Metody obliczeniowe w nauce i technice

SPRAWOZDANIE

Ćwiczenie 3
INTERPOLACJA

Łukasz Gajewski

I. Cel ćwiczenia

I. Porownanie wlasnych procedur interpolacji wielomianowej z funkcjonalonoscia GSL:
   A. Wygenerowac tablice N punktow (x,y)
   B. Uzyc funkcji gsl do interpolacji wielomianowej dla tych punktow - uzyc `gsl_interp_polynomial`. Narysowac jego wykres.
   C. Napisac wlasny program generujacy dane (recznie - bez korzystania z gsl) do narysowania wykresu wielomianu interpolujacego metoda Lagrange'a dla tych punktow w wybranym przedziale. Postarac sie zaprojektowac API podobnie do GSL - osobna funkcja *init* oraz *eval* Narysowac wykres.
   D. Zrobic to samo metoda Newtona. Porownac wszystkie 3 wyniki na jednym wykresie.
   E. Porownac metody poprzez pomiar czasu wykonania. Dokonac pomiaru 10 razy i policzyc wartosc srednia oraz oszacowac blad pomiaru za pomoca odchylenia standardowego. Narysowac wykresy w R.
   F. Poeksperymentowac z innymi typami interpolacji gsl (cspline, akima), zmierzyc czasy, narysowac wykresy i porownac z wykresami interpolacji wielomianowej.

II. Zbadac i zademonstrowac algorytmy interpolacji stosowane w grafice komputerowej (np. do zmiany wielkosci obrazu) - przyklad

III. Wymagane sprawozdanie zawierajace krotki opis zastosowanych metod Lagrange'a i Newtona a takze wykresy, wyniki i wnioski (forma dowolna), ktore nalezy pokazac w czasie nastepnego lab (nie wysylac)

II. Algorytmy interpolacji stosowane w grafice komputerowej

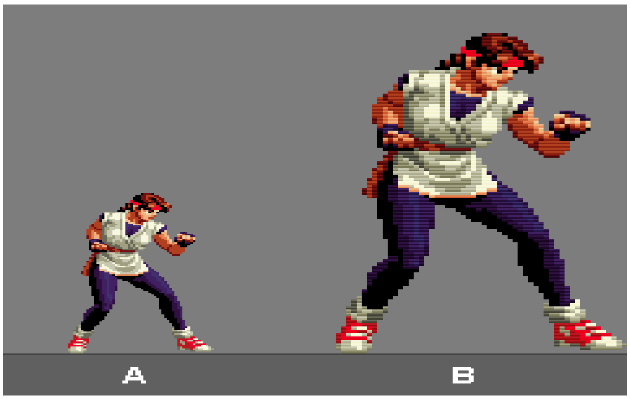
1. Metoda najbliższego sąsiada.
2. Metoda scale3x
3. Metoda hq3x

Interpolacja – w grafice komputerowej jest to proces mający na celu utworzenie nowego, wcześniej nieistniejącego piksela na podstawie pikseli sąsiadujących z pikselem tworzonym tak, aby był on jak najlepiej dopasowany optycznie do przetwarzanego obrazu.

1. Metoda najbliższego sąsiada

Zastępuje każdy piksel dziewięcioma pikselami o tym samym kolorze (przy powiększeniu 3x).

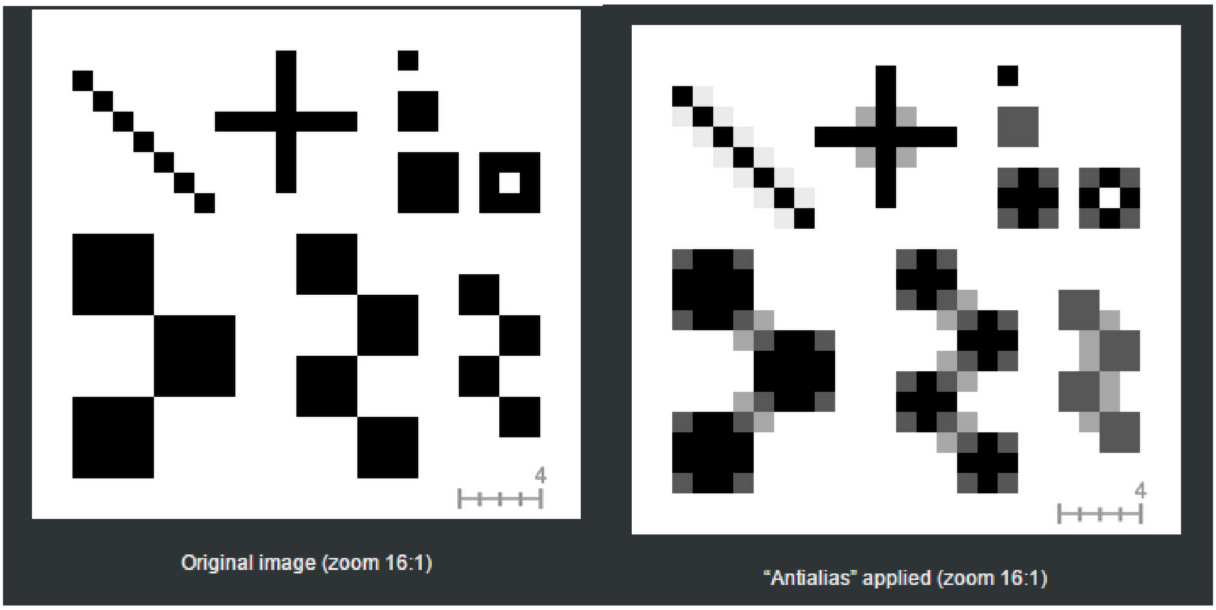
- najprostszy
- wydajny
- brak odpowiednich (płynnych) przebiegów między kolorami
- statyczne kopiowanie pikseli
- przy dużych powiększeniach widać grupy pikseli



2. Metoda scale3x

Implementacja algorytmu EPX.

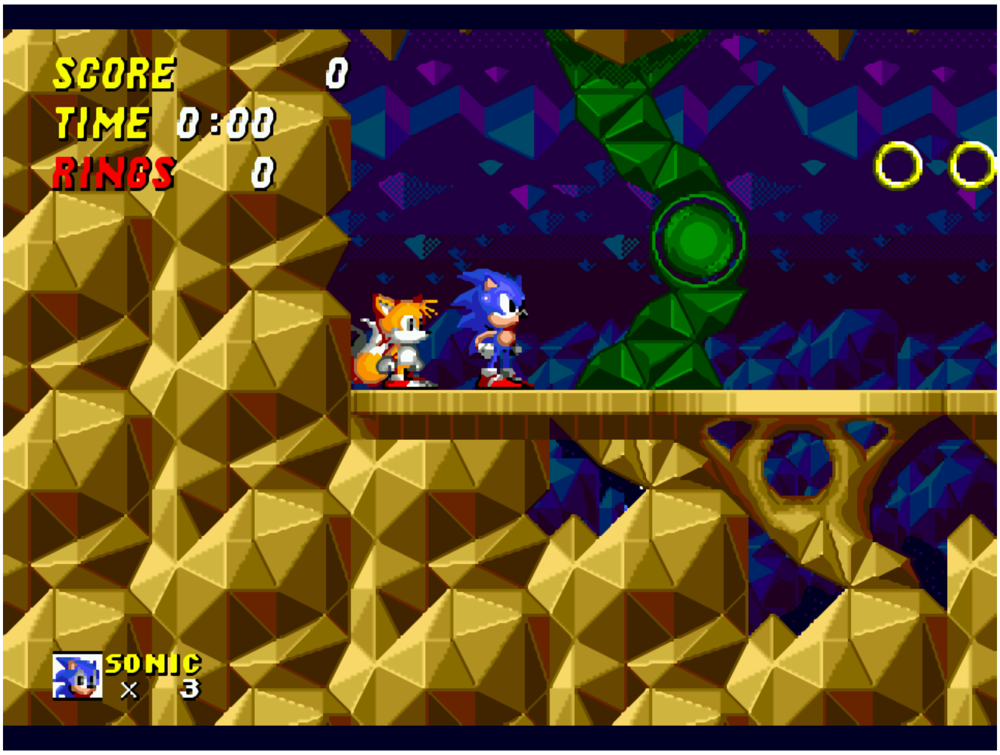
- wykorzystuje wartości sąsiadów piksela źródłowego do określenia kolorów pikseli w obrazie wyjściowym
- prosty, użyteczny algorytm
- dobre rezultaty przy stosunkowo niskim nakładzie obliczeniowym
- przebiegi gładsze niż w algorytmie najbliższego sąsiada



3. Metoda hq3x

Oblicza różnicę kolorów między każdym z ośmiu sąsiadów rozpatrywanego piksela (węzła). Inteligentnie dzieli piksele ze względu na pozycję.

- piksele: bliskie, dalekie
- do wyznaczania wartości wykorzystywana jest wartość w tablicy (ztablicowane wzorce)
- wygładza krzywizny
- gładki przebieg między ‘węzłami’
- nieduży koszt obliczeń



III. Rodzaje interpolacji w matematyce

Interpolacja – metoda numeryczna polegająca na wyznaczaniu w danym przedziale tzw. funkcji interpolacyjnej, która przyjmuje w nim z góry zadane wartości w ustalonych punktach, nazywanych węzłami

a) **Wielomian interpolacyjny Lagrange'a** – interpolacja liniowa

$P_1(x)$ - przez (x_0, y_0) i (x_1, y_1)

$$P_1(x) = \underbrace{\frac{(x - x_1)}{(x_0 - x_1)}}_{L_0(x)} y_0 + \underbrace{\frac{(x - x_0)}{(x_1 - x_0)}}_{L_1(x)} y_1 = \sum_{k/0}^{1} L_k(x) f(x_k)$$

- wielomian stopnia ≤ 1
- $x = x_0 \rightarrow P(x_0) = y_0$
$x = x_1 \rightarrow P(x_1) = y_1$
$L_k(x_l) = \delta_{k,l}$

Wielomian n-tego stopnia:

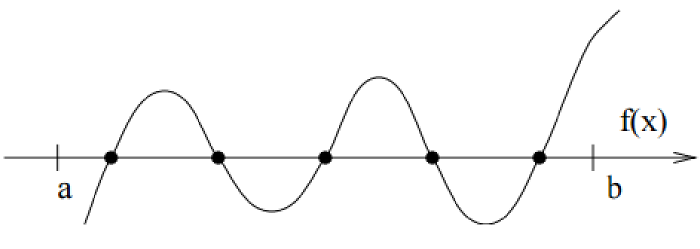
przez $x_0, x_1, x_2, \ldots, x_n$

$$L_k(x_l) = \delta_{k,l} = \begin{cases} 0, & k \neq l \quad (\star) \\ 1, & k = l \quad (\star\star) \end{cases} \text{ f. “wymierna”}$$

$(\star) \rightarrow licznik = (x - x_0)(x - x_1) \ldots (x - x_{k-1})^{\downarrow}(x - x_{k+1}) \ldots (x - x_n)$
$(\star\star) \rightarrow mianownik = (x_k - x_0)(x_k - x_1) \ldots (x_k - x_{k-1})(x_k - x_{k+1}) \ldots (x_k - x_n)$
LIP:

$$\boxed{L_k(x) = \prod_{i/0, i\neq k}^{n} \frac{x - x_i}{(x_k - x_i)} \quad , \quad P_n(x) = \sum_{k/0}^{n} L_k(x) f(x_k)}$$

Zadanie: wykres $L_k(x)$, sprawdzić $\sum_{k/0}^{n} L_k(x) = 1$



b) Wielomian postaci Newtona

Dla wielomianu stopnia n wybiera się $n+1$ punktów $x_0, x_1, \dots, x_n$ i buduje wielomian postaci:

$$w(x) = \sum_{i=0}^{n} a_i \prod_{j=0}^{i-1} (x - x_j)$$

$$= a_0 + a_1(x - x_0) + a_2(x - x_1)(x - x_0) + \cdots + a_n(x - x_{n-1}) \cdots (x - x_1)(x - x_0)$$

Metoda ilorazu różnicowego:

$$\begin{array}{llllll}
x_0 & f(x_0) & & & & \\
x_1 & f(x_1) & f[x_0, x_1] & & & \\
x_2 & f(x_2) & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
x_n & f(x_n) & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots & f[x_0, \dots, x_n]
\end{array}$$

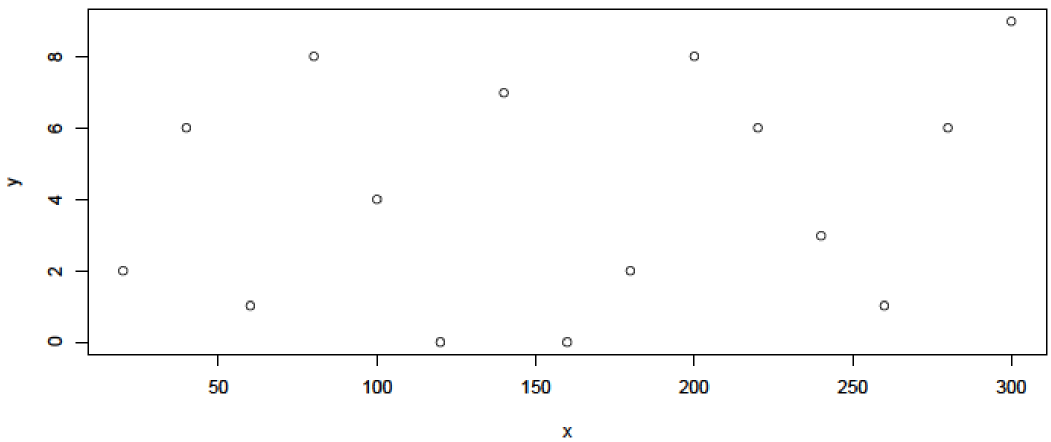
$$f(x_0) = y_0 \qquad oraz \quad f(x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0}$$

IV. Interpolacja za pomocą funkcji dostarczonych przez bibliotekę GSL oraz metodami Newtona, Lagrange'a.

Zadanie składało się na kilka podpunktów:

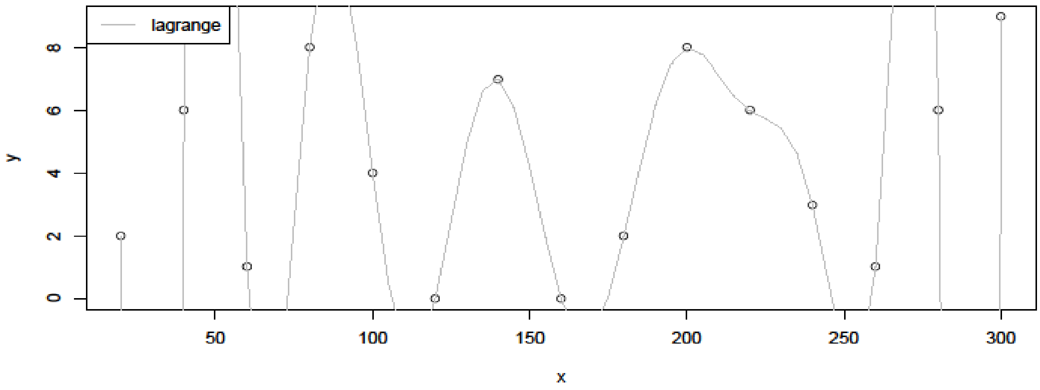
- generowanie tablicy N punktów
- wykorzystać bibliotekę GSL do interpolacji wielomianowej
- zaprojektować API podobnie do GSL i zaimplementować wyznaczanie wielomianu interpolacyjnego oraz ewaluację dla metod Lagrange'a i Newtona
- sprawdzić rezultaty wykorzystując inne metody interpolacji (cspline, akima)
- porównać wszystkie metody na jednym wykresie
- wyznaczyć czasy wykonania, obliczyć wartość średnią i oszacować błąd pomiaru za pomocą odchylenia standardowego

N punktów (x,y) - wykres

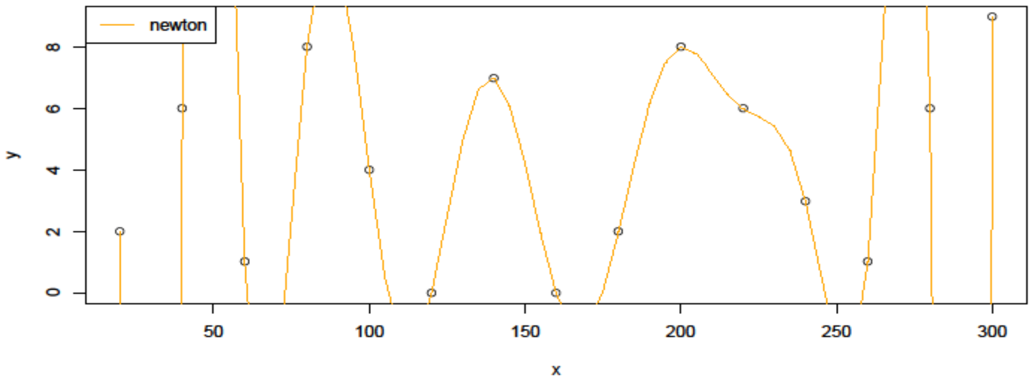


Łukasz Gajewski

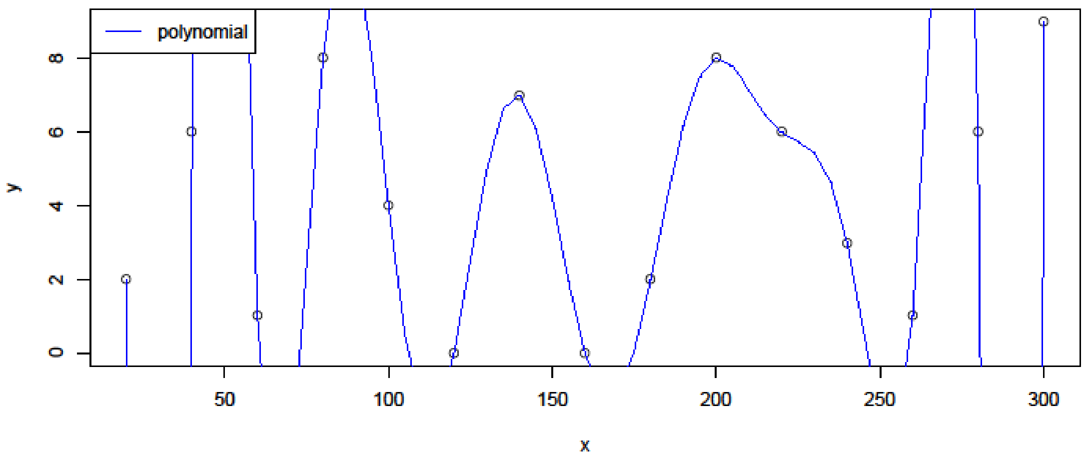
Interpolacja za pomocą metody Lagrange'a



Interpolacja za pomocą metody Newtona

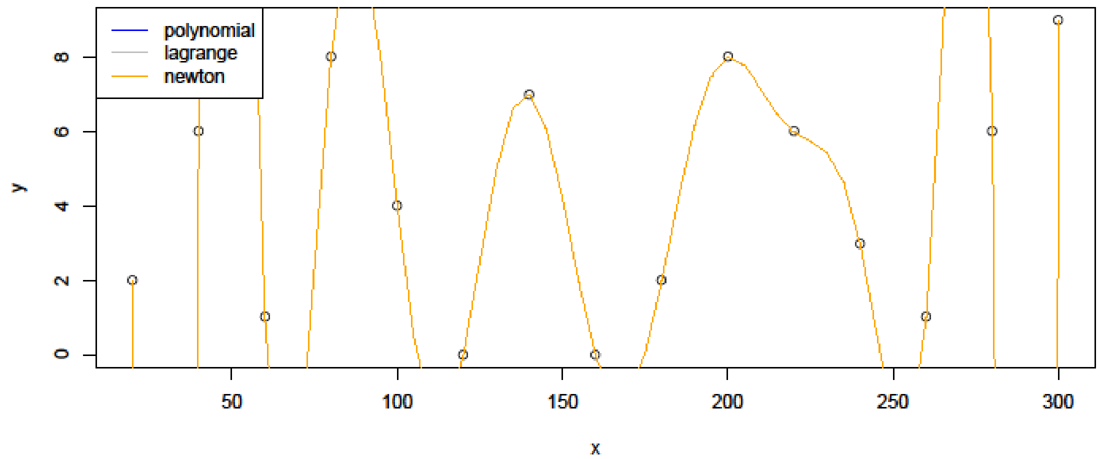


Interpolacja za pomocą biblioteki GSL (polynomial)

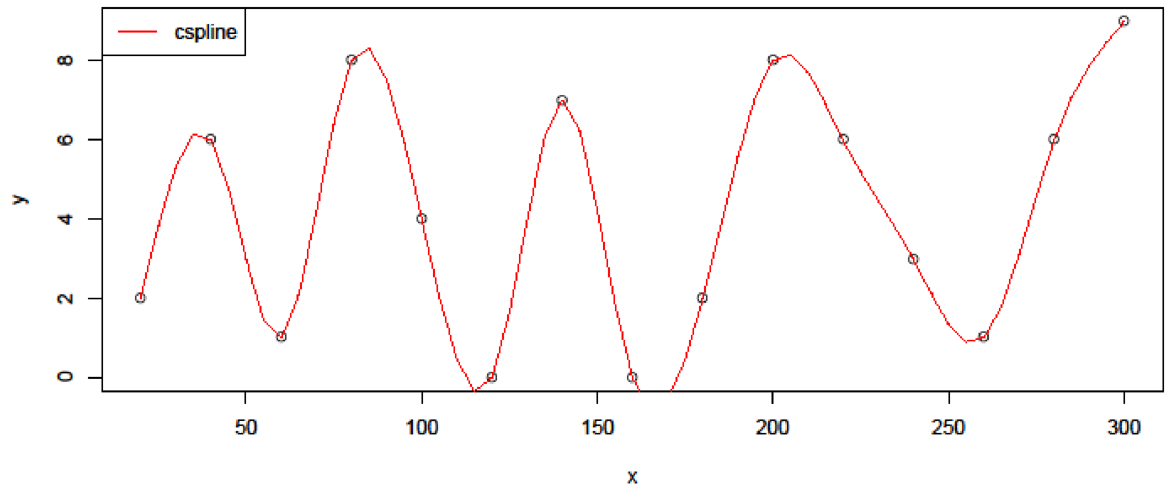


Porównanie interpolacji wielomianowej: polynomial, lagrange, newton.

Widzimy nakładanie się wielomianów interpolacyjnych (na wykresie trudno to zaobserwować).
Najlepiej porównać wykres z poprzednimi i zanalizować kształt funkcji wielomianowej.

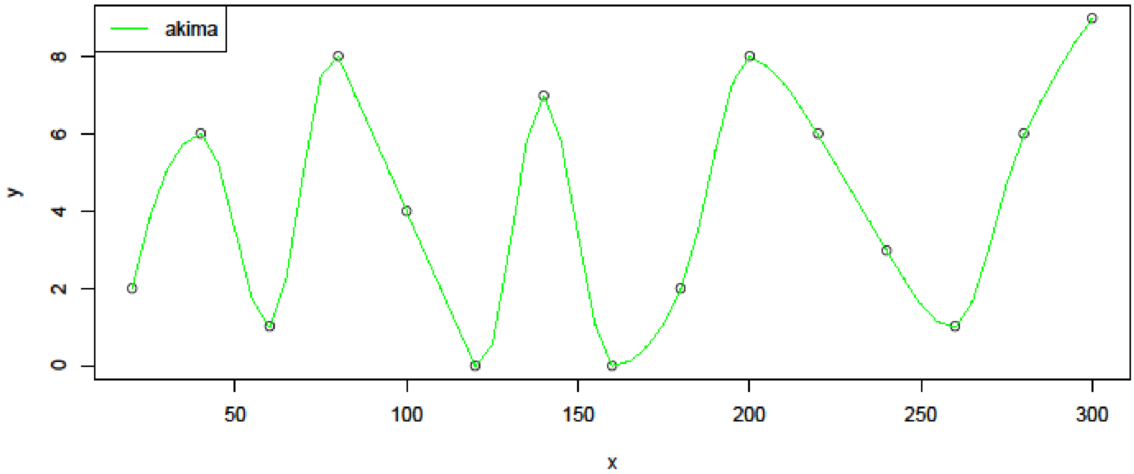


Interpolacja za pomocą biblioteki GSL (cspline)



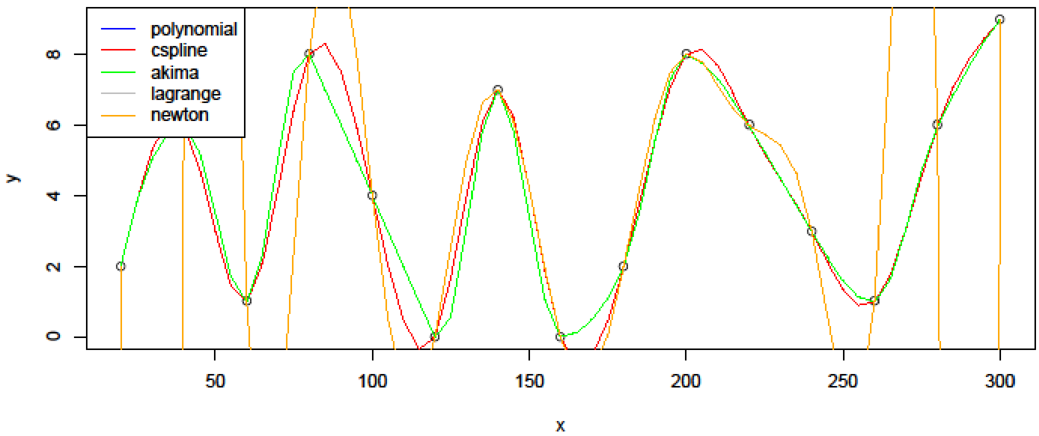
Łukasz Gajewski

Interpolacja za pomocą biblioteki GSL (akima)



Wykres wszystkich użytych metod interpolacyjnych.

Czarne punkty to tzw. węzły. Obserwujemy nakładanie się funkcji wielomianowych wyznaczonych za pomocą metod: polynomial, lagrange, newton.

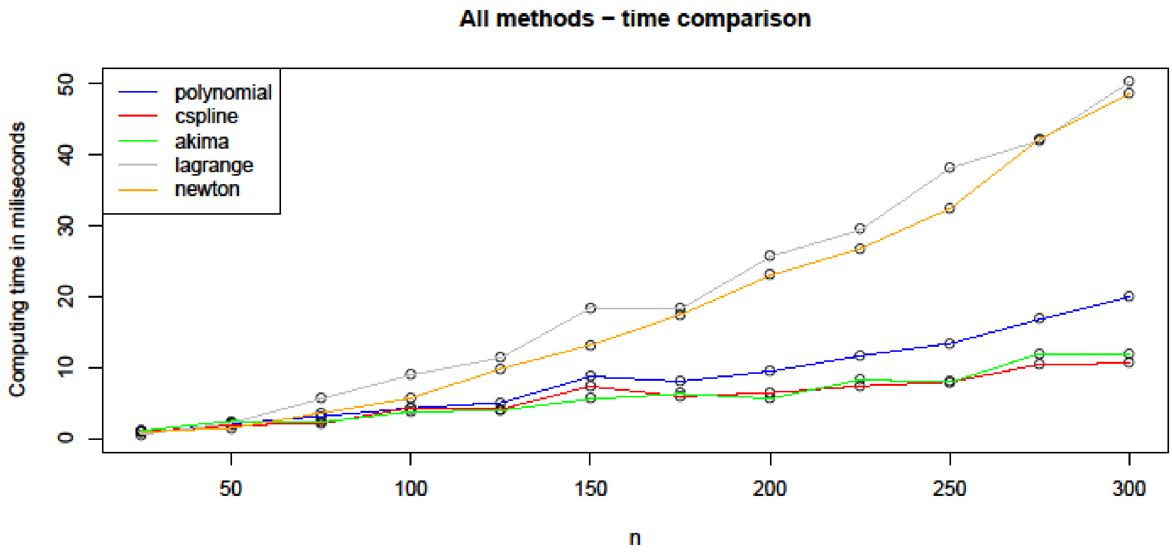


Łukasz Gajewski

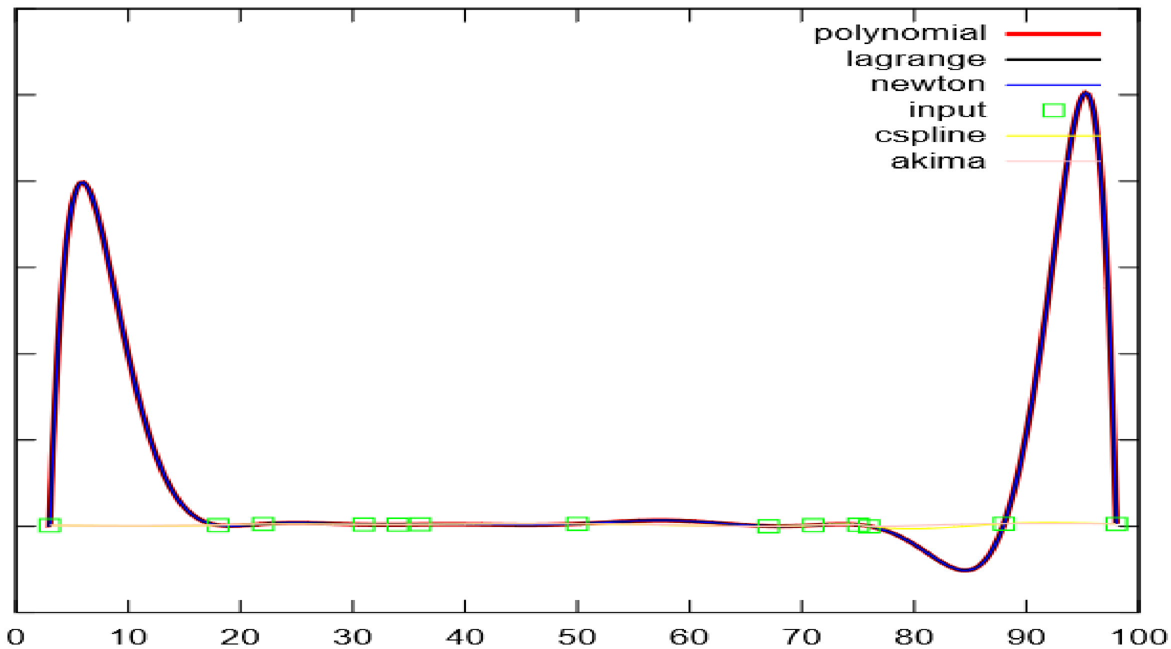
Podsumowanie czasów wykonania poszczególnych metod interpolacji

Czasy wyznaczone zostały za pomocą 10-krotnego pomiaru każdej z metod (dla każdej wartości n – liczby punktów do interpolacji). Przedstawione wartości to wartości średnie.

*Ze względów estetycznych błąd odchylenia standardowego nie został zaznaczony. Dokładniejsza analiza czasów wykonania wraz z błędami odchylenia standardowego zamieszczona jest w pliku '**plots/times.pdf**'*



Podsumowanie wielomianów interpolacyjnych – inna skala



Łukasz Gajewski

V. Algorytmy

a) Lagrange

```
zainicjuj wielomian W jako 0
dla każdego węzła
    zainicjuj wielomian Li jako 1
    zainicjuj mianownik jako 1
    dla każdego węzła
            jeżeli nie i'ty węzeł
                    mianownik *= (xi -xj)
                    wielomian Li *= jednomian (x-xj)
    dodaj Li do W
```

b) Newton

```
wypełnienie pierwszej kolumny ilorazów różnicowych
dla każdego wiersza począwszy od drugiego
        dla każdej kolumny począwszy od drugiej
        oblicz kolejny iloraz w tablicy //miejsce wyznacza aktualna kolumna i wiersz
zainicjalizuj wielomian omega jako 1
zainicjalizuj wielomian wynikowy W
dla każdego węzła xi
    skopiuj wielomian omega
    wynik wymnóż przez odpowiedni iloraz różnicowy z tablicy
    dodaj wynik do wielomianu wynikowego W
    wymnóż wielomian omega razy jednomian (x-xi)
```

VI. Wnioski

- interpolując punkty, które zostały wygenerowane za pomocą pewnej funkcji otrzymujemy stosunkowo duży błąd funkcji interpolującej do funkcji oryginalnej.
- dla metod interpolacji wielomianowej bardzo duże znaczenie ma dobór węzłów
- przy większej ilości węzłach obserwujemy efekt Runge'go – pogorszenie jakości interpolacji wielomianowej mimo zwiększenia jej liczby węzłów. Szczególnie widoczne na końcach przedziałów
- dla dużej ilości węzłów interpolacja wielomianowa staje się praktycznie bezużyteczna – efekt Rungego oraz złe uwarunkowanie numeryczne (duży stopień wielomianu)
- czas interpolacji znacznie szybszy w metodach zaimplementowanych przez bibliotekę GSL (wykres)
- czasowo interpolacja za pomocą metod Lagrange'a i Newton'a wygląda podobnie, jednak implementacyjnie dużo lepiej wypada Lagrange