

Wstęp do metod numerycznych

Interpolacja

P. F. Góra

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

2010

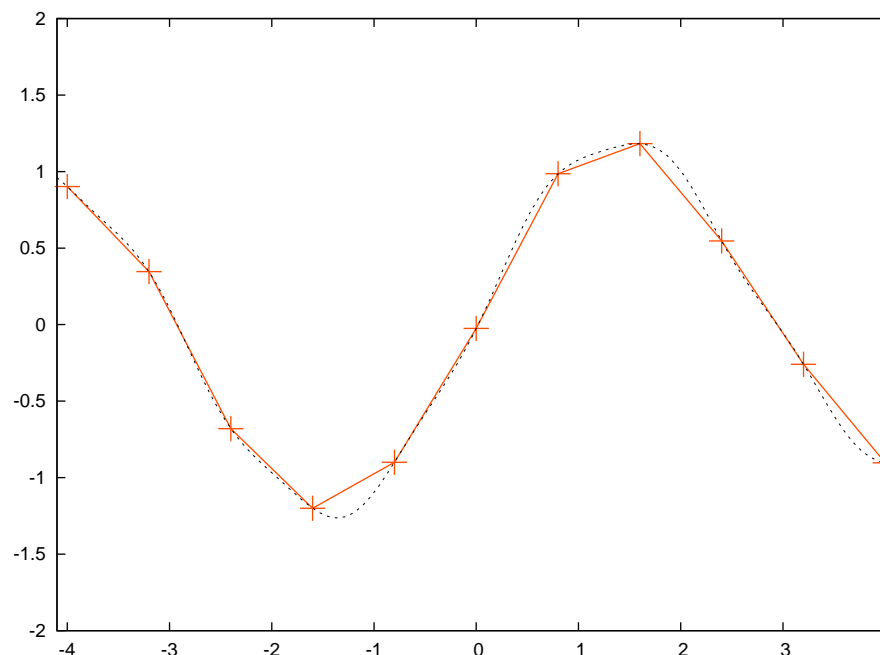
Interpolacja

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 początku wykładu, 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

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

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*. Widać, że jej dwa wiersze (k -ty i l -ty) są liniowo niezależne, jeżeli $x_k \neq x_l$, 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.

Wzór interpolacyjny Lagrange'a

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

$$f(x) = \sum_{j=1}^n l_j(x) f_j + E(x), \quad (4a)$$

gdzie

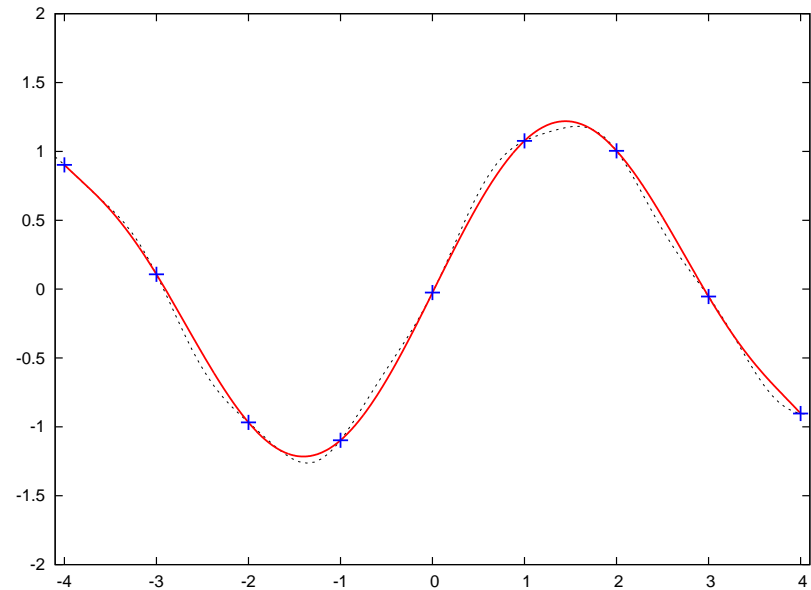
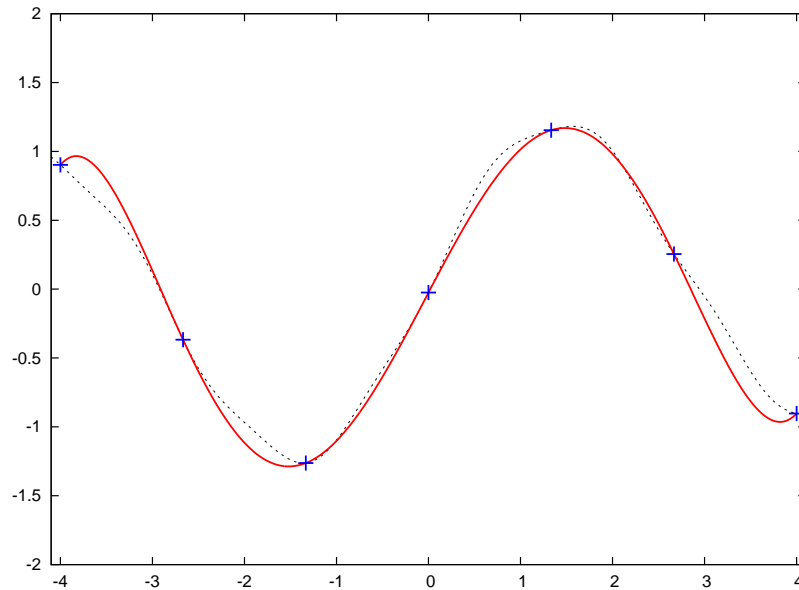
$$l_j(x) = \frac{(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}. \quad (4b)$$

$E(x)$ w (4a) jest nazywane *resztą* lub *błędem interpolacji*. Zauważmy, że $l_j(x)$ jest wielomianem stopnia $n-1$ oraz

$$l_j(x_k) = \delta_{jk}. \quad (5)$$

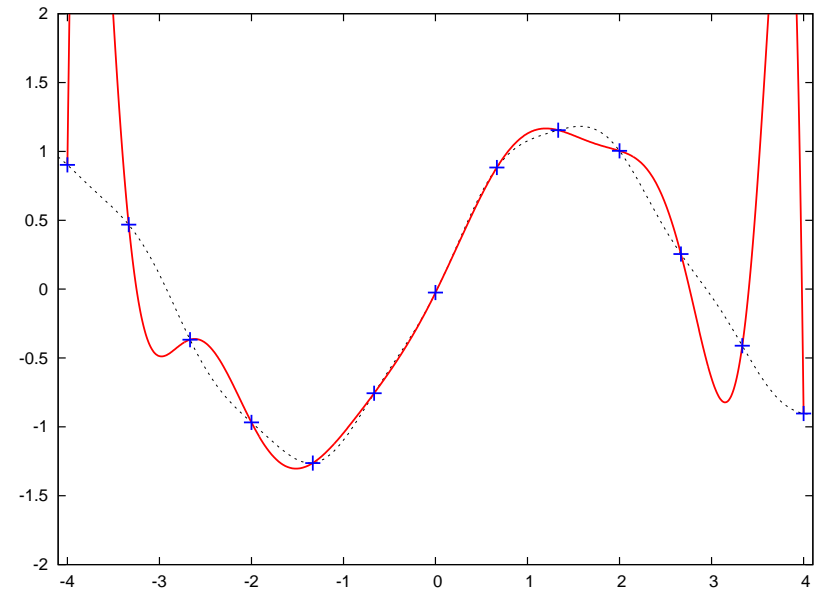
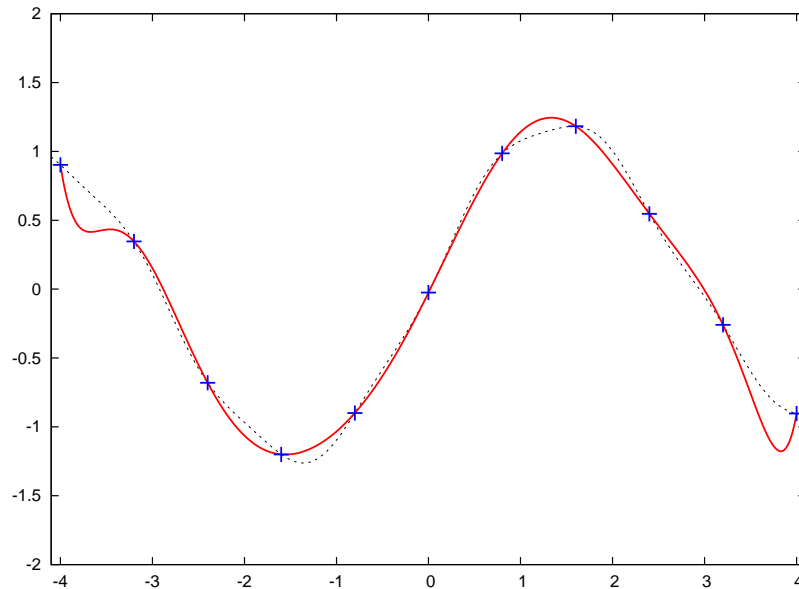
Jeżeli $f(x)$ jest wielomianem stopnia co najwyżej $n-1$, $E(x)$ znika tożsamościowo. Mówimy, że interpolacja (4a) ma dokładność $n-1$.

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.

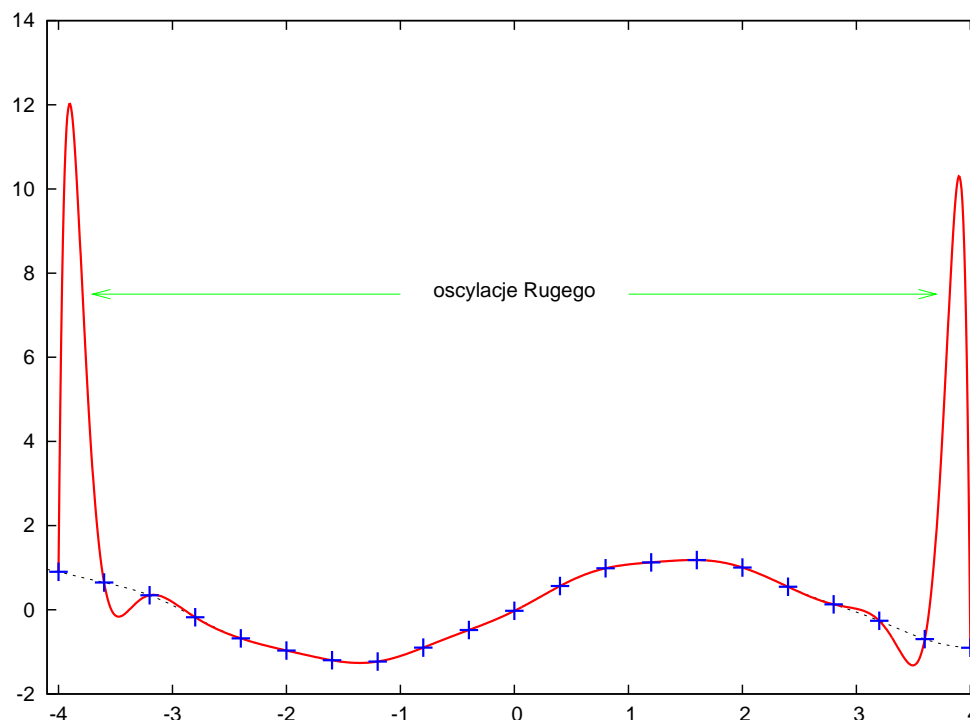
Przykłady



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

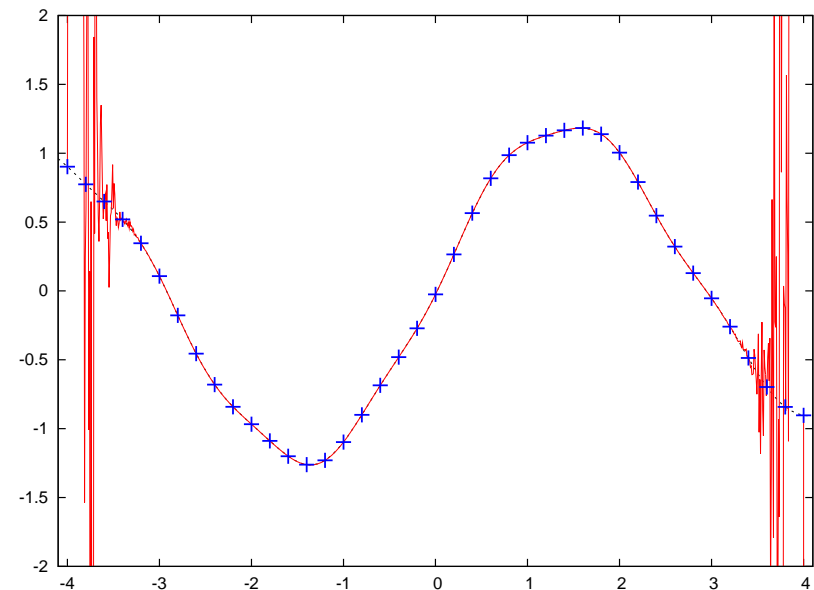
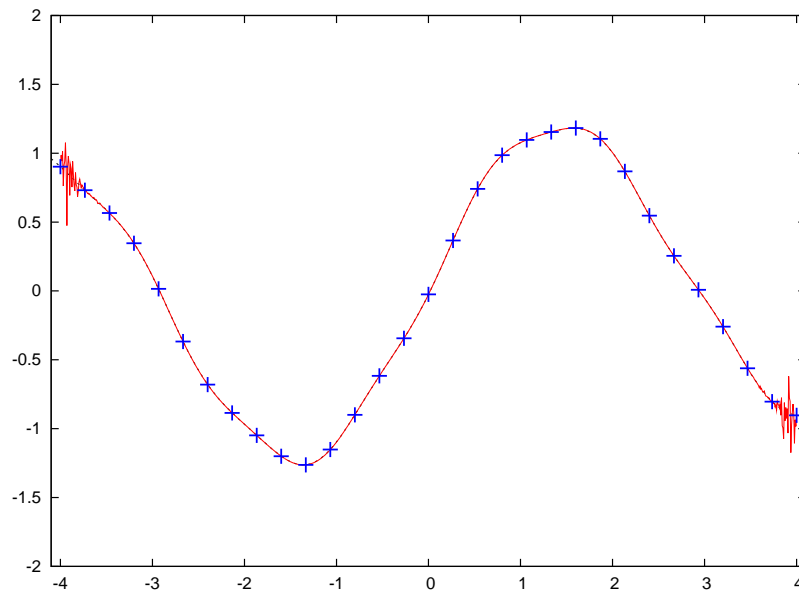
Dalsze zwiększanie liczby węzłów prowadzi do *pogorszenia* jakości interpolacji, zwłaszcza w pobliżu krańców przedziału zawierającego węzły.

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 ~ 200).

Błąd interpolacji

Wielomian (4b) wygodnie jest niekiedy zapisywać w postaci

$$l_j(x) = \frac{p_n(x)}{(x - x_j)p'_n(x_j)} \quad (6a)$$

gdzie

$$p_n(x) = \prod_{i=1}^n (x - x_i), \quad p'_n(x_j) = \left. \frac{dp_n}{dx} \right|_{x=x_j} \quad (6b)$$

Oznaczmy $y(x) = \sum_{j=1}^n l_j(x) f_j$ (jest to wielomianowa część wzoru interpolacyjnego Lagrange'a (4a), bez reszty).

Rozpatrzmy funkcję

$$F(z) = f(z) - y(z) - [f(x) - y(x)] \frac{p_n(z)}{p_n(x)}. \quad (7)$$

$F(z)$, 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

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

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 ξ . Zauważmy, że $y^{(n)}(z) = 0$, gdyż $y(z)$ jest wielomianem stopnia $n-1$. Ostatecznie otrzymujemy

$$0 = F^{(n)}(\xi) = f^{(n)}(\xi) - \underbrace{[f(x) - y(x)]}_{E(x)} \frac{n!}{p_n(x)} \quad (9)$$

czyli

$$E(x) = \frac{p_n(x)}{n!} f^{(n)}(\xi). \quad (10)$$

ξ 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.

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*. Można skonstruować wówczas interpolację wielomianową rzędu $2n - 1$, postaci

*Jest to zagadnienie o stosunkowo niewielkich zastosowaniach praktycznych, ale za to o dużym znaczeniu teoretycznym.

$$y(x) = \sum_{i=1}^n h_i(x) f_i + \sum_{i=1}^n \bar{h}_i(x) f'_i + E(x), \quad (11a)$$

gdzie

$$h_i(x) = \left(1 - 2(x - x_i)l'_i(x_i)\right) l_i^2(x), \quad (11b)$$

$$\bar{h}_i(x) = (x - x_i)l_i^2(x). \quad (11c)$$

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

$$E(x) = \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\xi), \quad (11d)$$

gdzie ξ jest pewnym punktem wewnętrznym przedziału rozpiętego na węzłach i wartości x . Wielomian interpolacyjny (11a) zgadza się z interpolowaną funkcją oraz jej pochodną w węzłach.

Interpolacja za pomocą funkcji sklejanych

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 sklejanych**.

Funkcją sklejaną 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 sklejanych rzędu 3, czyli “splajnów kubicznych” (ang. *cubic splines*).

Cubic splines

Założmy[†], że oprócz wartości funkcji w węzłach, znamy także wartości drugiej pochodnej interpolowanej funkcji 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
$f_i'' = f''(x_i)$	f_1''	f_2''	f_3''	\dots	f_n''

[†]W przeciwieństwie do interpolacji Hermite'a, jest to tylko *założenie robocze*!

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

$$y_j(x) = A f_j + B f_{j+1} + C f_j'' + D f_{j+1}'', \quad (12a)$$

gdzie

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j}, \quad B = \frac{x - x_j}{x_{j+1} - x_j}, \quad (12b)$$

$$C = \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2, \quad D = \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2. \quad (12c)$$

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

$$\frac{d^2 y_j}{dx^2} = A f_j'' + B f_{j+1}'', \quad (13)$$

także wartości drugiej pochodnej (12a) zgadzają się z zadanymi wartościami w węzłach.

Jest jednak pewien **problem**: w rzeczywistości **nie znamy wartości drugich pochodnych** f_j'' . Nie skorzystaliśmy jednak jeszcze z wymogu ciągłości **pierwszej** pochodnej (12a) w węzłach. 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

$$\begin{aligned} \frac{x_j - x_{j-1}}{6} f_{j-1}'' + \frac{x_{j+1} - x_{j-1}}{3} f_j'' + \frac{x_{j+1} - x_j}{6} f_{j+1}'' \\ = \frac{f_{j+1} - f_j}{x_{j+1} - x_j} - \frac{f_j - f_{j-1}}{x_j - x_{j-1}}. \end{aligned} \quad (14)$$

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

Naturalny splajn kubiczny

Jeżeli mamy n węzłów interpolacji, mamy $n-2$ wewnętrznych punktów zszycia, w których możemy żądać ciągłości pochodnej. W takim wypadku (14) stanowi układ $n-2$ równań z n niewiadomymi. Trzeba podać jakieś dodatkowe warunki.

Najczęściej przyjmuje się, że $f_1'' = f_n'' = 0$. 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 (14) przybiera szczególnie prostą postać:

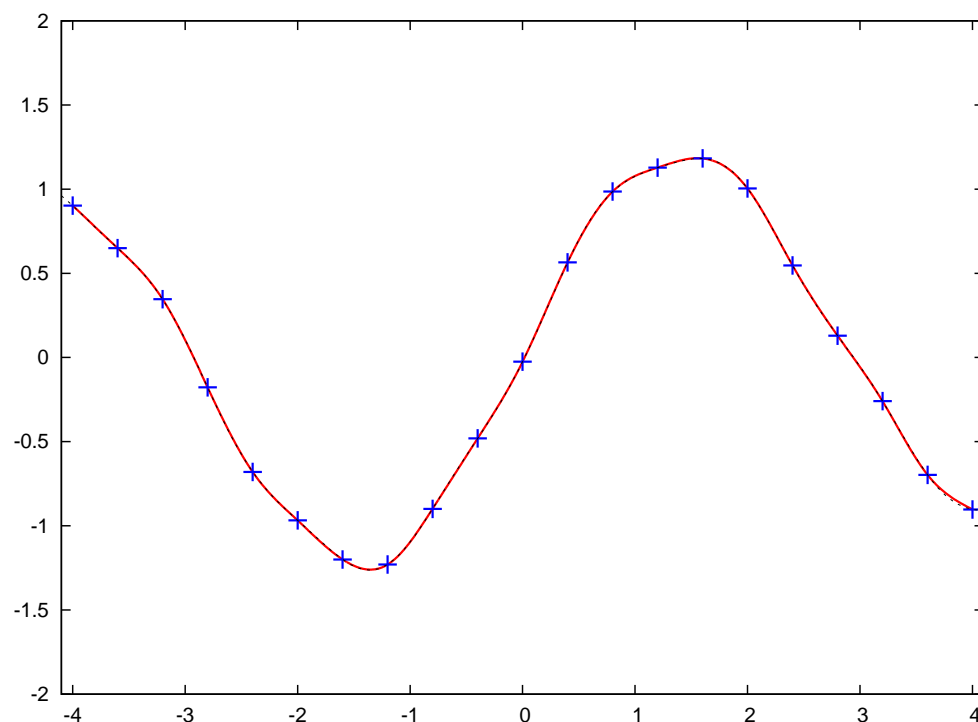
$$\begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} f_2'' \\ f_3'' \\ f_4'' \\ \vdots \\ f_{n-2}'' \\ f_{n-1}'' \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} f_1 - 2f_2 + f_3 \\ f_2 - 2f_3 + f_4 \\ f_3 - 2f_4 + f_5 \\ \vdots \\ f_{n-3} - 2f_{n-2} + f_{n-1} \\ f_{n-2} - 2f_{n-1} + f_n \end{bmatrix} \quad (15)$$

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

Praktyczne zastosowanie naturalnych splajnów kubicznych przebiega dwuetapowo:

1. Rozwiązujemy układ równań (14) (lub, jeżeli można, (15)) na $n-2$ nieznanymi wielkościami $\{f_j''\}_{j=2}^{n-1}$. 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 (12a) 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:

$$\begin{array}{ccccccccc} f_{11}=f(x_1, y_1) & f_{21}=f(x_2, y_1) & f_{31}=f(x_3, y_1) & \cdots & f_{n1}=f(x_n, y_1) \\ f_{12}=f(x_1, y_2) & f_{22}=f(x_2, y_2) & f_{32}=f(x_3, y_2) & \cdots & f_{n2}=f(x_n, y_2) \\ f_{13}=f(x_1, y_3) & f_{23}=f(x_2, y_3) & f_{33}=f(x_3, y_3) & \cdots & f_{n3}=f(x_n, y_3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{1n}=f(x_1, y_n) & f_{2n}=f(x_2, y_n) & f_{3n}=f(x_3, y_n) & \cdots & f_{nn}=f(x_n, y_n) \end{array} \quad (16)$$

“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

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

$$r(x) = \frac{P_\mu(x)}{Q_\nu(x)}. \quad (17)$$

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

$$r(x) = \frac{\sum_{i=0}^{n-d} \lambda_i(x) p_i(x)}{\sum_{i=0}^{n-d} \lambda_i(x)}, \quad (18a)$$

gdzie

$$\lambda_i(x) = \frac{(-1)^i}{(x - x_i) \cdots (x - x_{i+d})}. \quad (18b)$$

$r(x)$ 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*:

$$r(x) = \frac{\sum_{k=0}^n \frac{w_k}{x - x_k} f_k}{\sum_{k=0}^n \frac{w_k}{x - x_k}} \quad (19a)$$

$$w_k = \sum_{i \in J_k} (-1)^i \prod_{j=i, j \neq k}^{i+d} \frac{1}{x_k - x_j} \quad (19b)$$

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

Zastanówmy się, czy (19) 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 (19a). Wówczas w obu sumach dominować będzie tylko człon z $k = l$, a zatem

$$r(x \rightarrow x_l) \rightarrow \frac{\frac{w_l}{x - x_l} f_l}{\frac{w_l}{x - x_l}} = f_l, \quad (20)$$

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 (19a). 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.

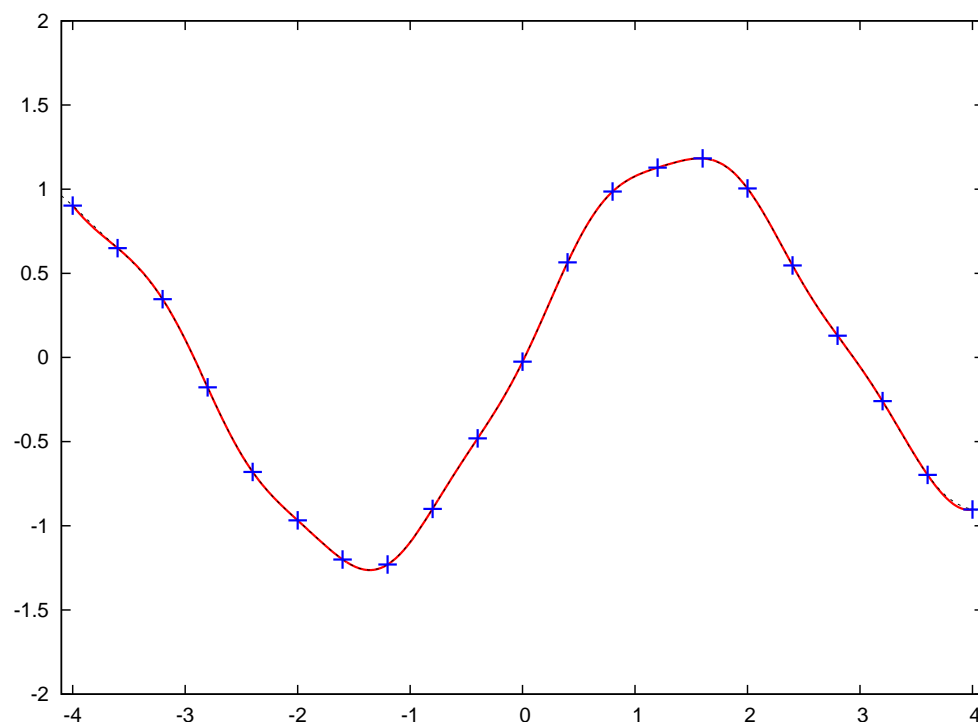
Jeżeli węzły interpolacji są równoodległe, $\forall i: x_i - x_{i-1} = h$, wyrażenia na wagi przyjmują szczególnie prostą postać:

$$w_k = \frac{(-1)^{k-d}}{h^d} \sum_{i \in J_k} \frac{1}{(k-i)!(i+d-k)!} \quad (21)$$

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

$$w_k = (-1)^{k-d} \sum_{i \in J_k} \binom{d}{k-i}. \quad (22)$$

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 znalezionego rozkładu. Ten zespół zagadnień wykracza poza ramy niniejszego wykładu.