(size,size)
print_tab(tab)
print('====================')
for x in range(size):
  nearest = neigh()
  res=tab
  for y in range(size):
      for x in range(size):
        if tab[y][x] == 0:
          if nearest[y][x]==1:
            res[y][x]=1
  tab=res
  print_tab(tab)
  print('====================')
```