# Sinusy są magiczne

# Sinusy są magiczne

- Każdą funkcję okresową o okresie **T** można przedstawić w postaci szeregu funkcji **sinus** i **cosinus** o okresach **T**, 2**T**, 3**T**, 4**T**, …

  (częstościach 1/T, 2/T, 3/T, 4/T, …)

  z odpowiednimi współczynnikami

# Sinusy są magiczne

- Każdą funkcję okresową o okresie **T** można przedstawić w postaci szeregu funkcji **sinus** i **cosinus** o okresach **T**, 2**T**, 3**T**, 4**T**, …

  (częstościach 1/T, 2/T, 3/T, 4/T, …)

  z odpowiednimi współczynnikami

- Ponieważ

$$e^{ix} = \cos x + i \sin x \quad \Rightarrow \begin{cases} \cos x = \frac{1}{2}(e^{ix} + e^{-ix}) \\ \sin x = \frac{1}{2i}(e^{ix} - e^{-ix}) \end{cases}$$

  to możemy stwierdzić, że…

# $e^{2\pi ift}$ są magiczne

- Każdą funkcję okresową x(t) o okresie **T** można przedstawić w postaci szeregu funkcji $X(t) = \exp(2\pi i\mathbf{f}t)$,

  gdzie f = 0, ±1/T, ±2/T, ±3/T, …

  z odpowiednimi zespolonymi współczynnikami

# $e^{2\pi i f t}$ są magiczne

- Każdą funkcję okresową x(t) o okresie **T** można przedstawić w postaci szeregu funkcji $X(t) = \exp(2\pi i \mathbf{f} t)$,

  gdzie f = 0, ±1/T, ±2/T, ±3/T, …

  z odpowiednimi zespolonymi współczynnikami, czyli…

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k \exp\left(2\pi i \frac{k}{T} t\right)$$

# Sygnały ciągłe

- Każdą funkcję okresową x(t) o okresie **T** można przedstawić w postaci szeregu funkcji $X(t) = \exp(2\pi i \mathbf{f} t)$,

  gdzie f = 0, ±1/T, ±2/T, ±3/T, …

  z odpowiednimi <u>zespolonymi</u> współczynnikami, czyli…

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k \exp\left(2\pi i \frac{k}{T} t\right)$$

**SZEREG FOURIERA**

# Sygnały dyskretne

- Każdą funkcję okresową x(t) o okresie **T** można przedstawić w postaci szeregu funkcji $X(t) = \exp(2\pi i \mathbf{f} t)$,

  gdzie f = 0, ± 1/T, ± 2/T, ± 3/T, … ± ½ $\mathbf{f_s}$

  z odpowiednimi zespolonymi współczynnikami, czyli…

$$x(t) = \sum_{k=-?}^{+?} c_k \exp\left(2\pi i \frac{k}{T} t\right)$$

# … liczymy …

- t = 0, Δt, 2Δt, 3Δt, …

  $\Delta t = 1 / f_s$

- $t = n \cdot \Delta t$
- $T = N \cdot \Delta t$

$$2\pi i k \frac{t}{T} = 2\pi i k \frac{n \Delta t}{N \Delta t} = 2\pi i k n / N$$

$$\frac{½ f_s}{1/T} = ½ f_s \cdot N \Delta t = ½ f_s N \frac{1}{f_s} = ½ N$$

# Sygnały dyskretne

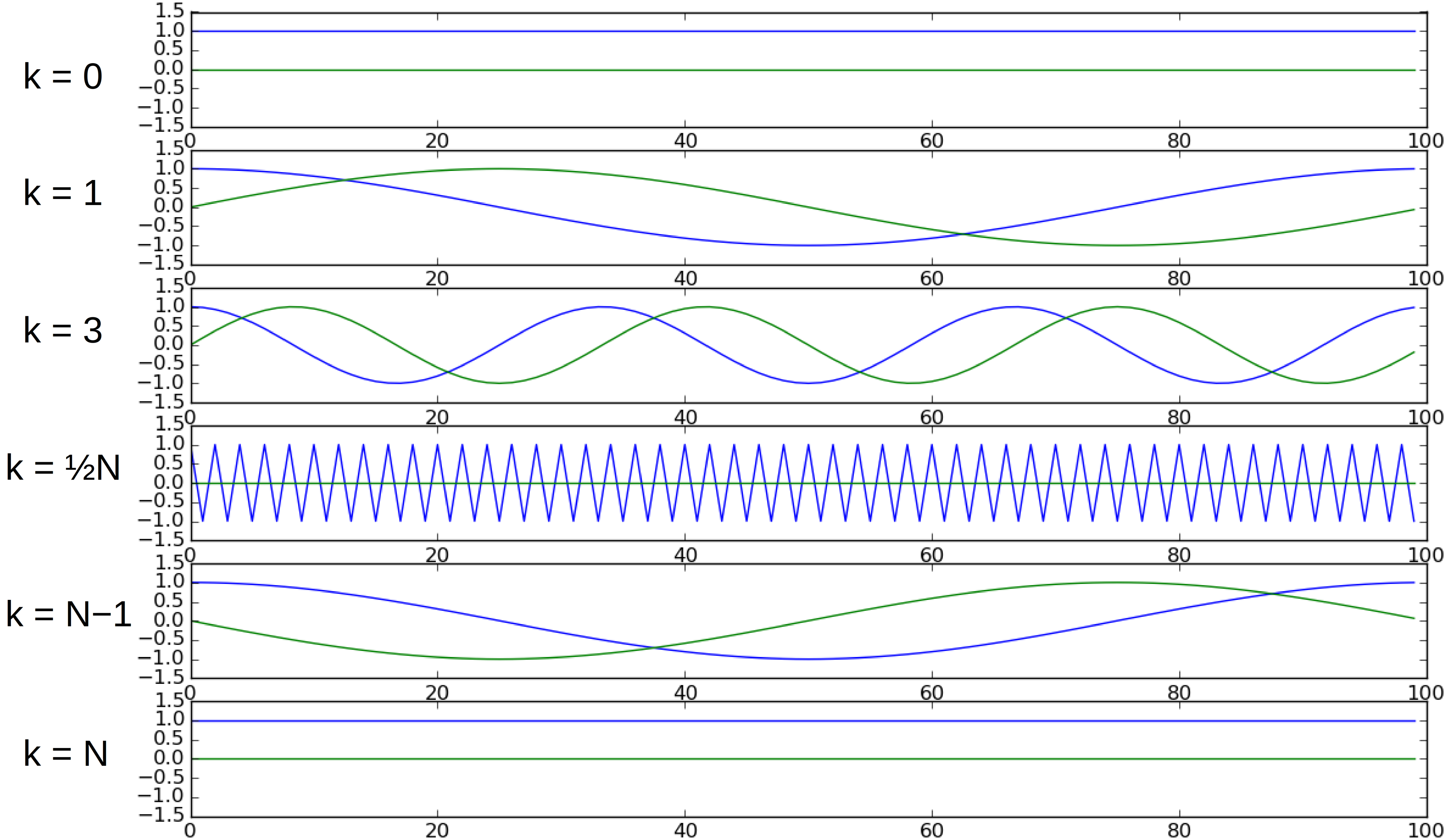
- Każdą funkcję okresową x(n) o okresie **N** można przedstawić w postaci szeregu funkcji $X(n) = \exp(2\pi ikn/N)$,

  gdzie k = 0, ± 1, ± 2, ± 3, … ± ½ N

  z odpowiednimi <u>zespolonymi</u> współczynnikami, czyli…

$$x(n)=\sum_{k=-N/2}^{+N/2} c_k \exp(2\pi i k n/N)$$

# Przypomnienie: aliasing



# Sygnały dyskretne

- Każdą funkcję okresową x(n) o okresie **N** można przedstawić w postaci szeregu funkcji $X(n) = \exp(2\pi ikn/N)$,

  gdzie k = 0, 1, 2, 3, … N−1

  z odpowiednimi <u>zespolonymi</u> współczynnikami, czyli…

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} c_k \exp(2\pi i k n / N)$$

**ODWROTNA TRANSFORMATA FOURIERA**

# Skąd wzią~~ś~~ć współczynniki

- Dyskretna Transformata Fouriera sygnału $x_n$ o długości **N**

  - **odwrotna**

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} c_k \exp\left(2\pi i k n / N\right)$$

# Skąd wzią~~ś~~ć współczynniki

- Dyskretna Transformata Fouriera (DTF / DFT) sygnału $x_n$ o długości **N**
  - **zwykła**

$$c_k = \sum_{n=0}^{N-1} x_n \exp\left(-2\pi i k n / N\right)$$

  - **odwrotna**

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} c_k \exp\left(2\pi i k n / N\right)$$

# Przykład

- Sygnał o długości 128 próbek, składający się z 28 jedynek i 100 zer.

```
x1 = numpy.ones(28)
x0 = numpy.zeros(100)
x = numpy.concatenate((x1,x0))
```

- Proszę wyświetlić sygnał.

# Przykład

- Sygnał o długości 128 próbek, składający się z 28 jedynek i 100 zer.

  ```
  x1 = numpy.ones(28)
  x0 = numpy.zeros(100)
  x = numpy.concatenate((x1,x0))
  ```

- Proszę obliczyć DTF:
  - bezpośrednio ze wzoru

- Proszę wyświetlić sygnał oraz amplitudę (przy użyciu funkcji `numpy.abs`) transformaty.

# Przykład

- Sygnał o długości 128 próbek, składający się z 28 jedynek i 100 zer.

  ```
  x1 = numpy.ones(28)
  x0 = numpy.zeros(100)
  x = numpy.concatenate((x1,x0))
  ```

- Proszę obliczyć DTF dwoma sposobami:
  - bezpośrednio ze wzoru
  - z użyciem funkcji `numpy.fft.fft(x)`

- Proszę wyświetlić sygnał oraz amplitudę (przy użyciu funkcji `numpy.abs`) obu transformat.

# Przykład

- Sygnał o długości 128 próbek, składający się z 28 jedynek i 100 zer.

```
x1 = numpy.ones(28)
x0 = numpy.zeros(100)
x = numpy.concatenate((x1,x0))
```

- Proszę obliczyć DTF trzema sposobami:
  - bezpośrednio ze wzoru
  - z użyciem funkcji `numpy.fft.fft(x)`
  - z użyciem funkcji `numpy.fft.rfft(x)`
- Proszę wyświetlić sygnał oraz amplitudę (przy użyciu funkcji `numpy.abs`) transformat.