

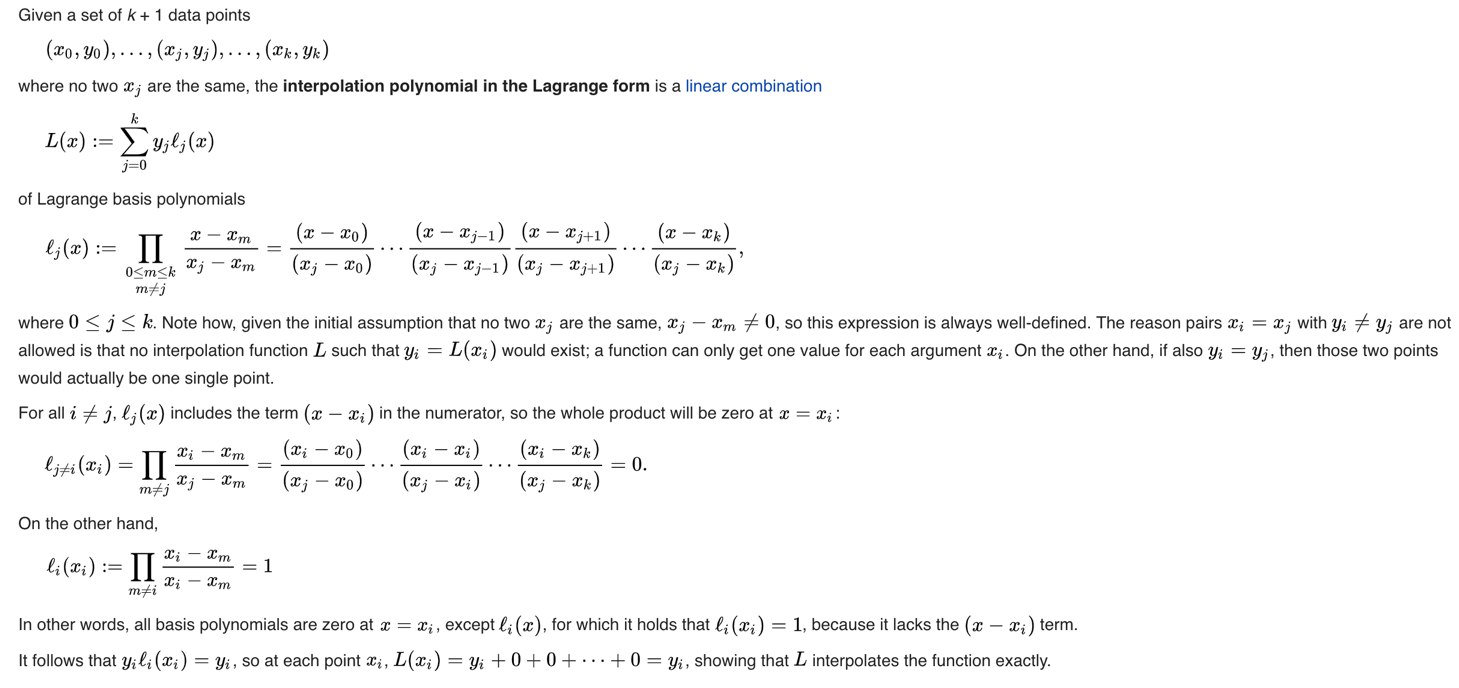
Metody Obliczeniowe w Nauce I Technice

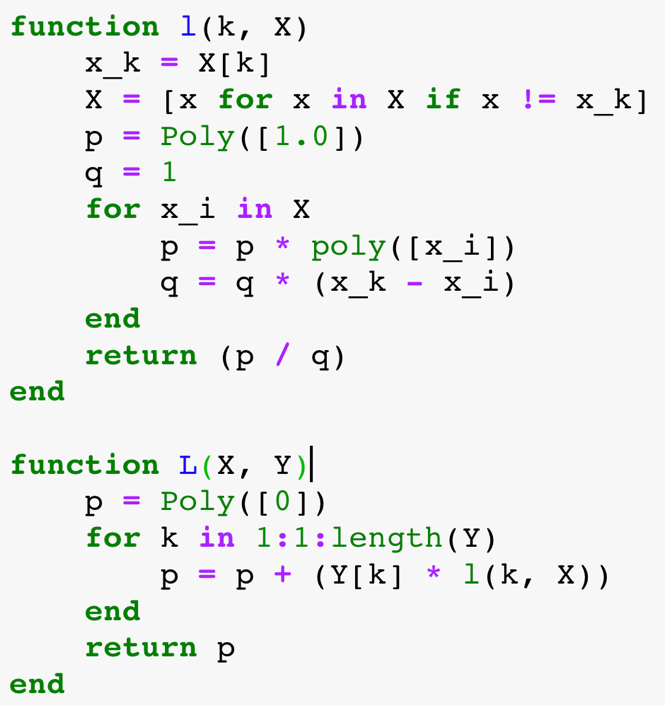
Laboratorium 5: Interpolacja – Sprawozdanie

Jakub Pajor

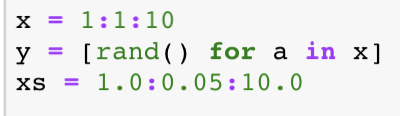
1. Napisać własną implementację interpolacji wielomianowej stosując wprost wzór na wielomian interpolacyjny Lagrange'a . Język implementacji do wyboru (Julia, C). Przetestować swoją implementację na wylosowanych węzłach interpolacji w wybranym przedziale. Narysować wykres wielomianu interpolacyjnego w tym przedziale wraz z wezlami interpolacji.

Definition:[[1]](#footnote-1)

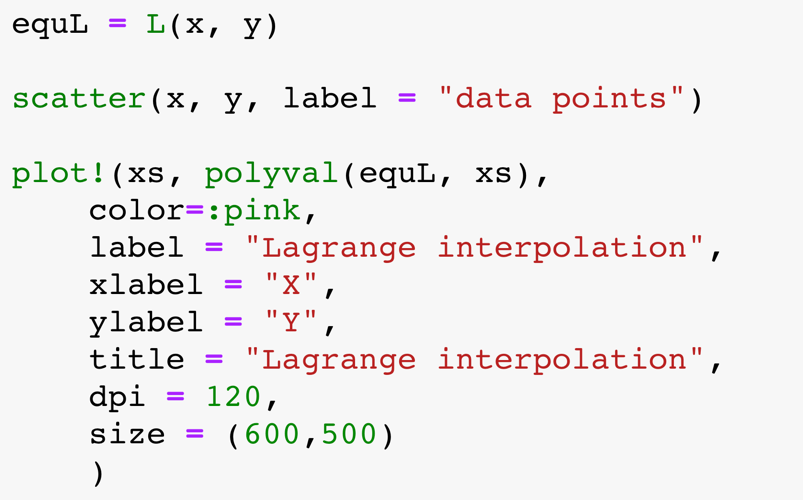
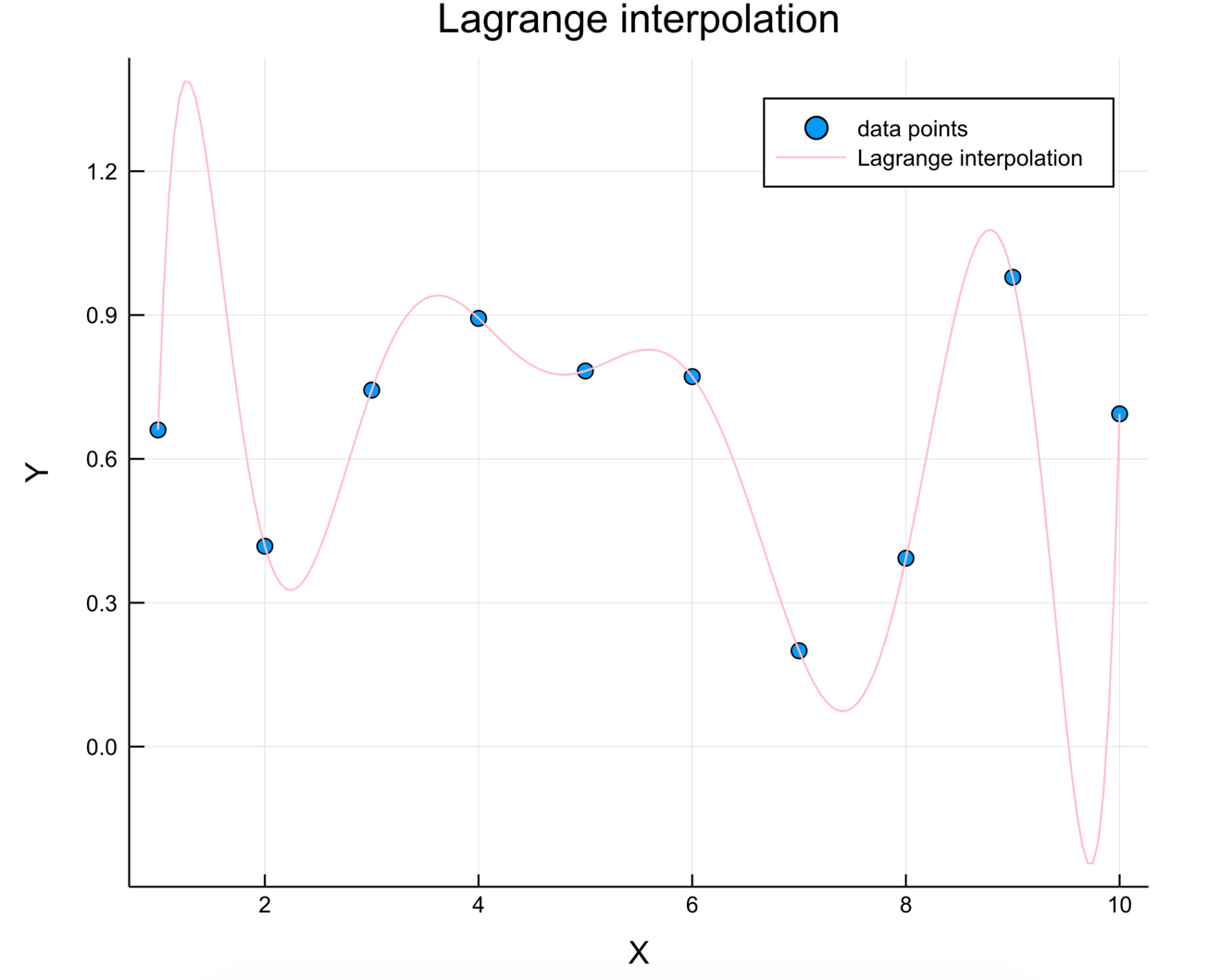


Implementation:

Code : implementation of given definition

Then I generated random points to carry out interpolations.

Code : interpolation points

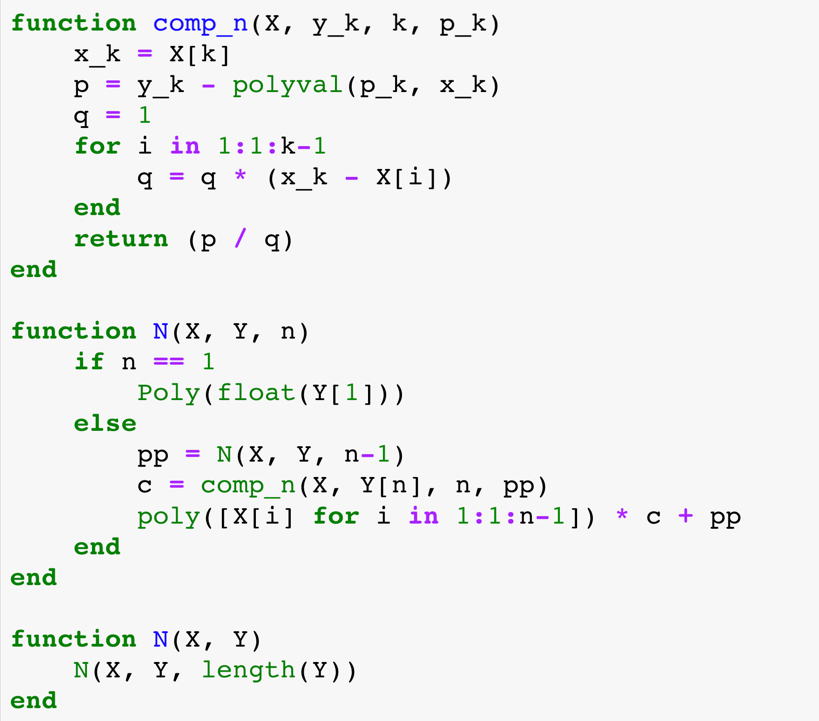
After that I used written function to calculate Lagrange interpolating polynomial, scatter created points and plot the polynomial.

Plot 1: Lagrange's interpolation

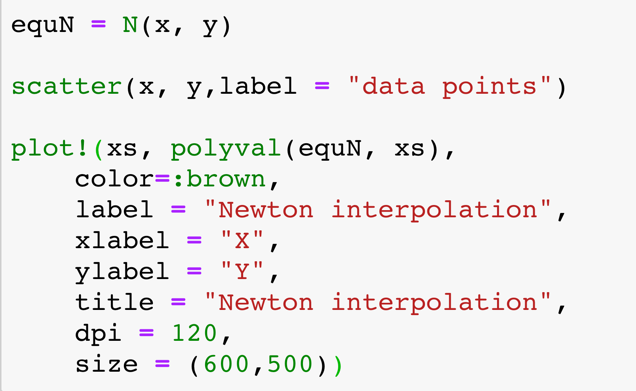
Code : polynomial and plot

1. Zrobić to samo dla metody Newtona (metoda ilorazów różnicowych). Zadbać o to, żeby ilorazy wyliczać tylko raz dla danego zbioru węzłów interpolacji. Język implementacji wybrać taki sam, jak w poprzednim punkcie. Narysować wykres wielomianu interpolacyjnego dla tych samych danych, co w poprzednim punkcie.

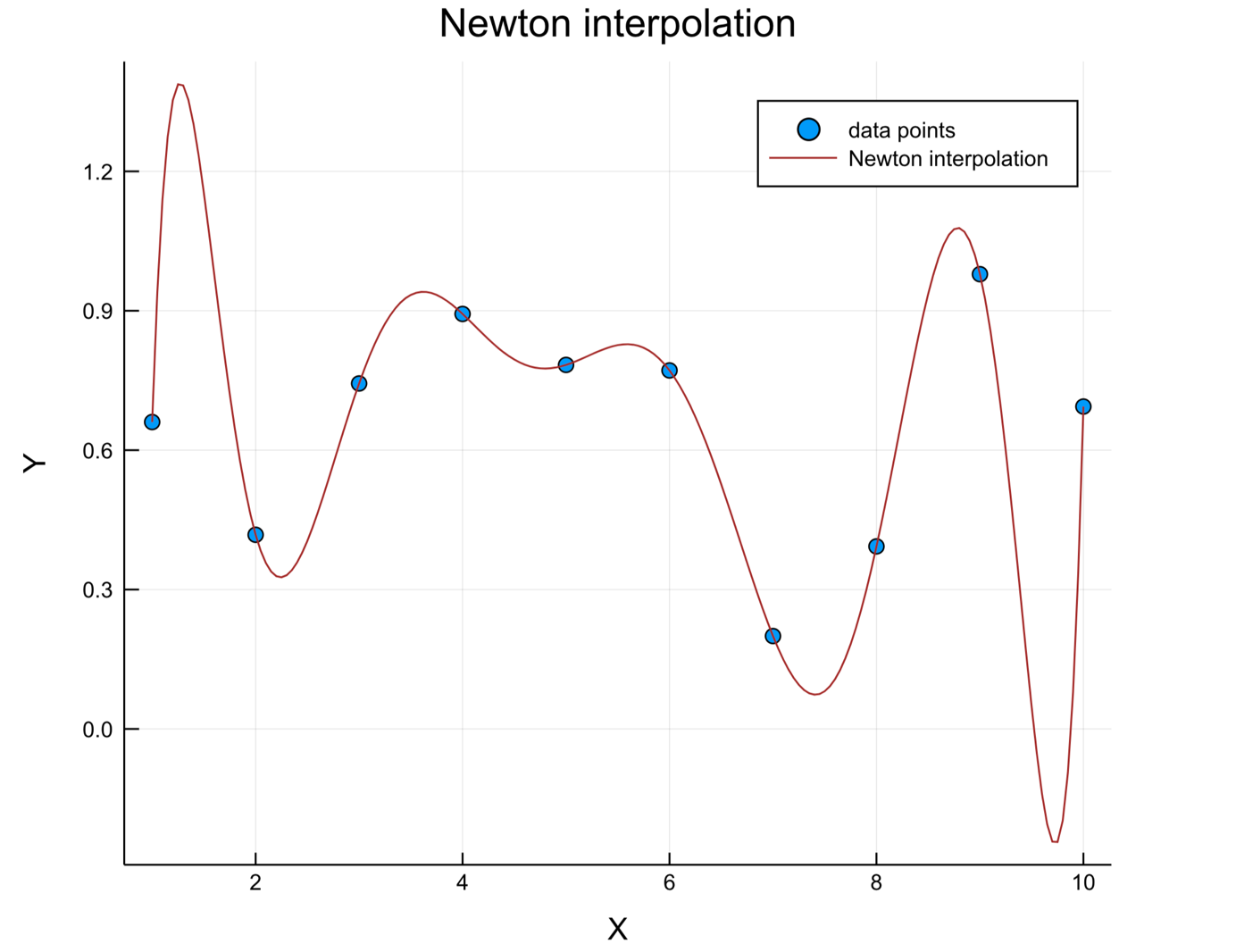
Definition:[[2]](#footnote-2)

Implementation:

Code : implementation of Newton method

After that I used written function to calculate Newton’s interpolating polynomial, scatter created points and plot the polynomial.

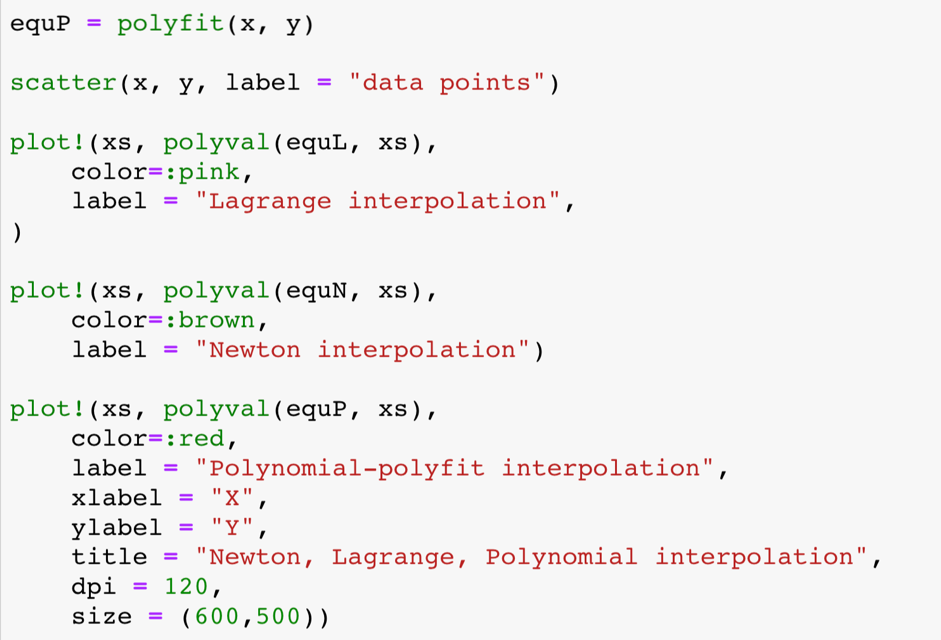
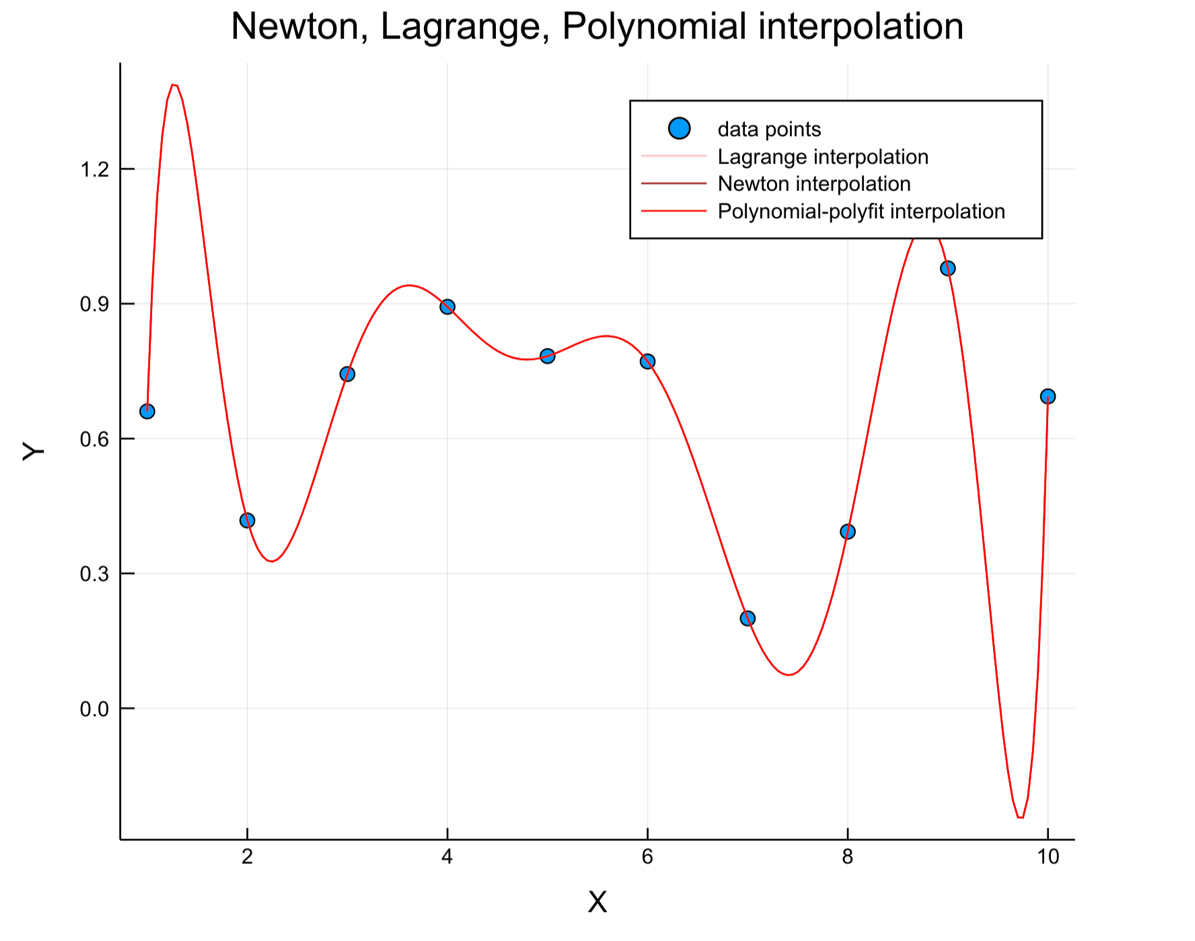
Code : polynomial and plot



Plot 2: Newton's interpolation

1. Zastosowac interpolację wielomianową z pakietu Polynomials do tych samych danych, co w poprzednich punktach. Porównać wszystkie 3 wyniki interpolacji wielomianowej na jednym wykresie. Co zauważamy? Dlaczego?

So I calculated Polynomial which fits generated points using embedded polyfit function and plotted each kind of interpolation.



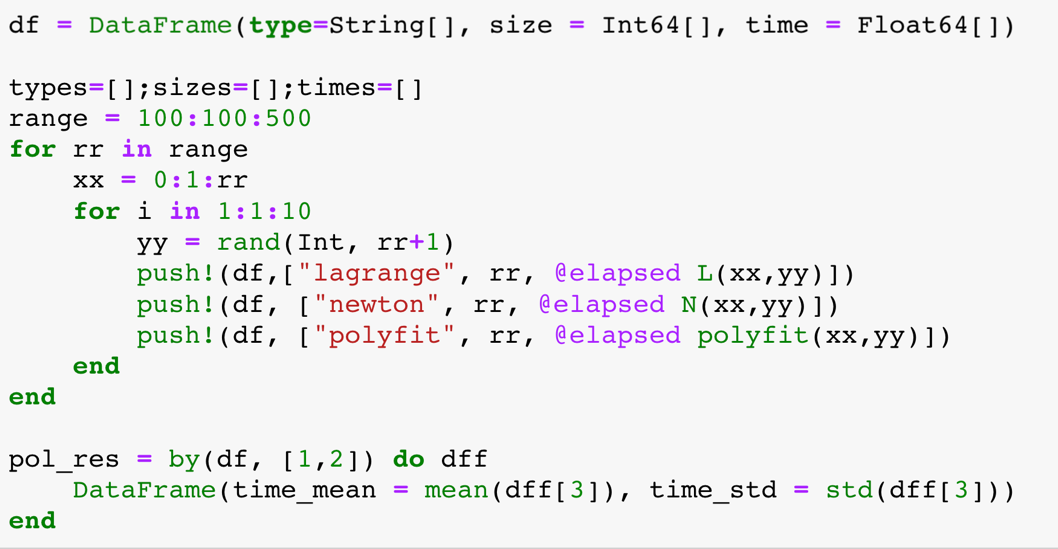
Plot 3: Langrange's, Newton's, Polyfit interpolation

Code : poly, Lagrange's, Newton's Interpolation

Observation: each of interpolating polynomials goes through the same points. It happens, because for each of given n + 1 points exists only one n’th degree polynomial.

1. Porównać metody poprzez pomiar czasu wykonania dla zmiennej ilości węzłów interpolacji. Dokonać pomiaru 10 razy i policzyć wartość średnią oraz oszacować błąd pomiaru za pomocą odchylenia standardowego.

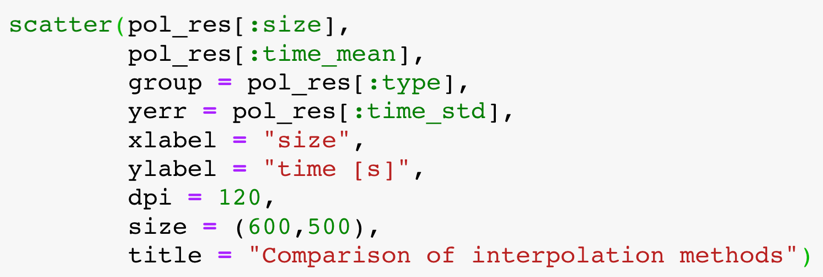
First I created specific DataFrame and then I filled it 10 times with 100 to 500 points; step 100.

In the end determined mean for each “bucket” and it’s standard deviation

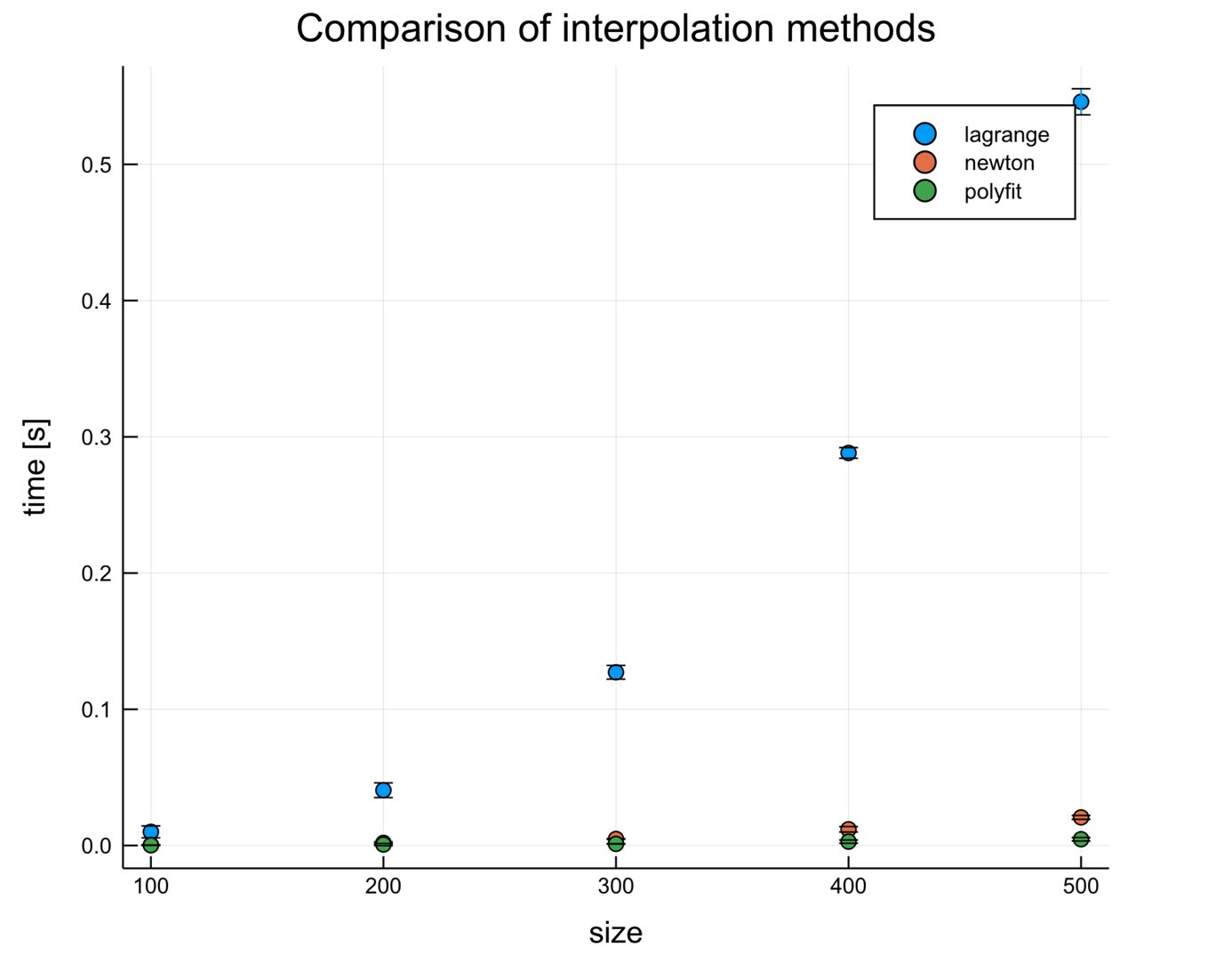
Code : dataframe, std, mean

Table : interpolation method times

|  | **type** | **size** | **time\_mean** | **time\_std** |
| --- | --- | --- | --- | --- |
|  | **String** | **Int64** | **Float64** | **Float64** |
| **1** | lagrange | 100 | 0.0100212 | 0.00435218 |
| **2** | newton | 100 | 0.000363283 | 0.000105236 |
| **3** | polyfit | 100 | 0.000195765 | 0.000147574 |
| **4** | lagrange | 200 | 0.0405791 | 0.00540252 |
| **5** | newton | 200 | 0.00191023 | 0.000790464 |
| **6** | polyfit | 200 | 0.000724788 | 0.000814945 |
| **7** | lagrange | 300 | 0.127135 | 0.00505647 |
| **8** | newton | 300 | 0.00477466 | 0.000192925 |
| **9** | polyfit | 300 | 0.00120259 | 0.000118685 |
| **10** | lagrange | 400 | 0.28814 | 0.0039224 |
| **11** | newton | 400 | 0.0118855 | 0.00200529 |
| **12** | polyfit | 400 | 0.00283653 | 0.00120733 |
| **13** | lagrange | 500 | 0.54594 | 0.0095752 |
| **14** | newton | 500 | 0.0206329 | 0.00141171 |
| **15** | polyfit | 500 | 0.00458597 | 0.00128937 |

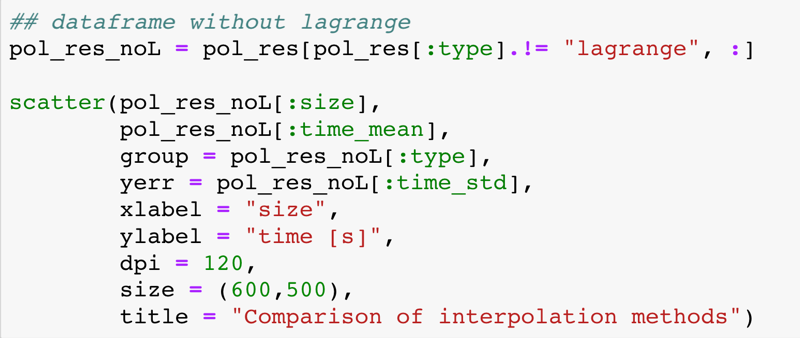


Code : scatter of table 1

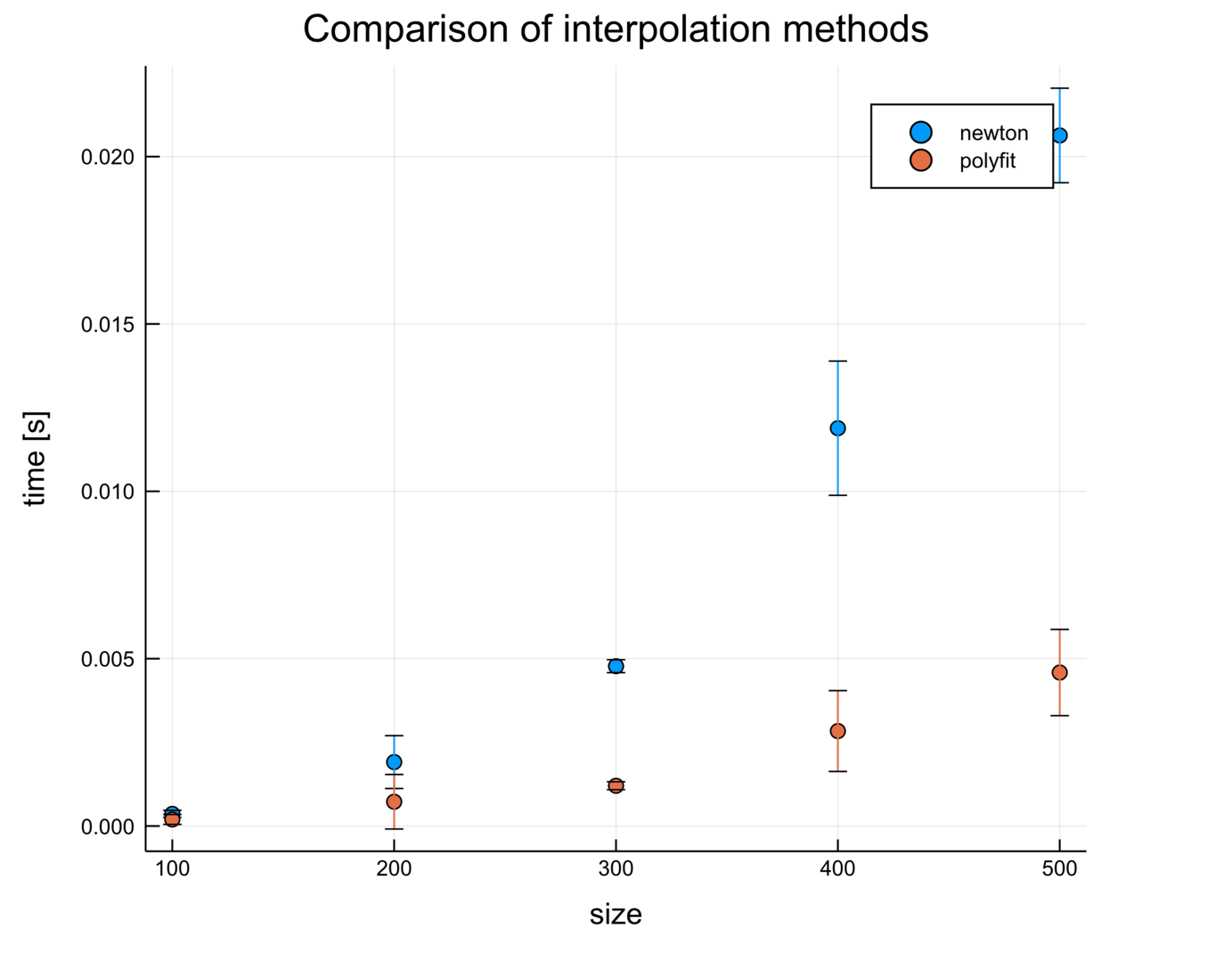


Plot 4: comparison of interpolation methods #1

As we can see using Langrange’s polynomial the time is much slower than Newton’s or using embedded polyfit.

After that I created another scatter to distinguish Newton’s and Polyfit times.

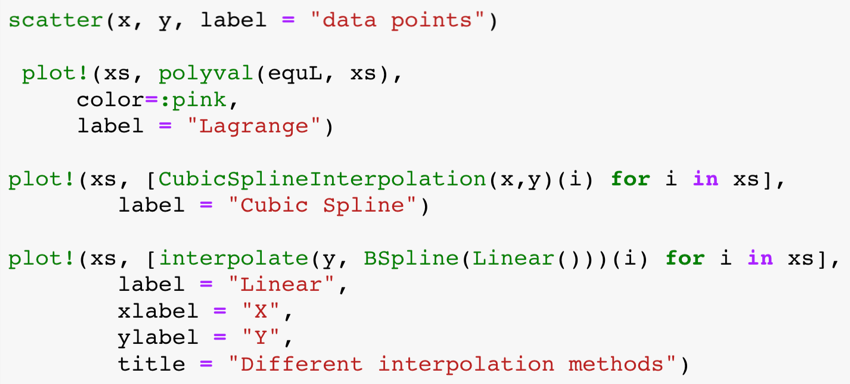
Code : Newton's, Polyfit times



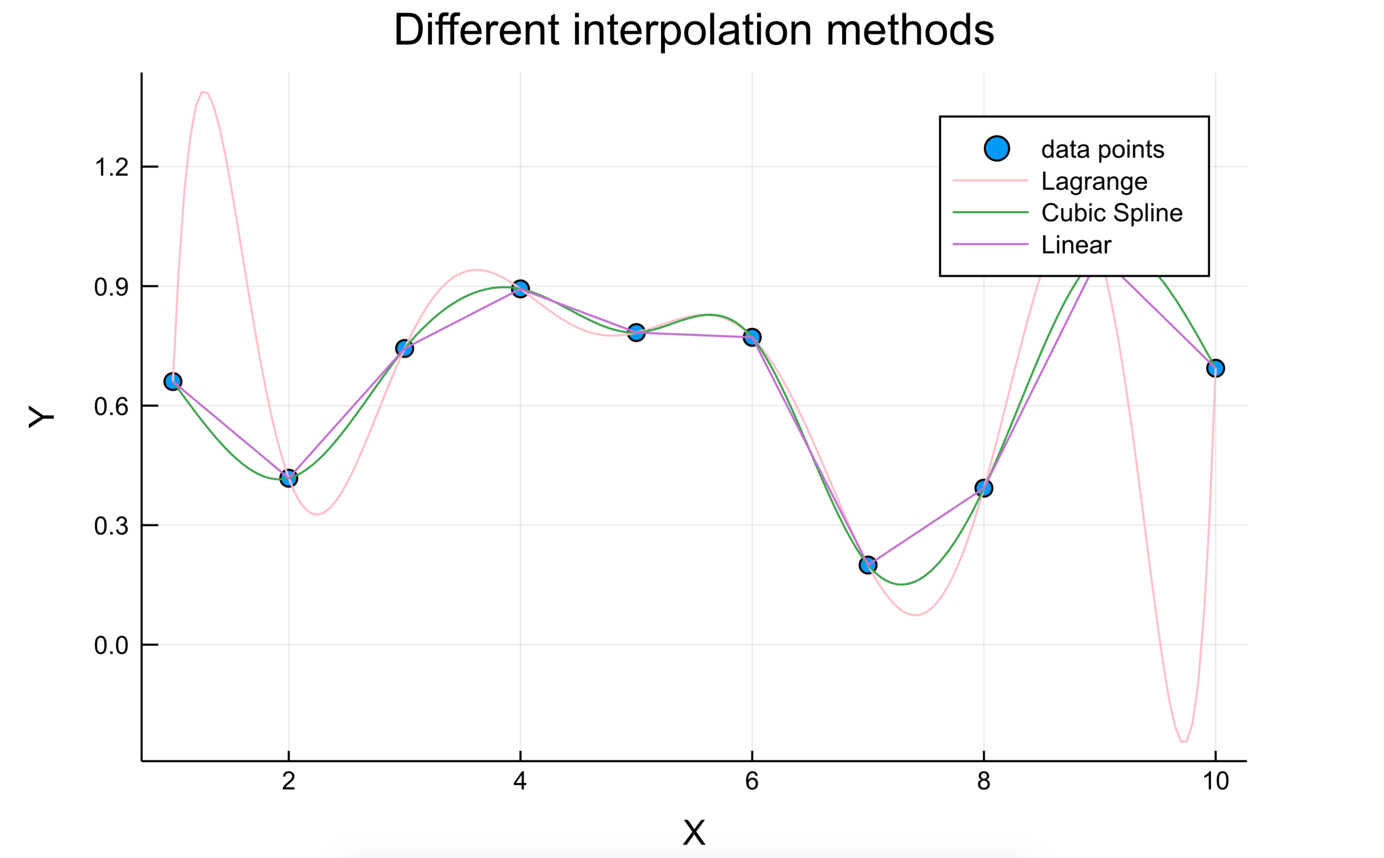
Plot 5: comparison of interpolations methods #2

Even Newton’s interpolation is slower by a square of polyfit time.

1. Poeksperymentować z interpolacją funkcjami sklejanymi (minimum dwie rożne funkcje sklejane), narysować wykresy i porównać z wykresami interpolacji wielomianowej.



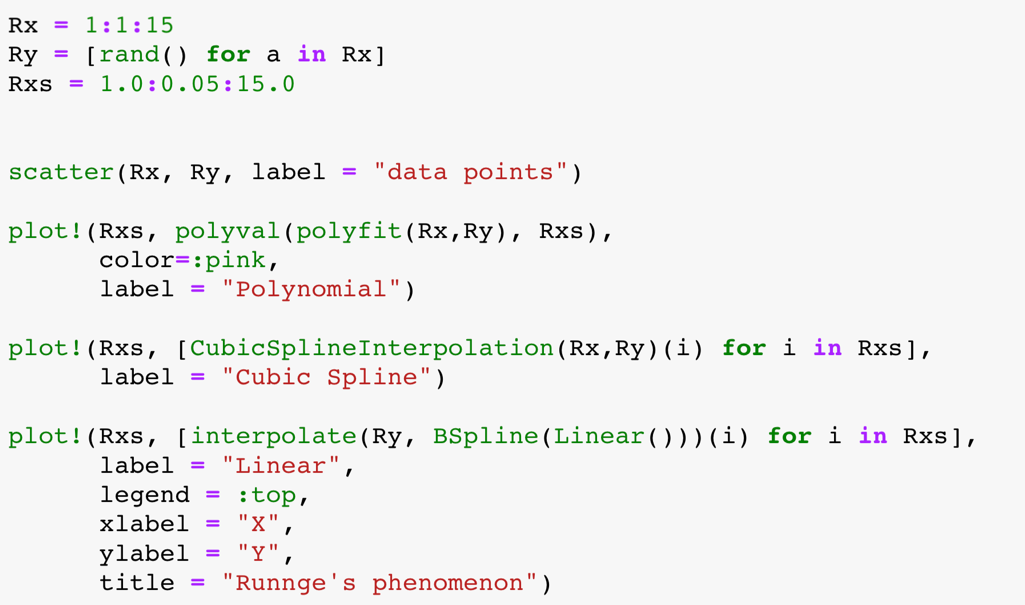
Code 10: plotting Lagrange, Cubic Spline, Linear interpolation



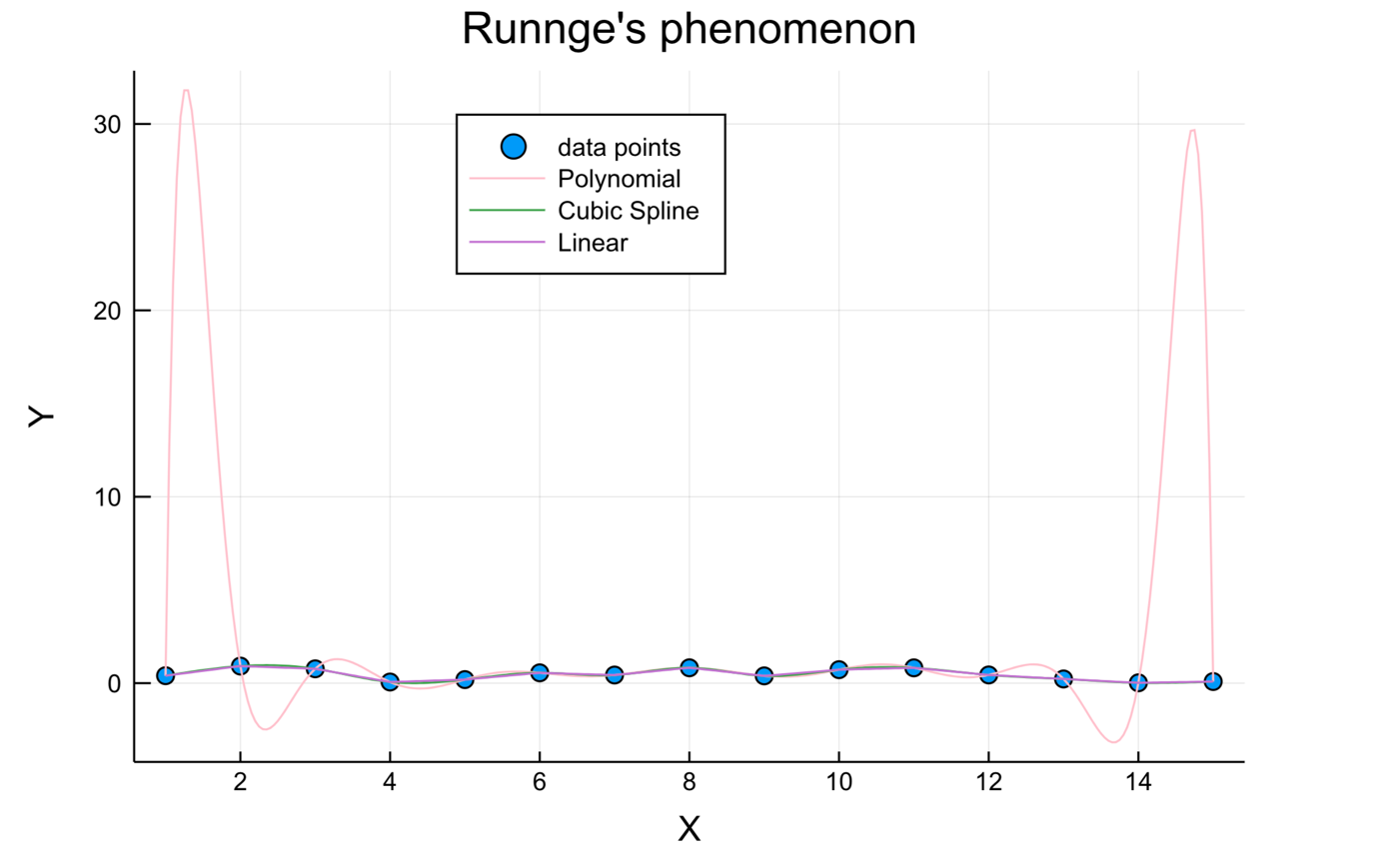
Plot 6: Lagrange, Cubic Spline, Linear interpolation

Lagrange’s interpolation (so also Newton’s interpolation) is in general the least accurate, but mostly on first and last edges. It is Runge’s phenomenon.

1. Zademonstrować efekt Rungego.

I generated again random points but this time more of them (15) and plotted them with Polynomial, Cubic Spline and linear interpolation.

Code 11: points, polyfit, cublic spline, linear



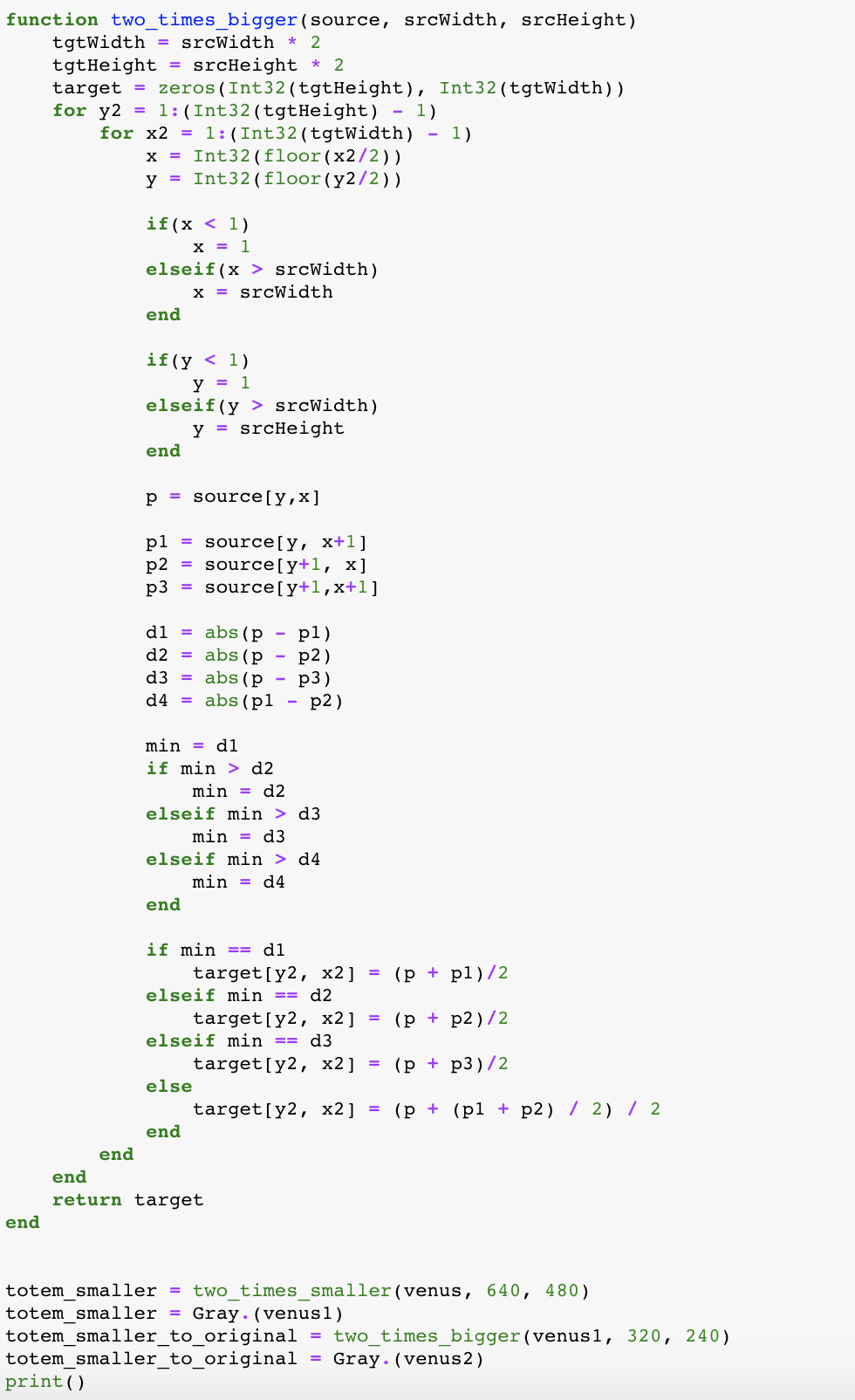
Plot 7: Runge's Phenomenon

Runge's phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points. It shows that going to higher degrees does not always improve accuracy.

1. Zbadać i zademonstrować podczas zajęć różne algorytmy interpolacji stosowane w grafice komputerowej (np. do zmiany wielkości obrazu). Można korzystać z gotowych rozwiązań, ale trzeba wiedzieć, jak te algorytmy działają. Do zaliczenia tego zadania potrzebne jest demonstracja i porównanie działania co najmniej dwóch metod.



Code : image size reducing



Code : image size enlarging

Picture : original size



Picture : reduced size



Picture : reduced and enlarged

1. https://en.wikipedia.org/wiki/Lagrange\_polynomial#Definition [↑](#footnote-ref-1)
2. https://en.wikipedia.org/wiki/Newton\_polynomial#Definition [↑](#footnote-ref-2)