

Laboratorium 1

Słowa kluczowe: Próbkowanie, akwizycja, normalizacja, kwantyzacja, zaszumienie i odszumienie sygnału

Biblioteki: *numpy, matplotlib*

Zadanie 1:

- Zainicjalizuj wykres (3 kolumny x 3 wiersze) o wymiarach 10×10 cali.
- Spróbuj funkcję sinus w zakresie $0-4\pi$ w 40 punktach.
 - Podpowiedź: W tym celu należy zadeklarować wartości z przedziału X (zmienną nazwać `x`) poleceniem `np.linspace` w podanym zakresie (od 0 do 4π), o podanej liczbie kwantów (40).
 - Następnie dla każdej z wartości wywołać funkcję sinus: `y = np.sin(x)`.
- W pierwszej komórce wykresu narysuj spróbkowaną funkcję przy użyciu polecenia `plot`.
- Utwórz dwuwymiarowe złożenie spróbkowanej funkcji.
 - Podpowiedź: Ponieważ `y` jest jednowymiarowym wektorem należy użyć `np.newaxis` lub alternatywnie `None` do adresacji w nowym wymiarze:

```
y[:, np.newaxis]*y[np.newaxis, :].
```

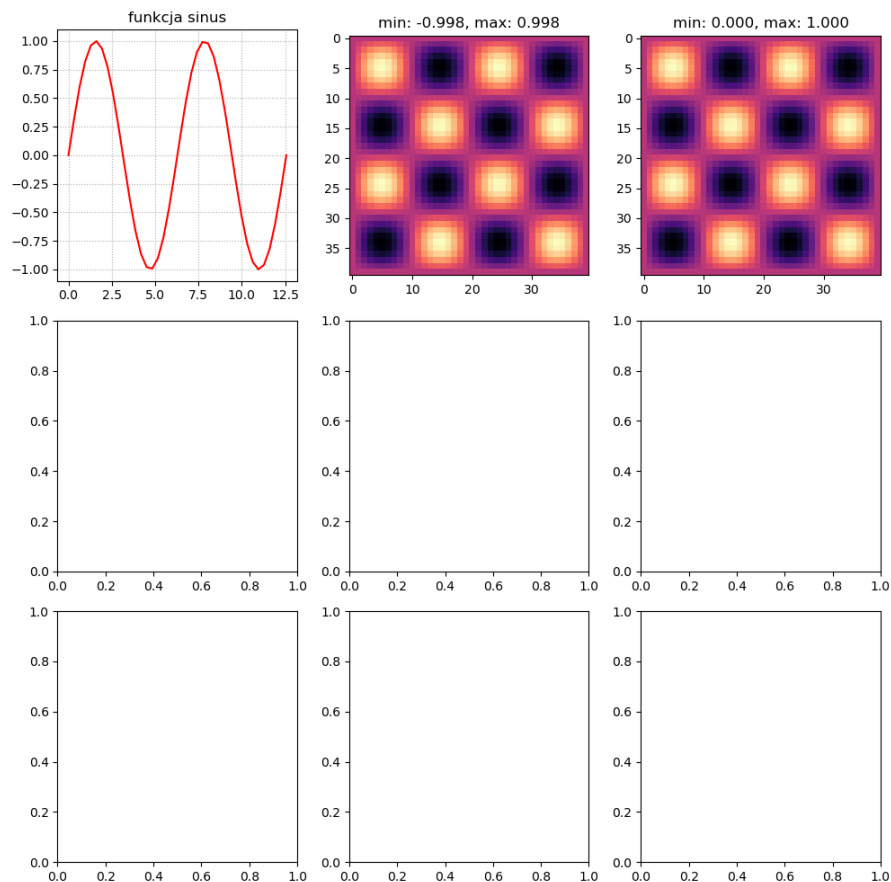
- Otrzymany obraz zaprezentuj w drugiej komórce wykresu wykorzystując `imshow`. Użyj colormapy *magma*.
- W podpisie umieść maksymalną i minimalną wartość intensywności obrazu, wyznaczone przy użyciu funkcji `np.min` oraz `np.max`, z dokładnością do 3 miejsc po przecinku.
 - Podpowiedź: Mamy w *pythonie* wiele sposobów na zagnieżdżenie wartości w ciągu znaków, a tutaj jest jedna z prostszych:

```
'min: %.3f, max: %.3f' % (value_min, value_max).
```

- W powyższym przypadku w miejsce `%.3f` wpisane będą wartości z `value_min` oraz `value_max` z dokładnością do 3 miejsc po przecinku.
- Wykonaj *normalizację przedziałową* obrazu do zakresu 0-1. Po tej operacji wartość maksymalna intensywności powinna wynosić 1, a minimalna 0.
 - Podpowiedź: Należy w tym celu wykonać dwie proste operacje arytmetyczne — pierwszą wykorzystującą wartość minimalną, a drugą wykorzystującą wartość maksymalną.
 - Proszę nie korzystać z gotowych metod do normalizacji — da się to zrobić w dwóch liniach.

- Znormalizowany obraz, wraz z podpisem zawierającym wartość minimalną i maksymalną, umieść w kolejnej komórce pierwszego wiersza wykresu.

Efekt zadania 1:

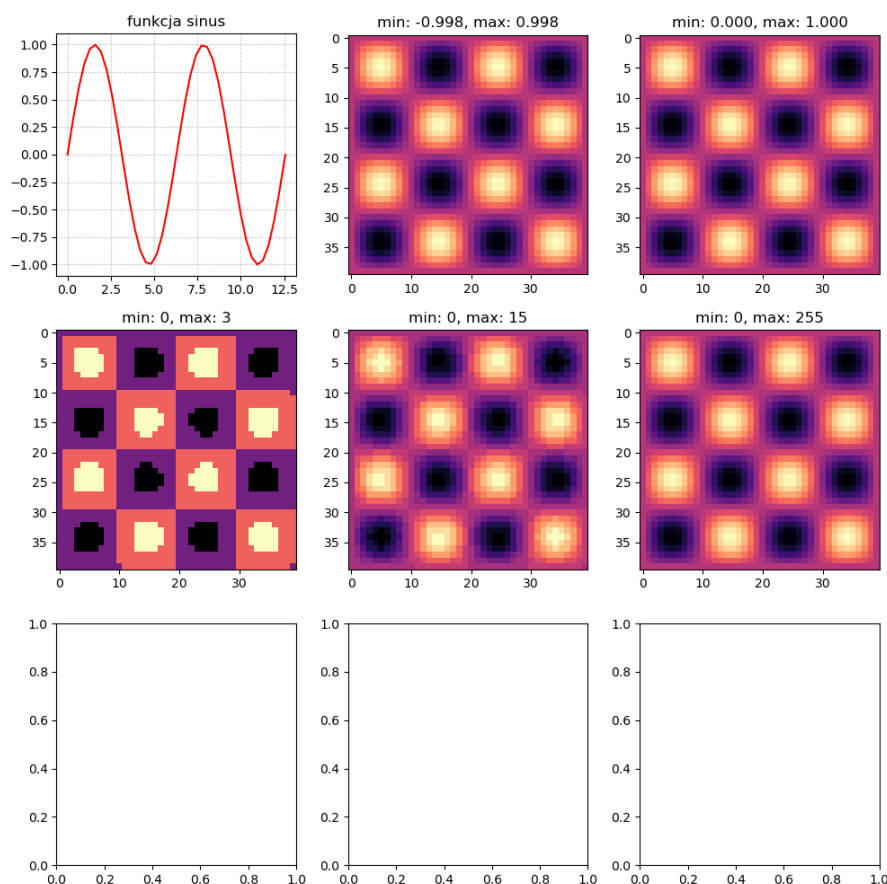


Zadanie 2:

Kolejnym etapem jest *dyskretyzacja* otrzymanego po normalizacji obrazu — czyli sygnał będzie *cyfrowy* lub *skwantowany*.

- W drugim wierszu wykresu narysuj sygnał z zadania pierwszego kolejno o głębi 2, 4 i 8-bitowej.
 - Podpowiedź: Do wygenerowania takiego obrazu należy w pierwszej kolejności obliczyć maksymalną wartość intensywności w danej reprezentacji oraz *przeskalować* sygnał (zwykłe mnożenie).
 - Następnie, aby wartości sygnału były typu całkowitoliczbowego należy użyć funkcji `np rint`.
- W podpisach wyświetlić wartość minimalną i maksymalną sygnału.

Efekt zadania 2:



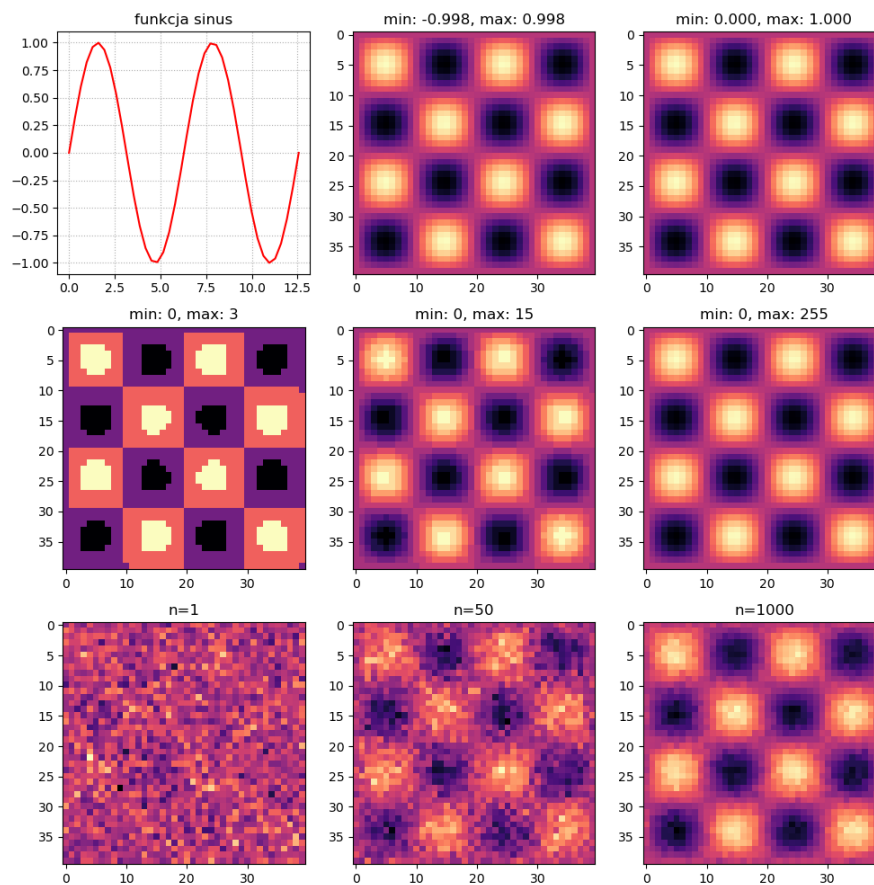
Zadanie 3:

- Zaszum sygnał po normalizacji (ten, który był na końcu zadania pierwszego) za pomocą jednokrotnie pobranego szumu normalnego.
 - Aby to zrobić wygeneruj szum z rozkładu normalnego przy użyciu funkcji `np.random.normal` o kształcie takim jak obraz.
 - Do obrazu dodaj wygenerowany szum.
 - Efekt narysuj w pierwszej kolumnie i trzecim wierszu wykresu.
- Aby odszumić sygnał, należy wykorzystać *wielokrotną akwizycję*.
 - Wygeneruj `n = [50, 1000]` zaszumionych rozkładem normalnym obrazów. Za każdym razem należy generować nowy, losowy szum.

Podpowiedź: Można tutaj wykorzystać pętlę `for` i zbierać zaszumione obrazy do listy lub tablicy.

- Odszumienie obrazu uzyskamy dzięki zsumowaniu n zaszumionych obrazów, i następnie obliczeniu ich średniej (przyda się funkcja `np.mean` i odpowiedni parametr *axis*).
- Wynik odszumienia dla 50 i 1000 akwizycji narysować w ostatnich dwóch komórkach wykresu.

Efekt zadania 3:



Co warto zapamiętać z tej instrukcji:

- Import bibliotek *numpy* i *matplotlib*.
- Inicjalizacja wykresu i prezentacja sygnałów (`subplots` , `plot` , `imshow`).
- Próbkowanie funkcji liniowej (`np.linspace`).
- Inicjalizacja nowego wymiaru tablicy (`np.newaxis`).
- Obliczenie wartości maksymalnej i minimalnej sygnału.
- Normalizacja przedziałowa.
- Kwantyzacja i zaokrąglenie do liczb całkowitych (`np rint`).
- Zaszumienie sygnału.
- Obliczenie średniej z wielu sygnałów.