

# Wstęp do metod numerycznych

## 7. Interpolacja

P. F. Góra

<http://th-www.if.uj.edu.pl/zfs/gora/>

2019/20

## Interpolacja

**Problem:** Dana jest funkcja **w postaci stabelaryzowanej** — określona w węzłach pewnej siatki (jedno-, dwu- lub trójwymiarowej) — a trzeba znaleźć jej wartości *pomiędzy* węzłami. Niekiedy też należy, jedynie na podstawie stabelaryzowanych wartości funkcji, określić pochodną tej funkcji w węzłach lub pomiędzy nimi (na przykład w grafice 3D trzeba znaleźć wektor normalny w środku plakietki rozpiętej na trzech lub czterech znanych punktach). Takie zadania pojawiają się bardzo często w zagadnieniach takich, jak

- obliczenia inżynierskie i numeryczne,
- statystyczna analiza danych,
- przetwarzanie i renderowanie grafiki,
- fotografia cyfrowa.

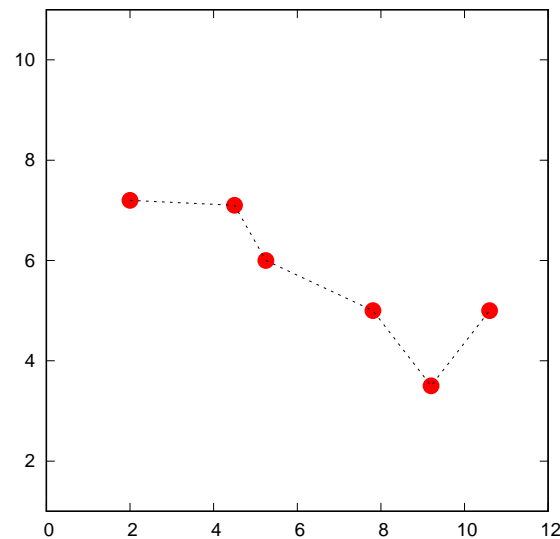
## Przykład — ruch ramienia robota

Naszym zadaniem\* jest oprogramowanie robota przemysłowego, który ma przemieszczać się pomiędzy sześcioma punktami o zadanych z góry i niezmiennych współrzędnych. Czas przebywania robota nad punktem jest mały w porównaniu z czasem potrzebnym na przemieszczanie się pomiędzy punktami.

Jakie rozwiązanie wybrać?

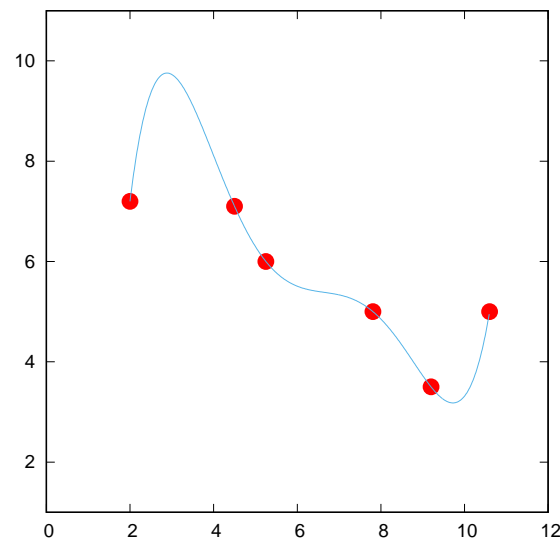
\*Przykład pochodzi z tego podręcznika.

Najprostsze rozwiązanie: Połączmy punkty odcinkami linii prostej, czyli skonstruujemy łamaną:



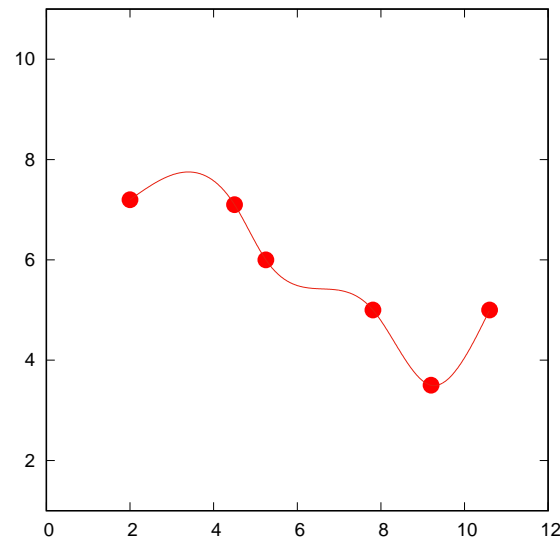
Wadą tego rozwiązania jest to, że prędkość zmienia się bardzo gwałtownie (skokowo), gdy ramię przechodzi przez kolejne punkty, co może doprowadzić do szybkiego zużycia i uszkodzenia mechanicznego.

Inne rozwiązanie: Przeprowadźmy przez podane sześć punktów wielomian piątego stopnia (wielomian taki jest określony jednoznacznie):



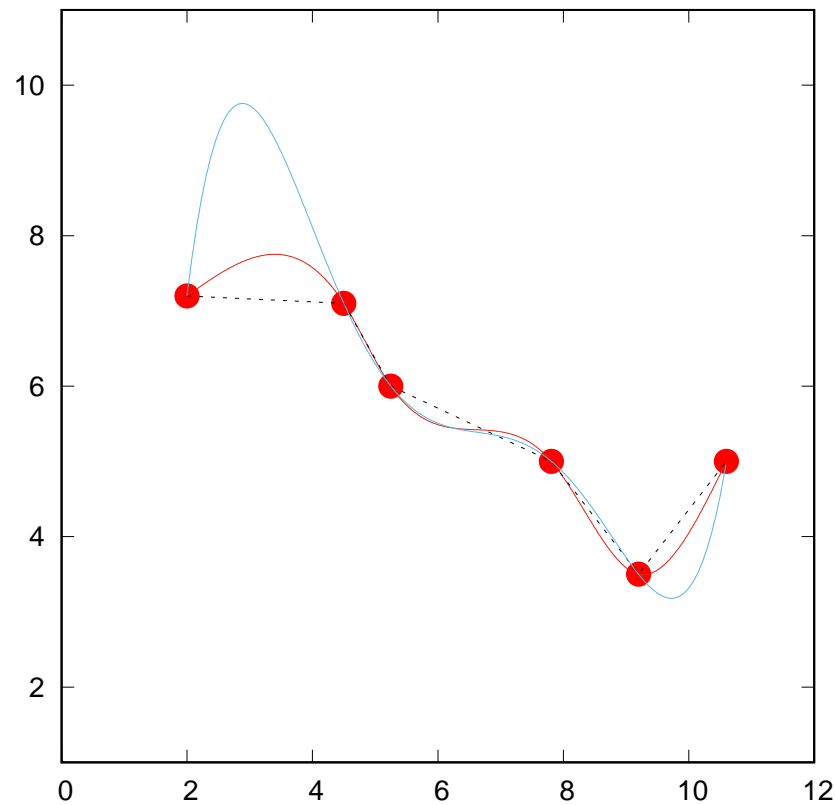
Nie ma już skokowych zmian prędkości (pochodna trajektorii ramienia robota jest ciągła), ale droga wydaje się być zbyt długa, zwłaszcza w okolicach skrajnych punktów.

Jeszcze inne rozwiązanie: Przeprowadźmy przez podane punkty naturalny splajn kubiczny:



To rozwiązanie jest najlepszym z trzech tutaj zaprezentowanych: Prędkość nie zmienia się skokowo, a droga nie jest byt długa.

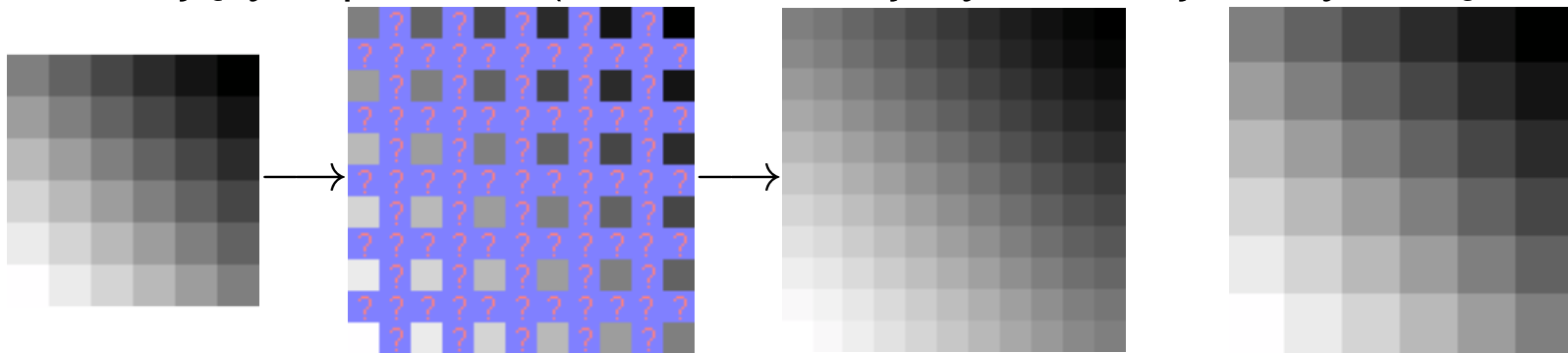
Wszystkie trzy zaprezentowane rozwiązania — linia łamana, wielomian, naturalny splajn kubiczny — są różnymi rodzajami interpolacji.



## Przykład — powiększenie cyfrowe

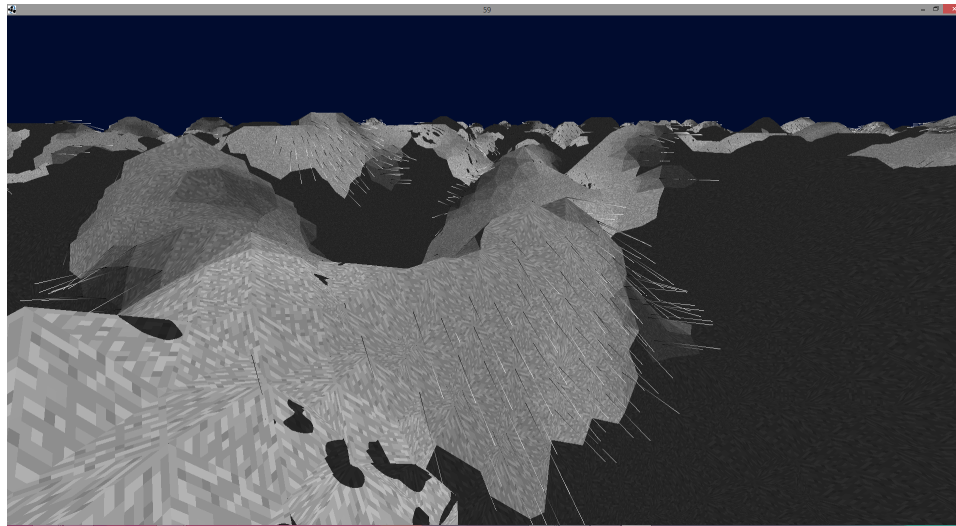
Fotografia cyfrowa określona jest poprzez pixele: najmniejsze punkty rozróżnialne na zdjęciu. Każdemu pixelowi przypisana jest pewna liczba (niekiedy zbiór liczb), określająca kolor i jasność danego punktu.

Jeśli dokonujemy powiększenia cyfrowego (ang. *digital zoom*), “rozsuwamy” istniejące pixele. Pojawiają się nowe pixele, puste, które trzeba czymś wypełnić. W tym celu należy w pewien sposób uwzględnić wartości w istniejących pixelach (niekoniecznie jedynie w najbliższych sąsiadach).





## Przykład



Przy renderowaniu grafiki 3D, aby uzyskać zadowalający efekt wizualny, trzeba użyć bardziej zaawansowanych metod interpolacji, wymagających — na przykład — obliczania gradientu (nachylenia) renderowanej powierzchni.

## Ekstrapolacja

Zagadnieniem pokrewnym do interpolacji jest *ekstrapolacja*: Jak, na podstawie znanych, stabilizowanych wartości funkcji, przewidzieć jej *przyszłe* wartości, to znaczy wartości w punktach leżących poza przedziałem zawierającym wszystkie węzły.

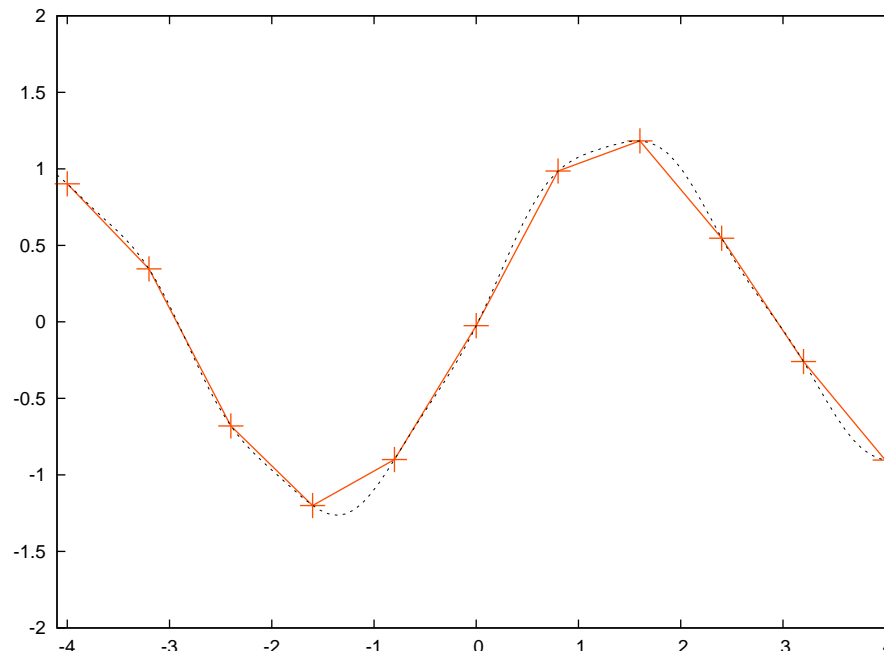
## Inerpolacja jedowymiarowa

Dana jest funkcja w postaci tabelaryzowanej

$x_i$	$x_1$	$x_2$	$x_3$	$\dots$	$x_n$
$f_i = f(x_i)$	$f_1$	$f_2$	$f_3$	$\dots$	$f_n$

Punkty  $x_i$  nazywamy **węzłami interpolacji**. Problem: chcemy znaleźć *łatwy* sposób na wyliczanie wartości funkcji pomiędzy węzłami (interpolacja) lub poza obszarem obejmującym węzły (ekstrapolacja).

## Interpolacja odcinkami liniowa



Interpolacja odcinkami liniowa na 11 węzłach. “Prawdziwa” funkcja interpolowana zaznaczona jest linią kropkowaną. Istnieje **nieskończenie wiele** funkcji ciągłych, które są sobie równe w skończonej liczbie węzłów!

Można to zrobić prowadząc łamaną pomiędzy poszczególnymi punktami węzłowymi  $(x_i, f_i)$ , ale jest to nieeleganckie i w pewnych przypadkach może powodować problemy, gdyż funkcja interpolująca ma ostrza w węzłach.

## Interpolacja wielomianowa

Tabela taka, jak na stronie 11, wyznacza jednoznacznie wielomian stopnia  $n-1$ . Istotnie, rozpatrzmy wielomian

$$a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0. \quad (1)$$

Jeśli do wielomianu (1) podstawimy za  $x$  kolejno  $x_1, x_2, \dots, x_n$ , przyjmując, że wartość wielomianu w tych punktach wynosi odpowiednio  $f_1, f_2, \dots, f_n$ , otrzymamy

$$\begin{array}{cccccccc} a_{n-1}x_1^{n-1} & + & a_{n-2}x_1^{n-2} & + & \dots & + & a_1x_1 & + & a_0 & = & f_1 \\ a_{n-1}x_2^{n-1} & + & a_{n-2}x_2^{n-2} & + & \dots & + & a_1x_2 & + & a_0 & = & f_2 \\ \dots & & \dots & & \dots & & \dots & & \dots & & \dots \\ a_{n-1}x_n^{n-1} & + & a_{n-2}x_n^{n-2} & + & \dots & + & a_1x_n & + & a_0 & = & f_n \end{array} \quad (2)$$

Układ (2) zapisany w postaci macierzowej ma postać

$$\begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \cdots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \cdots & x_2 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n^{n-1} & x_n^{n-2} & \cdots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad (3)$$

Rozwiązaniem układu równań (3) są *współczynniki* wielomianu (1). Macierz układu (3) nosi nazwę *macierzy Vandermonde'a*.

Można pokazać<sup>†</sup>, że wyznacznik macierzy Vandermonde'a ma postać

$$\prod_{1 \leq i < j \leq n} (x_i - x_j), \quad (4)$$

<sup>†</sup>Dziękuję internetowemu komentatorowi `juventusik_plus` za wskazanie błędu w poprzedniej wersji.

a zatem wyznacznik macierzy Vandermonde'a jest różny od zera, jeżeli żadne punkty  $x_1, x_2, \dots, x_n$  nie pokrywają się. Tym samym problem znalezienia współczynników wielomianu (1) ma jednoznaczne rozwiązanie.

Gdybyśmy chcieli rozwiązywać układ równań (3) za pomocą już poznanych metod, mogłoby się wydawać, że koszt interpolacji wielomianowej wynosi  $O(n^3)$ . W rzeczywistości układ ten można, korzystając z symetrii macierzy Vandermonde'a, rozwiązać w czasie  $O(n^2)$ . Przekonany się o tym konstruując interpolację wielomianową **w całkiem inny sposób**: Skonstruujemy wielomian interpolacyjny, który w arytmetyce dokładnej dawałby **dokładnie** takie same wyniki, co interpolacja za pomocą wielomianu o współczynnikach wyliczanych z układu (3). Zobaczymy jednak, że konstrukcja takiego wielomianu jest prostsza i ma mniejszą złożoność, niż  $O(n^3)$ .

## Wzór interpolacyjny Lagrange'a

Zamiast szukać rozwiązania równania (3), postulujemy, że poszukiwany wzór interpolacyjny ma postać

$$f(x) = \sum_{j=1}^n l_j(x) f_j + E(x), \quad (5a)$$

gdzie

$$l_j(x) = \frac{(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}. \quad (5b)$$

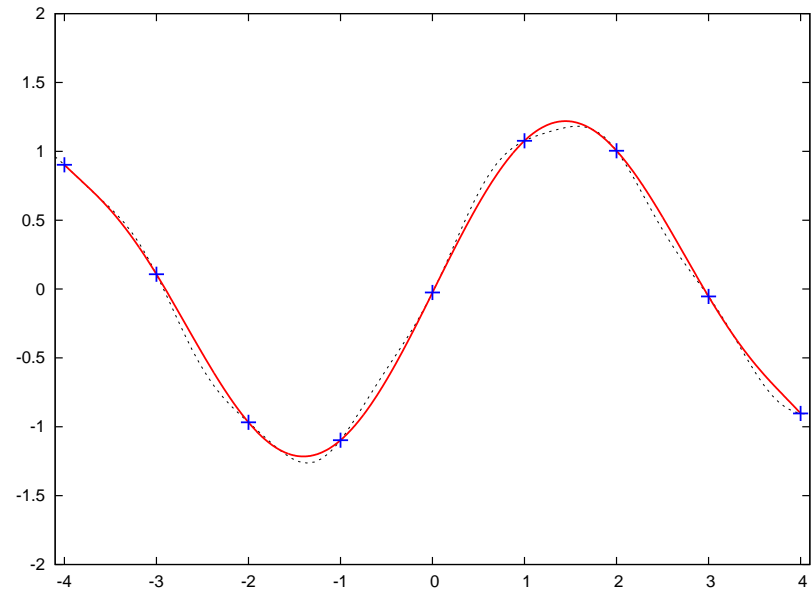
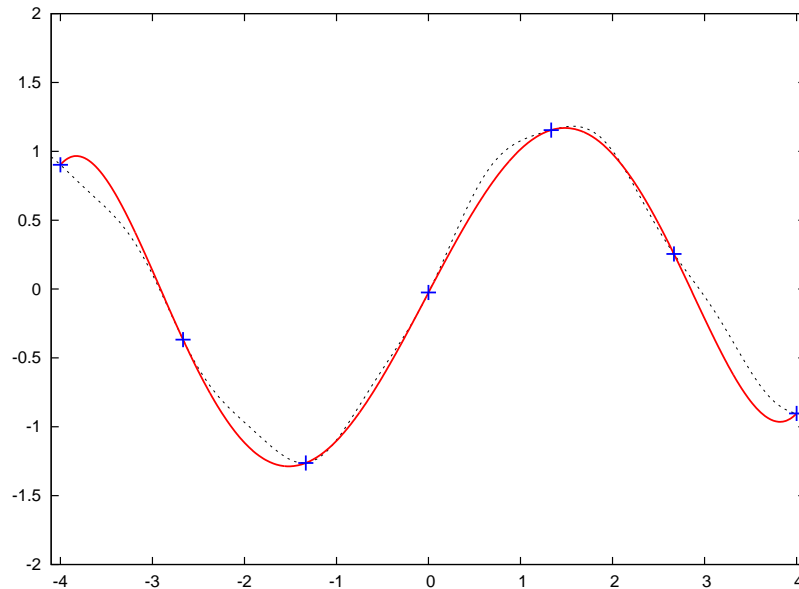
$E(x)$  w (5a) jest nazywane *resztą* lub *błędem interpolacji*. Zauważmy, że  $l_j(x)$  jest wielomianem stopnia  $n-1$  oraz

$$l_j(x_k) = \delta_{jk}. \quad (6)$$

Jeżeli  $f(x)$  jest wielomianem stopnia co najwyżej  $n-1$ ,  $E(x)$  znika tożsamościowo. Mówimy, że interpolacja (5a) ma dokładność  $n-1$ .

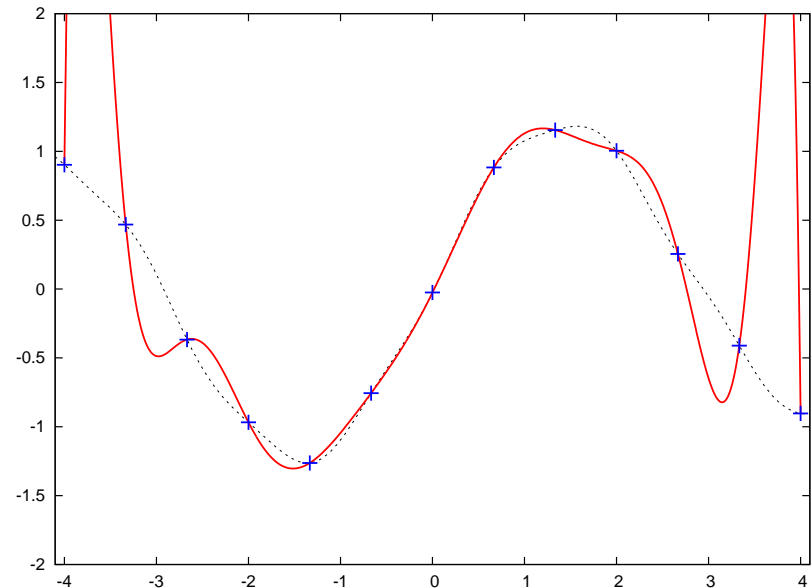
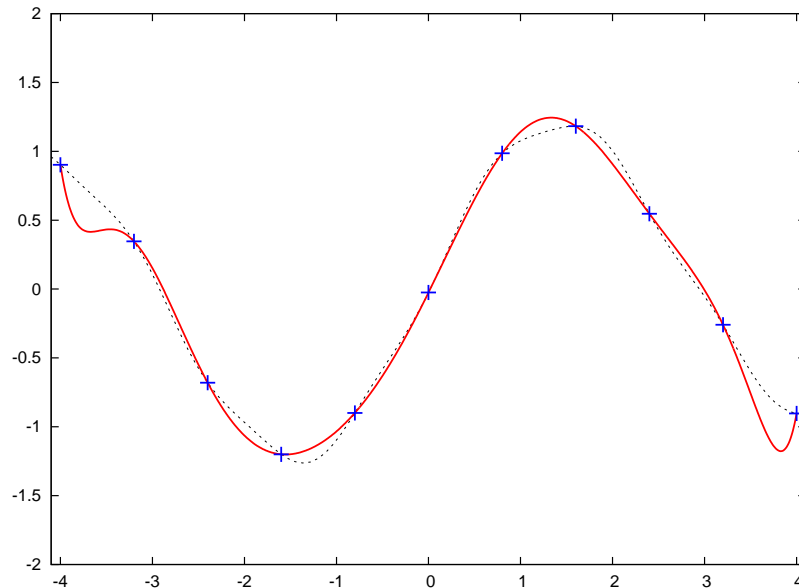


## Przykłady



Interpolacja pewnej funkcji (oznaczonej linią kropkowaną) oparta na 7 (lewy panel) i 9 (prawy panel) węzłach. W tym wypadku zwiększanie liczby węzłów poprawia jakość interpolacji.

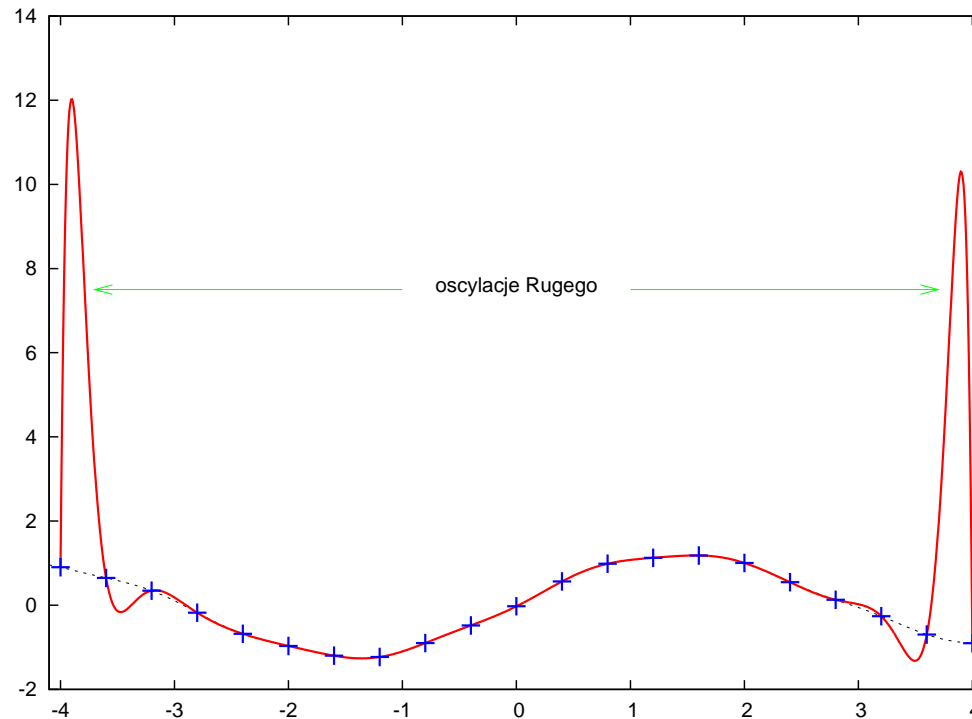
## Przykłady



Interpolacja tej samej funkcji, co poprzednio, oparta na 11 i 13 węzłach.

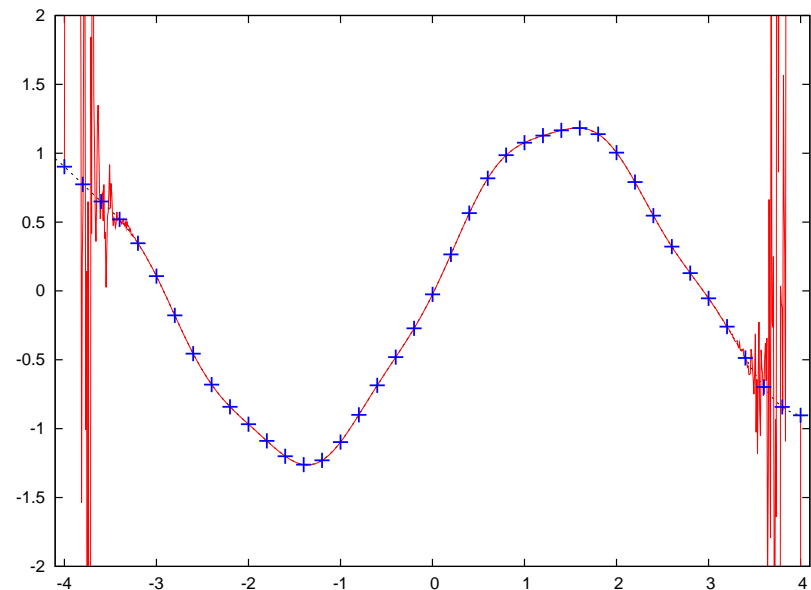
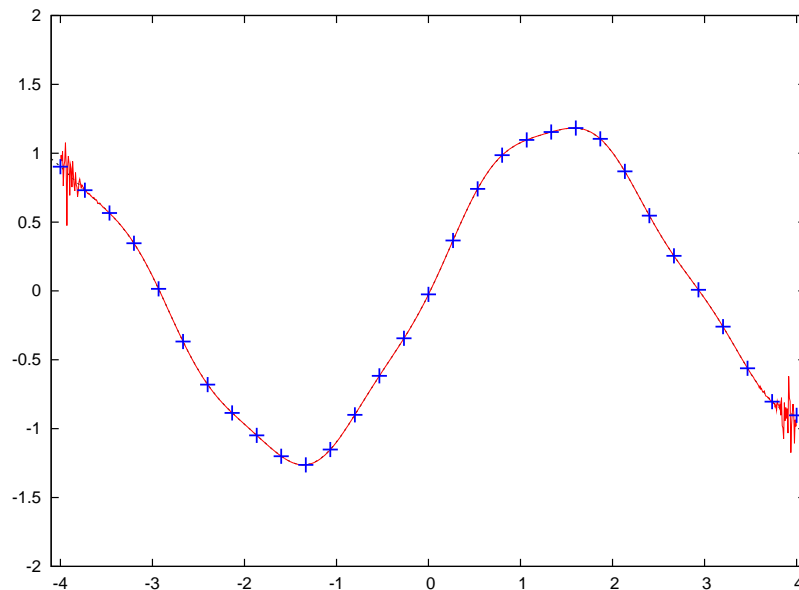
Dalsze zwiększanie liczby węzłów prowadzi do *pogorszenia* jakości interpolacji, zwłaszcza w pobliżu krańców przedziału zawierającego węzły.

## Oscylacje Rungego



Wielomiany wysokiego stopnia są “sztywne”. Jeżeli narzucić im warunek, że mają przechodzić przez ustalone z góry punkty, mogą to kompensować znacznymi wahaniami pomiędzy węzłami. Zjawisko to nazywa się *oscylacjami Rungego* i oznacza, że [interpolowanie za pomocą wielomianów wysokiego stopnia bywa niewskazane](#).

## Dalsze przykłady



Może się zdarzyć, że dalsze powiększanie liczby węzłów poprawi jakość interpolacji (interpolacja oparta na 31 węzłach, lewy panel), jednak w końcu zbyt duża liczba węzłów doprowadza do katastrofy (interpolacja oparta na 41 węzłach, prawy panel — oscylacje Rungego osiągają amplitudę  $\sim 200$ ).

## Błąd interpolacji

Wielomian (5b) wygodnie jest niekiedy zapisywać w postaci

$$l_j(x) = \frac{p_n(x)}{(x - x_j)p'_n(x_j)} \quad (7a)$$

gdzie

$$p_n(x) = \prod_{i=1}^n (x - x_i), \quad p'_n(x_j) = \left. \frac{dp_n}{dx} \right|_{x=x_j} \quad (7b)$$

Oznaczmy  $y(x) = \sum_{j=1}^n l_j(x) f_j$  (jest to wielomianowa część wzoru interpolacyjnego Lagrange'a (5a), bez reszty).

Rozpatrzmy funkcję

$$F(z) = f(z) - y(z) - [f(x) - y(x)] \frac{p_n(z)}{p_n(x)}. \quad (8)$$

$F(z)$ , jako funkcja zmiennej  $z$ , ma  $n+1$  miejsc zerowych:  $x_1, x_2, \dots, x_n$  oraz  $x$ . Zakładając, że funkcja  $f(\cdot)$  jest dostatecznie gładka, stosujemy do funkcji  $F(z)$   $n$ -krotnie twierdzenie Rolle'a i stwierdzamy, że pochodna

$$F^{(n)}(z) = f^{(n)}(z) - y^{(n)}(z) - [f(x) - y(x)] \frac{n!}{p_n(x)} \quad (9)$$

ma co najmniej jedno miejsce zerowe w najmniejszym przedziale domkniętym zawierającym punkty  $x_1, x_2, \dots, x_n$  oraz  $x$ . Oznaczmy to miejsce zerowe przez  $\xi$ . Zauważmy, że  $y^{(n)}(z) = 0$ , gdyż  $y(z)$  jest wielomianem stopnia  $n-1$ . Ostatecznie otrzymujemy

$$0 = F^{(n)}(\xi) = f^{(n)}(\xi) - \underbrace{[f(x) - y(x)]}_{E(x)} \frac{n!}{p_n(x)} \quad (10)$$

czyli

$$E(x) = \frac{p_n(x)}{n!} f^{(n)}(\xi). \quad (11)$$

$\xi$  jest pewnym punktem wewnętrznym przedziału zawierającego węzły i  $x$  (to ostatnie jest ważne w wypadku ekstrapolacji). Nie wiemy, którym punktem, zatem dla bezpieczeństwa należałoby brać największą (co do modułu) wartość  $f^{(n)}(\xi)$ . Trudność w szacowaniu błędu interpolacji polega na **trudności w szacowaniu wysokich pochodnych interpolowanych funkcji**. Praktyka pokazuje, że wysokie pochodne nawet “porządnych” funkcji (niewielomianowych) mogą przybierać znaczne wartości.

Przykład: Maksimum dziesiątej pochodnej funkcji  $e^{-x^2}$  jest rzędu 15 000.

## Złożność obliczeniowa interpolacji Lagrange'a

Jak wynika ze wzorów (5), złożoność obliczeniowa interpolacji Lagrange'a *na jeden punkt*, w którym wyliczamy interpolowaną wartość, wynosi  $O(n^2)$ : obliczenie każdego  $l_j(x)$  wymaga  $O(n)$  operacji, a trzeba wyliczyć wartość  $n$  wielomianów  $l_j$ . Gdybyśmy mieli wyliczyć wartość funkcji interpolowanej w  $m > n$  punktach, otrzymalibyśmy złożoność większą, niż  $O(n^3)$ . Z drugiej strony, gdybyśmy znali *współczynniki* wielomianu interpolacyjnego, koszt wyinterpolowania wartości w jednym punkcie byłby rzędu  $O(n)$ , gdyż taki jest koszt obliczania wartości wielomianu. Zapowiedzieliśmy na stronie 15, że problem znalezienia współczynników wielomianu interpolacyjnego można rozwiązać w  $O(n^2)$  krokach, bez konieczności jawnego rozwiązywania układu (3). Wobec tego koszt wyinterpolowania wartości w  $m$  punktach wynosi  $O(\max(n^2, m \cdot n))$ .



## Współczynniki wielomianu interpolacyjnego

Przypuśćmy, że mamy funkcję w postaci stabelaryzowanej, jak na stronie 11. Dodatkowo założmy, że żaden z węzłów interpolacji nie jest zerem. Wiemy, że wielomian interpolacyjny ma postać

$$y(x) = \sum_{j=1}^n l_j(x) f_j. \quad (12a)$$

Jednocześnie *wiemy*, że

$$y(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_2x^2 + a_1x + a_0. \quad (12b)$$

Natychmiast widzimy, że współczynnik  $a_0$  znajdziemy obliczając wartość wielomianu (12a) w zerze:

$$a_0 = y(0) = \sum_{j=1}^n l_j(0) f_j. \quad (13)$$

Do obliczenia  $a_0$  musimy wykonać  $O(n^2)$  operacji: dla każdego z  $n$  węzłów  $O(n)$  w celu obliczenia  $l_j(0)$ ,  $j = 1, \dots, n$ . Zapamiętujemy obliczone wartości  $l_j(0)$ .

Zauważmy, że

$$\frac{y(x) - a_0}{x} = a_{n-1}x^{n-2} + a_{n-2}x^{n-3} + \dots + a_2x + a_1. \quad (14)$$

Niestety, w celu znalezienia wartości  $a_1$  nie można *obliczyć* wartości  $(y(x) - a_0)/x$  w zerze, gdyż dostalibyśmy symbol nieoznaczony  $0/0$ . Możemy jednak skorzystać... z interpolacji. Przede wszystkim zauważmy, że w węzłach interpolacji

$$\frac{y(x_i) - a_0}{x_i} = \frac{1}{x_i} \left( \sum_{j=1}^n l_j(x_i) f_j - a_0 \right) = \frac{1}{x_i} \left( \sum_{j=1}^n \delta_{ij} f_j - a_0 \right) = \frac{f_i - a_0}{x_i}. \quad (15)$$

Tworzymy wobec tego tabelę:

$x_i$	$x_1$	$x_2$	$x_3$	$\dots$	$x_n$
$f_i^{(1)}$	$\frac{f_1 - a_0}{x_1}$	$\frac{f_2 - a_0}{x_2}$	$\frac{f_3 - a_0}{x_3}$	$\dots$	$\frac{f_n - a_0}{x_n}$

a na jej podstawie wielomian interpolacyjny

$$y^{(1)}(x) = \sum_{j=1}^n l_j(x) f_j^{(1)}. \quad (16a)$$

**Wiemy** (patrz wyżej (14)), że

$$y^{(1)}(x) = a_{n-1}x^{n-2} + a_{n-2}x^{n-3} + \dots + a_2x + a_1, \quad (16b)$$

a zatem możemy obliczyć

$$a_1 = y^{(1)}(0) = \sum_{j=1}^n l_j(0) f_j^{(1)}. \quad (17)$$

Podobnie

$$\frac{y^{(1)}(x) - a_1}{x} = a_{n-1}x^{n-3} + a_{n-2}x^{n-4} + \dots + a_2, \quad (18)$$

a zatem w celu obliczenia  $a_2$  tworzymy odpowiednią tabelę i wyinterpolujemy wartość wyrażenia (18) w zerze. I tak dalej.

W ogólności,

$$a_k = \sum_{j=1}^n l_j(0) f_j^{(k)}, \quad k = 0, \dots, n-1, \quad (19a)$$

gdzie

$$f_j^{(k)} = \begin{cases} \frac{f_j^{(k-1)} - a_{k-1}}{x_j} & k = 1, \dots, n-1, \\ f_j & k = 0. \end{cases} \quad (19b)$$

Jaka jest zatem złożoność obliczeniowa znalezienia współczynników wielomianu interpolacyjnego? Zauważmy, że węzły interpolacji nie zmieniają się w trakcie wykonywania obliczeń według schematu (19), a wartości  $l_j(0)$  zależą tylko od węzłów. Zatem obliczając  $a_0$ , wykonujemy  $O(n^2)$  operacji, obliczając przy tym  $l_j(0)$ . Obliczając według (19a)  $a_{k>0}$  wykonujemy dla każdego  $k$  tylko  $O(n)$  operacji, gdyż wartości  $l_j(0)$  są już w tym momencie znane. Ostatecznie całkowity koszt wyliczenia współczynników wielomianu interpolacyjnego wynosi  $O(n^2)$ .

Założyliśmy, że żaden z węzłów interpolacji nie jest równy zero. Jeżeli założenie to nie jest spełnione, tworzymy wielomian interpolacyjny (12a), za jego pomocą wyinterpolowujemy wartość w jakimś punkcie  $\tilde{x} \neq 0$ , usuwamy węzeł interpolacji równy zero i odpowiadającą mu wartość funkcji, zastępując go przez  $\tilde{x}$  i wyliczoną wartość  $\tilde{f} = y(\tilde{x})$ . W arytmetyce dokładnej wielomiany interpolacyjne wyliczone na podstawie pierwotnej i tak zmodyfikowanej tabeli są identyczne. Dalej postępujemy jak powyżej.

## Interpolacja Hermite'a

Jeżeli znamy nie tylko wartości funkcji interpolowanej w węzłach, ale także *wartości pochodnej*,

$x_i$	$x_1$	$x_2$	$x_3$	$\dots$	$x_n$
$f_i = f(x_i)$	$f_1$	$f_2$	$f_3$	$\dots$	$f_n$
$f'_i = f'(x_i)$	$f'_1$	$f'_2$	$f'_3$	$\dots$	$f'_n$

narzuca to  $2n$  warunków na wielomian interpolacyjny<sup>‡</sup>. Można skonstruować wówczas interpolację wielomianową rzędu  $2n - 1$ , postaci

<sup>‡</sup>Jest to zagadnienie o stosunkowo niewielkich zastosowaniach praktycznych, ale za to o dużym znaczeniu teoretycznym.

$$y(x) = \sum_{i=1}^n h_i(x) f_i + \sum_{i=1}^n \bar{h}_i(x) f'_i + E(x), \quad (20a)$$

gdzie

$$h_i(x) = \left(1 - 2(x - x_i)l'_i(x_i)\right) l_i^2(x), \quad (20b)$$

$$\bar{h}_i(x) = (x - x_i)l_i^2(x). \quad (20c)$$

$l_j(x)$  oznacza to samo, co w interpolacji Lagrange'a, natomiast

$$E(x) = \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\xi), \quad (20d)$$

gdzie  $\xi$  jest pewnym punktem wewnętrznym przedziału rozpiętego na węzłach i wartości  $x$ . Wielomian interpolacyjny (20a) zgadza się z interpolowaną funkcją oraz jej pochodną w węzłach.

## Interpolacja za pomocą funkcji sklepanych

Interpolacja wielomianowa jest koncepcyjnie najprostszym sposobem interpolacji, prowadzić jednak może, jak to pokazaliśmy, do niepożądanych zachowań. Najbardziej popularnym sposobem uniknięcia oscylacji Rungego, związanych z trudnością szacowania błędu interpolacji wielomianowej, jest interpolacja za pomocą [funkcji sklepanych](#).

**Funkcją sklepaną rzędu  $k$** , czyli “splajnem” (ang. *spline*), nazywam funkcję, która

1. lokalnie jest wielomianem rzędu  $k$ ,
2. jest  $(k-1)$ -krotnie różniczkowalna w węzłach (z czego wynika, że jej pochodne rzędu  $k-2$  i niższych są ciągłe).

Najczęściej używa się funkcji sklepanych rzędu 3, czyli “splajnów kubicznych” (ang. *cubic splines*).



## Cubic splines

Założmy, że oprócz wartości funkcji w węzłach, znamy także pewne inne wartości  $\xi_i$  w węzłach. Mamy więc tabelkę postaci:

$x_i$	$x_1$	$x_2$	$x_3$	$\dots$	$x_n$
$f_i = f(x_i)$	$f_1$	$f_2$	$f_3$	$\dots$	$f_n$
$\xi_i$	$\xi_1$	$\xi_2$	$\xi_3$	$\dots$	$\xi_n$

Chcemy skonstruować takie wyrażenie interpolacyjne, które

- będzie zgadzać się z wartościami  $f_i$  w węzłach,
- wielkości  $\xi_i$  będą drugimi pochodnymi [wyrażenia interpolacyjnego](#) (nie funkcji interpolowanej!) w węzłach.

W każdym przedziale  $[x_j, x_{j+1}]$ ,  $j = 1, 2, \dots, n - 1$ , konstruujemy wielomian trzeciego stopnia

$$y_j(x) = A f_j + B f_{j+1} + C \xi_j + D \xi_{j+1}, \quad (21a)$$

gdzie

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j}, \quad B = \frac{x - x_j}{x_{j+1} - x_j}, \quad (21b)$$

$$C = \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2, \quad D = \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2. \quad (21c)$$

Łatwo sprawdzić, że  $y_j(x_j) = f_j$ ,  $y_j(x_{j+1}) = f_{j+1}$ , a ponieważ, co można wykazać prostym rachunkiem,

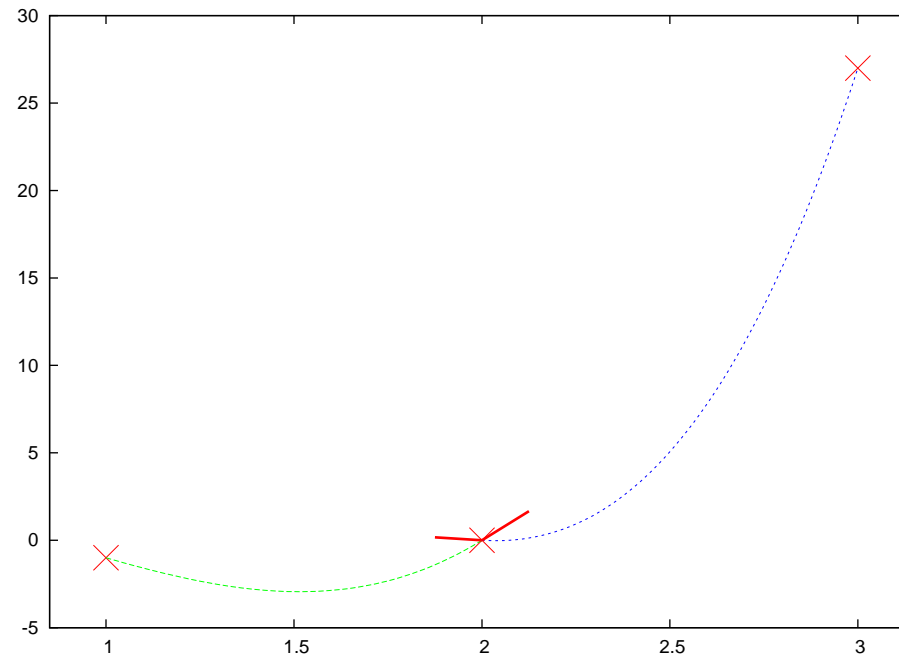
$$\frac{d^2 y_j}{dx^2} = A \xi_j + B \xi_{j+1}, \quad (22)$$

także wartości drugiej pochodnej (21a) zgadzają się z zadanymi wartościami  $\xi_j$  w węzłach.

Jest jednak pewien **problem**: w rzeczywistości **nie znamy wartości**  $\xi_j$ . Nie skorzystaliśmy jednak jeszcze z wymogu ciągłości **pierwszej** pochodnej (21a) w węzłach (aby można było mówić o drugich pochodnych  $y_j(x)$ , pierwsza pochodna tego wyrażenia musi być ciągła). W tym celu żądamy, aby pochodna  $y_j(x)$  obliczana w **prawym** krańcu przedziału równała się pochodnej  $y_{j+1}(x)$  obliczanej w **lewym** krańcu odpowiedniego przedziału. Gdy to zrobimy, otrzymamy równanie

$$\begin{aligned} \frac{x_j - x_{j-1}}{6} \xi_{j-1} + \frac{x_{j+1} - x_{j-1}}{3} \xi_j + \frac{x_{j+1} - x_j}{6} \xi_{j+1} \\ = \frac{f_{j+1} - f_j}{x_{j+1} - x_j} - \frac{f_j - f_{j-1}}{x_j - x_{j-1}}. \end{aligned} \quad (23)$$

Jest to, w istocie, trójdzielny układ równań na nieznane wielkości  $\{\xi_j\}$ .



Zasada obliczania wielkości  $\xi_j$ : W dwu sąsiednich przedziałach odpowiadającym trzem kolejnym punktom danych przeprowadzono wielomiany trzeciego stopnia (21a) z *jakoś* (raczej kiepsko) dobranymi wielkościami  $\xi_j$ . Trzeba je dobrać tak, aby pochodne obu wielomianów, zaznaczone grubymi liniami w środkowym punkcie, zgadzały się ze sobą w tym punkcie (w punkcie zszycia).

## Naturalny splajn kubiczny

Jeżeli mamy  $n$  węzłów interpolacji, mamy  $n-2$  wewnętrznych punktów zszycia, w których możemy żądać ciągłości pochodnej. W takim wypadku (23) stanowi układ  $n-2$  równań z  $n$  niewiadomymi. Trzeba podać jakieś dodatkowe warunki.

Najczęściej przyjmuje się, że  $\xi_1 = \xi_n = 0$ . Jest to wówczas tak zwany **naturalny splajn kubiczny**. Jeżeli sytuacja tego wymaga — lub jeżeli mamy jakieś przesłanki, aby tak zrobić — możemy narzucić inne warunki na drugie pochodne na brzegach lub na kombinacje liniowe drugich pochodnych.

## Równoodległe węzły

Jeżeli węzły interpolacji są równoodległe,  $x_{j+1} - x_j = h$ , równanie (23) przybiera szczególnie prostą postać:

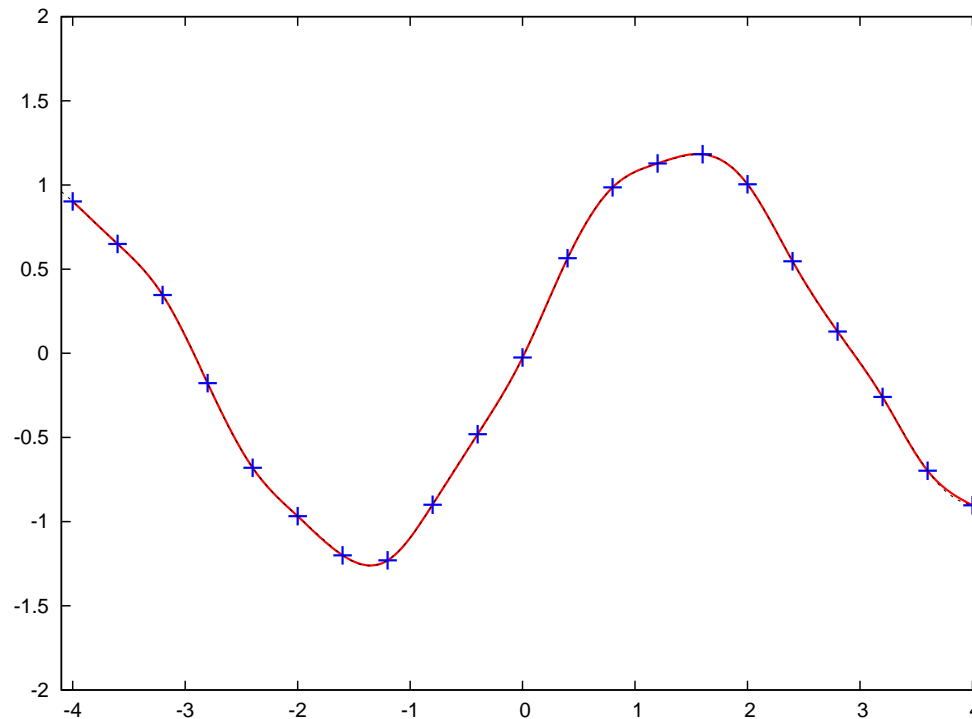
$$\begin{bmatrix} 4 & 1 & & & & & & \\ 1 & 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ & & & & 1 & 4 & 1 & \\ & & & & & 1 & 4 & \end{bmatrix} \begin{bmatrix} \xi_2 \\ \xi_3 \\ \xi_4 \\ \vdots \\ \xi_{n-2} \\ \xi_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} f_1 - 2f_2 + f_3 \\ f_2 - 2f_3 + f_4 \\ f_3 - 2f_4 + f_5 \\ \vdots \\ f_{n-3} - 2f_{n-2} + f_{n-1} \\ f_{n-2} - 2f_{n-1} + f_n \end{bmatrix} \quad (24)$$

Macierz po lewej stronie tego równania posiada łatwą do znalezienia faktoryzację Cholesky'ego. Z dokładnością do czynników "6", prawa strona zawiera drugie ilorazy różnicowe interpolowanej funkcji.

**Praktyczne zastosowanie** naturalnych splajnów kubicznych przebiega dwu-etapowo:

1. Rozwiązujemy układ równań (23) (lub, jeżeli można, (24)) na  $n-2$  nieznanymi wielkościami  $\{\xi_j\}_{j=2}^{n-1}$ . Ponieważ układ ten jest trójdzielny, koszt obliczeniowy wynosi  $O(n)$ . Krok ten wykonujemy tylko raz na początku całej procedury.
2. W celu znalezienia wartości pomiędzy węzłami, wykorzystujemy równanie (21a) tyle razy, ile wartości chcemy znaleźć. W każdym przedziale  $[x_j, x_{j+1}]$  używamy *odpowiedniego* wielomianu  $y_j(x)$ !

## Przykład



Interpolacja za pomocą naturalnych splajnów kubicznych z 21 węzłami. Wynik interpolacji praktycznie pokrywa się z funkcją interpolowaną (tą samą, co w poprzednich przykładach), oznaczoną linią kropkowaną.



## Interpolacja na płaszczyźnie — splajny bikubiczne

Przypuśćmy, że pewną funkcję dwu zmiennych,  $f(x, y)$ , mamy stabelaryzowaną w węzłach dwuwymiarowej siatki kwadratowej:

$$\begin{array}{cccccc} f_{11}=f(x_1, y_1) & f_{21}=f(x_2, y_1) & f_{31}=f(x_3, y_1) & \cdots & f_{n1}=f(x_n, y_1) \\ f_{12}=f(x_1, y_2) & f_{22}=f(x_2, y_2) & f_{32}=f(x_3, y_2) & \cdots & f_{n2}=f(x_n, y_2) \\ f_{13}=f(x_1, y_3) & f_{23}=f(x_2, y_3) & f_{33}=f(x_3, y_3) & \cdots & f_{n3}=f(x_n, y_3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{1n}=f(x_1, y_n) & f_{2n}=f(x_2, y_n) & f_{3n}=f(x_3, y_n) & \cdots & f_{nn}=f(x_n, y_n) \end{array} \quad (25)$$

“Wiersze” tej siatki odpowiadają **ustalonym wartościom zmiennej  $y$** .

“Kolumny” tej siatki odpowiadają **ustalonym wartościom zmiennej  $x$** .

Przypuśćmy, że chcemy znaleźć wartość funkcji  $f(x^*, y^*)$ , gdzie  $(x^*, y^*)$  jest wewnętrznym punktem siatki. W tym celu postępujemy jak następuje:

1. Przeprowadzamy splajn wzdłuż każdego “wiersza”. W każdym wierszu wartość zmiennej  $y$  jest ustalona, więc jest to za każdym razem zwykły splajn jednowymiarowy. Każdy splajn pociąga koszt numeryczny rzędu  $O(n)$ , a zatem obliczenie splajnów wzdłuż wszystkich wierszy pociąga koszt rzędu  $O(n^2)$ .
2. Obliczamy wartość każdego z powyższych splajnów w punkcie  $x = x^*$ . W ten sposób dostajemy  $n$  wartości funkcji w punktach  $(x^*, y_1)$ ,  $(x^*, y_2)$ ,  $\dots$ ,  $(x^*, y_n)$ .
3. Przez powyższe punkty przeprowadzmy splajn w kierunku  $y$  (przy ustalonej wartości  $x = x^*$ ) i wyliczamy wartość tego splajnu w punkcie  $(x^*, y^*)$ . Wymaga to dodatkowych  $O(n)$  operacji, zatem cały koszt jest zdominowany przez  $O(n^2)$ .

## Interpolacja za pomocą funkcji wymiernych

“Szttywności” interpolacji wielomianowej można uniknąć interpolując za pomocą funkcji wymiernych, to znaczy ilorazów wielomianów:

$$r(x) = \frac{P_\mu(x)}{Q_\nu(x)}. \quad (26)$$

Funkcje wymierne z łatwością modelują większe bogactwo zachowań, niż wielomiany. Zagadnienie interpolacji wymiernej jest opracowane od strony teoretycznej gorzej niż interpolacji wielomianowej, a ponieważ problem interpolacji wymiernej *nie ma* jednoznacznego rozwiązania, istnieje szereg konkurencyjnych podejść.

W tym wykładzie skorzystam z opublikowanego w 2007 algorytmu Floatera i Hormanna (zobacz także tutaj).

## Algorytm Floatera i Hormanna

Niech  $x_0, x_1, \dots, x_n$  będą wzajemnie różnymi punktami (węzłami interpolacji) i niech  $f_j = f(x_j)$  będą stabelaryzowanymi wartościami pewnej funkcji w węzłach. Wybieramy parametr interpolacji  $d$ ,  $0 \leq d \leq n$ . Niech  $p_i(x)$  będzie wielomianem interpolującym rozpiętym na punktach  $x_i, \dots, x_{i+d}$ . Wówczas

$$r(x) = \frac{\sum_{i=0}^{n-d} \lambda_i(x) p_i(x)}{\sum_{i=0}^{n-d} \lambda_i(x)}, \quad (27a)$$

gdzie

$$\lambda_i(x) = \frac{(-1)^i}{(x - x_i) \cdots (x - x_{i+d})}. \quad (27b)$$

$r(x)$  jest gładką “mieszanką” lokalnych wielomianów interpolacyjnych.

Można pokazać, że  $r(x)$  nie ma biegunów na osi rzeczywistej oraz że można go zapisać w następującej postaci barycentrycznej:

$$r(x) = \frac{\sum_{k=0}^n \frac{w_k}{x - x_k} f_k}{\sum_{k=0}^n \frac{w_k}{x - x_k}} \quad (28a)$$

$$w_k = \sum_{i \in J_k} (-1)^i \prod_{j=i, j \neq k}^{i+d} \frac{1}{x_k - x_j} \quad (28b)$$

gdzie  $J_k = \{i \in I : k - d \leq i \leq k\}$ ,  $I = \{0, 1, \dots, n - d\}$ .

Zastanówmy się, czy (28) istotnie daje *interpolację*, to znaczy czy zgadza się z funkcją interpolowaną w węzłach. Niech  $x \rightarrow x_l$ , gdzie  $x_l$  jest którymś węzłem, a więc zerem mianownika któregoś z ułamków występujących w liczniku i mianowniku (28a). Wówczas w obu sumach dominować będzie tylko człon z  $k = l$ , a zatem

$$r(x \rightarrow x_l) \rightarrow \frac{\frac{w_l}{x - x_l} f_l}{\frac{w_l}{x - x_l}} = f_l, \quad (29)$$

a zatem badane wyrażenie odtwarza interpolowaną funkcję w węzłach.

**Praktyczne zastosowanie** algorytmu Floatera i Hormanna wygląda tak:

- Sprawdzamy, czy  $x$  jest blisko (z dokładnością do błędu obcięcia) węzła  $x_k$ ; jeśli tak, wynikiem jest stabelaryzowana wartość funkcji  $f_k$ ;
- jeżeli nie, obliczamy  $r(x)$  według wzoru (28a). Wagi  $w_k$  obliczamy tylko raz, na początku całej procedury.

Jak dobrać parametr  $d$ ? Praktyka pokazuje, że w większości typowych przypadków wystarczy brać  $d = 3$ , aczkolwiek niekiedy potrzebne jest nawet  $d = 8$ . Jeżeli interpolowana funkcja jest dostatecznie gładka, błąd interpolacji nie przekracza  $O(h_{\max}^{d+1})$ , gdzie  $h_{\max}$  jest największą odległością pomiędzy węzłami.

Jeżeli węzły interpolacji są równoodległe,  $\forall i: x_i - x_{i-1} = h$ , wyrażenia na wagi przyjmują szczególnie prostą postać:

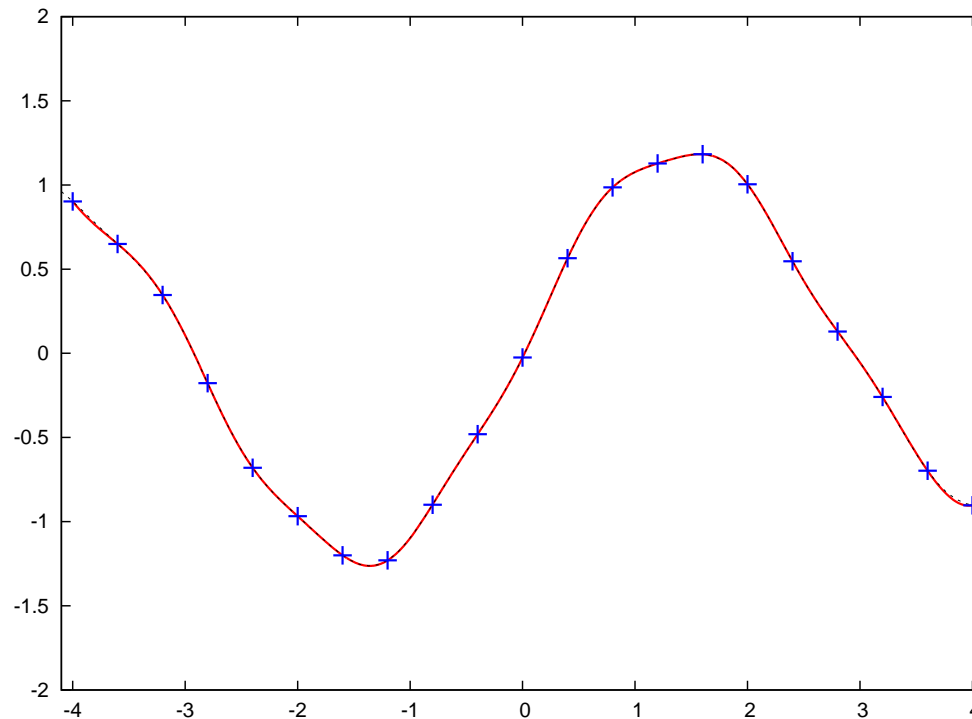
$$w_k = \frac{(-1)^{k-d}}{h^d} \sum_{i \in J_k} \frac{1}{(k-i)!(i+d-k)!} \quad (30)$$

Ponieważ ostateczny wynik nie zmieni się, jeśli wszystkie wagi przemnożymy przez tę samą stałą (obliczamy stosunek dwóch wyrażeń!), dla równoodległych węzłów interpolacji możemy wybrać wagi całkowite postaci

$$w_k = (-1)^{k-d} \sum_{i \in J_k} \binom{d}{k-i}. \quad (31)$$



## Przykład



Interpolacja funkcjami wymiernymi wg algorytmu Floatera i Hormanna z 21 równoodległymi węzłami i parametrem  $d = 3$ . Dokładność interpolacji jest efektywnie taka sama, jak dla splajnów, ale algorytm Floatera i Hormanna jest *numerycznie mniej złożony*.

## Inne rodzaje interpolacji

W specyficznych sytuacjach, głównie w kontekście analizy sygnałów, stosuje się też zupełnie inne rodzaje interpolacji: Interpolację trygonometryczną i falkową (waveletową). Mianowicie, “sygnał” (ciąg zdyskretyzowanych wartości w węzłach) rozkłada się na ciąg funkcji trygonometrycznych lub falek, a następnie wartości pomiędzy węzłami (lub brakujące wartości w węzłach!) znajduje się korzystając ze znalezionej rozkładu. Ten zespół zagadnień wykracza poza ramy niniejszego wykładu.

## Różniczkowanie numeryczne

Zdarza się, że mając tylko stabelaryzowane wartości funkcji z jakimś skończonym krokiem — tak, jak w wypadku interpolacji — chcemy numerycznie wyznaczyć wartość pochodnej funkcji w węzłach. Jest to zagadnienie bardzo podatne na błędy i należy go unikać, ale czasami nie ma wyjścia. . .

Pochodna zdefiniowana jest jako granica ilorazu różnicowego:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (32)$$

Na potrzeby numeryczne, mając do dyspozycji tylko zdyskretyzowane wartości funkcji, granicę możemy zastąpić wyrażeniem skończonym na (co najmniej) trzy różne sposoby (dla uproszczenia zakładamy, że odległość między węzłami jest stała):

Iloraz różnicowy “do przodu”:

$$f'_j = \frac{f_{j+1} - f_j}{h} \quad (33a)$$

Wsteczny iloraz różnicowy:

$$f'_j = \frac{f_j - f_{j-1}}{h} \quad (33b)$$

Symetryczny iloraz różnicowy:

$$f'_j = \frac{f_{j+1} - f_{j-1}}{2h} \quad (33c)$$

Wszystkie te przybliżenia mogą dać *różne* wyniki. Co więcej, niesymetryczne przybliżenia (33a),(33b) mogą wprowadzać pewien błąd systematyczny, zależny od wypukłości (od drugiej pochodnej) analizowanej funkcji. Przybliżenie symetryczne (33c) jest pod tym względem najbezpieczniejsze.

Jakość uzyskanego przybliżenia pochodnej mocno zależy od kroku interpolacji: im większy krok, tym przybliżenie pochodnej jest gorsze.

## Przykład

Niech  $f(x) = \frac{1}{2}x^2$ . Wówczas poszczególne ilorazy różnicowe dają

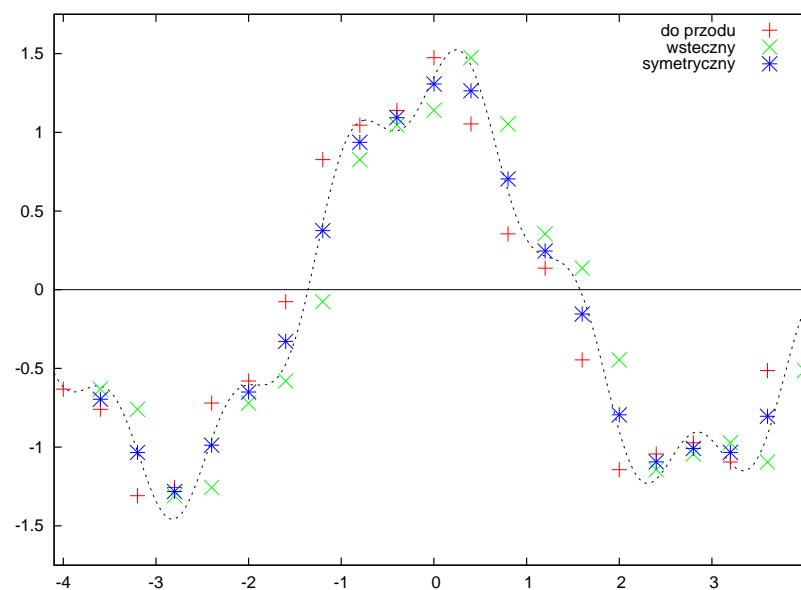
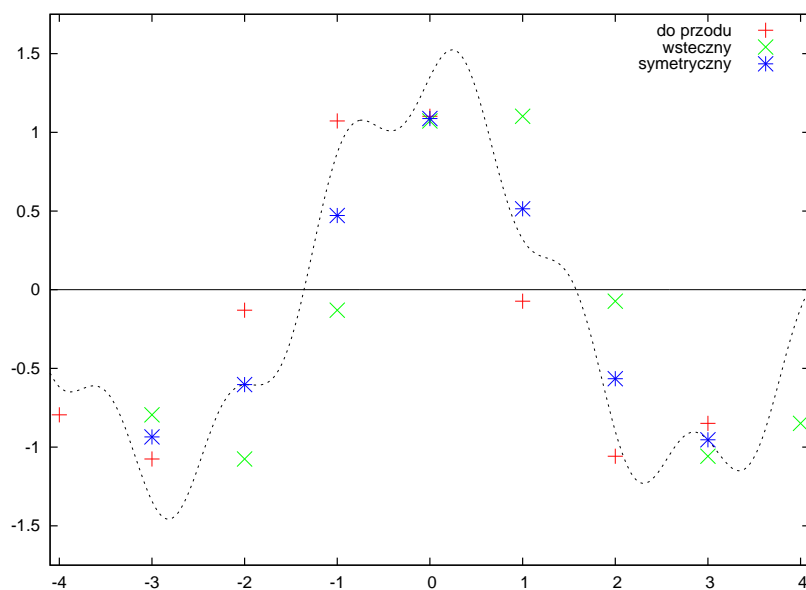
$$f'_{\text{do przodu}} = x + \frac{1}{2}h, \quad (34a)$$

$$f'_{\text{wsteczny}} = x - \frac{1}{2}h, \quad (34b)$$

$$f'_{\text{symetryczny}} = x. \quad (34c)$$

Symetryczny iloraz różnicowy daje w tym wypadku wartość dokładną, ale jest to przypadek, wynikający ze szczególnie prostej postaci funkcji (z tego, że różniczkowana funkcja jest wielomianem stopnia drugiego).

## Przykłady



Wyniki różniczkowania numerycznego funkcji w 9 (lewy panel) i 21 węzłach (prawy panel) za pomocą wzorów (33a)-(33c). Linia kropkowana oznacza dokładną wartość pochodnej. Widoczne są systematyczne różnice pomiędzy przybliżeniami “do przodu” a “wstecznym”. W tym przykładzie użyto tej samej funkcji, co w poprzednich.

## Różniczkowanie splajnów

Najlepszym, a zarazem numerycznie tanim, sposobem różniczkowania numerycznego jest **przeprowadzenie splajnu, a następnie zróżniczkowanie go w węzłach**. Korzystamy przy tym z wszystkich własności splajnów, a więc z semi-analitycznych wzorów i z *pewności*, że pochodna jest ciągła w węzłach, co pozwala nam uniknąć niejednoznaczności związanej ze stosowaniem wzorów (33).

Dla uproszczenia w dalszym ciągu będziemy zakładać, że mamy do czynienia z naturalnymi splajnami kubicznymi z równoodległymi węzłami.

Różniczkując wyrażenie (21a), właściwe dla przedziału  $[x_j, x_{j+1}]$ , i obliczając pochodną w *prawym* krańcu przedziału, otrzymujemy

$$y'_{j+1} = \frac{1}{h}(f_{j+1} - f_j) + \frac{1}{6}h(2\xi_{j+1} + \xi_j). \quad (35)$$



Równie dobrze moglibyśmy jednak wziąć wielomian dla przedziału  $[x_{j+1}, x_{j+2}]$ , zróżniczkować go i obliczyć pochodną w *lewym* krańcu przedziału:

$$y'_{j+1} = \frac{1}{h}(f_{j+2} - f_{j+1}) - \frac{1}{6}h(2\xi_{j+1} + \xi_{j+2}). \quad (36)$$

*Wiemy*, że wyrażenia (35), (36) są sobie równe. Istotnie, porównując ich prawe strony, dostajemy wyrażenie

$$\xi_j + 4\xi_{j+1} + \xi_{j+2} = \frac{6}{h^2}(f_j - 2f_{j+1} + f_{j+2}), \quad (37)$$

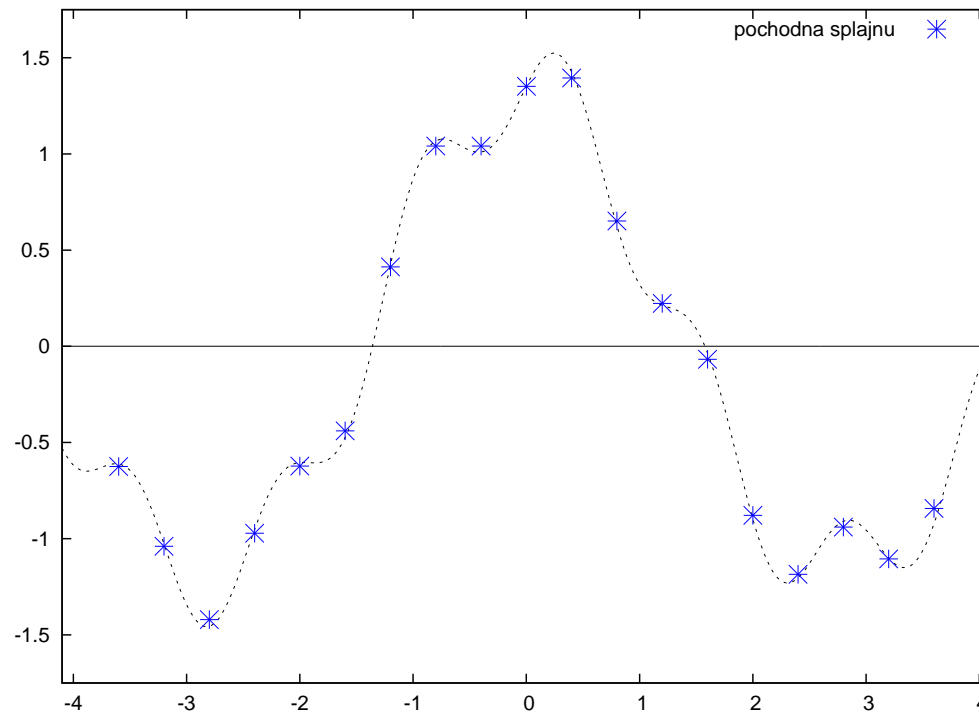
odpowiadające  $(j+1)$  wierszowi równania (23).

Wyrażenia (35), (36) są sobie równe w arytmetyce dokładnej. W praktycznych obliczeniach numerycznych mogą wystąpić jakieś różnice. Aby je zminimalizować, bierzemy średnią prawych stron (35), (36). Ostatecznie jako przybliżenie pochodnej dostajemy

$$y'(x_j) = \frac{f_{j+1} - f_{j-1}}{2h} - \frac{1}{12}h (\xi_{j+1} - \xi_{j-1}), \quad (38)$$
$$j = 2, 3, \dots, n - 1$$

Wyrażenie (38) ma postać symetrycznego ilorazu różnicowego z poprawkami wynikającymi ze znajomości *przybliżenia* drugiej pochodnej. Wielkości  $\xi_k$  wyliczamy z równania (23).

## Przykład



Różniczkowanie numeryczne na podstawie naturalnego splajnu kubicznego z 21 równoodległymi węzłami. Linia kropkowana oznacza prawdziwy przebieg pochodnej interpolowanej funkcji.