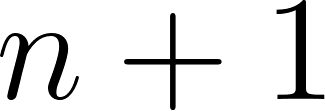
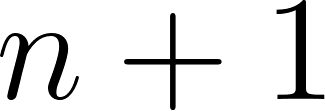
Metody numeryczne – projekt 3

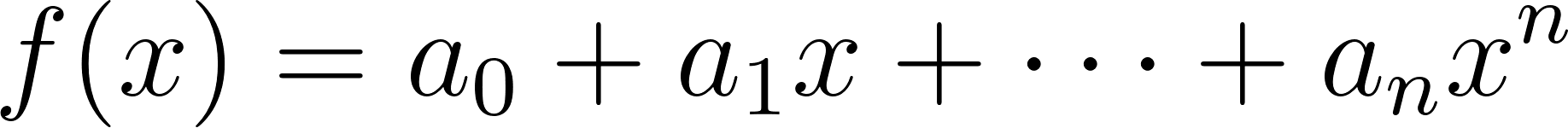
W tym projekcie wykorzystałem dwie metody interpolacji: interpolację Lagrange'a oraz interpolację funkcjami sklejanymi.

# Teoria

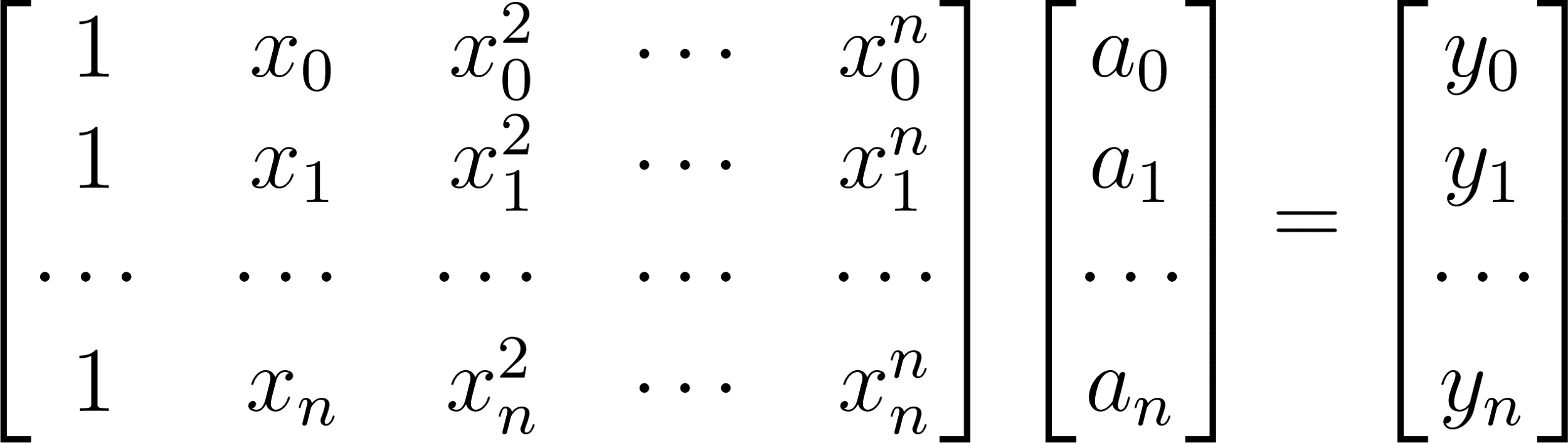
## Interpolacja Lagrange’a

Interpolacja Lagrange’a to inaczej interpolacja wielomianowa. W tej metodzie szukamy wielomianu n-tego stopnia, który przechodzi przez [](https://www.codecogs.com/eqnedit.php?latex=n%2B1#0) punktów (węzłów):

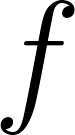
W tym celu układamy [](https://www.codecogs.com/eqnedit.php?latex=n%2B1#0) równań, żeby znaleźć [](https://www.codecogs.com/eqnedit.php?latex=n#0) współczynników [](https://www.codecogs.com/eqnedit.php?latex=a#0).

[](https://www.codecogs.com/eqnedit.php?latex=f(x)%20%3D%20a_0%20%2B%20a_1x%20%2B%20%5Ccdots%20%2B%20a_nx%5En#0)

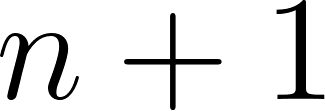
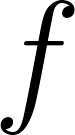
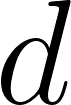
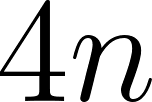
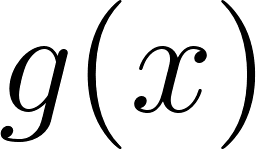
Ten układ rownan można zapisac macierzowo:

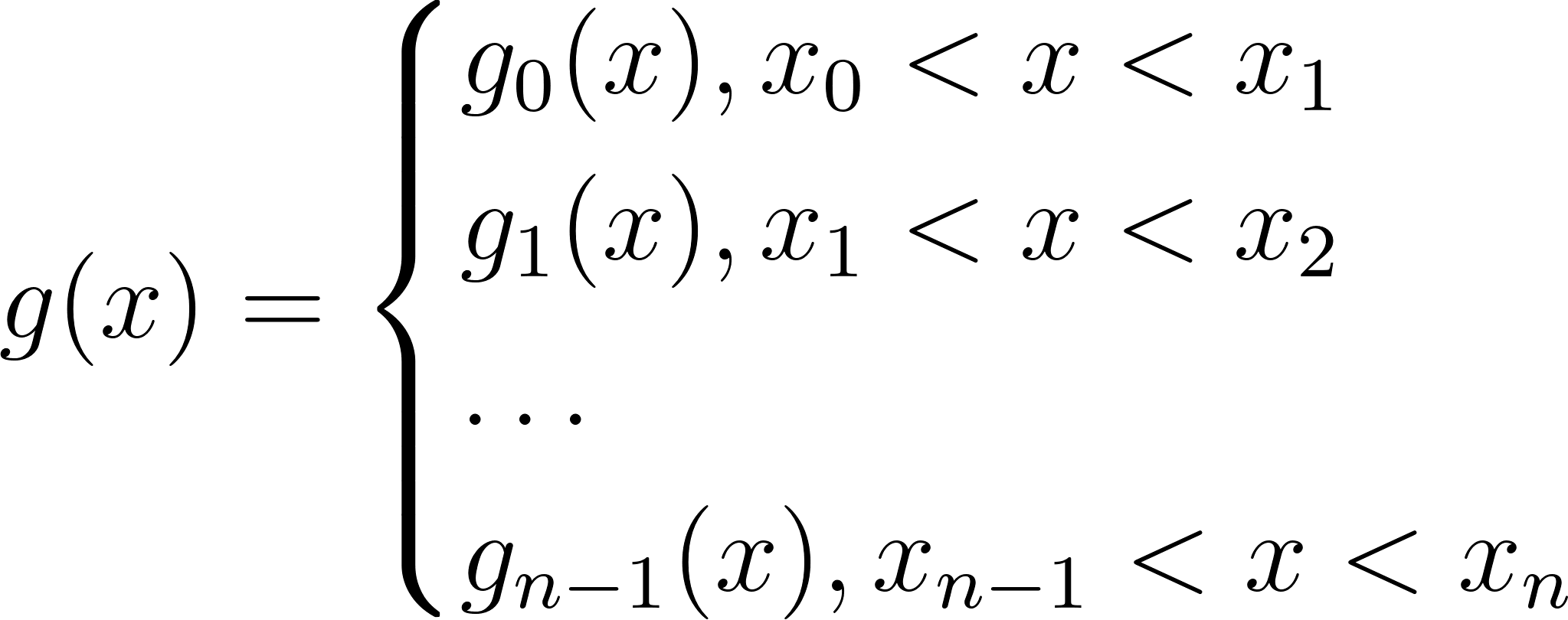
[](https://www.codecogs.com/eqnedit.php?latex=%5Cbegin%7Bbmatrix%7D%201%20%26%20x_0%20%26%20x_0%5E2%20%26%20%5Ccdots%20%26%20x_0%5En%20%5C%5C%5C%5C%201%20%26%20x_1%20%26%20x_1%5E2%20%26%20%20%5Ccdots%20%26%20x_1%5En%20%5C%5C%5C%5C%20%5Ccdots%20%26%20%20%5Ccdots%20%26%20%5Ccdots%20%26%20%5Ccdots%20%26%20%20%5Ccdots%20%5C%5C%5C%5C%201%20%26%20x_n%20%26%20x_n%5E2%20%26%20%5Ccdots%20%26%20x_n%5En%20%5Cend%7Bbmatrix%7D%20%5Cbegin%7Bbmatrix%7D%20a_0%20%5C%5C%5C%5C%20a_1%20%5C%5C%5C%5C%20%5Ccdots%20%5C%5C%5C%5C%20a_n%20%5C%5C%5C%5C%20%5Cend%7Bbmatrix%7D%20%3D%20%20%5Cbegin%7Bbmatrix%7D%20y_0%20%5C%5C%5C%5C%20y_1%20%5C%5C%5C%5C%20%5Ccdots%20%5C%5C%5C%5C%20y_n%20%5C%5C%5C%5C%20%5Cend%7Bbmatrix%7D%20#0)

## Interpolacja funkcjami sklejanymi

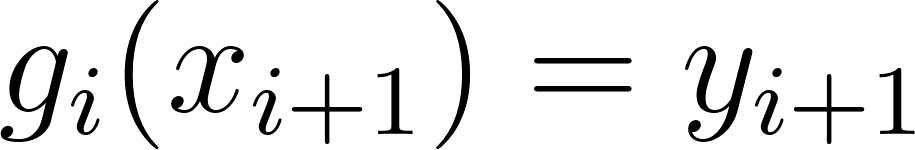
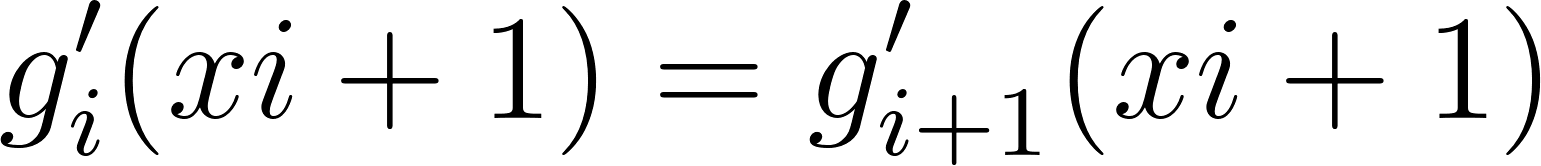
Interpolacja funkcjami sklejanymi to metoda w której przybliżamy pewną funkcję [](https://www.codecogs.com/eqnedit.php?latex=f#0), wieloma funkcjami postaci:

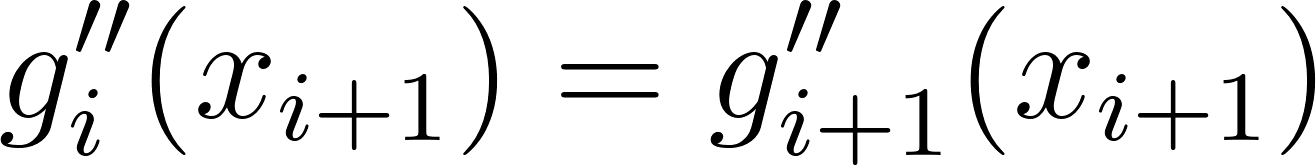
[](https://www.codecogs.com/eqnedit.php?latex=g_i(x)%20%3D%20a_i(x-x_i)%5E3%20%2B%20b_i(x-x_i)%5E2%20%2B%20c_i(x-x_i)%20%2B%20d_i#0)

Jest to wielomian 3 stopnia, k[](https://www.codecogs.com/eqnedit.php?latex=g_i#0)tórego początek jest przesunięty na [](https://www.codecogs.com/eqnedit.php?latex=x_i#0). Dla [](https://www.codecogs.com/eqnedit.php?latex=n%2B1#0) punktów ([](https://www.codecogs.com/eqnedit.php?latex=x_i#0)) z funkcji [](https://www.codecogs.com/eqnedit.php?latex=f#0) jest [](https://www.codecogs.com/eqnedit.php?latex=n#0) funkcji [](https://www.codecogs.com/eqnedit.php?latex=g_i#0). Zeby znaleźć wspolczynniki [](https://www.codecogs.com/eqnedit.php?latex=a#0), [](https://www.codecogs.com/eqnedit.php?latex=b#0), [](https://www.codecogs.com/eqnedit.php?latex=c#0), [](https://www.codecogs.com/eqnedit.php?latex=d#0) dla [](https://www.codecogs.com/eqnedit.php?latex=n#0) funkcji nalezy ulozyc [](https://www.codecogs.com/eqnedit.php?latex=4n#0) rownan. Ostatecznie funkcja [](https://www.codecogs.com/eqnedit.php?latex=g(x)#0) wyglada nastepujaco:

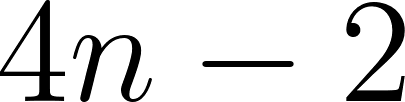
[](https://www.codecogs.com/eqnedit.php?latex=g(x)%20%3D%20%5Cbegin%7Bcases%7D%20g_0(x)%20%2C%20x_0%20%3C%20x%20%3C%20x_1%20%5C%5C%5C%5C%20g_1(x)%20%2C%20x_1%20%3C%20x%20%3C%20x_2%20%5C%5C%5C%5C%20%5Cldots%5C%5C%5C%5C%20g_%7Bn-1%7D(x)%20%2C%20x_%7Bn-1%7D%20%3C%20x%20%3C%20x_n%20%5C%5C%5C%5C%20%5Cend%7Bcases%7D#0)

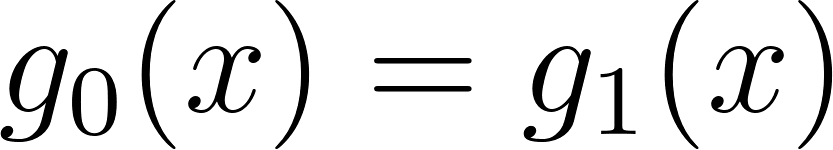
Warunkami jakie chcemy zeby funkcje [](https://www.codecogs.com/eqnedit.php?latex=g_i#0) spelnialy sa:

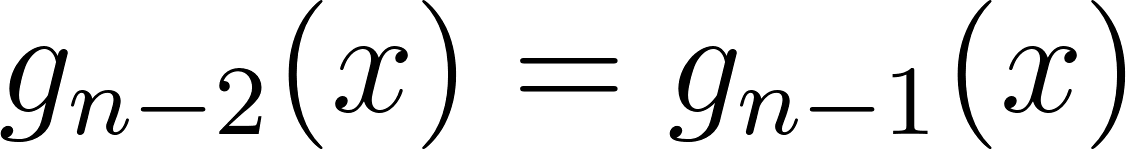
* funkcje powinny przechodzic przez punkty ([](https://www.codecogs.com/eqnedit.php?latex=x_i#0))
  + [](https://www.codecogs.com/eqnedit.php?latex=g_i(x_i)%20%3D%20y_i#0) → [](https://www.codecogs.com/eqnedit.php?latex=n#0) rownan
  + [](https://www.codecogs.com/eqnedit.php?latex=g_i(x_%7Bi%2B1%7D)%20%3D%20y_%7Bi%2B1%7D#0) → [](https://www.codecogs.com/eqnedit.php?latex=n#0) rownan
* funkcje niepowinny nagle zmieniac nachylenia calej funkcji g przechodzac przez punkt [](https://www.codecogs.com/eqnedit.php?latex=x_i#0), czyli pochodzna [](https://www.codecogs.com/eqnedit.php?latex=g_i#0) powinna byc ciagla w punkcie [](https://www.codecogs.com/eqnedit.php?latex=x_i#0)
  + → [](https://www.codecogs.com/eqnedit.php?latex=n-1#0) rownan

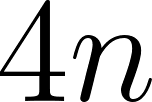
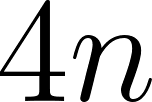
Zeby jeszcze bardziej wygladzic funkcje g, mozemy dac warunek na ciaglosc drugiej pochodnej:

→ [](https://www.codecogs.com/eqnedit.php?latex=n-1#0) rownan

[Z tych wszystkich warunkow wyszlo](https://www.codecogs.com/eqnedit.php?latex=g_i''(x_%7Bi%2B1%7D)%20%3D%20g_%7Bi%2B1%7D''(x_%7Bi%2B1%7D)#0) [](https://www.codecogs.com/eqnedit.php?latex=4n%20-%202#0) rownan. Te 2 pozostale rownania mozna uzyskac z warunku not-a-knot, czyli dwie sasiednie funkcje [](https://www.codecogs.com/eqnedit.php?latex=g_i#0) (tutaj pierwsze i ostatnie) maja te same wspolczynniki:

[](https://www.codecogs.com/eqnedit.php?latex=g_0(x)%20%3D%20g_1(x)#0)

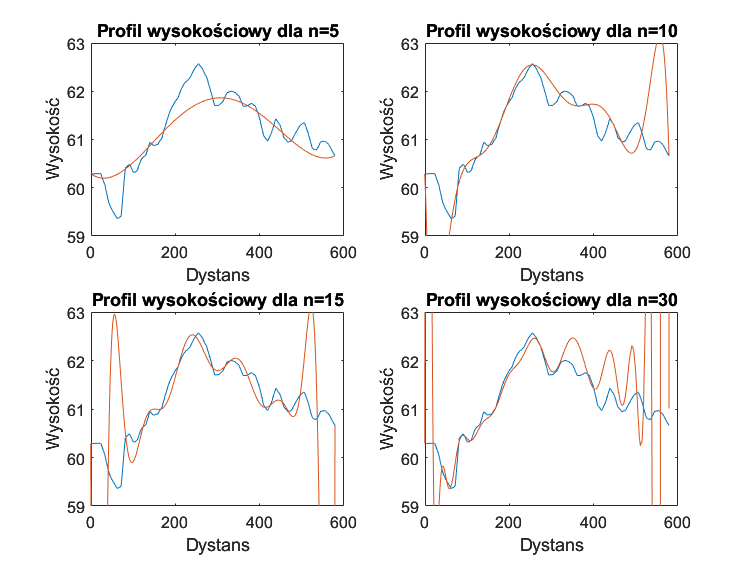
[](https://www.codecogs.com/eqnedit.php?latex=g_%7Bn-2%7D(x)%20%3D%20g_%7Bn-1%7D(x)#0)

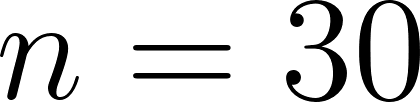
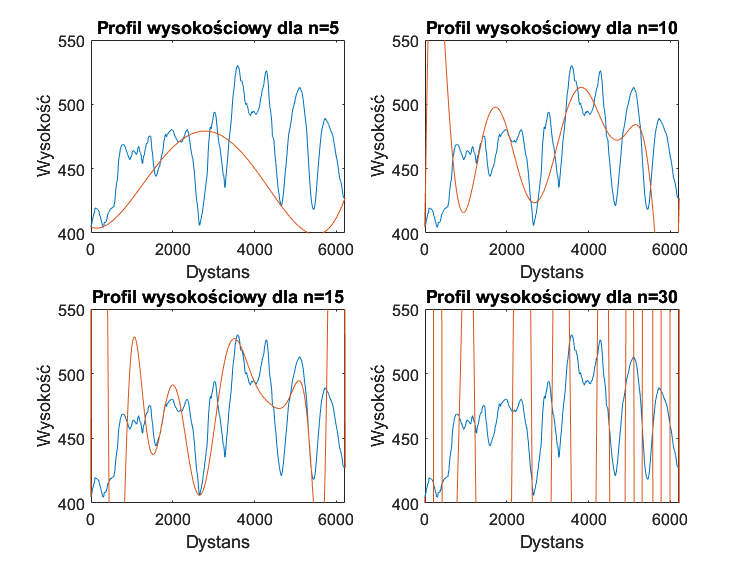
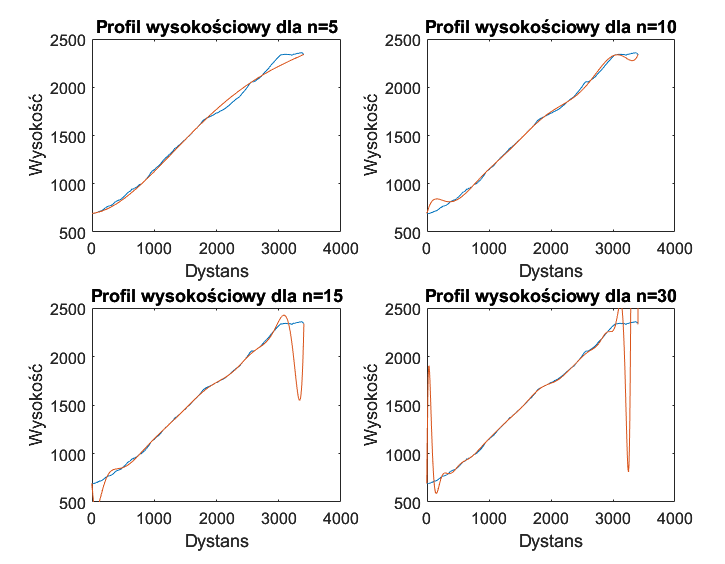
Majac [](https://www.codecogs.com/eqnedit.php?latex=4n#0) rownan mozemy znalezc [](https://www.codecogs.com/eqnedit.php?latex=4n#0) wspolczynnikow i podstawic do funkcji [](https://www.codecogs.com/eqnedit.php?latex=g_i#0).

## Zachowanie metod względem zmiany liczby węzłów

### Metoda wielomianowa

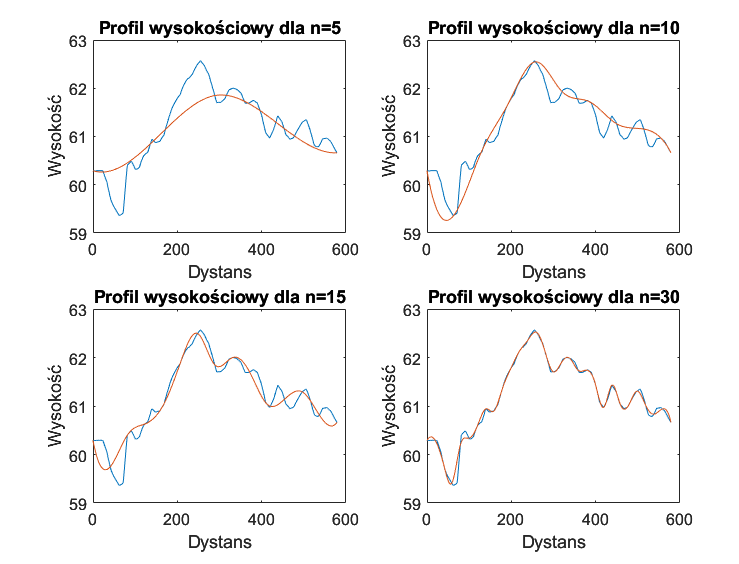
W metodzie wielomianowej przyblizenie wraz z zwiekszeniem liczby wezlow poczatkowo staje sie lepsze, ale przy wiekszych wartosciach [](https://www.codecogs.com/eqnedit.php?latex=n#0), pojawia sie efekt Rungego, czyli pogorszenie interpolacji na krawedziach przedzialu.



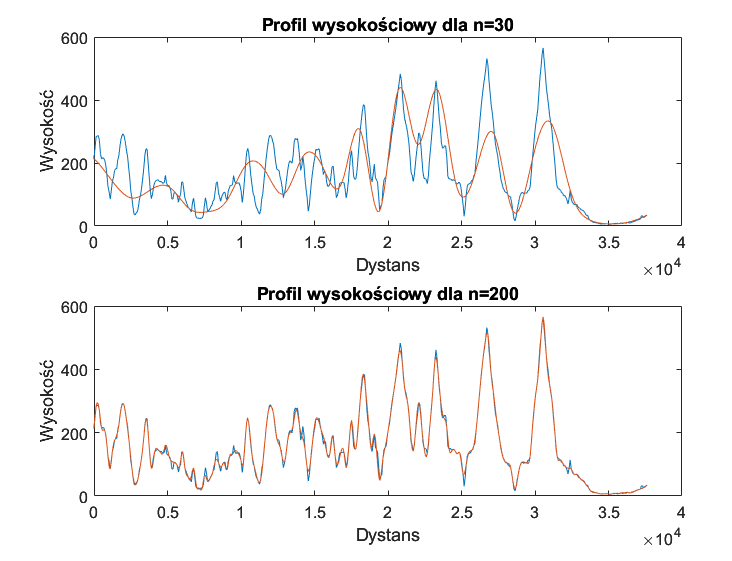
Jesli profil wysokosciowy jest bardzo zroznicowany, to metoda langrag’a zachowuje sie jeszcze gorzej. Widac to na wykresie z [](https://www.codecogs.com/eqnedit.php?latex=n%20%3D%2030#0).Natomiast gdy jest male zroznicowanie, to dla wiekszych wartosci [](https://www.codecogs.com/eqnedit.php?latex=n#0) zachowuje sie dobrze, ale wciaz na krawedziach sa nagle wzrosty i spadki wartosci.

### Interpolacja funkcjami sklejanymi

W metodzie funkcji sklejanych przybliżenie jest o wiele lepsze:

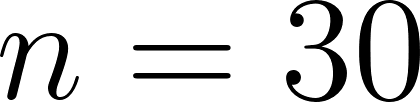


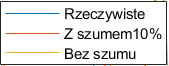
Ta metoda radzi sobie nawet, gdy wysokość zmienia się nagle i często.

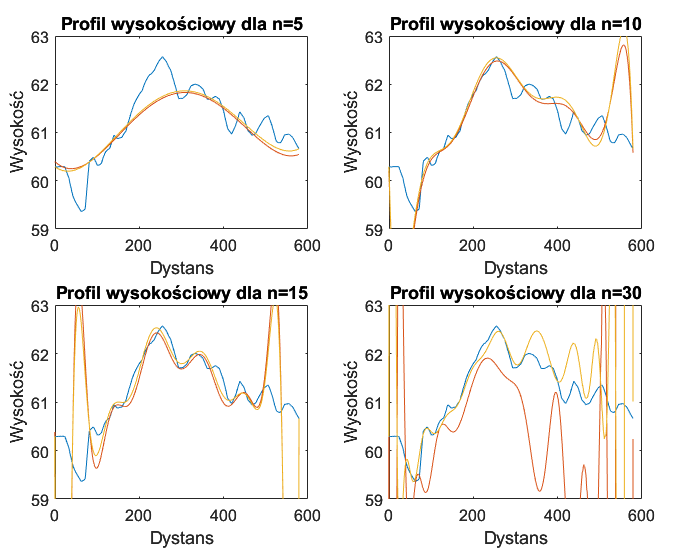


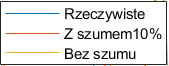
## 

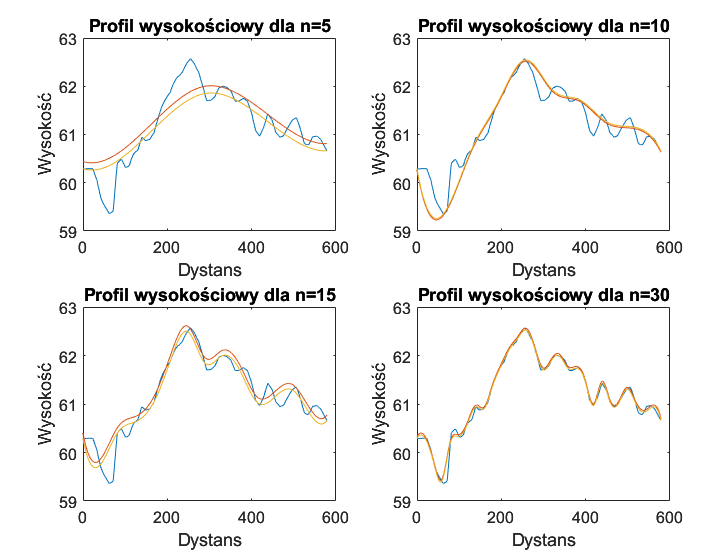
## Interpolacja a szum

Tutaj, zastosowalem szum 10% roznicy najmniejszej i najwiekszej wysokosci z danych rzeczywistych. Na wykresie widoczne jest przyblizenie metoda Lagrang’a. Przy malych [](https://www.codecogs.com/eqnedit.php?latex=n#0) przyblizenie z szumem nieznacznie roznia sie od siebie. Jednak dla wiekszych wartosci takich jak [](https://www.codecogs.com/eqnedit.php?latex=n%3D30#0) roznica jest duzo wieksza.





W przypadku metody funkcji sklejanych jest na odwrot - wraz ze zwiekszeniem sie liczby wezlow roznica miedzy przyblizeniem, a przyblizeniem z szumem 10% maleje:



# Podsumowanie

Z tych dwoch metod interpolacyjnych najlepiej przybliza interpolacja funkcjami sklejanymi.