# TECHNIKI PROGRAMOWANIA - projekt 3

Jan Bullmann 197574

Roman Masiuk 206439

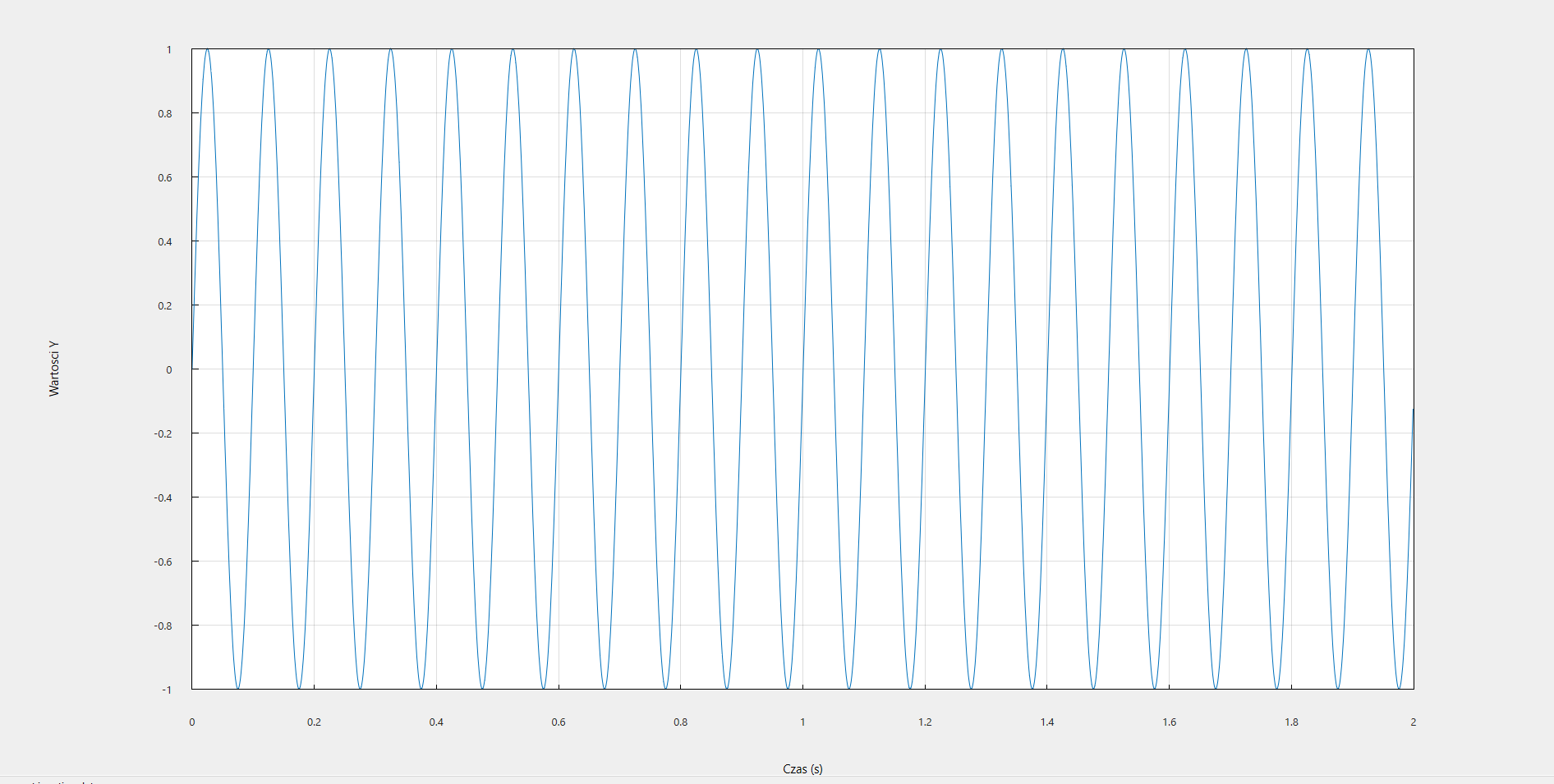
# Wizualizacja wykresów 1D

>>> import signal\_processing as b

1. Generowanie sin

>>> sygnal = b.generate\_sin(10,1000,0,2)

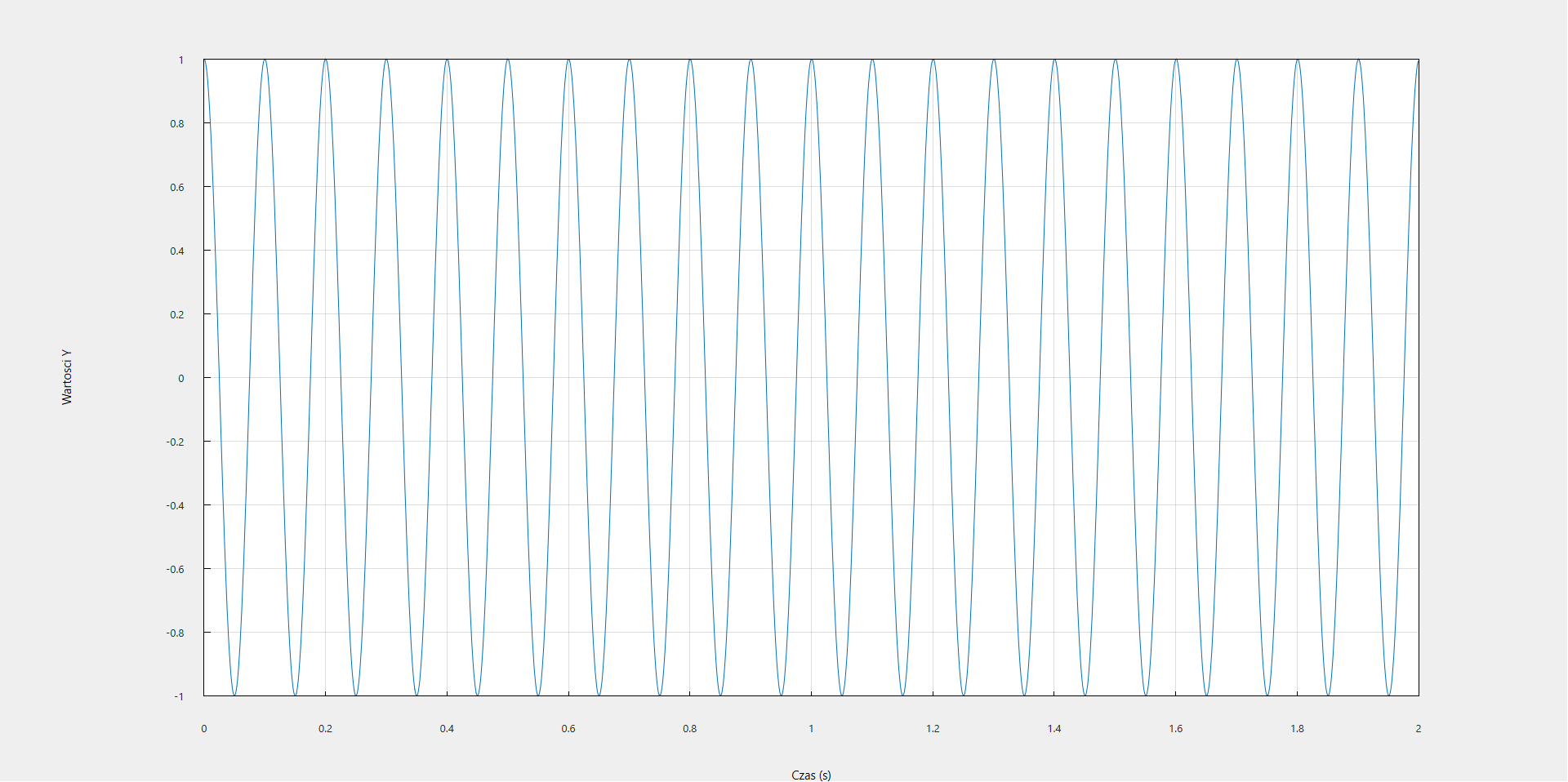
>>> b.visualize\_signal(sygnal,1000,0,2)



1. Generowanie cos()

>>> sygnal = b.generate\_cos(10,1000,0,2)

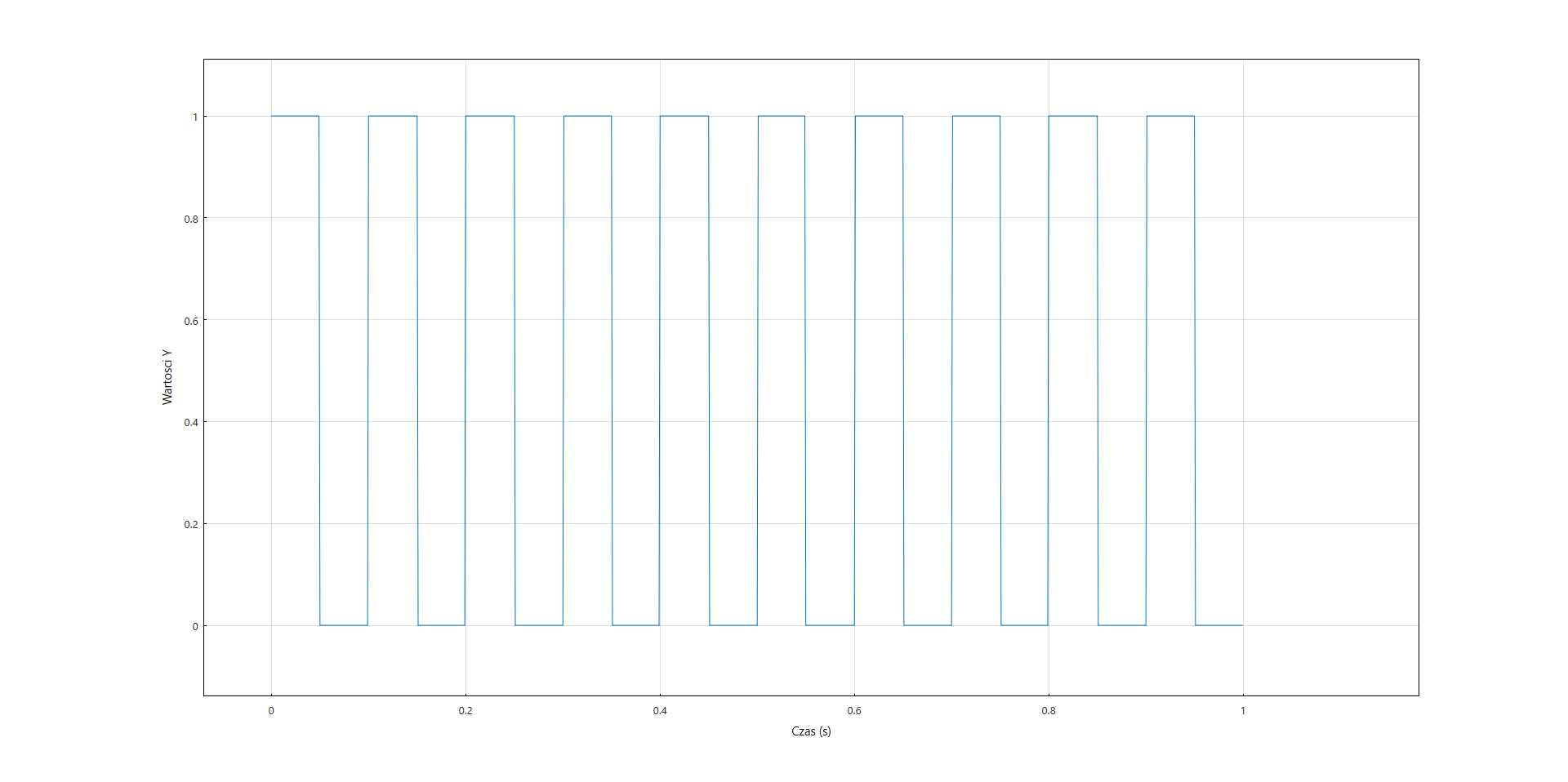
>>> b.visualize\_signal(sygnal,1000,0,2)



1. Generowanie sygnału prostokątnego

>>> sygnal = b.generate\_square(10,1000,0,1,1,0.5)

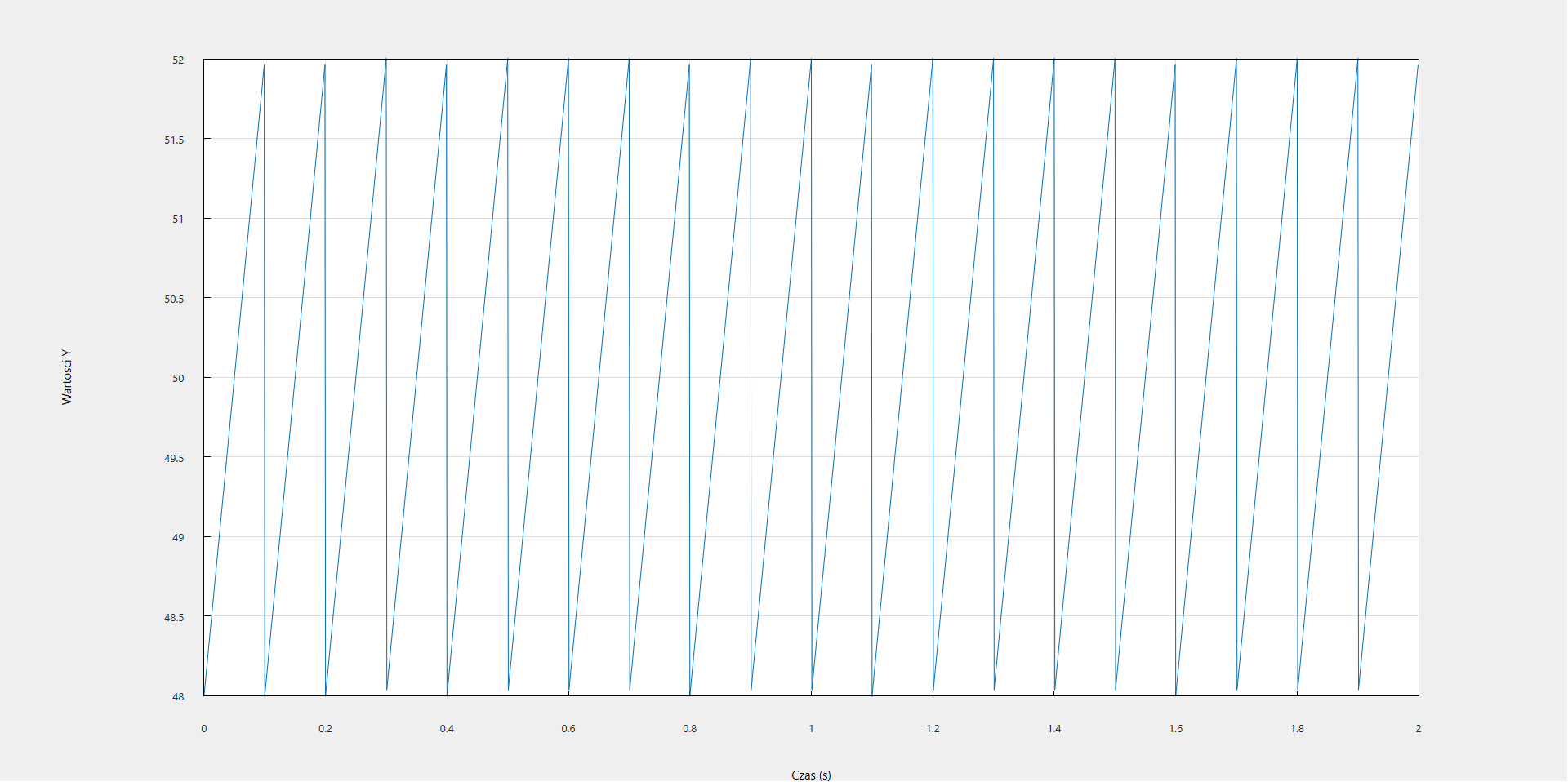
>>> b.visualize\_signal(sygnal,1000,0,1)



4.Generowanie sygnału piłokształtnego

>>> sygnal = b.generate\_sawtoothe(10,1000,0,2,2,50)

>>> b.visualize\_signal(sygnal,1000,0,2)



# 

# 

# 

# 

# 

# 

# 

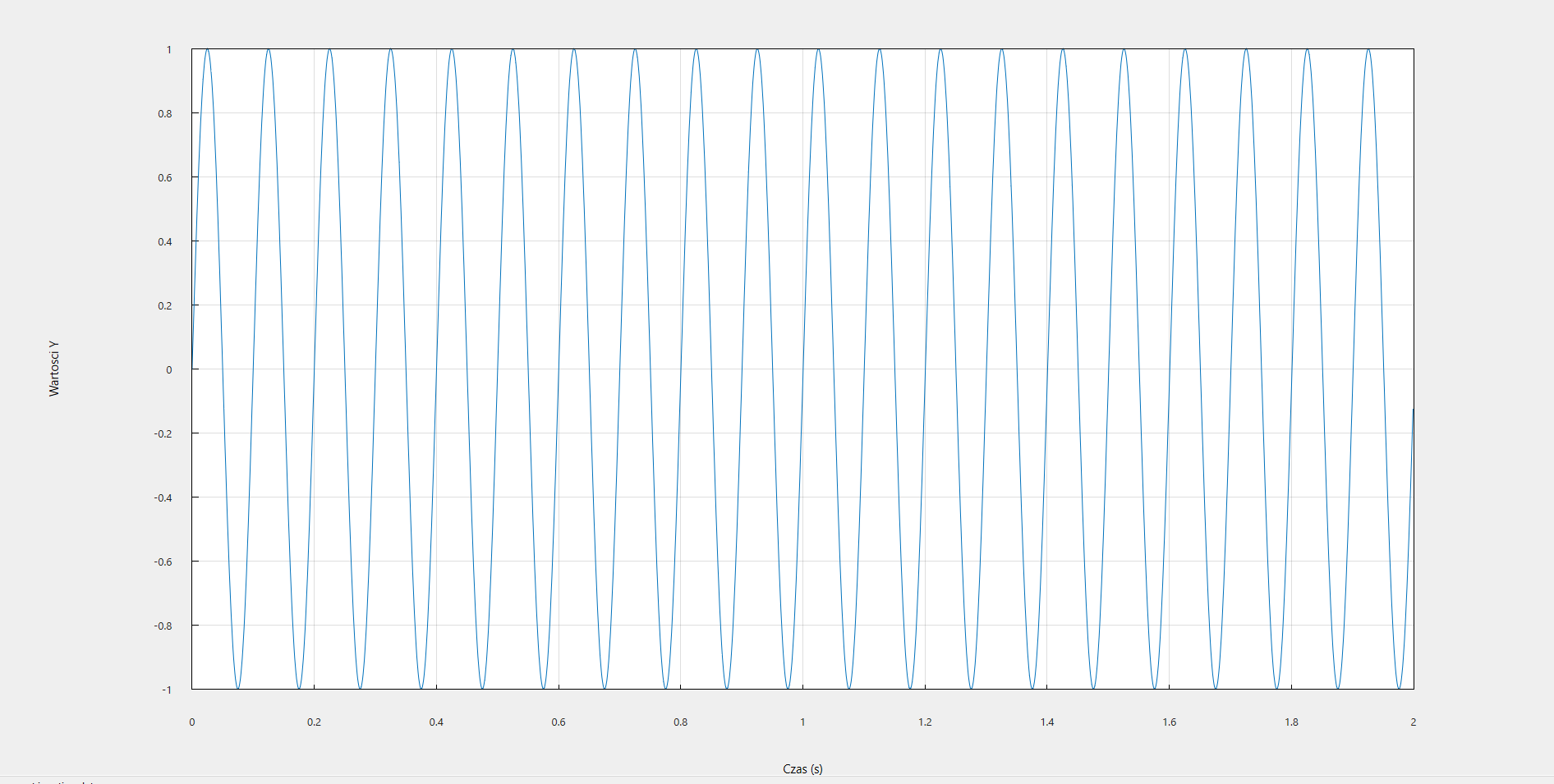
# 

# DFT i transformata odwrotna

Najpierw generujemy sygnał sinusoidalny:

>>> sygnal = b.generate\_sin(10,1000,0,2)

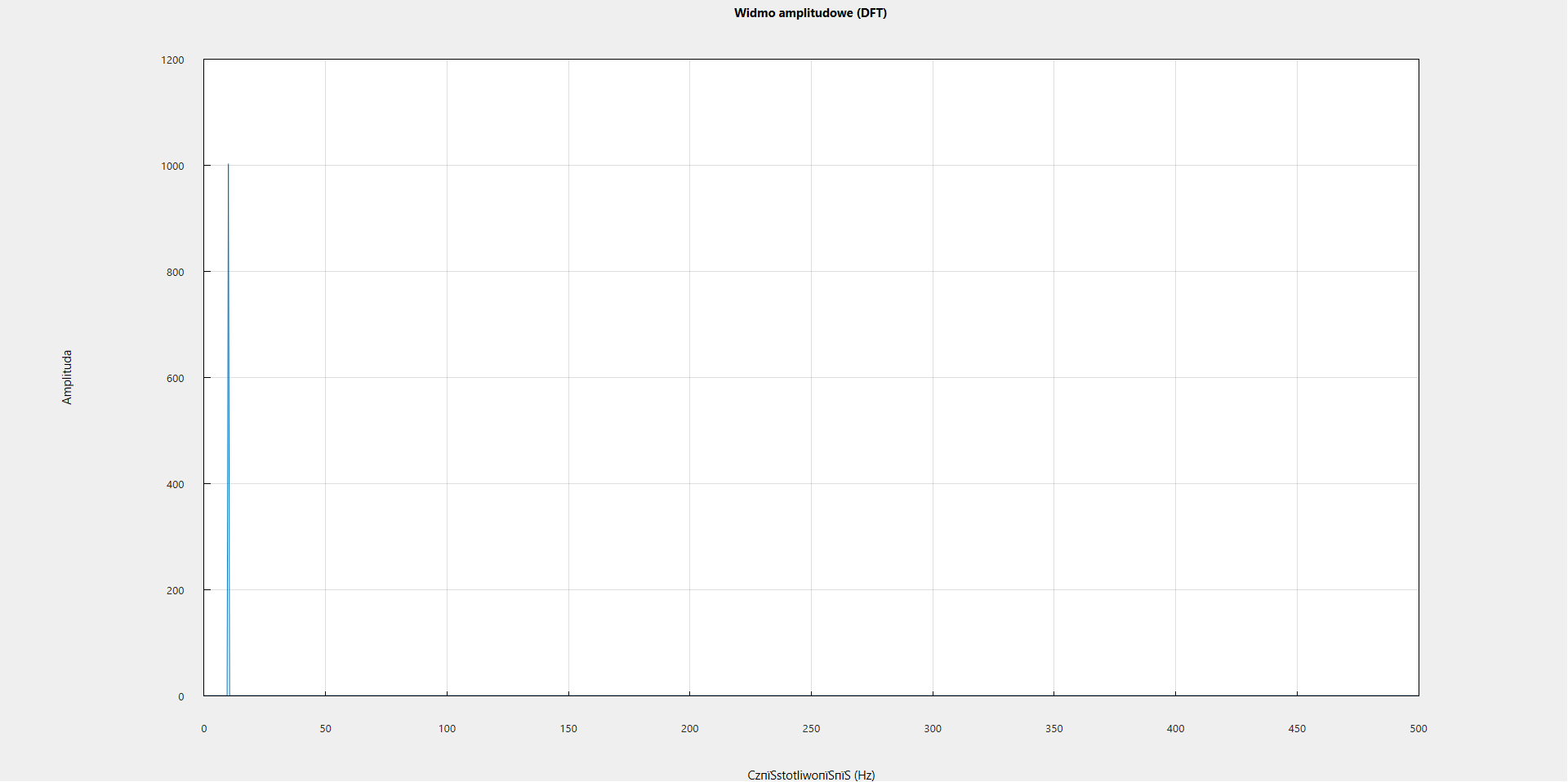
>>> b.visualize\_signal(sygnal,1000,0,2)



DFT sygnału sinusoidalnego:

>>> sygnaldft = b.dft(sygnal)

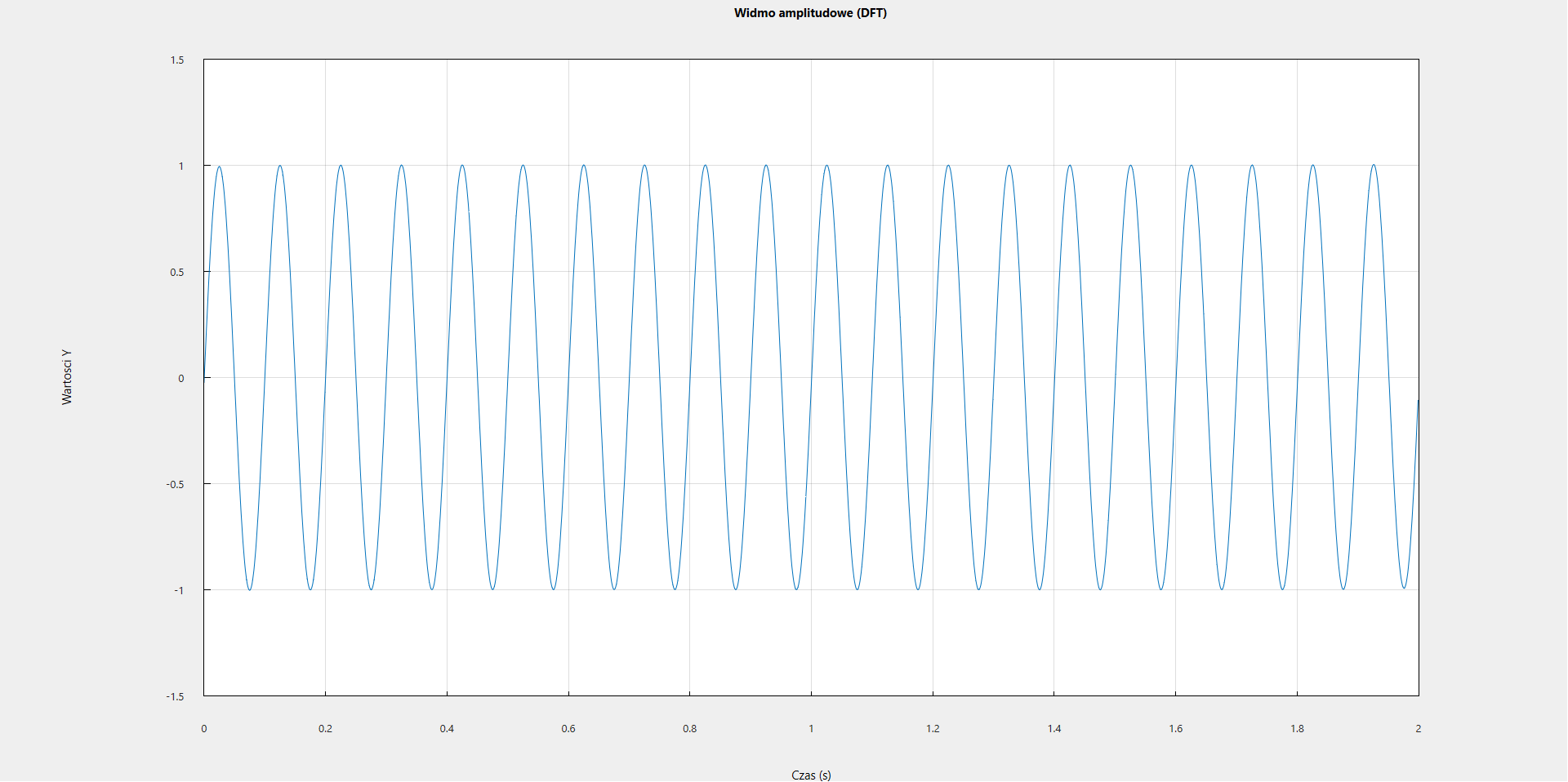
>>> b.visualize\_signal\_dft(sygnaldft, 1000)



Transformata odwrotna:

>>> sygnalidft = b.idft(sygnaldft)

>>> b.viusalize\_signal(sygnalidft,1000,0,2)



Otrzymaliśmy ten sam wykres sinusoidalny co na początku.

# Progowanie sygnału

Progujemy sygnał sinusoidalny z pierwszego zadania. Próg jest zerowy.

>>> sygnal = b.generate\_sin(10,1000,0,2)

>>> progowanie = b.progowanie(sygnal,0)

>>> b.visualize\_signal(progowanie,1000,0,2)

