Politechnika Wrocławska

Sprawozdanie 2

Sygnały i Obrazy Cyfrowe — Laboratorium

Interpolacja funkcji oraz skalowanie obrazów

Michał Białek
Nr indeksu: 264285
Grupa: 6, WT/NP godz. 18:55
Kod zajęć: W04ISA-SI0009G
Data: 05 grudzień 2023

1. Interpolacja funkcji

Pierwsze zadanie polega na generacji dodatkowych punktów dla zadanych funkcji, tak, aby nowe punkty były możliwie blisko oryginalnej funkcji. Generujemy funkcje podstawową o różnych ilościach punktów (N=100,200,400,1000) na konkretnym odcinku [−π, π], oraz wykonujemy do niej interpolację za pomogą jednowymiarowej konwoluji, korzystając z różnych jąder konwolucji.

Do zadania został wykonany kod, który jest załączony w repozytorium na github.

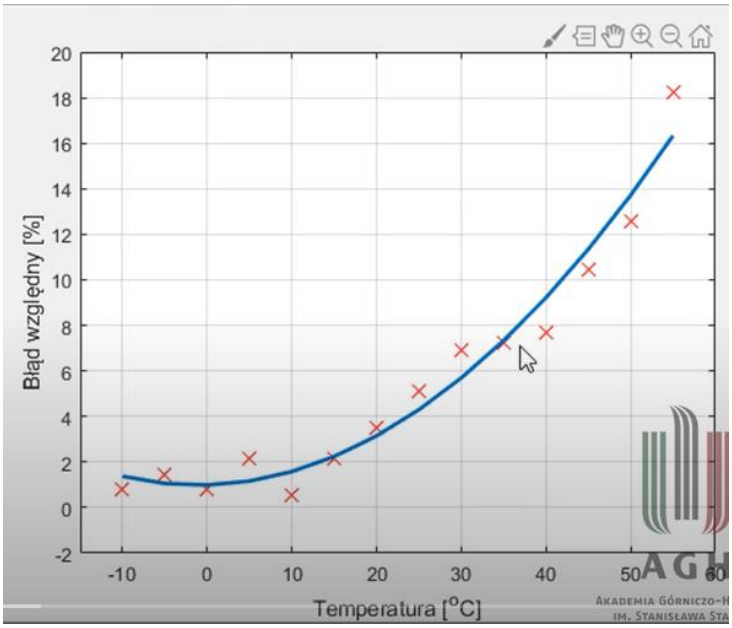
2. Funkcje matematyczne

Co to jest aproksymacja

Aproksymacja polega na dopasowaniu do punków pomiarowych krzywej, pewnego modelu, przy czym ta krzywa nie przechodzi przez wszystkie punkty, tylko pokazuje ogólny trend zmian wielkości, czyli dopasowujemy modelu matematycznego. Jego realizacja składa się z 2 etapów – przyjęcie klasy modelu np. wielomian n-tego stopnia, oraz obliczenie współczynników klasy modelu (iteracyjne poszukiwanie minimum funkcji celu)

Python posiada predefiniowane funkcje takie jak (scipy.optimize.curve_fit – do generalnego dopasowywania funkcji , które mają różnice implementacyjne) scipy.optimize.curve_fit używa metody nieliniowej, dopasowania poprzez metodę najmniejszych kwadratów, aby dopasować funkcje do danych. Podaje się model funkcji, xdata, czyli dane pomiarowe w postaci tablicy, oraz data, czyli dane zależne (wartość y funkcji danych punktach x) w postaci tablicy, oraz zwracana jest tablica optymalnych wartości parametrów, która minimalizuje sumę reszt kwadratów nazywana popt lub pcov czyli szacowana, przybliżona kowariancja.

Aby obliczyć standardowe odchylenie błędu, należy użyć perr = np.sqrt(np.diag(pcov)) (zalecenie z manuala sciPy) lub numpy.polynomial.polynomial.polynomial.fit (starsza wersja numpy.polyfit)

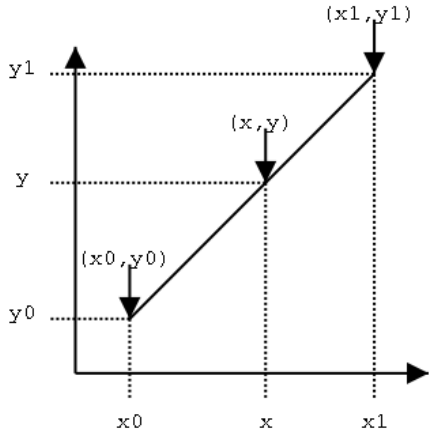


Co to jest interpolacja

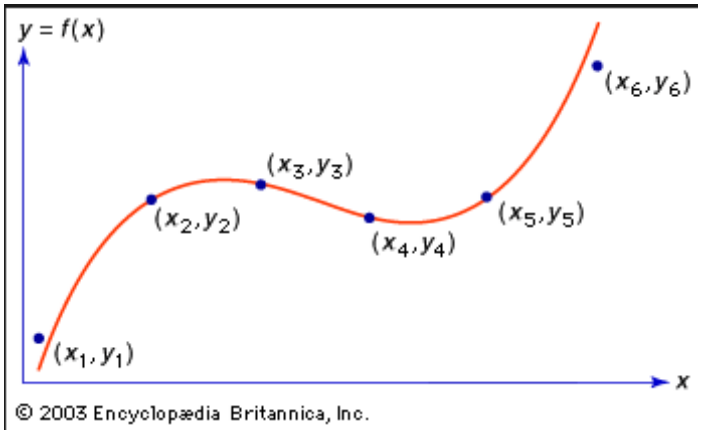
Interpolacja jest to również dopasowanie funkcji na podstawie podanych punków na wykresie, z tą różnicą, że w interpolacji funkncja koniecznie przechodzi przez podane punkty (punkty te nazywane są inaczej węzłami) Ważną zasadą jest to, że interpolowana funkcja, (w której jest M-punków) może być stopnia M-1. Python posiada bibliotekę Interpolation (scipy.interpolate), która funkcje sklejania, i funkcje interpolacyjne 1-D, wielowymiaroe, inerpolatory Lagrange'a i Taylor'a.

Biblioteka ta posiada funkcje interp1d (która jest oznaczona jako Deprecated, czyli jako przestarzała funkcja. Funkcja ta przyjmuje tablice punktów x , tablicę punktów y, opcjonalnie rodzaj interpolacji w postaci stringu lub integeru.
Do dyspozycji mamy: ‘linear’, ‘nearest’, ‘nearest-up’, ‘zero’, ‘slinear’, ‘quadratic’, ‘cubic’, ‘previous’, or ‘next’. ‘zero’, ‘slinear’, ‘quadratic’ and ‘cubic’

W poleceniu mamy zastosować wybrane konwolucje h1,h2 (Box) h3 (Triangle), h4 (Sinc function)

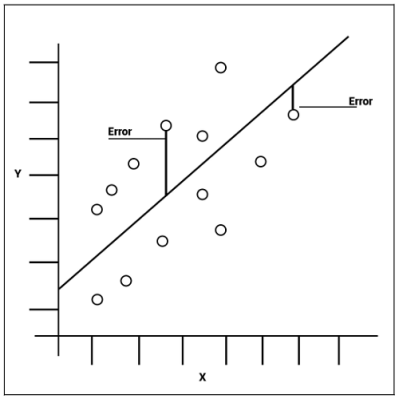


1. Interpolacja liniowa



2. Interpolacja wielomianowa 3-go stopnia

Co to jest MSE (mean squared error / błąd średniokwadratowy)



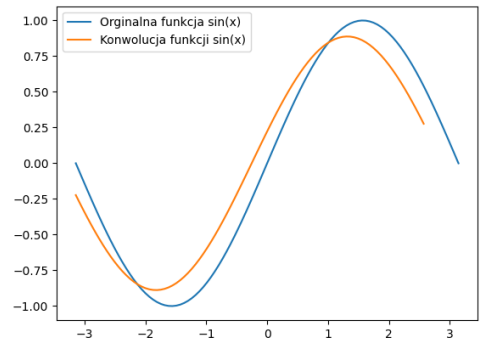
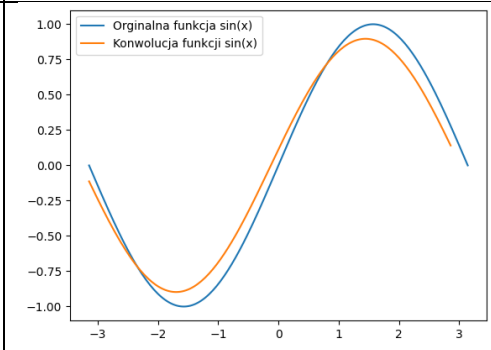
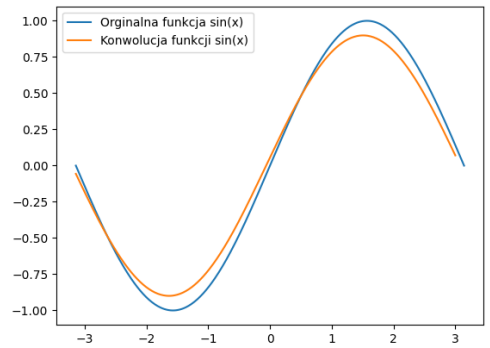
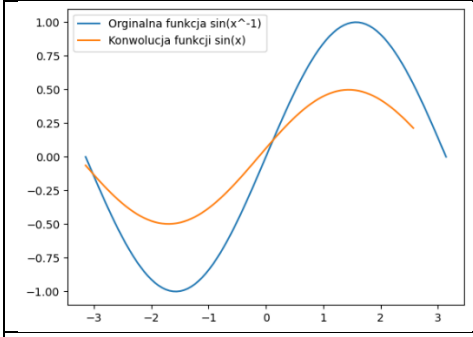
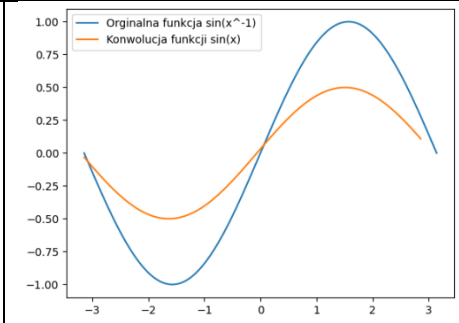
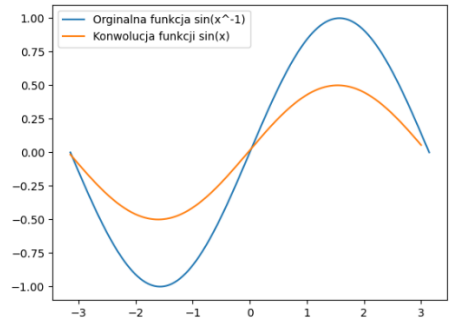
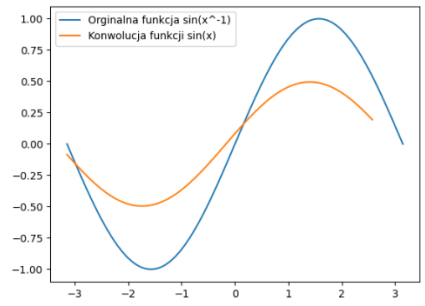
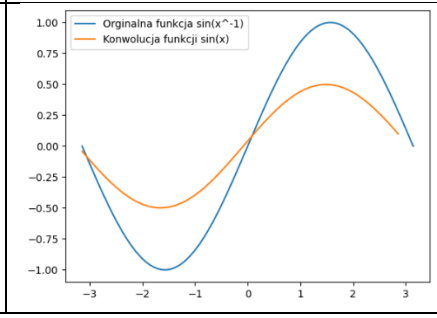
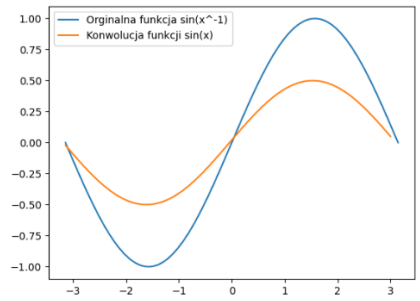
Błąd średniokwadratowy jest to estymator określający różnice między estymatorem (w naszym przypadku funkcją interpolacyjną) a wartością estymowaną (generowane funkcje). Im mniejsza wartość estymatora, tym dokładność odwzorowania prawdziwej funkcji większa.

Splot funkcji

9	5	7	6	3	1	2	8	4
4	1	2	3	8	7	2	6	5	9
6	2	4	8	4	5	9	7	3	1
5	1	3	2	9	7	8	6	4	1	2
8	4	2	9	1	5	3	6	7
7	6	1	2	4	3	5	9	8
2	8	5	3	9	7	1	4	6
1	7	4	5	6	8	9	2	3
3	9	6	1	2	4	8	7	5

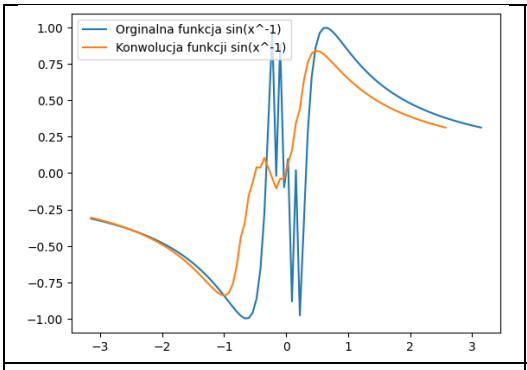
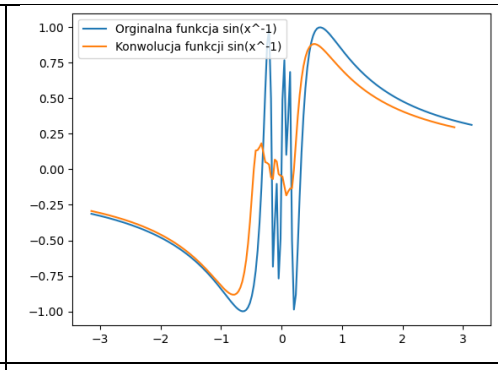
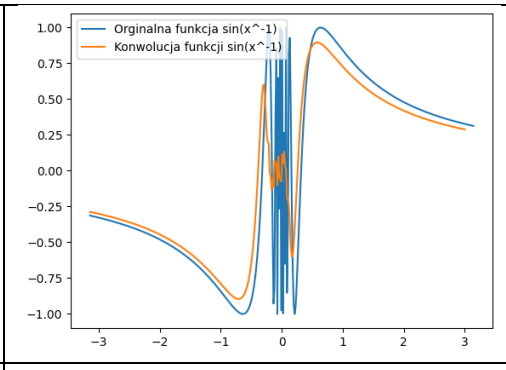
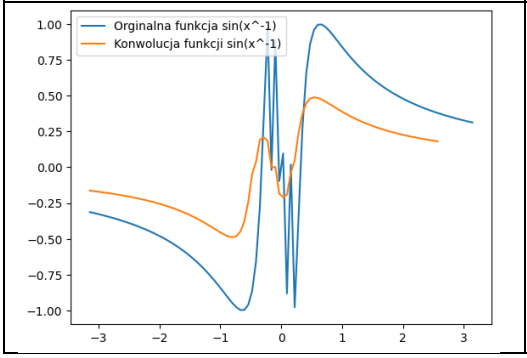
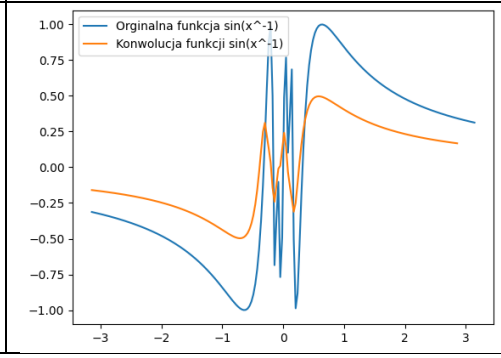
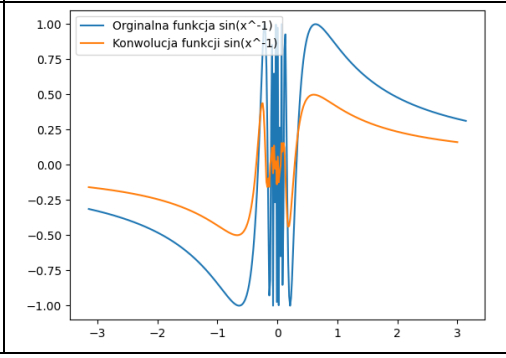
3. Rezultaty działania programu

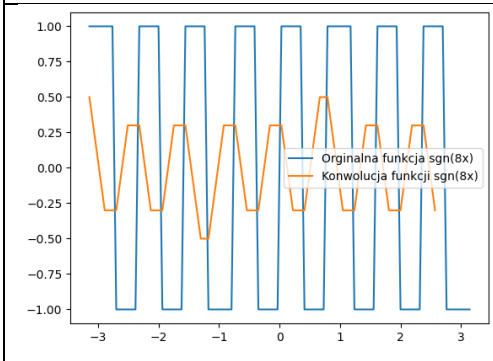
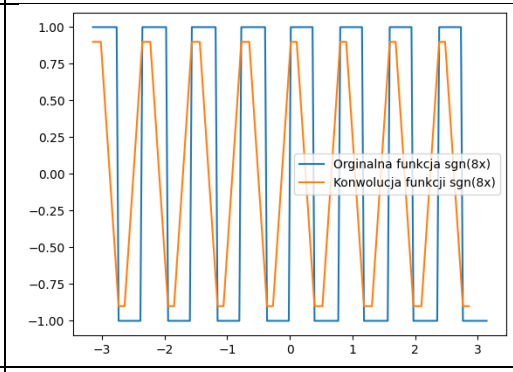
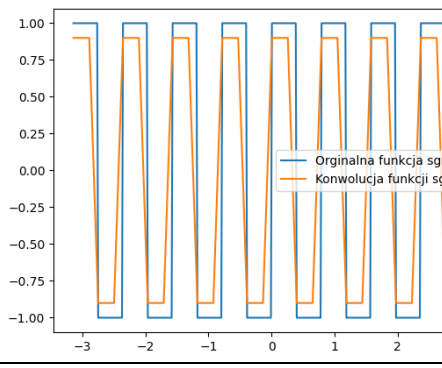
Interpolacja funkcji sinus do dla określonej ilości punktów N, określonego jądra konwolucji

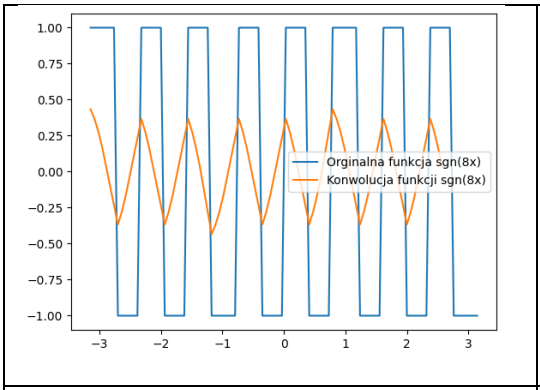
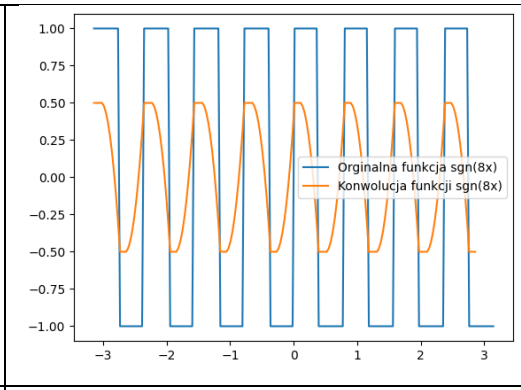
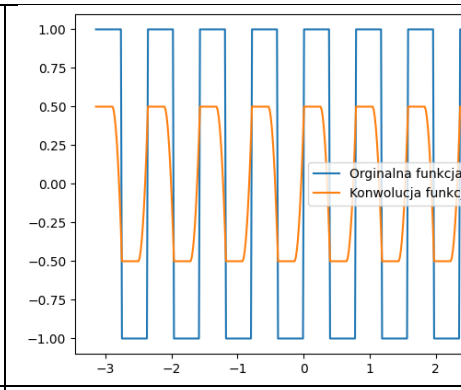
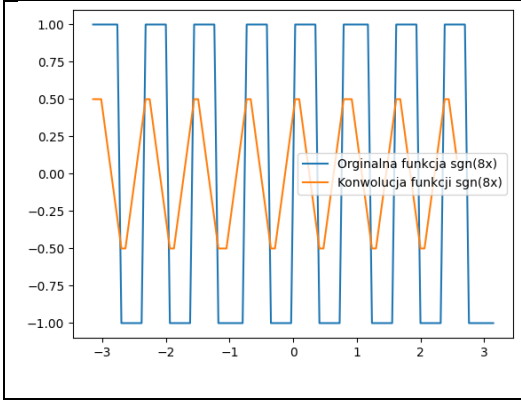
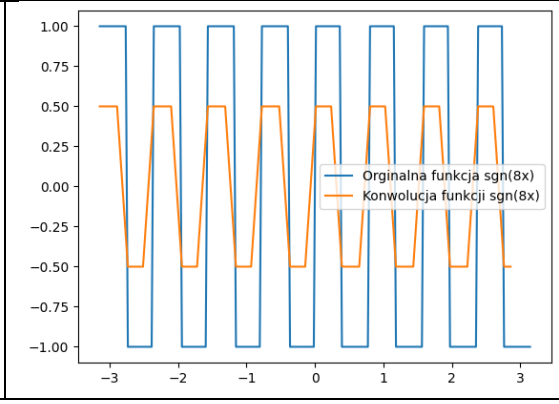
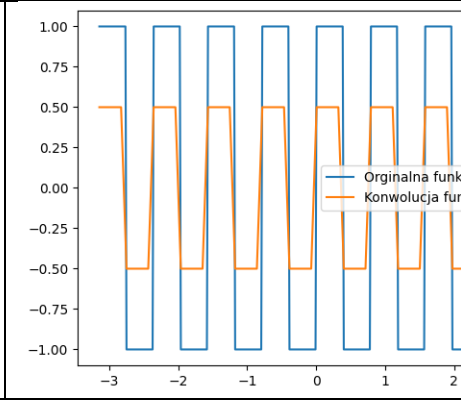
N=100 Kernel 1 MSE = 3.22 %	N=200 Kernel 1 MSE = 1.23 %	N=1000 Kernel 1 MSE = 0.69 %
		
N=100 Kernel 2 MSE = 13.75 %	N=200 Kernel 2 MSE = 13.10%	N=1000 Kernel 2 MSE = 12.78 %
		
N=100 Kernel 3 MSE = 14.16%	N=200 Kernel 3 MSE = 13.21%	N=1000 Kernel 3 MSE = 12.81%
		

Interpolacja funkcji sin(x^-1) do dla określonej ilości punktów N, określonego jądra konwolucji

N=100 Kernel 1 MSE = 11.95 %	N=200 Kernel 1 MSE = 7.35 %	N=1000 Kernel 1 MSE = 6.08 %

		
N=100 Kernel 2 MSE = 14.66 %	N=200 Kernel 2 MSE = 12.92	N=1000 Kernel 2 MSE = 12.95
		

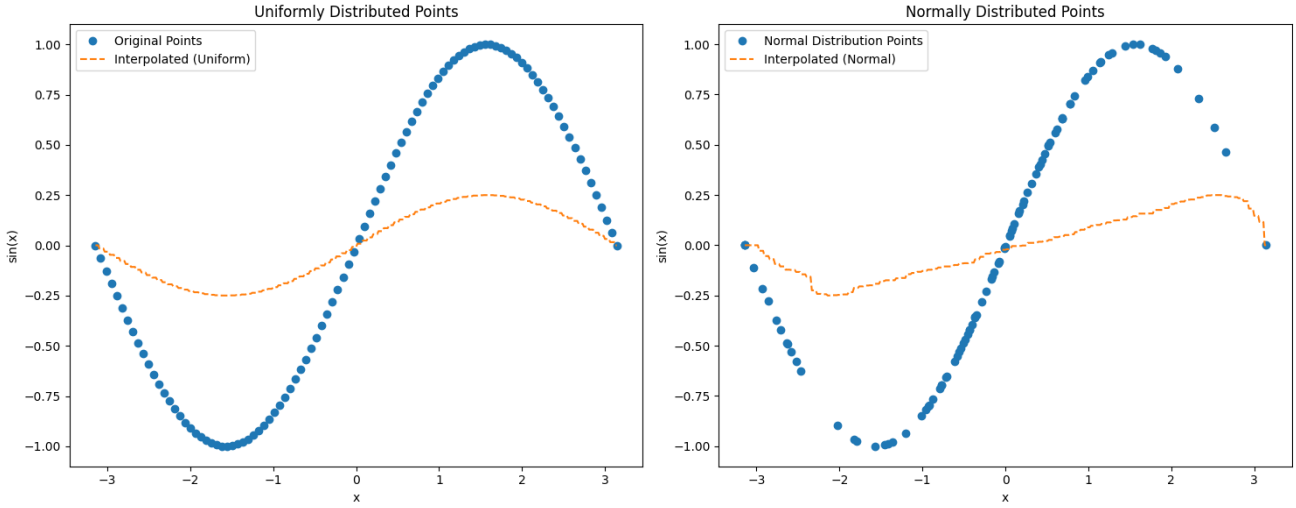
Interpolacja funkcji signum(8x)do dla określonej ilości punktów N, określonego jądr konwolucji		
N=100 Kernel 1 MSE = 124,51 %	N=200 Kernel 1 MSE = 76,39 %	N=1000 Kernel 1 MSE = 37,82 %
		
N=100 Kernel 2 MSE = 85,49	N=200 Kernel 2 MSE = 57,16	N=1000 Kernel 2 MSE = 40,70

		
N=100 Kernel 3 MSE = 75,11 %	N=200 Kernel 3 MSE = 50,13 %	N=1000 Kernel 3 MSE = 37,28 %
		

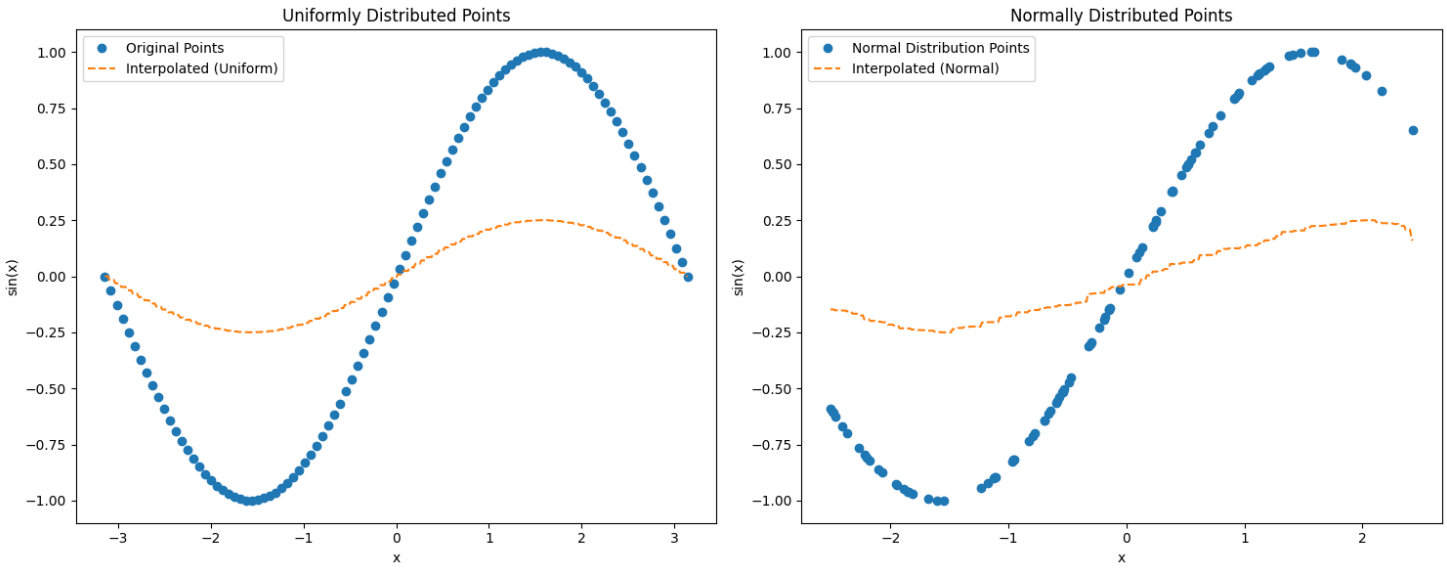
4. Analiza kodu

Do zrozumienia działania zasady programu wymagana jest znajomość interpolacji, czyli dla naszego programu estymacja wartości pomiędzy pikselami, podczas gdy operacja splotu dyskretnego polega na przesuwaniu jądra poprzez każdy piksel. W przypadku skalowania obrazów wykorzystujemy jądro uśredniające.

5. Zadanie dodatkowe- zbadać wpływ rozmieszczenia punktów na jakość interpolacji (uniformly distributed points comparison with normally distributed points)



Próba 1 - MSE: 59,23 pomiędzy interpolacjami



Próba 2 - MSE: 67,72 pomiędzy interpolacjami

W tym przypadku porównujemy interpolację funkcji sin(x) wygenerowaną przy pomocy N = 100 punktów przy równomiernym rozkładzie tych punktów, do funkcji o wylosowanych zgodnie z rozkładem normalnym punktach, co widać na obrazie.

6. Podsumowanie i wnioski

1. Za pomocą jedno-wymiarowej konwolucji, korzystając z różnych jąder konwolucji możemy uzyskać funkcję interpolacyjną danego przebiegu. W zależności od charakterystyki przebiegu, a raczej jej dużej zmienności przebiegu (np. dla szybko oscylującej funkcji sinus) występuje duża różnica pomiędzy wartością rzeczywistą a wartością prawdziwą.
2. Jak idzie zauważyć w każdym z przykładów, zwiększenie ilości punków tworzących funkcję w każdym z przypadków spowodowało zmniejszenie się parametru MSE (mean squred error), czyli zwiększała się dokładność odwzorowania interpolowanej funkcji.
3. Aby uzyskać dobrą interpolację, należy odpowiednio dobrać odpowiednią funkcję jądrową, ponieważ idzie zauważyć wizualnie, jak i również poprzez badanie parametru MSE znaczącą różnicę estymacji, w zależności od wybranej funkcji. Bez znaczenia jest również wielość zdefiniowanego jądra.
4. Ważna jest znajomość rozróżnienia pomiędzy aproksymacją, a interpolacją
5. Mean squared error (MSE) jest bardzo dobrą miarą do pomiaru różnicy pomiędzy wartością estymowaną, a wstymatą. Inną przydatną miarą jest RMSE (Root-mean-square deviation) czyli pierwiastek z MSE, który niestety jest bardziej wrażliwy na szybkie zmiany, przez co częściej wykorzystywanym parametrem jest MSE.
6. W przyszłości muszę się nauczyć pisać kod w notebookach, czyli używania Google colab / Jupyter, ponieważ aktualnie napisany kod strukturalnie nie jest napisany w funkcjonalny sposób.

7. Wykorzystane źródła

https://www.youtube.com/watch?v=QzY57FaENXg&t=264s

https://ksopyla.com/python/operacja-splotu-przetwarzanie-obrazow/#splot_numpy_python