Równanie falowe

Weronika Biadała
Aleksandra Gajda

Lipiec 5, 2019

Plan prezentacji

Ruch falowy, fale w ośrodkach sprężystych

Ruch drgający zaczyna rozprzestrzeniać się w postaci fali, kiedy pewien obszar ośrodka sprężystego zostanie pobudzony do drgań, które przekazane zostaje innym cząstkom w tym ośrodku. Rodzaje fal:

- fala kulista - kiedy czoło fali na kształt sferyczny,
- fala płaska - kiedy czoło fali ma kształt płaszczyzny,
- fala poprzeczna - kiedy drgania, rozchodzące się w postaci fali, odbywają się w kierunku prostopadłym do kierunku ruchu fali,
- fala podłużna - kiedy drgania odbywają się w kierunku równoległym do kierunku ruchu fali.

Ruch falowy, fale w ośrodkach sprężystych

Ruch drgający zaczyna rozprzestrzeniać się w postaci fali, kiedy pewien obszar ośrodka sprężystego zostanie pobudzony do drgań, które przekazane zostaje innym cząstkom w tym ośrodku. Rodzaje fal:

- fala kulista - kiedy czoło fali na kształt sferyczny,
- fala płaska - kiedy czoło fali ma kształt płaszczyzny,
- fala poprzeczna - kiedy drgania, rozchodzące się w postaci fali, odbywają się w kierunku prostopadłym do kierunku ruchu fali,
- fala podłużna - kiedy drgania odbywają się w kierunku równoległym do kierunku ruchu fali.

Ruch falowy, fale w ośrodkach sprężystych

Ruch drgający zaczyna rozprzestrzeniać się w postaci fali, kiedy pewien obszar ośrodka sprężystego zostanie pobudzony do drgań, które przekazane zostaje innym cząstkom w tym ośrodku. Rodzaje fal:

- fala kulista - kiedy czoło fali na kształt sferyczny,
- fala płaska - kiedy czoło fali ma kształt płaszczyzny,
- fala poprzeczna - kiedy drgania, rozchodzące się w postaci fali, odbywają się w kierunku prostopadłym do kierunku ruchu fali,
- fala podłużna - kiedy drgania odbywają się w kierunku równoległym do kierunku ruchu fali.

Ruch falowy, fale w ośrodkach sprężystych

Ruch drgający zaczyna rozprzestrzeniać się w postaci fali, kiedy pewien obszar ośrodka sprężystego zostanie pobudzony do drgań, które przekazane zostaje innym cząstkom w tym ośrodku. Rodzaje fal:

- fala kulista - kiedy czoło fali na kształt sferyczny,
- fala płaska - kiedy czoło fali ma kształt płaszczyzny,
- fala poprzeczna - kiedy drgania, rozchodzące się w postaci fali, odbywają się w kierunku prostopadłym do kierunku ruchu fali,
- fala podłużna - kiedy drgania odbywają się w kierunku równoległym do kierunku ruchu fali.

Równanie różniczkowe dla ruchu falowego

Równanie różniczkowe, opisujące wychylenie w ruchy falowym, oparte jest na pochodnych cząstkowych drugiego rzędu względem czasu i położenia. Dla fali rozchodzącej się w kierunku x

$$\frac{^2u}{t^2} = c^2\frac{^2u}{x^2}$$

Wyprowadzenie równania falowego

Równanie różniczkowe drgań opisuje drgania punktu:
$\frac{d^2u}{dt^2} + \omega^2 u = 0$
u - kierunek drgań,
Ponieważ: $\omega = 2\varpi f = 2\varpi \frac{1}{T}$
To równanie: $\frac{d^2u}{dt^2} + (\frac{2\varpi}{T})^2 u = 0$

Wyprowadzenie równania falowego

Rozwiązanie: $u = A\sin\omega t = A\sin\frac{2\varpi}{T}t$
Opisuje ono tylko zmiany wychylenia u w czasie, bez zmiany położenia środka drgań.

Wyprowadzenie równania falowego

Kolejna pochodna dotyczy odchyleń stałych w czasie, względem x:
$\frac{d^2u}{dx^2} + (\frac{2\varpi}{\lambda})^2 u = 0$
λ - długość fali, okres zmian przestrzennych,
Rozwiązanie: $u = A \sin kx = A \sin \frac{2\varpi}{\lambda} x$

Wyprowadzenie równania falowego

Zmiany w ruchu falowym odbywają się w czasie i w przestrzeni, oba równania zostają połączone:

$\frac{d^2u}{dx^2} + (\frac{2\varpi}{\lambda})^2(-\frac{T}{2\varpi})\frac{d^2u}{dt^2} = 0$

Ponieważ: $\frac{T}{\lambda} = \frac{T}{vT} = \frac{1}{v}$

Zmiana pochodnych na cząstkowe:

$\frac{^2u}{x^2} - \frac{1}{v^2}\frac{^2u}{t^2} = 0$

Wyprowadzenie równania falowego

Równanie różniczkowe dla ruchu falowego:
$\frac{^2u}{t^2} = c^2 \frac{^2u}{x^2}$

Fale

Przyspieszenie każdego elementu łańcucha jest proporcjonalne do frontu fali.

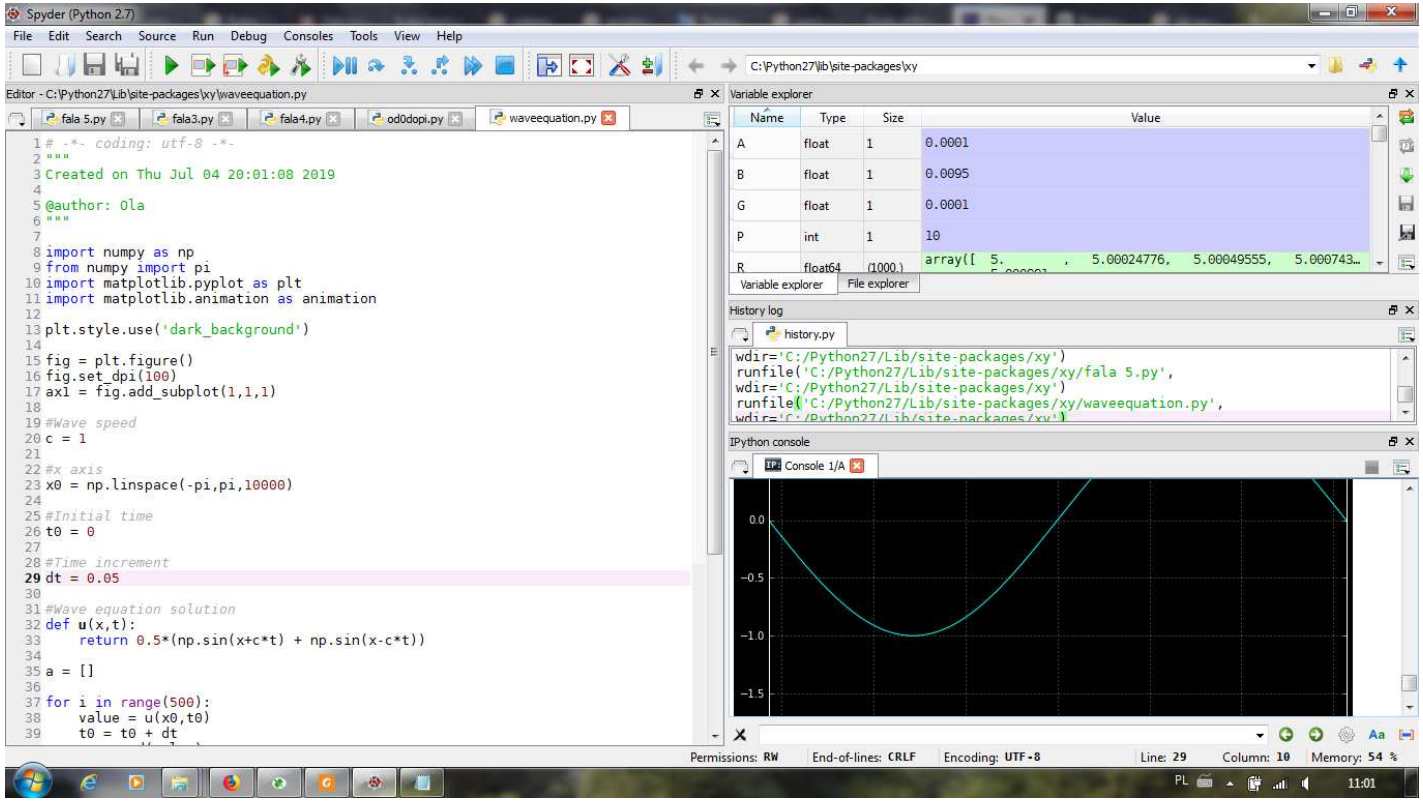
Równanie falowe

Poniższe równanie jest równaniem różniczkowym cząstkowym zwanym równaniem falowym i może być użyte do modelowania różnych zjawisk na przykład drgających strun i fal propagujących.
$\frac{^2u}{t^2} = c^2\frac{^2u}{x^2} 0 < t, L \geq x \geq 0\ u(x,0) = f(x) L \geq x \geq 0$
$u(0,t) = u(L,t) = 0 t > 0$
c - interpretacja prędkości fali $[\frac{m}{s}]$

Ogólne rozwiązanie równania falowego

$$u(x,t) = f(x - ct) + g(x + ct)$$

Python


Spyder (Python 2.7)
File Edit Search Source Run Debug Consoles Tools View Help
Editor - C:\Python27\Lib\site-packages\xy\waveequation.py
fala 5.py
fala3.py
fala4.py
od0dopi.py
waveequation.py
-*- coding: utf-8 -*-
"""
Created on Thu Jul 04 20:01:08 2019
@author: Ola
"""
import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib.animation as animation
plt.style.use('dark_background')
fig = plt.figure()
fig.set_dpi(100)
ax1 = fig.add_subplot(1,1,1)
#Wave speed
c = 1
#x axis
x0 = np.linspace(-pi,pi,10000)
#Initial time
t0 = 0
#Time increment
dt = 0.05
#Wave equation solution
def u(x,t):
return 0.5*(np.sin(x+c*t) + np.sin(x-c*t))
a = []
for i in range(500):
value = u(x0,t0)
t0 = t0 + dt
Variable explorer
Name Type Size Value
A float 1 0.0001
B float 1 0.0095
G float 1 0.0001
P int 1 10
R float64 (1000,) array([5. , 5.00024776, 5.00049555, 5.000743...
Variable explorer
File explorer
History log
history.py
wdir='C:/Python27/Lib/site-packages/xy')
runfile('C:/Python27/Lib/site-packages/xy/fala 5.py',
wdir='C:/Python27/Lib/site-packages/xy')
runfile('C:/Python27/Lib/site-packages/xy/waveequation.py',
IPython console
Console 1/A
0.0
−0.5
−1.0
−1.5
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 29
Column: 10
Memory: 54 %
PL
11:01

Przykładowe rozwiązanie dla sinusa

$$u(x,t) = \frac{1}{2}\Big(\sin(x-ct) + \sin(x+ct)\Big)$$

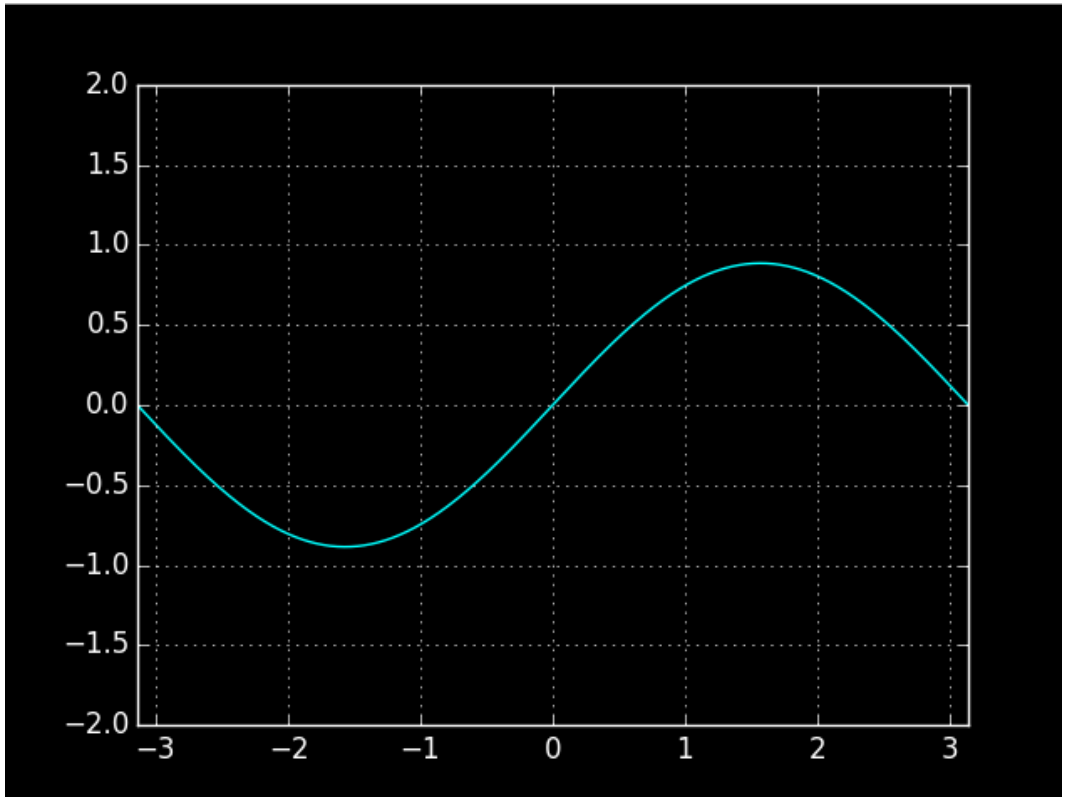
$$c = \frac{1m}{s} \quad L = 2\varpi$$

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jul 04 20:01:08 2019

@author: Ola
"""

import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib.animation as animation

plt.style.use('dark_background')

fig = plt.figure()
fig.set_dpi(100)
ax1 = fig.add_subplot(1,1,1)

#Wave speed
c = 1

#x axis
x0 = np.linspace(-pi,pi,10000)

#Initial time
t0 = 0

#Time increment
dt = 0.05

#Wave equation solution
def u(x,t):
    return 0.5*(np.sin(x+c*t) + np.sin(x-c*t))

a = []

for i in range(500):
    value = u(x0,t0)
    t0 = t0 + dt
```

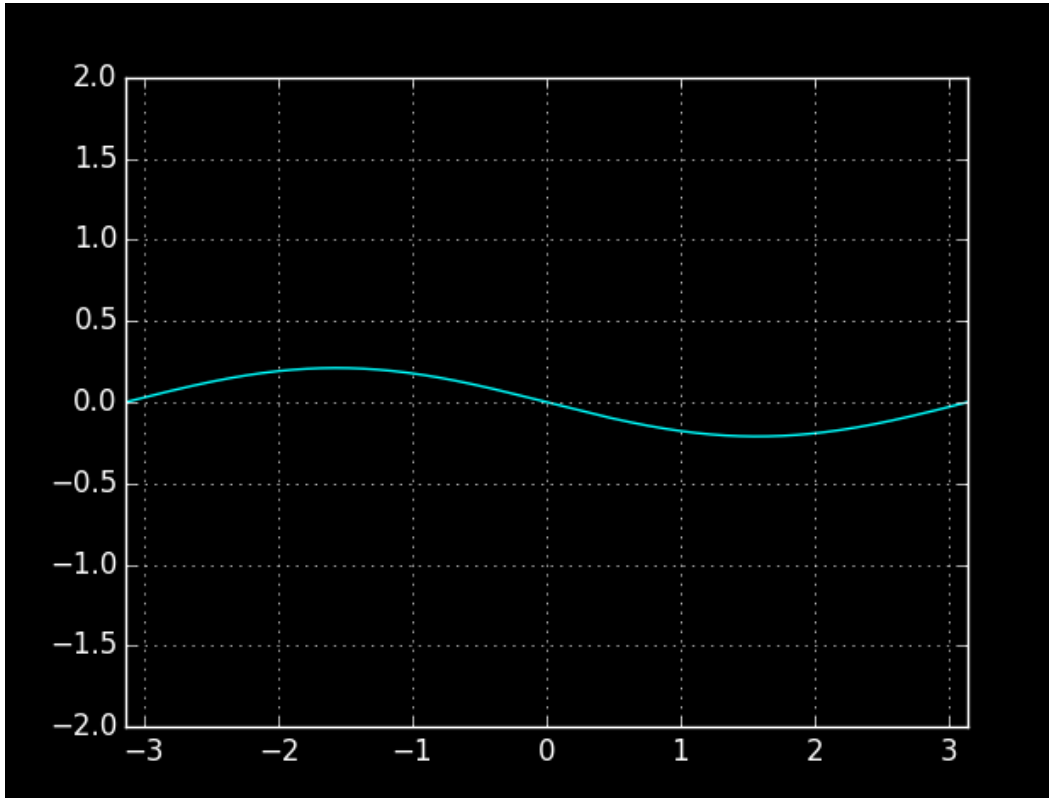
Przykładowe rozwiązanie dla sinusa

```
39      t0 = t0 + dt
40      a.append(value)
41
42  k = 0
43  def animate(i):
44      global k
45      x = a[k]
46      k += 1
47      ax1.clear()
48      plt.plot(x0,x,color='cyan')
49      plt.grid(True)
50      plt.ylim([-2,2])
51      plt.xlim([-pi,pi])
52
53  anim = animation.FuncAnimation(fig,animate,frames=360,interval=20)
54  plt.show()
```

Przykładowe rozwiązanie dla sinusa



Przykładowe rozwiązanie dla sinusa



Przykładowe rozwiązanie dla sinusa

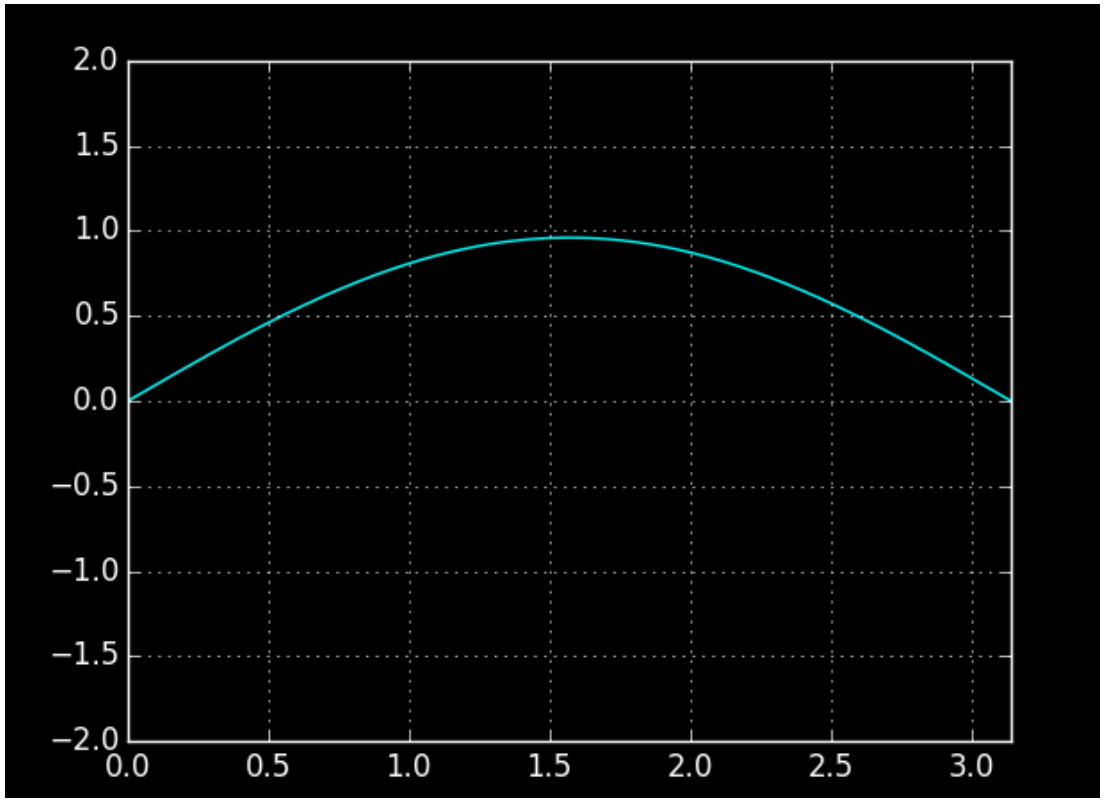
$$c = \frac{1m}{s} \; L = \varpi$$

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 05 09:23:28 2019

@author: Ola
"""

import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib.animation as animation

plt.style.use('dark_background')

fig = plt.figure()
fig.set_dpi(100)
ax1 = fig.add_subplot(1,1,1)

#Wave speed
c = 1

#x axis
x0 = np.linspace(0,pi,10000)

#Initial time
t0 = 0

#Time increment
dt = 0.05

#Wave equation solution
def u(x,t):
    return 0.5*(np.sin(x+c*t) + np.sin(x-c*t))

a = []

for i in range(500):
    value = u(x0,t0)
    t0 = t0 + dt
```

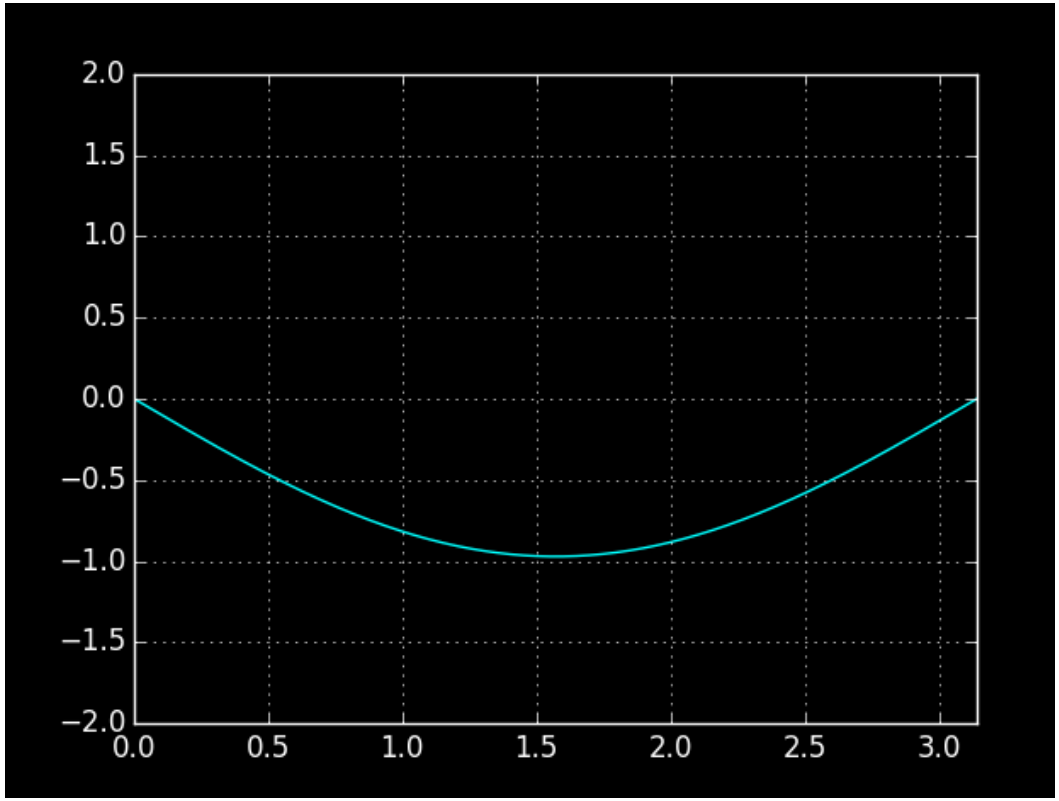
Przykładowe rozwiązanie dla sinusa

```
40      a.append(value)
41
42  k = 0
43  def animate(i):
44      global k
45      x = a[k]
46      k += 1
47      ax1.clear()
48      plt.plot(x0,x,color='cyan')
49      plt.grid(True)
50      plt.ylim([-2,2])
51      plt.xlim([0,pi])
52
53  anim = animation.FuncAnimation(fig,animate,frames=360,interval=20)
54  plt.show()
```

Przykładowe rozwiązanie dla sinusa



Przykładowe rozwiązanie dla sinusa



Przykładowe rozwiązanie dla cosinusa

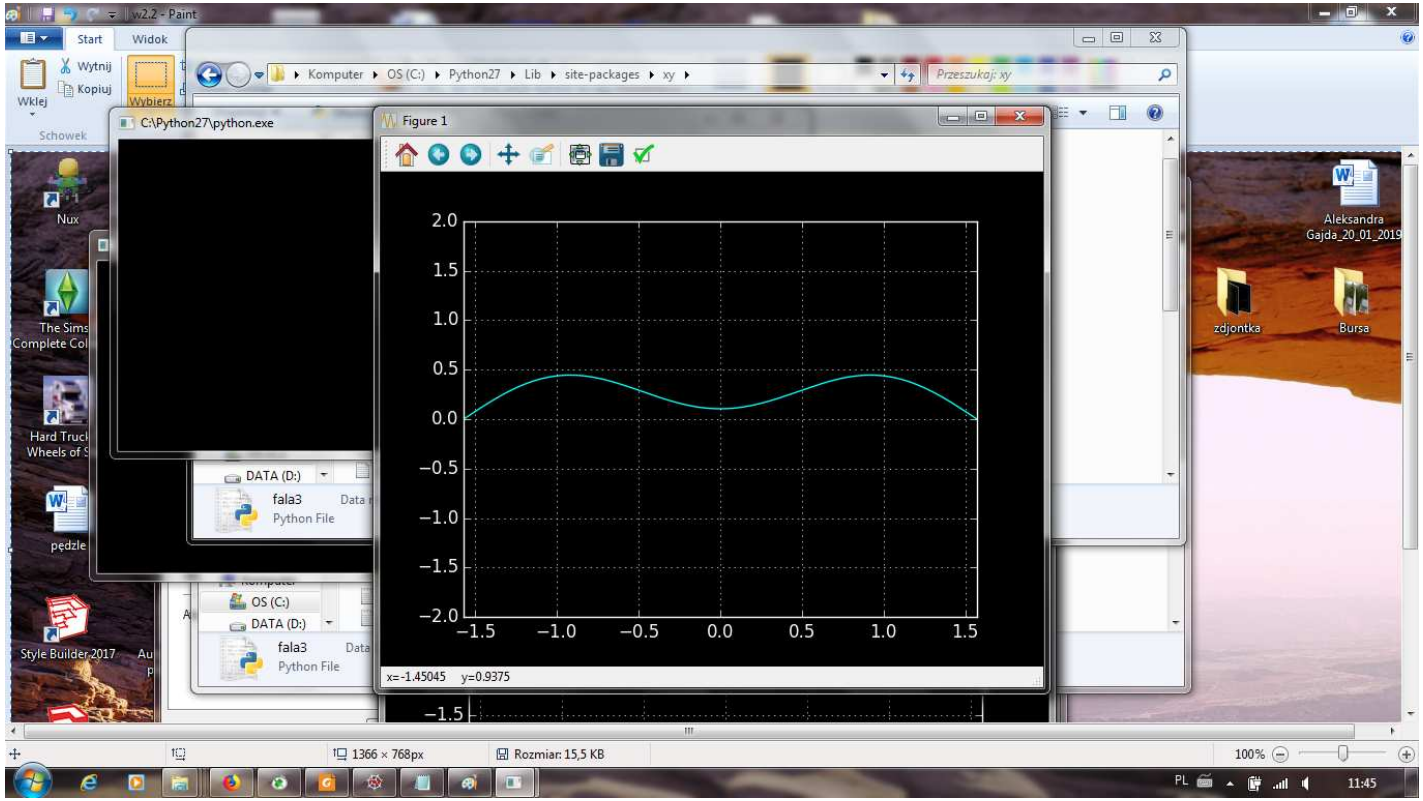
$$f(x-ct)+g(x+ct)=\cos(x-ct)^3+\cos(x+ct)^3$$

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 05 09:31:09 2019

@author: Ola
"""

import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib.animation as animation

plt.style.use('dark_background')

fig = plt.figure()
fig.set_dpi(100)
ax1 = fig.add_subplot(1,1,1)

#Wave speed
c = 2

#x axis
x0 = np.linspace(-pi/2,pi/2,10000)

#Initial time
t0 = 0

#Time increment
dt = 0.03

def phi(x):
    return np.cos(x)**3

#Wave
def u(x,t):
    return 0.5*(phi(x+c*t)+phi(x-c*t))

a = []

```

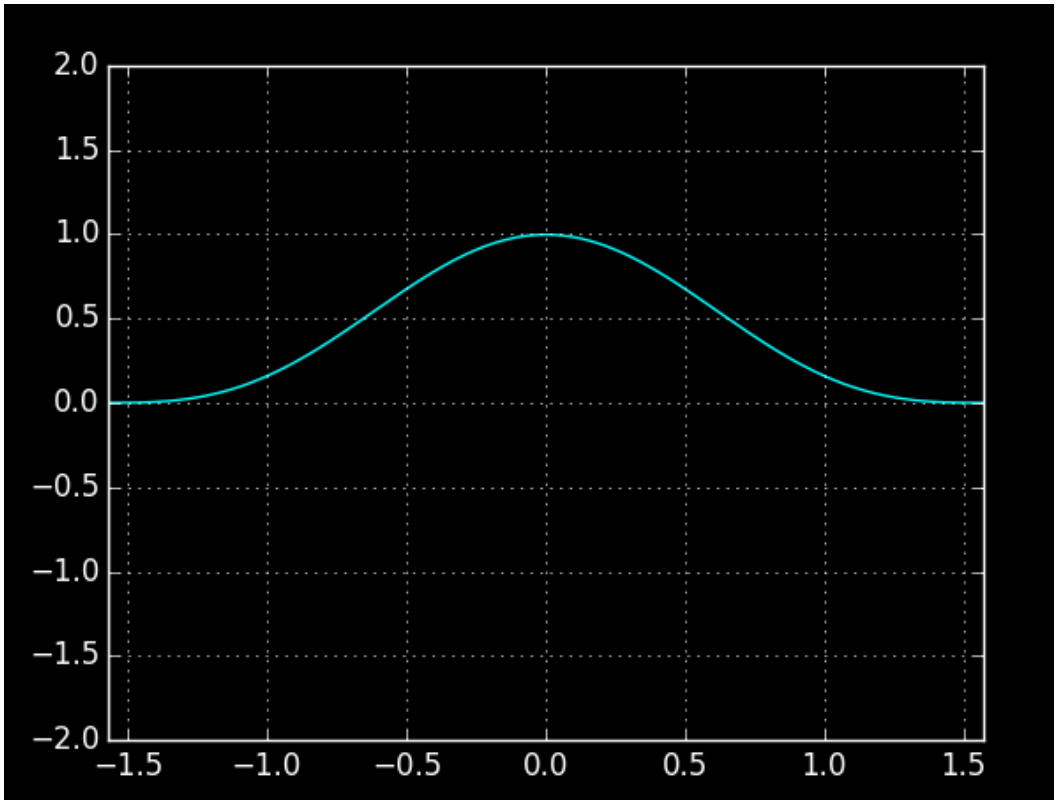
Przykładowe rozwiązanie dla cosinusa

```
40 for i in range(500):
41     value = u(x0,t0)
42     t0 = t0 + dt
43     a.append(value)
44
45 k = 0
46 def animate(i):
47     global k
48     x = a[k]
49     k += 1
50     ax1.clear()
51     plt.plot(x0,x,color='cyan')
52     plt.grid(True)
53     plt.ylim([-2,2])
54     plt.xlim([-pi/2,pi/2])
55
56 anim = animation.FuncAnimation(fig,animate,frames=360,interval=20)
57 plt.show()
```

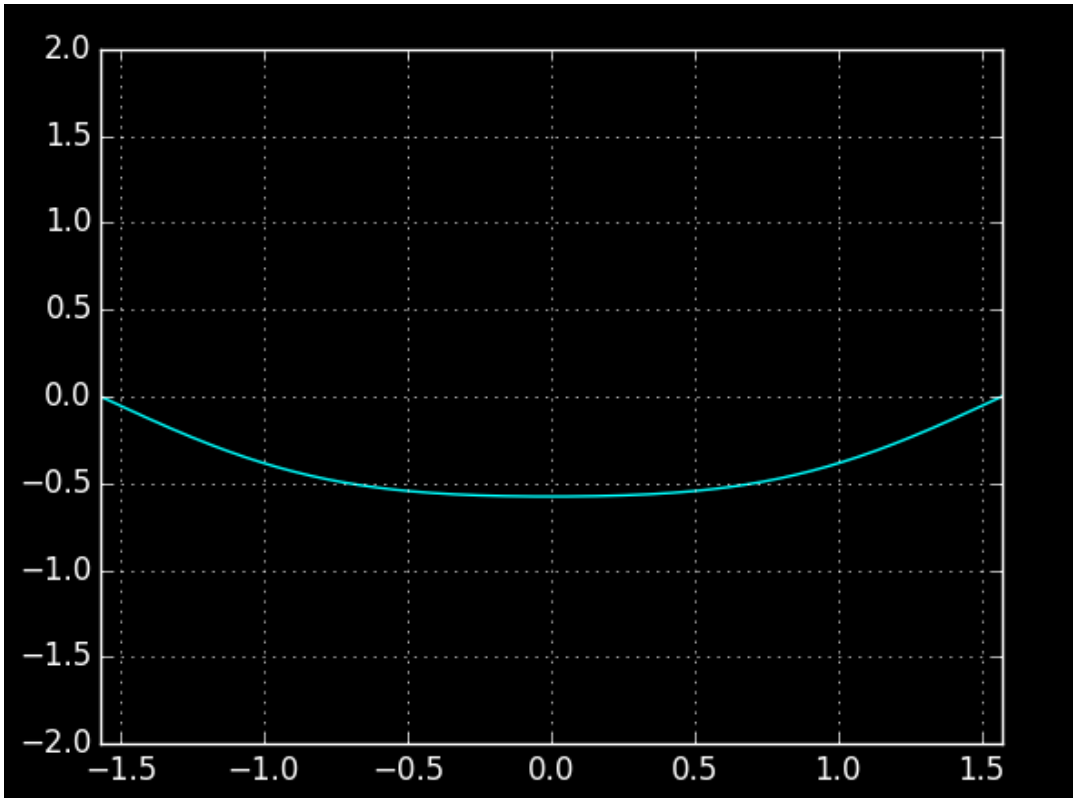
Przykładowe rozwiązanie dla cosinusa



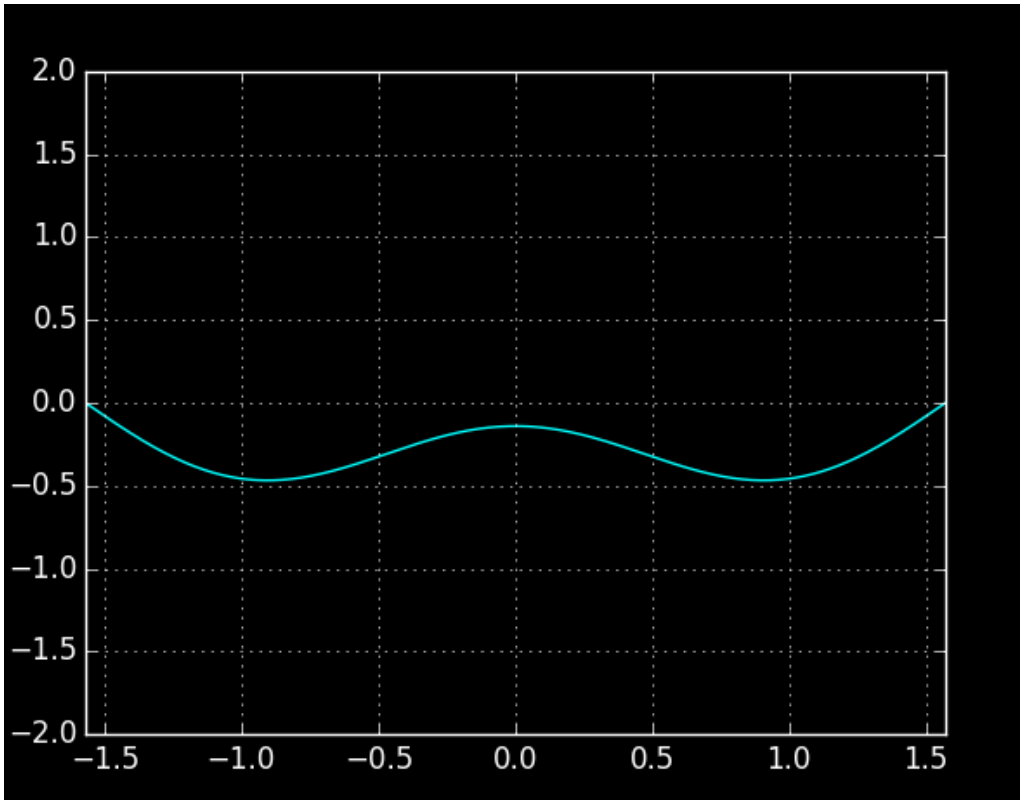
Przykładowe rozwiązanie dla cosinusa



Przykładowe rozwiązanie dla cosinusa



Przykładowe rozwiązanie dla cosinusa



Przykładowe rozwiązanie dla nakładania się fal

$$u(x,t) = \frac{1}{2}\Big(\cos(x-ct) + \cos(x+ct)\Big)$$

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 05 09:33:12 2019

@author: Ola
"""

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

plt.style.use('dark_background')

fig = plt.figure()
fig.set_dpi(100)
ax1 = fig.add_subplot(1,1,1)

#x axis
x0 = np.linspace(-3*np.pi,3*np.pi,10000)

#Initial time
t0 = 0

#Increment
dt = 0.1

def u(x,t):
    return 0.5*(np.cos(x+t)+np.cos(x-t))

a = []
si = []
co = []

for i in range(500):
    value = u(x0,t0)
    s = 0.5*np.cos(x0+t0)
    c = 0.5*np.cos(x0-t0)
    t0 = t0 + dt
    a.append(value)
```

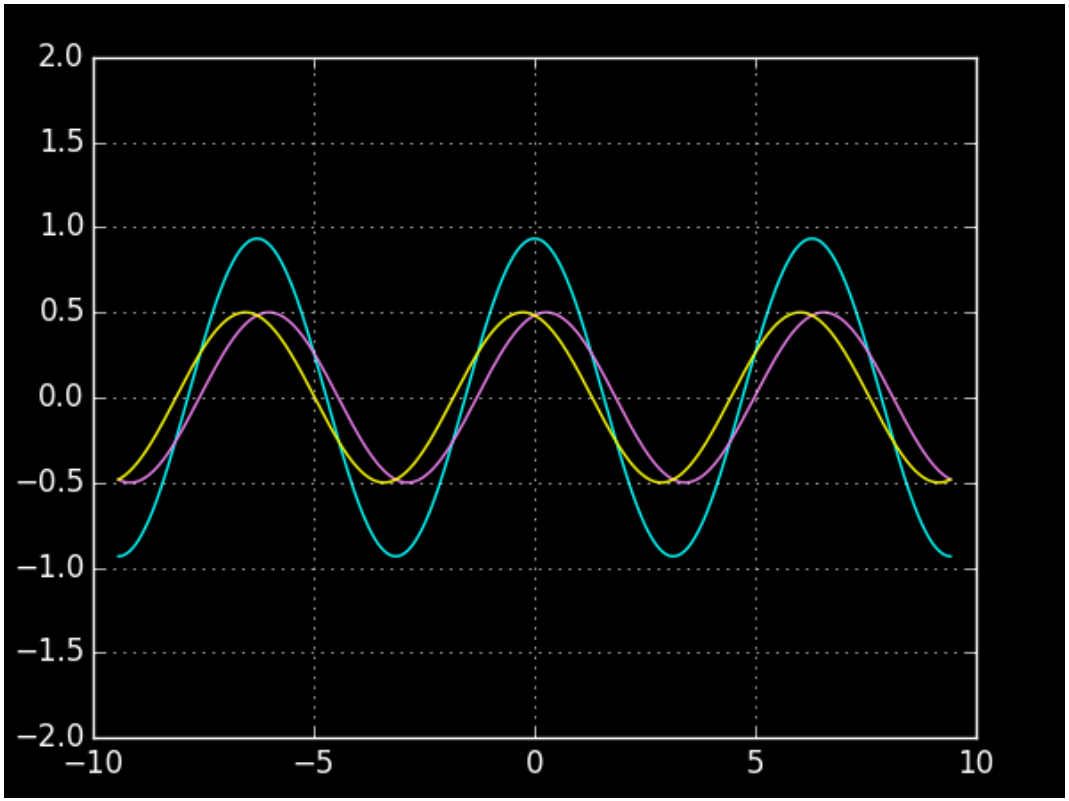
Przykładowe rozwiązanie dla nakładania się fal

$$u(x,t) = \frac{1}{2}\Big(\cos(x-ct) + \cos(x+ct)\Big)$$

```
40      si.append(s)
41      co.append(c)
42
43  k = 0
44  def animate(i):
45      global k
46      x = a[k]
47      k += 1
48      ax1.clear()
49      plt.plot(x0,x,color='cyan')
50      plt.plot(x0,si[k],color='violet')
51      plt.plot(x0,co[k],color='yellow')
52      plt.grid(True)
53      plt.ylim([-2,2])
54
55  anim = animation.FuncAnimation(fig,animate,frames=360,interval=20)
56  plt.show()
```

Przykładowe rozwiązanie dla nakładania się fal

$$u(x,t) = \frac{1}{2}\Big(\cos(x-ct) + \cos(x+ct)\Big)$$



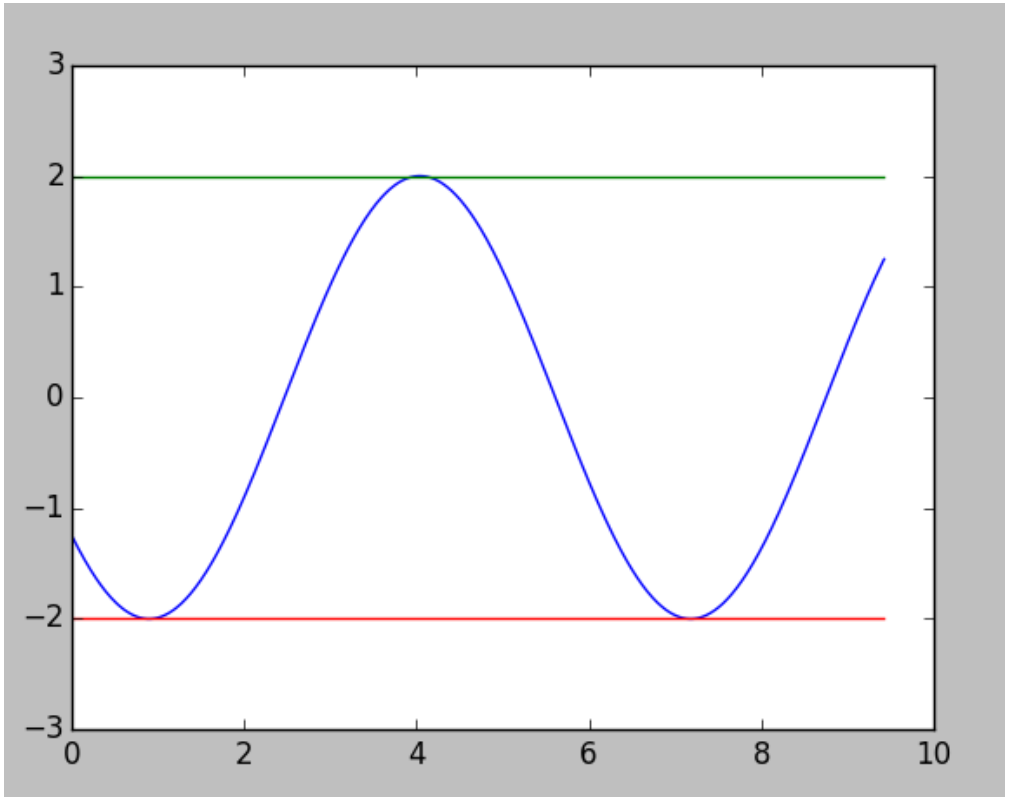
Przykładowe rozwiązanie dla nakładania się fal

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 05 09:36:16 2019

@author: Ola
"""

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

fig = plt.figure()
fig.set_dpi(100)
ax1 = fig.add_subplot(1,1,1)

#Speed of the wave
c = 20
scale = 2

#Initial conditions
x0 = np.linspace(0,3*np.pi,1000)
t0 = 0

#Increment
dt = 0.01

#Onda
def u(x,t):
    return scale*np.sin(x - c*t)

a = []

for i in range(500):
    value = u(x0,t0)
    t0 = t0 + dt
    a.append(value)

k = 0
def animate(i):
```

Przykładowe rozwiązanie dla nakładania się fal

```
40      global k
41      x = a[k]
42      k += 1
43      ax1.clear()
44      plt.plot(x0,x)
45      plt.plot(x0,np.ones((x0.shape[0],1))*scale)
46      plt.plot(x0,np.ones((x0.shape[0],1))*(-scale
47      plt.ylim([-scale-1,scale+1])
48
49 anim = animation.FuncAnimation(fig,animate,frame
50 plt.show()
```

Przykładowe rozwiązanie dla nakładania się fal



Bibliografia


https://matematyka.pl/latex.htm


http://www.mif.pg.gda.pl/homepages/sylas/students/wdi/do


http://firsttimeprogrammer.blogspot.com/2015/07/the-wave


http://www.latex-kurs.x25.pl/paper/wyrazenia_matematyczn


http://www.if.pw.edu.pl/~anadam/WykLadyFO/FoWWW_16.html

Koniec

Dziękujemy za uwagę