Piotr Trybisz, 193557
13/06/2024

Metody Numeryczne, informatyka 2023/2034

Aproksymacja Profilu Wysokościowego

Spis treści

1. Wstęp

Celem projektu jest porównanie wybranych algorytmów interpolacji:

- wielomianowej Lagrange'a
- przy użyciu funkcji sklejanych trzeciego stopnia

Algorytmy zostały zaimplementowane w języku C++ bez użycia zewnętrznych bibliotek. Do wyświetlenia wykresów użyłem biblioteki Matplotlib z języka Python.

1.1. Używane metody

1.1.1. Interpolacja wielomianowa Lagrange'a

Metoda interpolacji wielomianowej Lagrange'a dla n punktów polega na znalezieniu wielomianu stopnia $n-1$ przechodzącego przez te punkty:

$$F(x) = \sum_{i=1}^{n+1} y_i \phi_i(x) \text{ , gdzie } \phi_i(x) = \prod_{j=1, j\neq i}^{n+1} \frac{(x - x_j)}{(x_i - x_j)}$$

Jednakże, metoda ta ma zasadniczą wadę - efekt Rungego, czyli oscylacje wielomianu interpolującego na krańcach przedziału. Można go jednak wyeliminować stosując np. węzły Czybyszewa, zamiast rozmieszczone równomiernie.

1.1.2. Interpolacja funkcjami sklejanymi trzeciego stopnia

W przypadku interpolacji funkcjami sklejanymi trzeciego stopnia należy znaleźć funkcję przechodzącą przez wszystkie punkty, mającą ciągłe pochodne pierwszego i drugiego stopnia. $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$, gdzie a, b, c, d wyznaczamy tak, że:

- W x_i wartość funkcji sklejanej jest równa funkcji oryginalnej:

$$S_i(x_i) = y_i$$

- Pierwsze i drugie pochodne funkcji na krańcach przedziałów $[x_i, x_{i+1}]$ są równe:

$$S_i{'}(x_{i+1}) = S_{i+1}{'}(x_{i+1})$$

$$S_i{''}(x_{i+1}) = S_{i+1}{''}(x_{i+1})$$

- Drugie pochodne w punktach skrajnych są równe zero:

$$S_1{''}(x_1) = 0$$

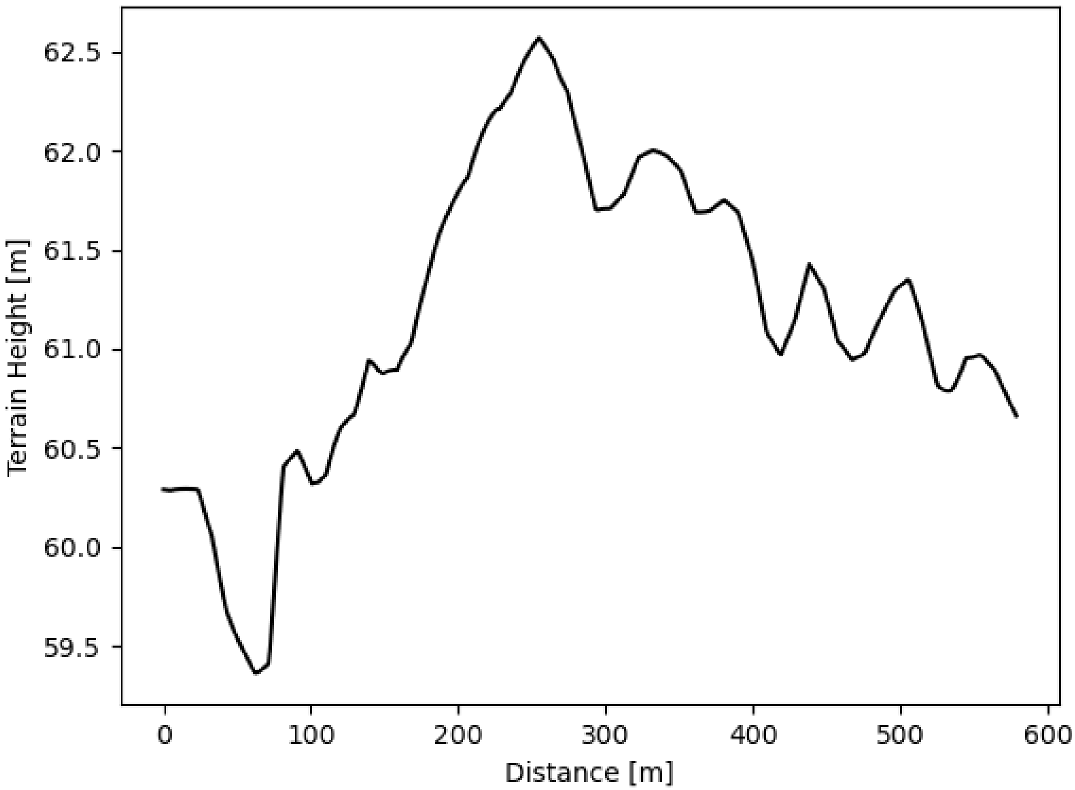
$$S_n{''}(x_n) = 0$$

Do rozwiązania powyższego układu równań wykorzystałem metodę LU zaimplementowaną w poprzednim projekcie. Niewątpliwą przewagą tej metody nad interpolacją Lagrange'a jest niewystępowanie efektu Rungego, jednak jest ona dosyć skomplikowana implementacyjnie.

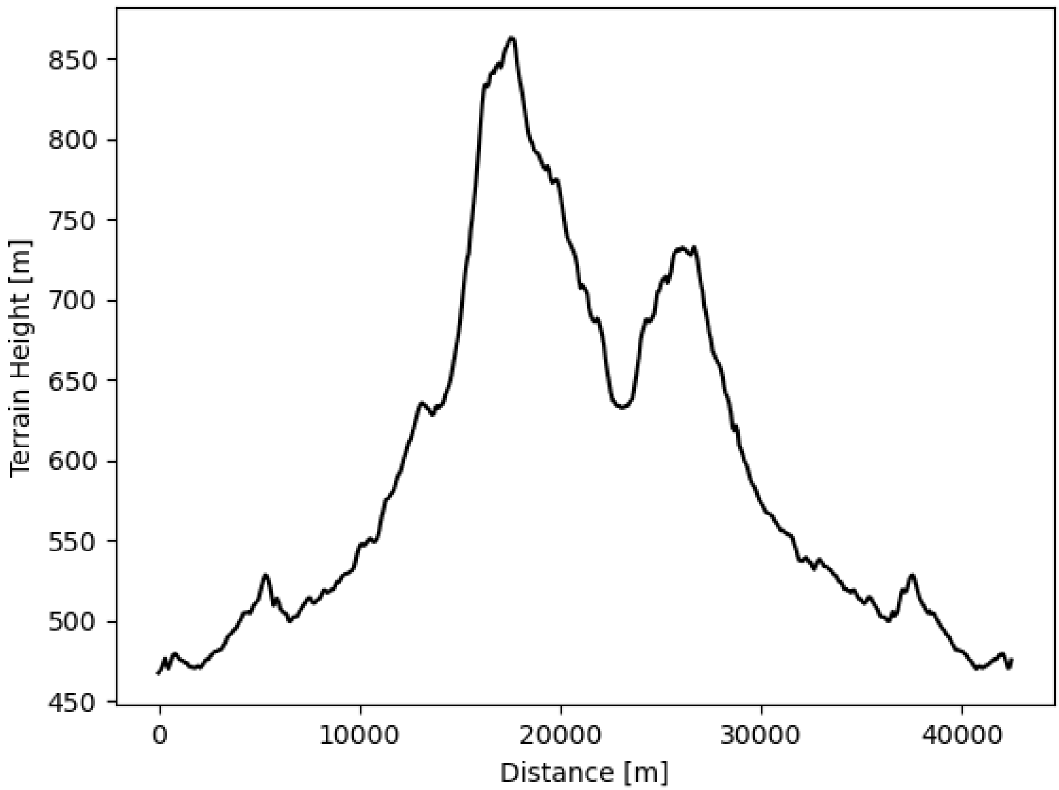
2. Badane trasy

Dane użyte w projekcie pochodzą z wyścigów kolarskich. Do badania metod interpolacji wybrałem dwie trasy:

- `chelm.txt` - płaska, stosunkowo niezróżnicowana trasa do podstawowych testów działania algorytmów interpolacji

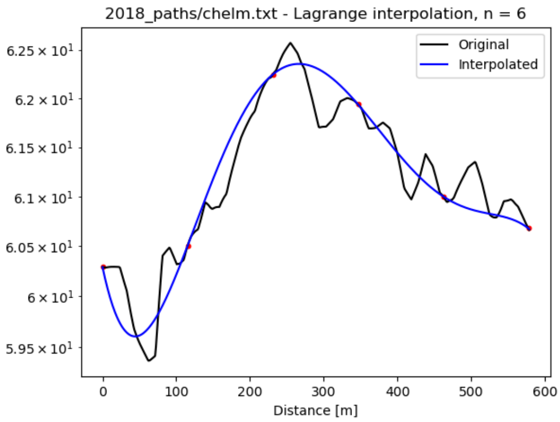


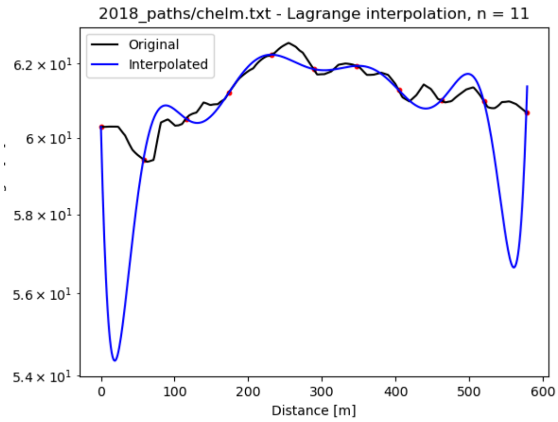
- `rozne_wniesienia.txt` - trudniejsza trasa, zawierająca ostre wahania wysokości terenu

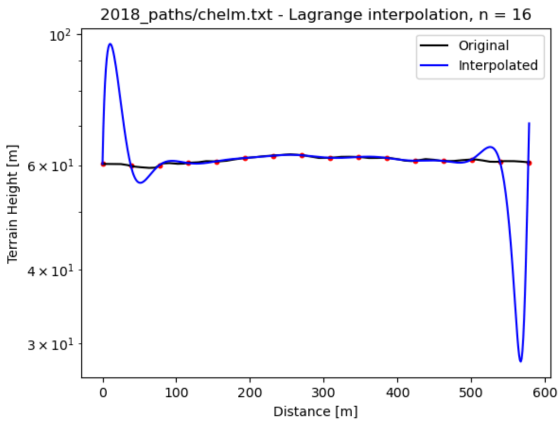


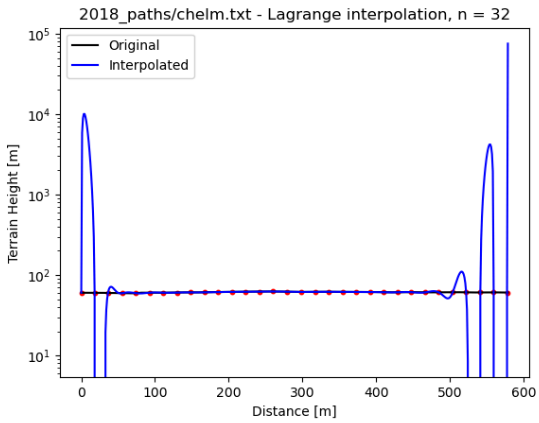
3. Interpolacja wielomianowa Lagrange'a

3.1. Trasa 1 - `chelm.txt`



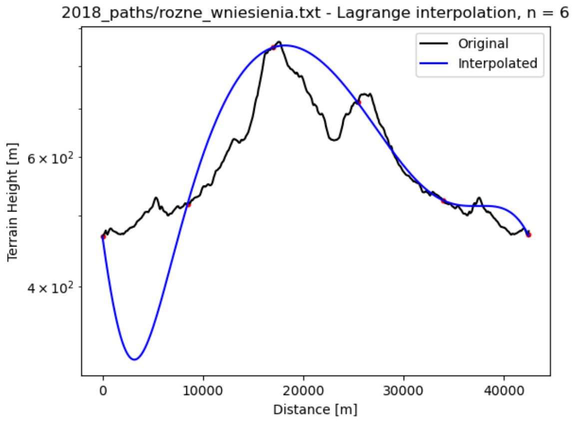


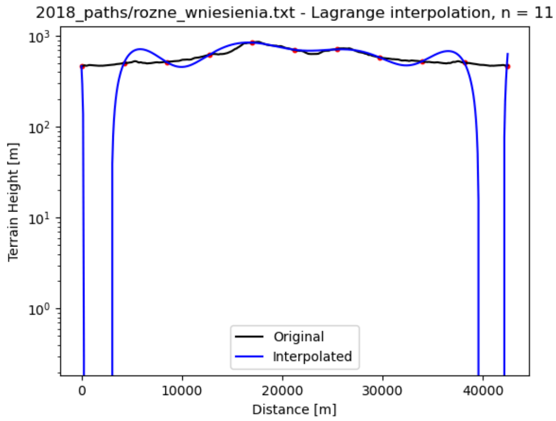


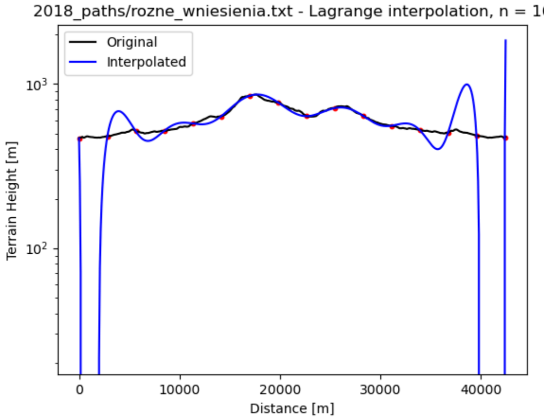


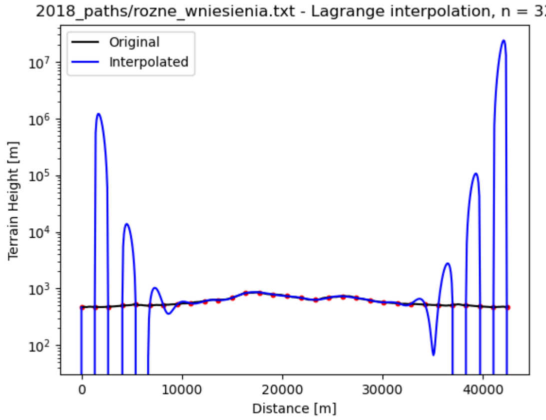
Dla niskiego n interpolacja przebiega poprawnie, jednakże wraz ze wzrostem n zaczyna być zauważalny efekt Rungego.

3.2. Trasa 2 - `rozne_wzniesienia.txt`





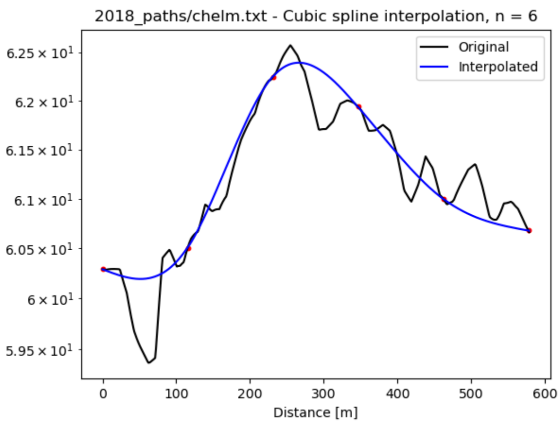


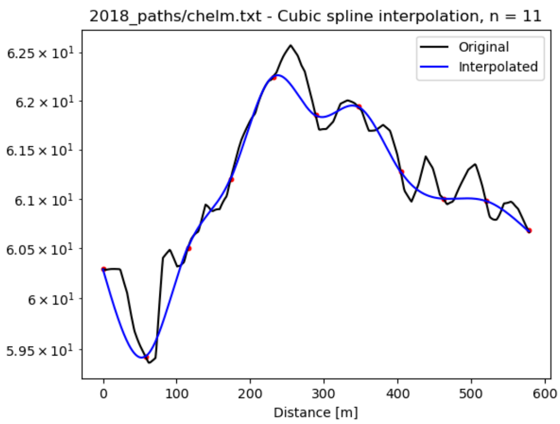


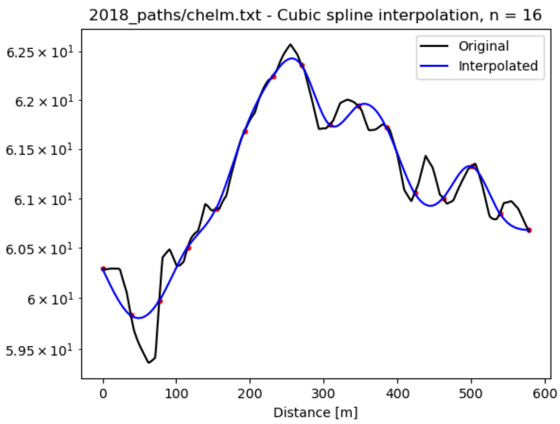
Już dla niewielkich wartości n na krańcach przedziału interpolacji widoczny jest efekt Rungego.

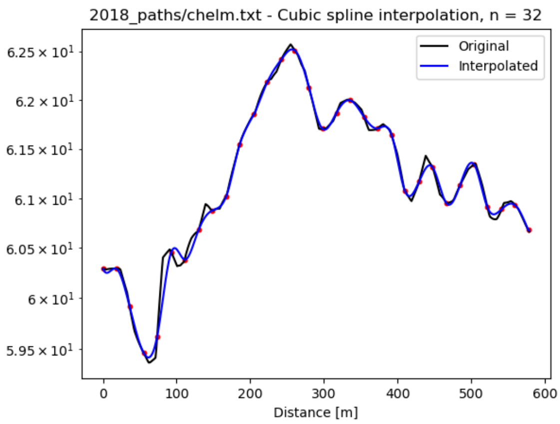
4. Interpolacja przy użyciu kubicznych funkcji sklejanych

4.1. Trasa 1 - `chelm.txt`



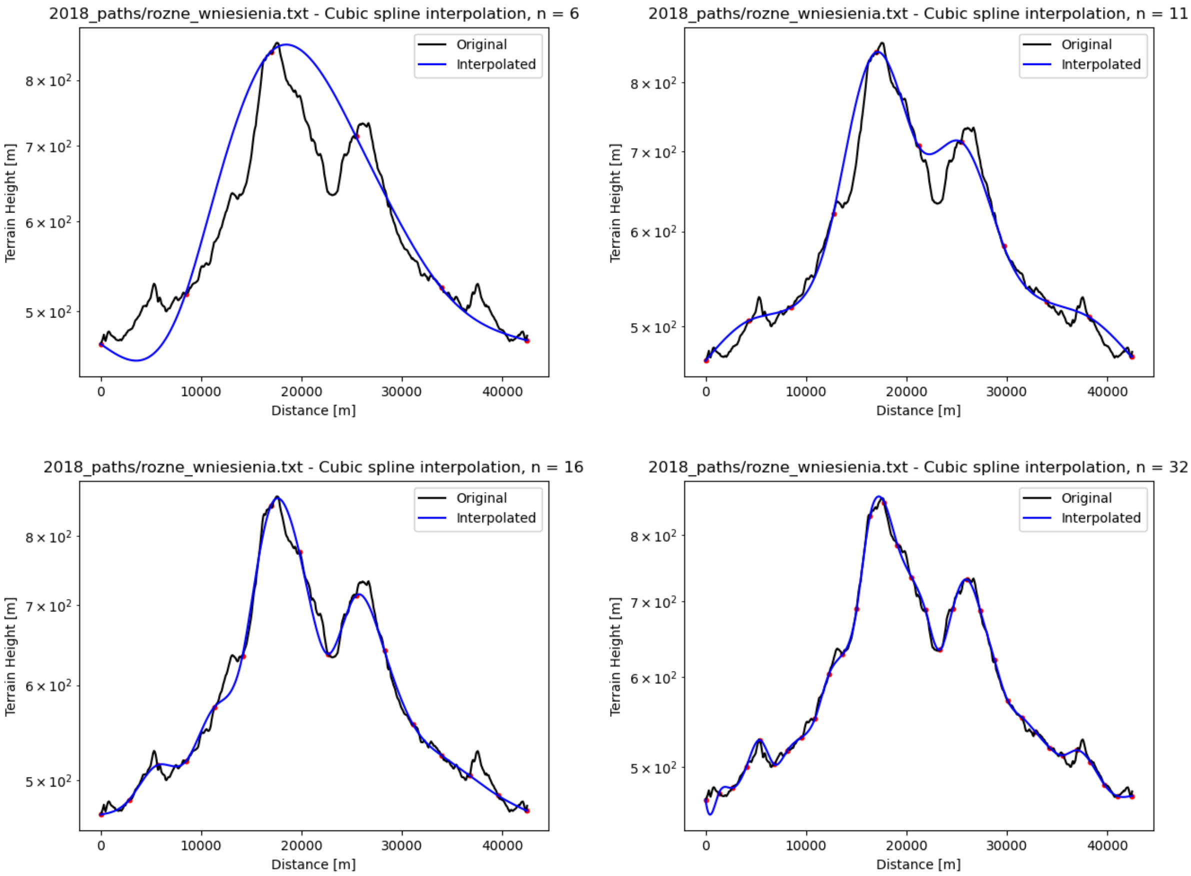






Widać wyraźnie, że interpolacja jest zadowalająca już przy niewielkiej liczbie węzłów. Ponadto, nie występuje efekt Rungego.

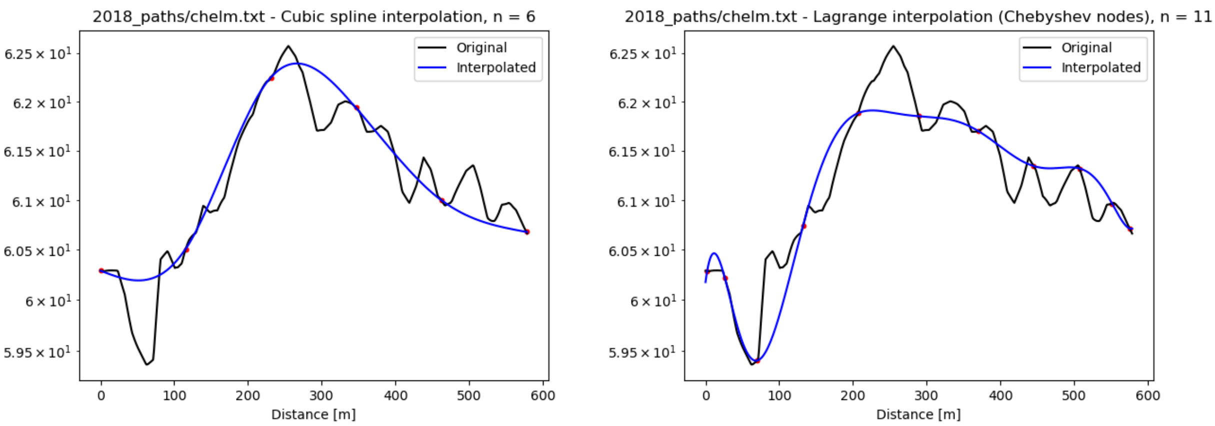
4.2. Trasa 2 - `rozne_wzniesienia.txt`

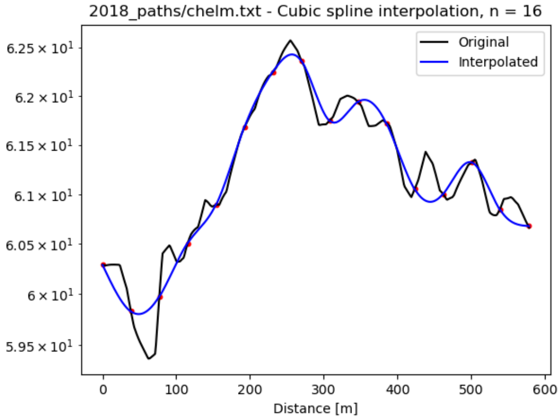


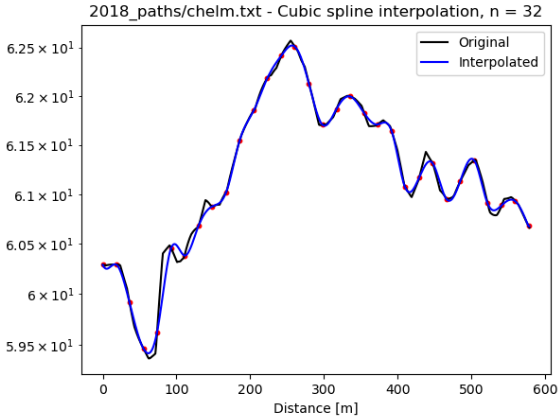
Nawet dla bardziej skomplikowanych tras funkcje sklejane dobrze się spisują.

5. Interpolacja wielomianowa Lagrange'a z użyciem węzłów Czybyszewa

5.1. Trasa 1 - `chelm.txt`

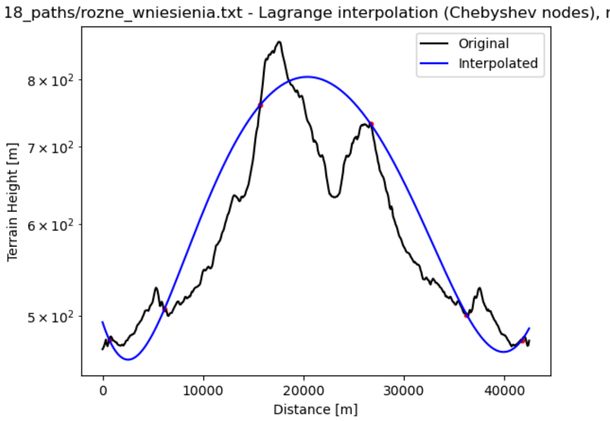


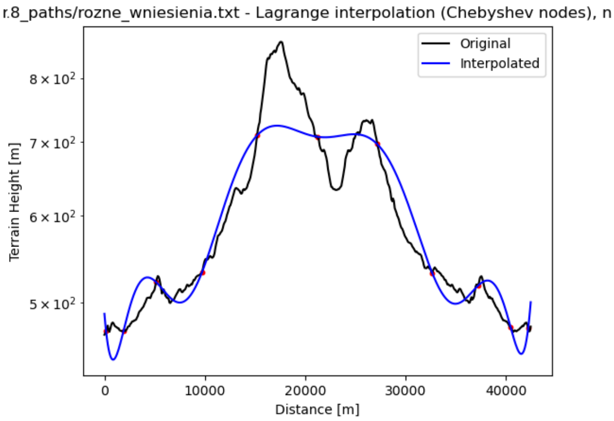


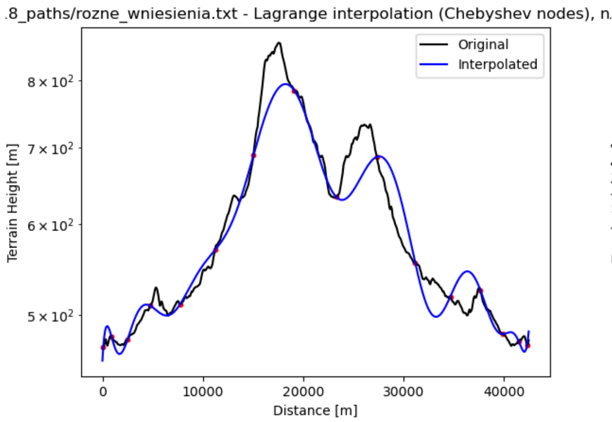


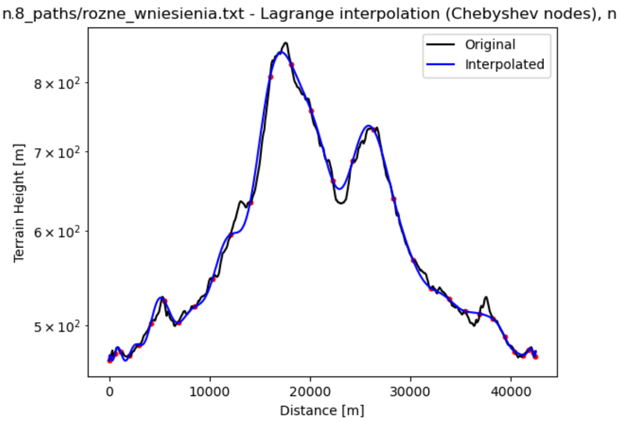
Zastosowanie węzłów Czybyszewa pozwala uniknąć pojawiania się efektu Rungego na krańcach przedziału interpolacji.

5.2. Trasa 2 - `rozne_wzniesienia.txt`









Dla funkcji o większym stopniu skomplikowania użycie węzłów Czybyszewa również pozwala uniknąć niepożądanego efektu Rungego.

6. Wnioski

Do funkcji o niewielkim stopniu zróżnicowania odpowiednią metodą wydaje się interpolacja wielomianowa Lagrange'a. Jeśli chcemy interpolować funkcję bardziej zróżnicowaną lub zależy nam na dokładności interpolacji, należy zastosować węzły Czybyszewa lub funkcje sklejane, co jest nieco trudniejsze implementacyjnie, ale przynosi bardzo dobre rezultaty.