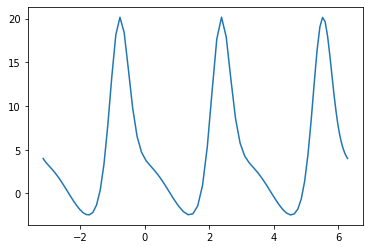
Sprawozdanie z ćwiczenia 2 - interpolacja

Konrad Pękala

# 1. Wstęp

W tym ćwiczeniu miałem za zadanie zaimplementować algorytm interpolacji funkcji f który wyznaczy wielomian interpolujący w postaci Lagrange’a i Newtona.



Rysunek 1. Wykres funkcji f

Do obliczeń korzystałem z języka Python 3 oraz projektu Jupyter Notebook

Korzystałem ze standardowej precyzji typu float oferowanej przez język Python(odpowiednik typu double w języku C).

## Postać Lagrange’a

## Postać Newtona

## Pomiar błędów obliczeniowych

1. maksymalny\_błąd =

gdzie - wartość wielomianu interpolującego w punkcie

- wartość funkcji f w punkcie

– i-ty punkt ze zbioru 100 punktów równomiernie rozłożonych na dziedzinie funkcji f

# 2. Zagadnienie Lagrange’a

## 2.1 Wizualizacja dla postaci Lagrange

(Pominę wykresy dla wielomianu postaci Newtona ponieważ nie różnią się końcowym rezultatem)

|  |  |  |
| --- | --- | --- |
| **Liczba węzłów** | **Równoległe rozłożenie węzłów** | **Rozłożenie węzłów metodą Czebyszewa** |
| **3** |  |  |
| **4** |  |  |
| **5** |  |  |
| **7** |  |  |
| **10** |  |  |
| **15** |  |  |
| **20** |  |  |

## 2.2 Porównanie średniego błędu

|  |  |
| --- | --- |
| **Równoległe rozłożenie węzłów** | **Rozłożenie węzłów metodą Czebyszewa** |
|  |  |

## 2. 3 Porównanie maksymalnego błędu

|  |  |
| --- | --- |
| **Równoległe rozłożenie węzłów** | **Rozłożenie węzłów metodą Czebyszewa** |
|  |  |

## 2.4 Wnioski

### Efekt Runge’go

Ten efekt możemy zauważyć już przy wielomianie stopnia 11(liczba węzłów=12) ale tylko gdy będziemy równomiernie wybierać węzły. Już dla wielomianu o 20 węzłach powstanie maksymalny błąd bliski 10 000. Rozłożenie węzłów metodą Czebyszewa radzi sobie całkiem dobrze w minimalizowaniu tego efektu

### Wielomian z dobrym przybliżeniem

Jeśli chcemy żeby wielomian przybliżał bardzo dokładnie (z maksymalnym błędem rzędu 0.5) to musimy wybrać wielomian stopnia co najmniej 39. Oczywiście bierzemy pod uwagę tylko wielomiany z węzłami Czebyszewa, ponieważ gdybyśmy wybierali węzły równomiernie to zwiększając stopień wielomianu zwiększalibyśmy maksymalny błąd wykładniczo.

# 4. Zagadnienie Hermita

Algorytm Hermita został zaimplementowany w postaci Newtona z tylko jedną pochodną funkcji f. Pochodna została obliczona ręcznie.

## 4.1 Wizualizacja

|  |  |  |
| --- | --- | --- |
| **Liczba węzłów** | **Równoległe rozłożenie węzłów** | **Rozłożenie węzłów metodą Czebyszewa** |
| **3** |  |  |
| **4** |  |  |
| **5** |  |  |
| **7** |  |  |
| **10** |  |  |
| **15** |  |  |
| **20** |  |  |

## 4.2 Porównanie średniego błędu

|  |  |
| --- | --- |
| **Równoległe rozłożenie węzłów** | **Rozłożenie węzłów metodą Czebyszewa** |
|  |  |

## 4.3 Porównanie maksymalnego błędu

|  |  |
| --- | --- |
| **Równoległe rozłożenie węzłów** | **Rozłożenie węzłów metodą Czebyszewa** |
|  |  |

## 4.4 Wnioski

### Znacznie większe błędy

Dla zagadnienia Hermita błędy interpolacji były znacznie większe(dla n > 19). Dla n = 20 maksymalny błąd wynosi , gdzie dla zagadnienia Lagrange’a był to błąd rzędu 10 000. Dla równoodległych węzłów nadal wystepuje efekt Rungego, który jest potęgowany przez błędy arytmetyki komputerowej, które są widoczne także dla węzłów Czebyszewa(ale dopiero dla n > 19).

### Wielomian z dobrym przybliżeniem

Jeśli chcemy żeby wielomian przybliżał bardzo dokładnie (z maksymalnym błędem rzędu 0.5) to musimy obejść się ze smakiem, ponieważ najniższą maksymalną wartość udało mi się uzyskać dla n=19, która wynosi ok 1.40, od tego momentu maksymalny błąd będzie się zwiększał.