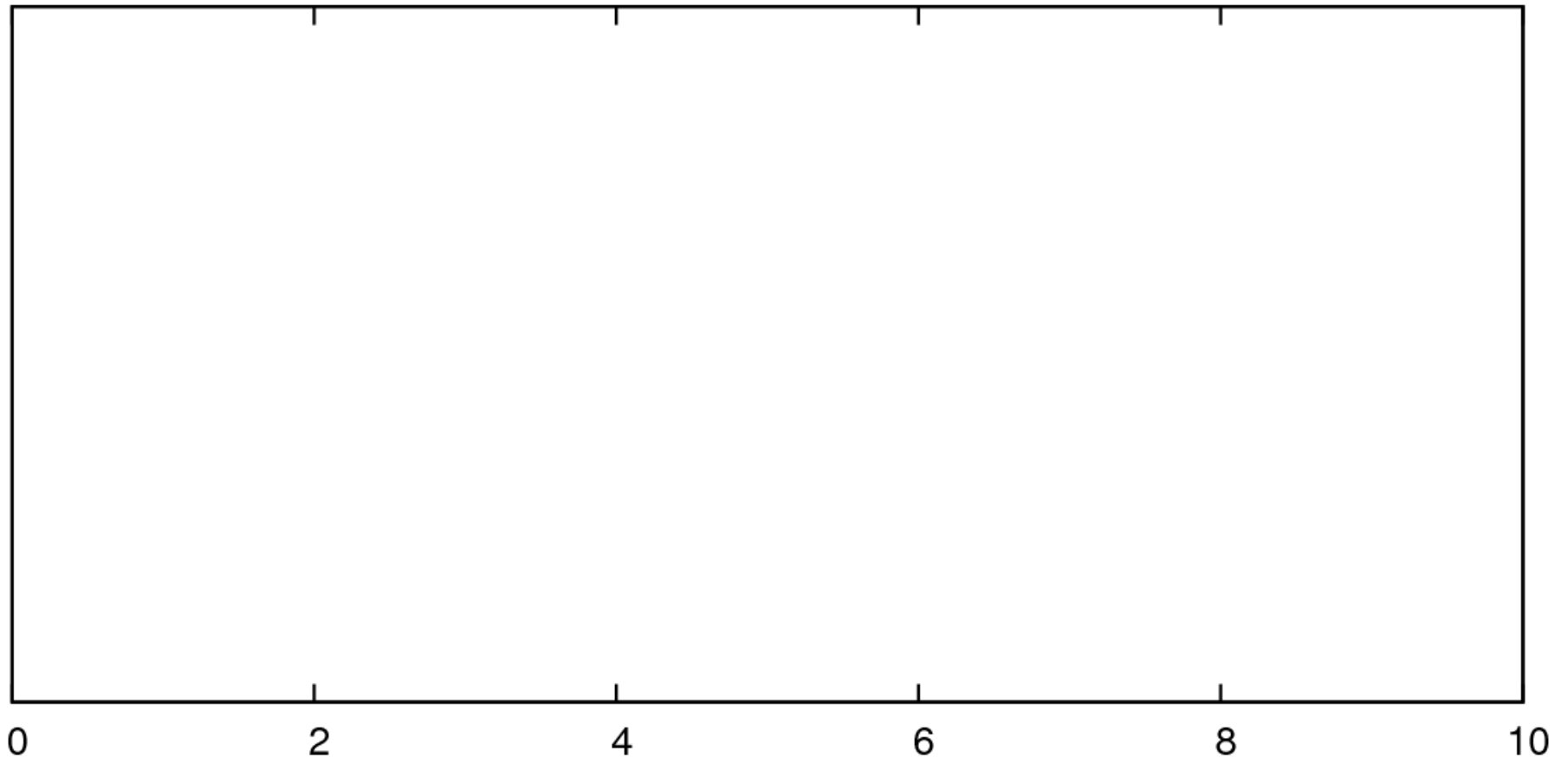


Generowanie sygnału

```
delta_t = 1.0/Fs
```

```
t = numpy.arange(0, T, delta_t)
```

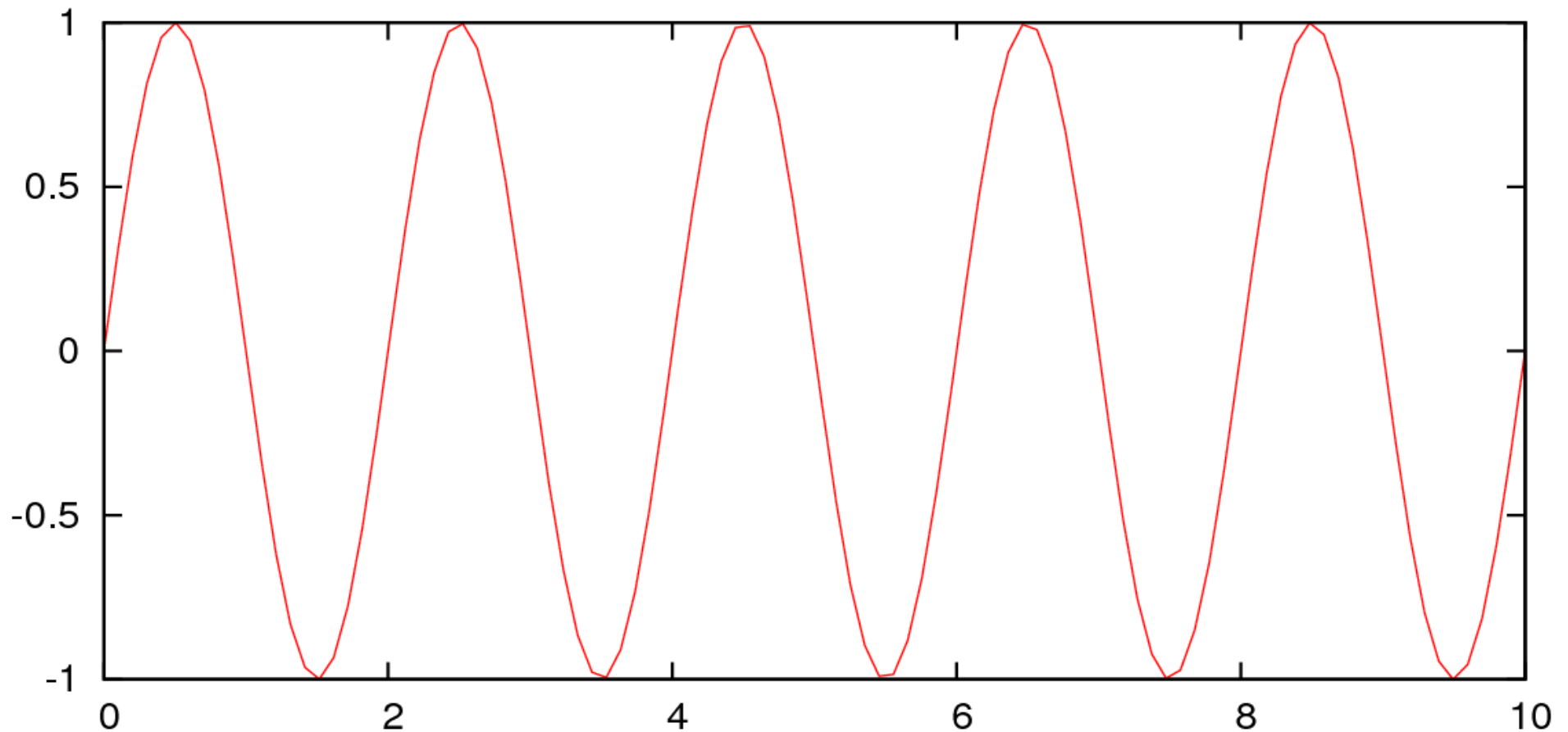


Generowanie sygnału

```
delta_t = 1.0/Fs
```

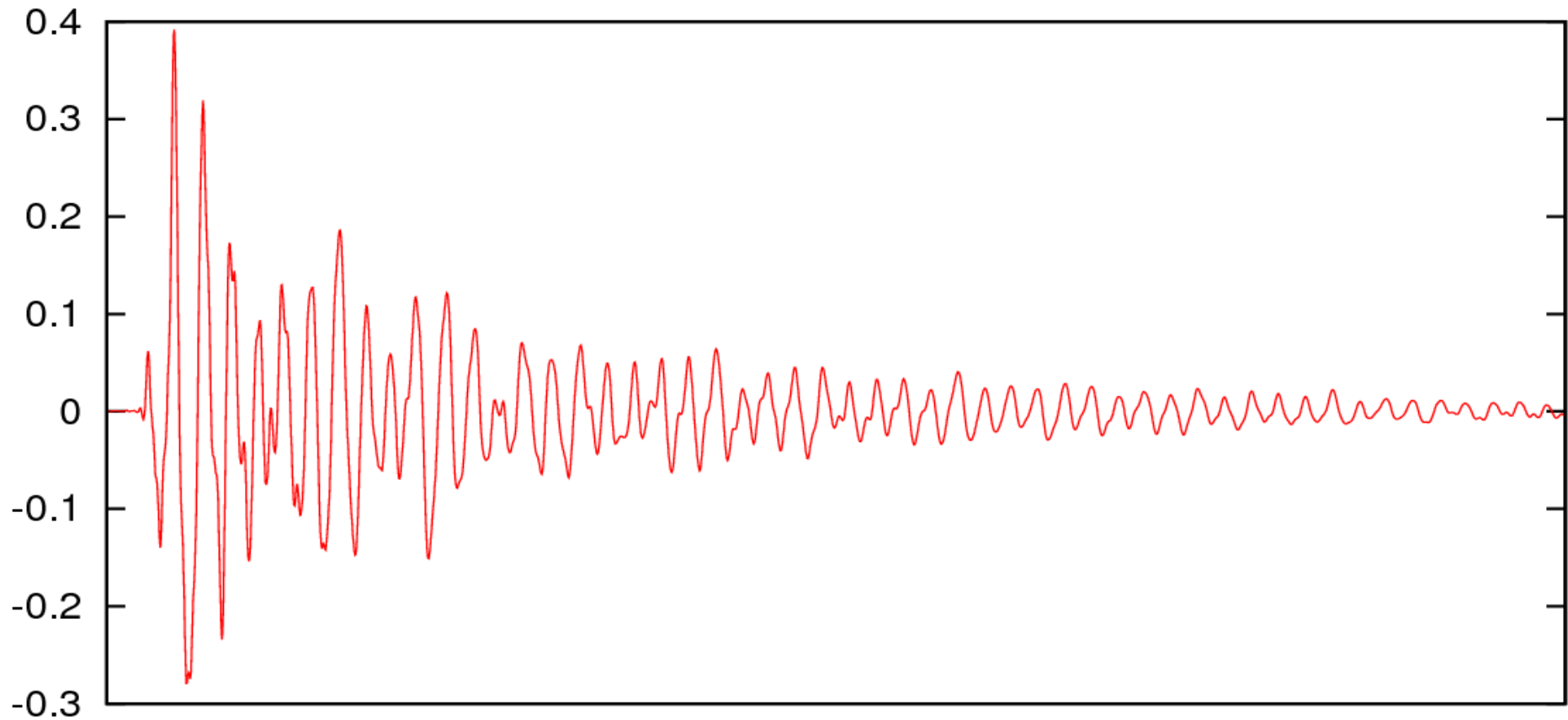
```
t = numpy.arange(0, T, delta_t)
```

```
x = numpy.sin(2*numpy.pi*f*t)
```



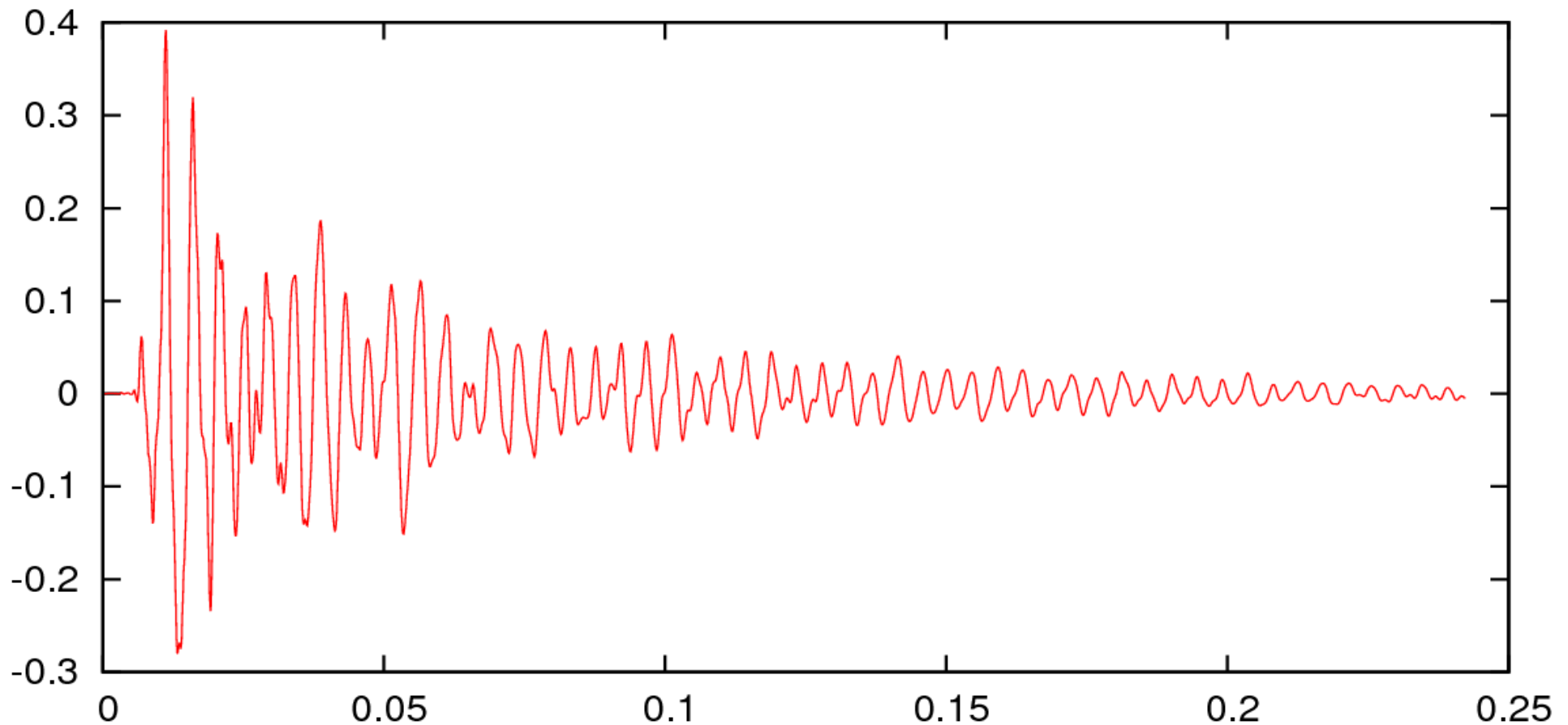
Wczytywanie sygnału

```
x = numpy.fromfile('pc.raw')
```



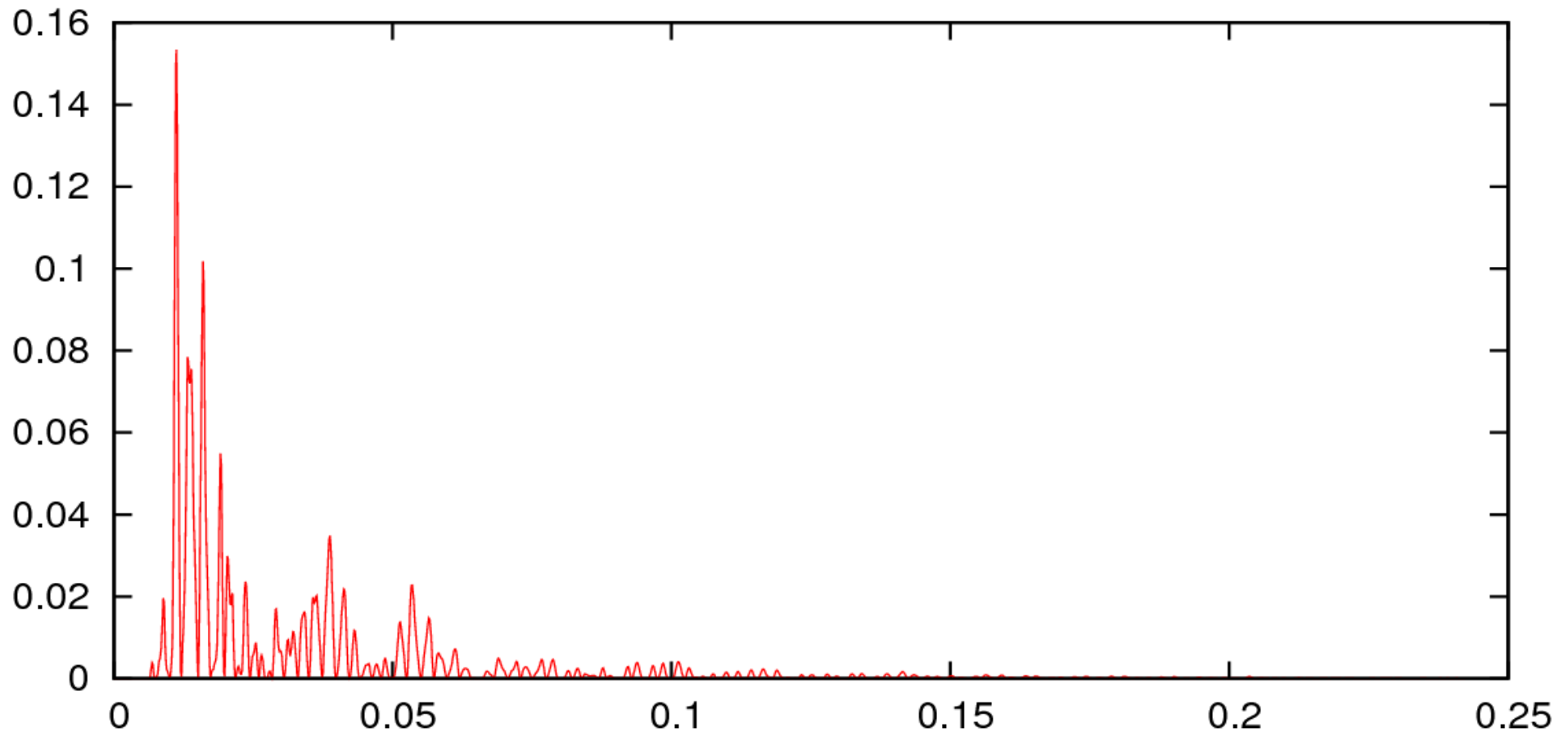
Wczytywanie sygnału

```
x = numpy.fromfile('pc.raw')  
delta_t = 1.0/Fs  
t = numpy.arange(len(x)) * delta_t
```



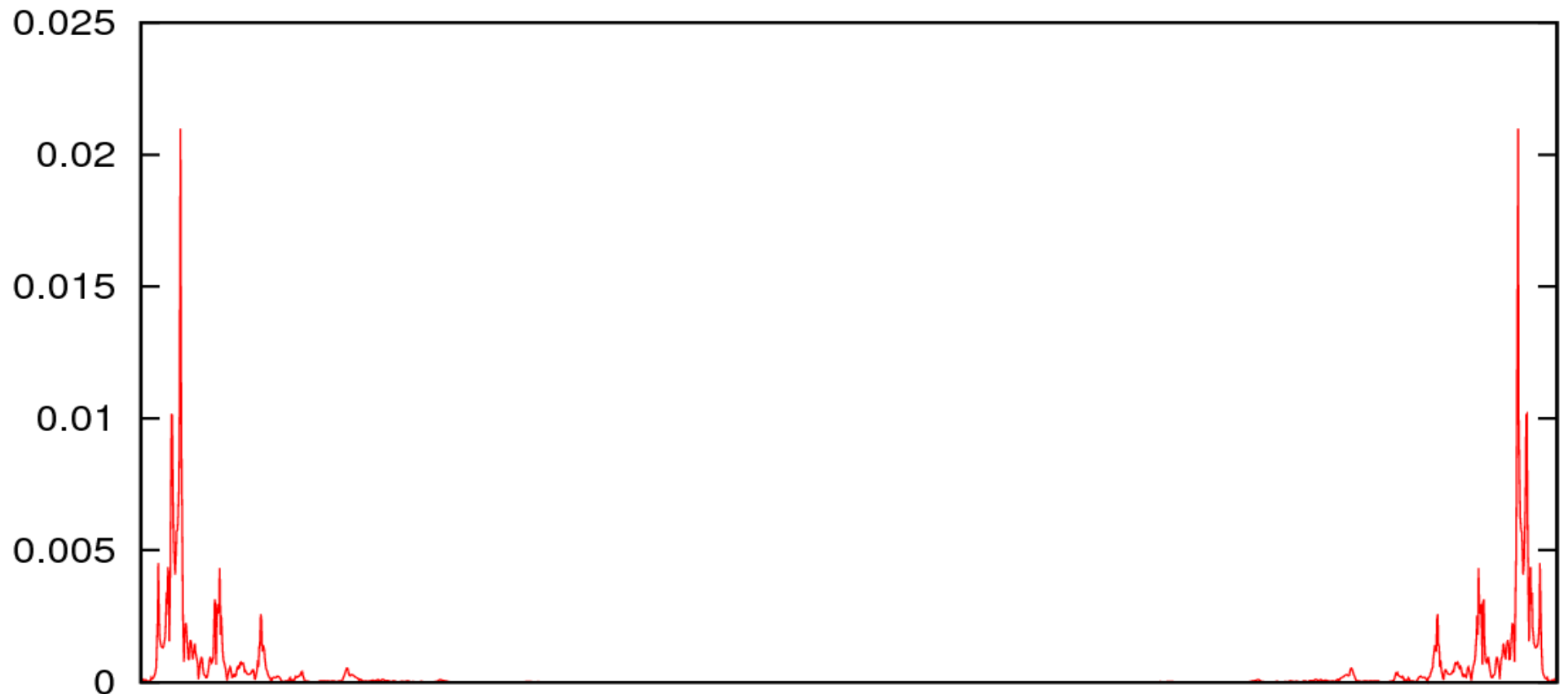
Moc chwilowa sygnału

```
moc_chwilowa = numpy.abs(x)**2  
# nadal zależność od t
```



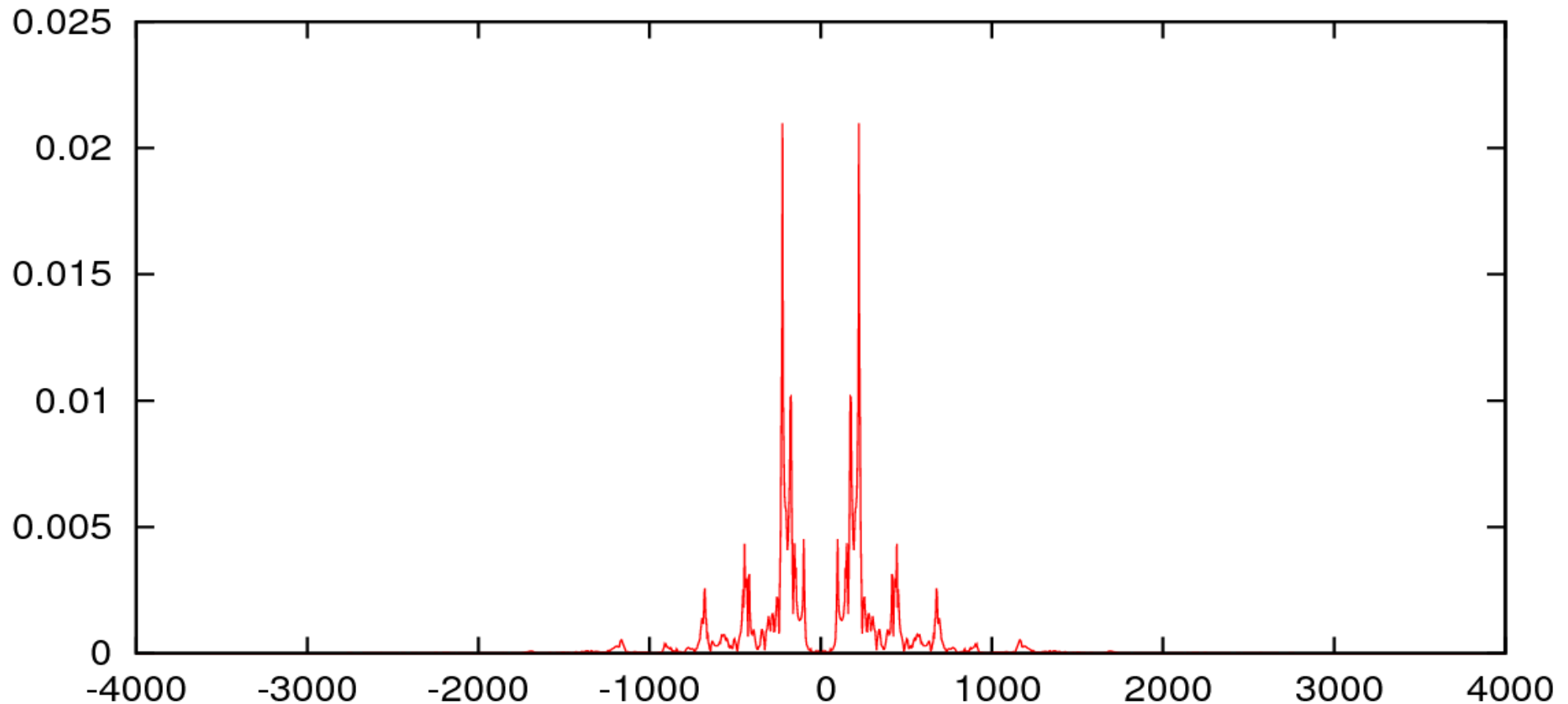
Widmo amplitudowe

```
X = numpy.fft.fft(x)  
X_amp1 = abs(X) / len(X)
```



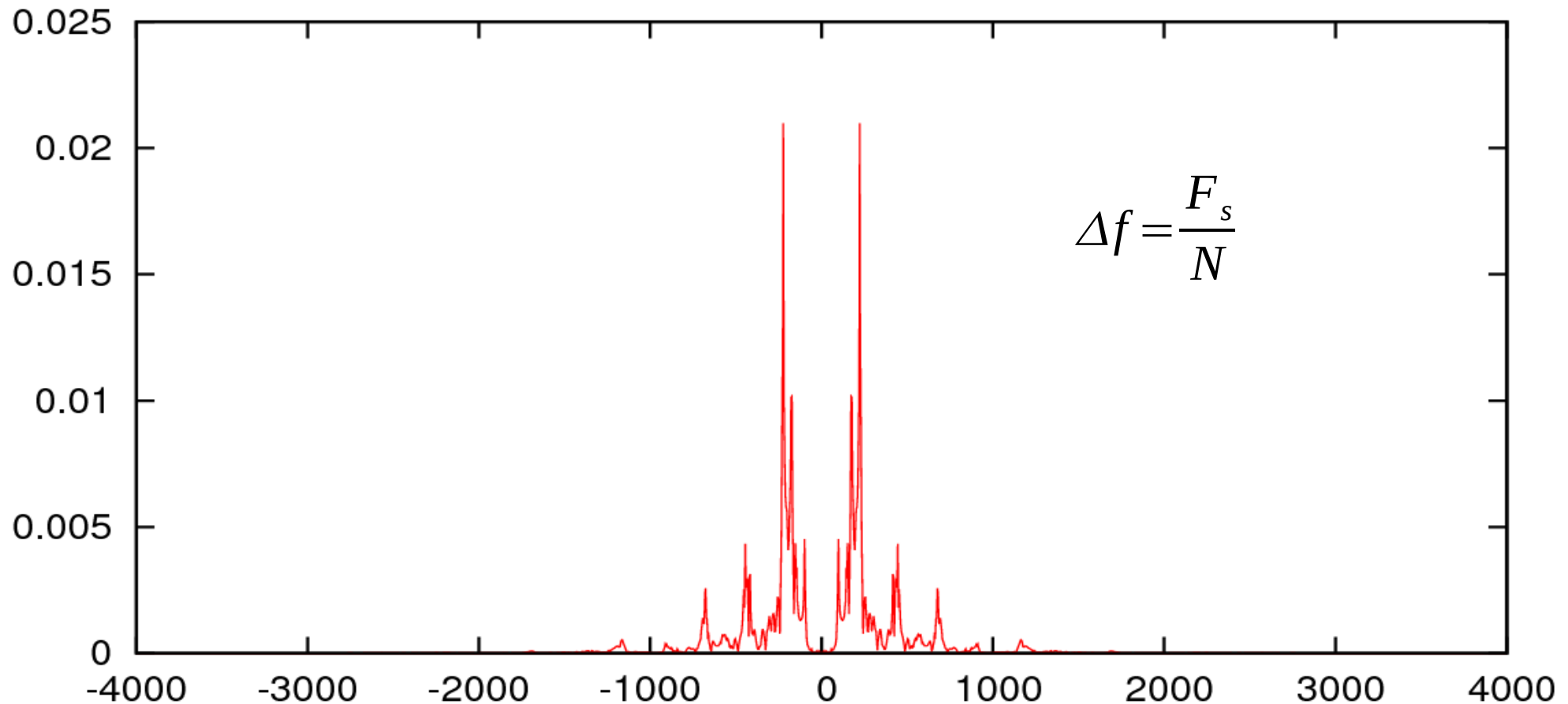
Widmo amplitudowe

```
X = numpy.fft.fft(x)
X_amp1 = abs(X) / len(X)           (delta_t)
freq = numpy.fft.fftfreq(len(X), 1.0/Fs)
```



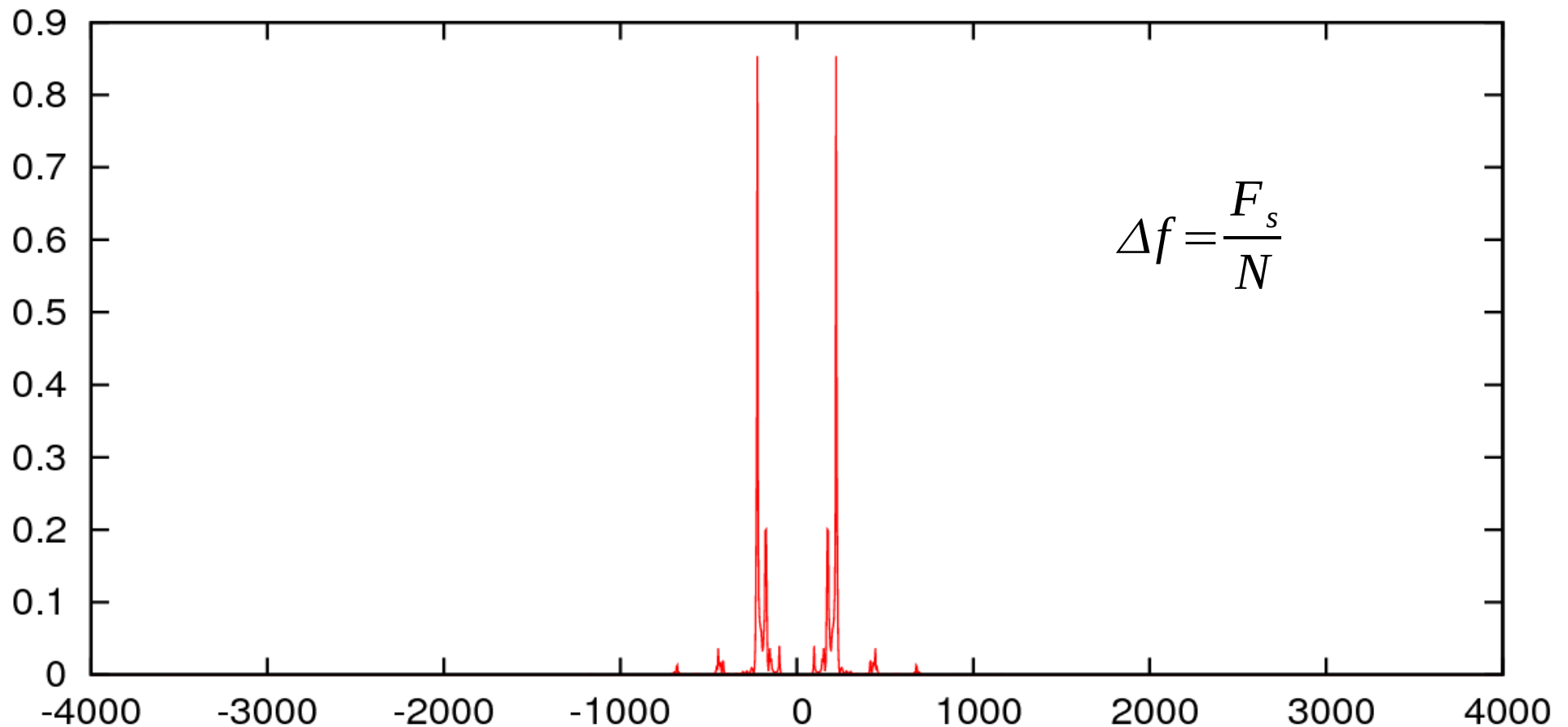
Widmo amplitudowe

```
X = numpy.fft.fft(x)
X_amp1 = abs(X) / len(X)          (delta_t)
freq = numpy.fft.fftfreq(len(X), 1.0/Fs)
```



Widmo mocy

```
X = numpy.fft.fft(x)
X_moc = abs(X)**2 / len(X)          (delta_t)
freq = numpy.fft.fftfreq(len(X), 1.0/Fs)
```



Zadanie (proste)

- Ile wynosi suma całego przebiegu mocy chwilowej?

$$E_1 = \sum_i |x_i|^2$$

- Ile wynosi suma całego widma mocy?

$$E_2 = \sum_i \left(\frac{1}{N} |X_i|^2 \right) = \frac{1}{N} \sum_i |X_i|^2$$

- Proszę obliczyć dla przykładowego sygnału (pc.raw, $F_s = 8000$ Hz)

Twierdzenie Parsevala

$$\sum_i |x_i|^2 = \frac{1}{N} \sum_i |X_i|^2$$

gdzie $X = \text{DFT}(x)$.

Twierdzenie Parsevala

$$\sum_i |x_i|^2 = \frac{1}{N} \sum_i |X_i|^2$$

gdzie $X = \text{DFT}(x)$.

- Całkowita energia sygnału

$$E = \frac{1}{F_s} \sum_i |x_i|^2 = \frac{1}{F_s N} \sum_i |X_i|^2$$

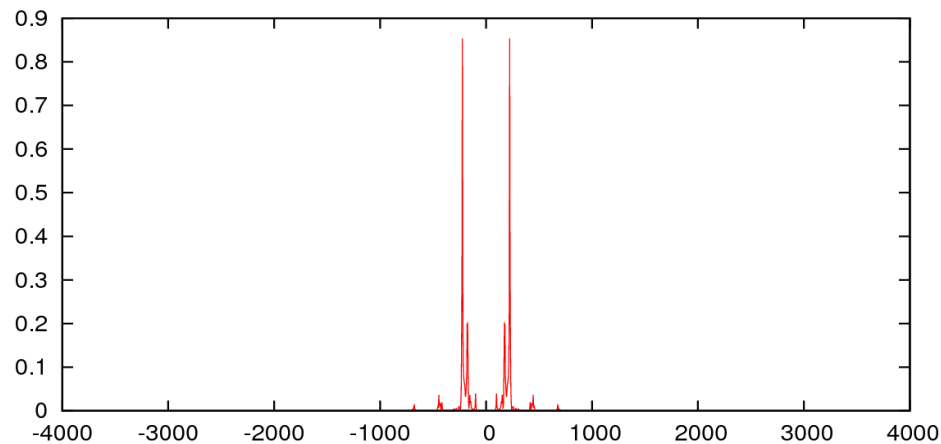
Zadanie

- Wygeneruj sygnał „sinus” na odcinku $T = 5 \text{ s}$ o częstotliwości $f = 0.3 \text{ Hz}$ z częstotliwością próbkowania $F_s = 10 \text{ Hz}$
- Wykreśl periodogram (widmo mocy) dla tego sygnału w skali decybelowej $= 10 \log_{10}(\text{moc})$
- + okienkowanego znormalizowanym oknem
 - Blackmana
 - Hamminga
- + dla sygnału przedłużonego zerami do długości $N = 1000$

$$\hat{w}[n] = \frac{1}{\sqrt{\sum_{n=0}^{N-1} (w[n])^2}} w[n]$$

Znajdowanie maksimum widma

```
X = numpy.fft.fft(x)
X_moc = abs(X)**2 / len(X)          (delta_t)
freq = numpy.fft.fftfreq(len(X), 1.0/Fs)
```



$$\Delta f = \frac{F_s}{N}$$

```
i_max = numpy.argmax(X_moc)
moc_max = X_moc[i_max]
freq_max = freq[i_max] # lub delta_f * i_max
```