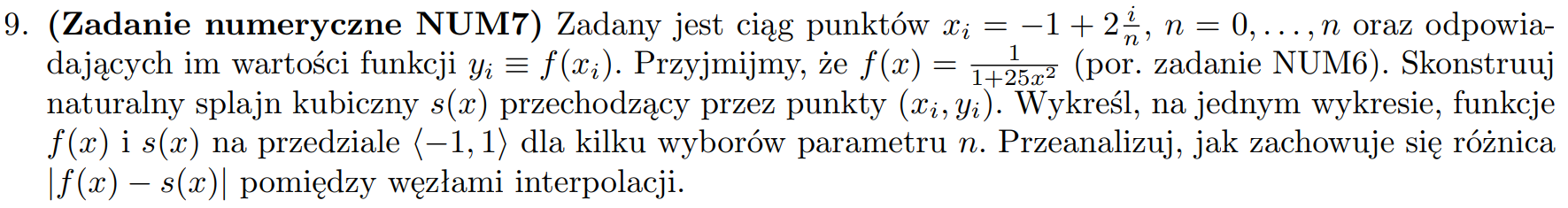
# Zadanie numeryczne 7



# Wprowadzenie

Celem zadania jest napisanie programu znajdującego interpolacje danej funkcji za pomocą naturalnych splajnów kubicznych.

# Wynik

Funkcja generująca splajny znajduje się w pliku shared.py, z którego korzystają pozostałe programy.

Tworzenie splajnów kubicznych przechodzących przez przebiega następująco:

Tworzona jest siatka punktów oraz odpowiadających im .

Wtedy wartość funkcji sklejanej wynosi:

dla

Gdzie , , ,

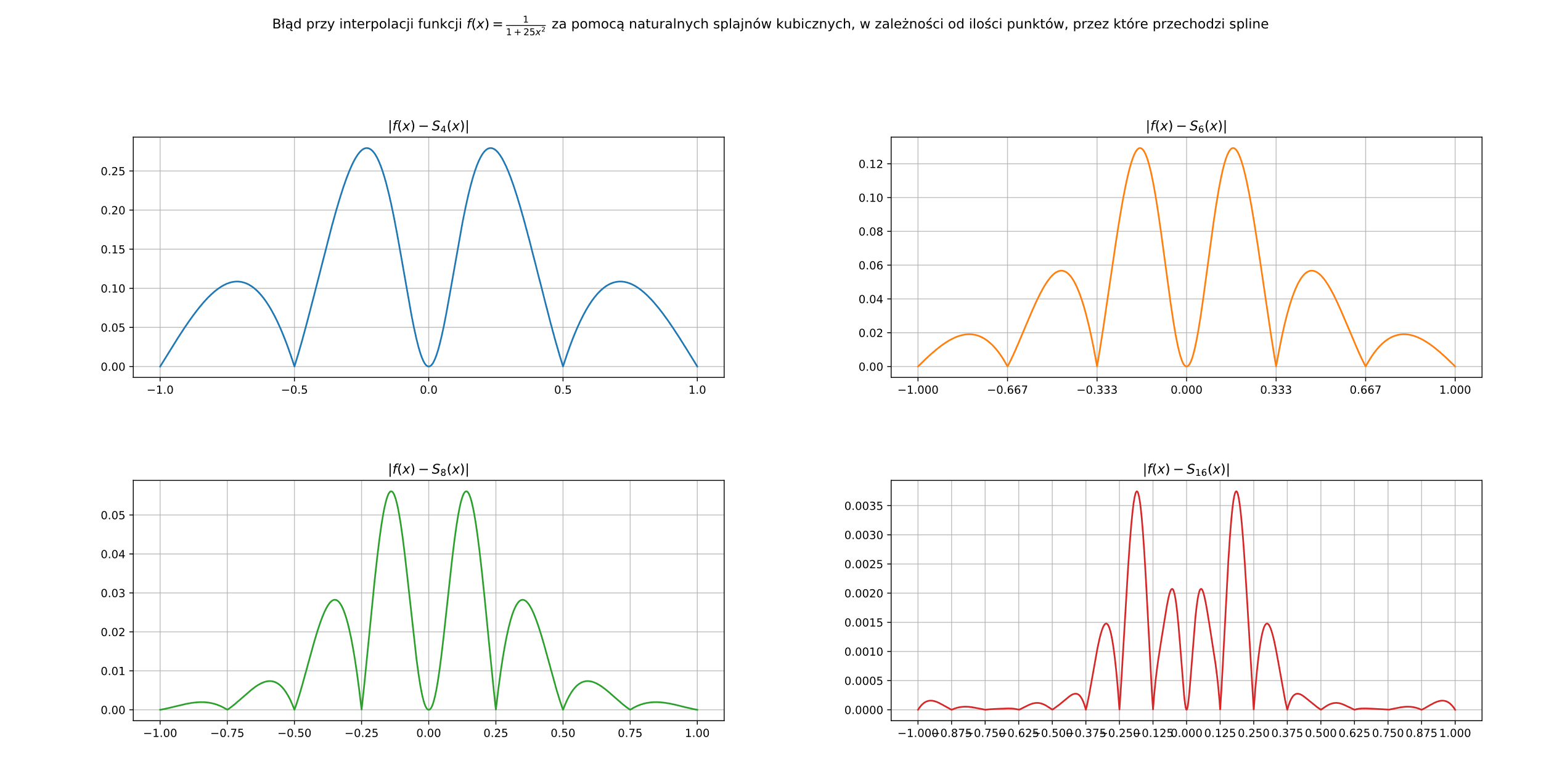
, a pozostałe wartości można wyliczyć rozwiązując równanie z macierzą:

W celu rozwiązania równania z macierzą została wykorzystana procedura scipy.sparse.linalg.solve z biblioteki scipy, służąca do rozwiązywania równań liniowych z macierzą rzadką. Sama macierz jest generowana za pomocą procedury scipy.sparse.diags i przechowywana w formacie CSR.

Program drawSplines.py generuje i wykreślna na wykresie funkcje oraz kilka funkcji dla kilku wybranych .

Program splineDiff.py oblicza i wykreśla na wykresach różnicę dla kilku wybranych .

# Otrzymane wykresy:



# Dyskusja wyników

Na wykresie widać, że interpolacja za pomocą splajnów jest dobra nawet dla takiej funkcji, dla której interpolacja wielomianami przy jednorodnej siatce skutkowała oscylacjami na krańcach przedziału.

Już przy widać, że interpolacja jest bardzo precyzyjna, a dla nieco niższych n też nie ma żadnych sporych odchyleń. Widać też, że wybranie nieparzystego skutkuje parzystą ilością punktów w siatce, co dla niskich skutkuje sporym odchyleniem na środku przedziału spowodowanym braku punktu na wierzchołku wykresu.

Na wykresach analizujących różnicę widać, że pomiędzy węzłami interpolacji różnica rośnie w miarę oddalania się od najbliższego węzła, podczas gdy w samych węzłach wynosi 0. Widać także, że wraz ze wzrostem ilości węzłów interpolacji maleje maksymalne odchylenie któremu podlega funkcja sklejana.