

Krzysztof Wesołowski
Automatyka i Robotyka
Grupa: środa, 14.00 - LD_3
Data: 26 XI 2008

Interpolacja funkcji

Dane do obliczeń

Interpolowana funkcja:

$$f(x) := \frac{1}{25x^2 + 1}$$

Ciąg na potrzeby wykresów

$$z := -3, -3 + 10^{-2} .. 3$$

Obliczanie wektora węzłów równo odległych

p - początek przedziału (pierwszy pkt)

k - ostatni punkt

ile - ile węzłów (czyli węzły dzieli na ilość-1 przedziałów)

$$\text{rownoodlegle}(p, k, \text{ile}) := \left| \begin{array}{l} \text{for } i \in 0 .. \text{ile} - 1 \\ w_i \leftarrow \frac{(k - p)}{\text{ile} - 1} \cdot i + p \\ \text{return } w \end{array} \right.$$

Obliczanie węzłów czebyszewa, argumenty j.w.

$$\text{czebyszew}(p, k, \text{ile}) := \left| \begin{array}{l} \text{for } i \in 0 .. \text{ile} - 1 \\ w_i \leftarrow \frac{1}{2} \left[(k - p) \cdot \cos\left(\frac{2 \cdot i + 1}{\text{ile}} \cdot \frac{\pi}{2}\right) + k + p \right] \\ \text{sort}(w) \end{array} \right.$$

Definiowanie ilości węzłów:

$$\text{ile} := 7$$

Sprawdzenie wygenerowanych węzłów

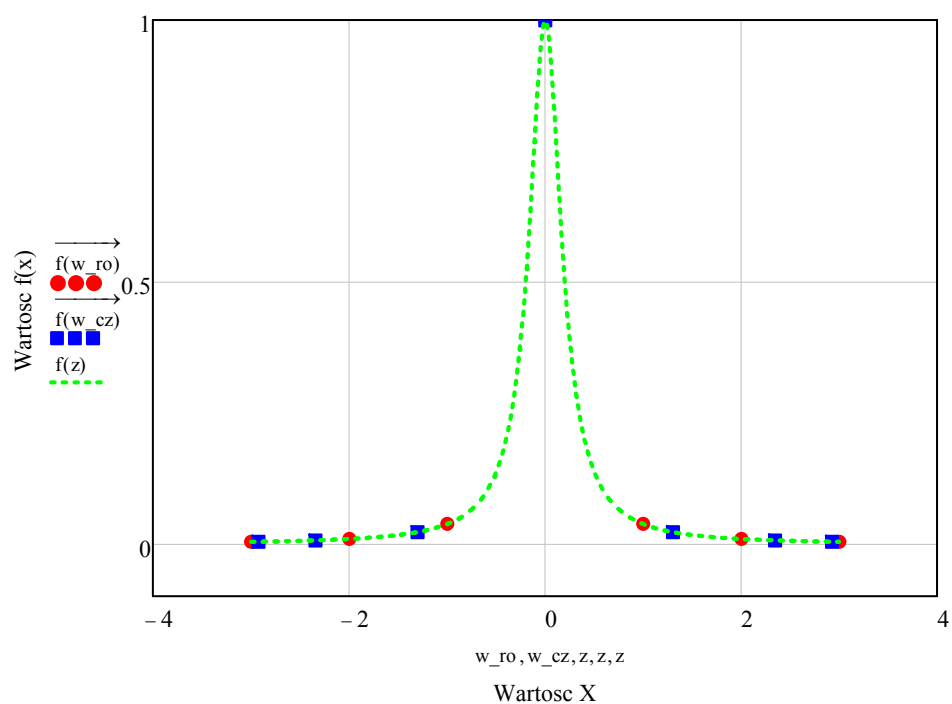
$$\text{czebyszew}(-3, 3, \text{ile}) = \begin{pmatrix} -2.925 \\ -2.345 \\ -1.302 \\ 0 \\ 1.302 \\ 2.345 \\ 2.925 \end{pmatrix} \quad \text{rownoodlegle}(-3, 3, \text{ile}) = \begin{pmatrix} -3 \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}$$

Przypisanie wektorow do przyszlego uzycia:

`w_cz := czebyszew(-3, 3, ile)`

`w_ro := rownoodlegle(-3, 3, ile)`

Wykres przedstawiajacy funkcje interpolowana, oraz wezly interpolacji,
Czerwone: Rownoodlegle
Niebieskie: Czebyszewa

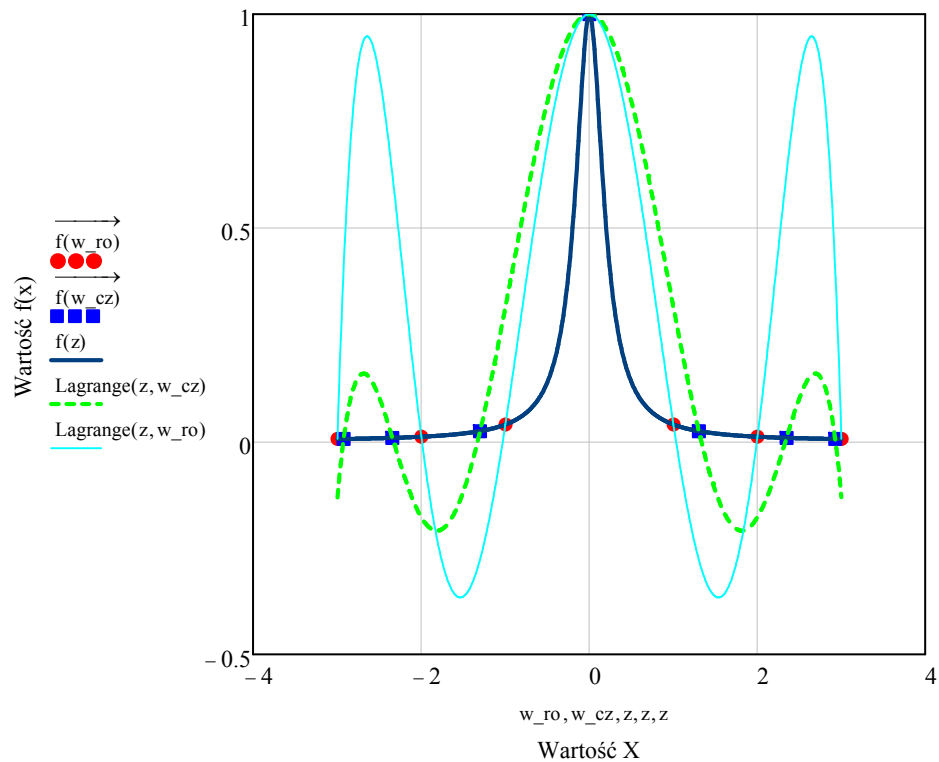


Implementacja metody Lagrange'a

x oznacza argument w którym interpolujemy funkcje
w oznacza wektor wezlow interpolacji wielomianem
warto od razu zauwazyc ze ta implementacja jest raczej demonstracyjna,
gdyz jest b.niefektywna: wsp wielomianu sa obliczane od nowa dla kazdego argumentu

$$\text{Lagrange}(x, w) := \sum_{j=0}^{\text{last}(w)} \left(f(w_j) \cdot \prod_{i=0}^{\text{last}(w)} \text{if} \left(i \neq j, 1, \frac{x - w_i}{w_j - w_i} \right) \right)$$

Ponizej na wykres nalożono węzły interpolacji, funkcję interpolowaną, oraz wykresy funkcji interpolujących. C



Implementacja metody Newtona

Funkcja zwracająca wektor ilorazów różnicowych, poprzez wypoziomienie pierwszej kolumny macierzy, obliczenie macierzy trojkatnej i zwrot pierwszego wiersza jako wektora

```

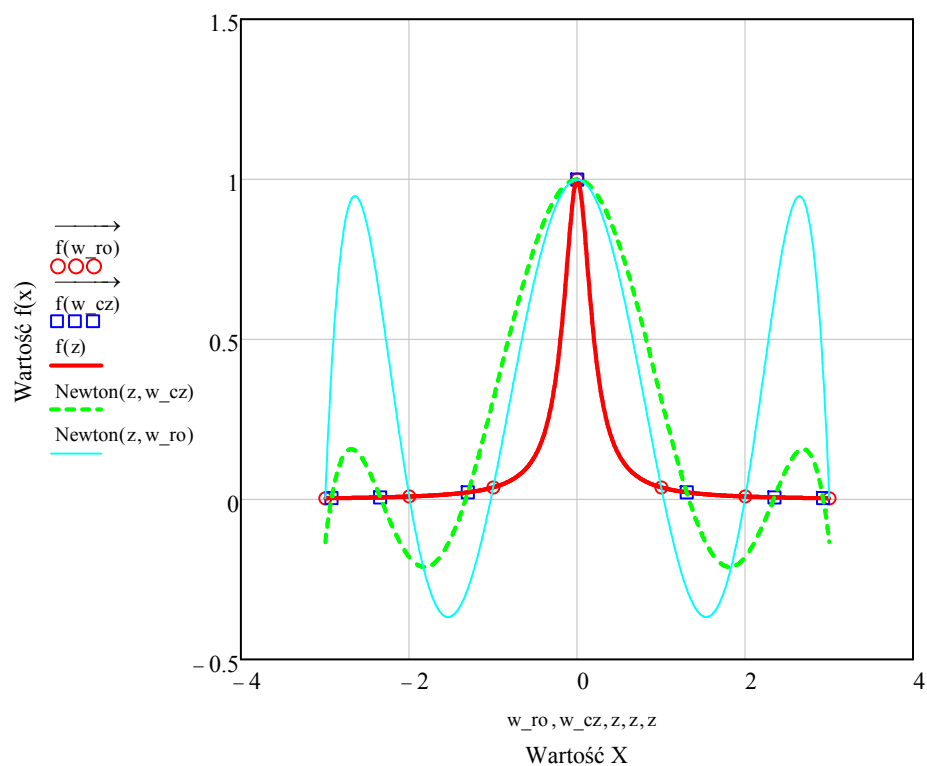
IlorazyRoznicowe(w) :=
  for i ∈ 0 .. last(w)
    Ai,0 ← f(wi)
  for j ∈ 1 .. last(w)
    for i ∈ 0 .. last(w) - j
      Ai,j ←  $\frac{A_{i+1,j-1} - A_{i,j-1}}{w_{i+j} - w_i}$ 
  (AT)<0>

```

I funkcja interpolująca:

$$\text{Newton}(x, w) := \sum_{i=1}^{\text{last}(w)} \left[\text{IlorazyRoznicowe}(w)_i \cdot \prod_{j=0}^{i-1} (x - w_j) \right] + \text{IlorazyRoznicowe}(w)_0$$

Co było do przewidzenia wielomiany interpolujące są identyczne

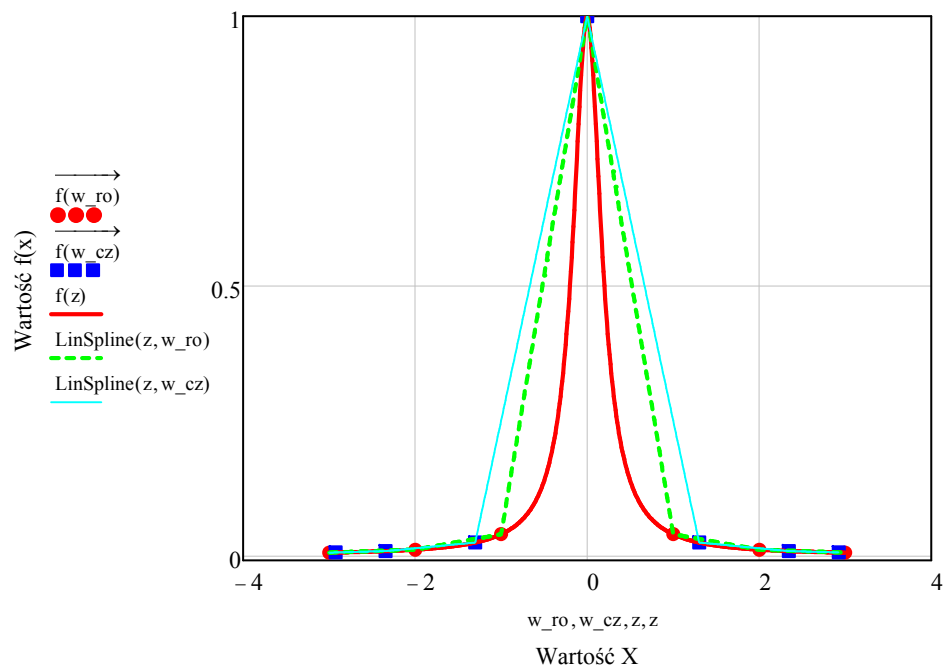


Funkcje sklejane I rzędu

Implementacja interpolacji funkcjami sklejanymi I stopnia (Linear-Splines)

$$\text{LinSpline}(x, w) := \begin{cases} \text{for } i \in 0 \dots \text{last}(w) \\ (p \leftarrow i) \text{ if } x \geq w_i \\ \frac{(w_p - x) \cdot (f(w_p) - f(w_{p+1}))}{w_p - w_{p+1}} + f(w_p) \end{cases}$$

Dla porownania uzylem tutaj zarowno wezlow rownoogleglych jak i wezlow czebyszewa, te drugie nie maja jednak specjalnych wlasnosci dla takiej interpolacji



Funkcje sklepane III rzędu

$$h(k, w) := w_{k+1} - w_k$$

$$d(k, w) := \frac{f(w_{k+1}) - f(w_k)}{h(k, w)}$$

$$\lambda(k, w) := \frac{h(k, w)}{h(k-1, w) + h(k, w)}$$

$$\rho(k, w) := 1 - \lambda(k, w)$$

Deklaracja zmiennych pomocniczych, wszystkie zaleza od wektora, gdyz chce latwo sprawdzic zachowanie funkcji dla roznej ilosci wezlow, zmieniajac po prostu wartosc zmiennej ile (o samej gory)

```

BigMatrix(w) :=
  for i ∈ 0 .. last(w) - 2
    Ai,i ← 2
  for j ∈ 1 .. last(w) - 2
    Aj-1,j ← λ(j, w)
    Aj,j-1 ← ρ(j + 1, w)
  A

```

Funkcja tworząca dużą macierz współczynników, potrzebna do wyznaczenia współczynników m' do $m(n-1)$

$m_0 := 0$ $m_n := 0$

Zmienne narzucające dodatkowe ograniczenia na brancach (tutaj dla funkcji naturalnej)

Funkcja wyznaczająca wektor V czyli wektor wyrazów wolnych równania z którego wyznaczamy współczynniki m

```

BigVector(w) :=
  a0 ← 6 ·  $\frac{d(1, w) - d(0, w)}{h(1, w) + h(0, w)} - \rho(1, w) m_0$ 
  for i ∈ 1 .. last(w) - 3
    ai ← 6 ·  $\frac{d(i + 1, w) - d(i, w)}{h(i + 1, w) + h(i, w)}$ 
  alast(w)-2 ← 6 ·  $\frac{d(\text{last}(w) - 1, w) - d(\text{last}(w) - 2, w)}{h(\text{last}(w) - 1, w) + h(\text{last}(w) - 2, w)} - \lambda[(\text{last}(w) - 1), w] \cdot m_n$ 
  a

```

$T_{ro} := \text{BigMatrix}(w_{ro})$

$V_{ro} := \text{BigVector}(w_{ro})$

Macierz i wektor dla węzłów równoodległych

Macierz i wektor dla węzłów równoodległych, przypominam że węzły te nie mają żadnego szczególnego znaczenia, zamieszczam je w celach porównawczych

$T_{cz} := \text{BigMatrix}(w_{cz})$

$V_{cz} := \text{BigVector}(w_{cz})$

Wektor rozwiązań dla wartości m_0 do m_n

$M_{ro} := \text{stack}(m_0, \text{lsolve}(T_{ro}, V_{ro}), m_n)$

$M_{cz} := \text{stack}(m_0, \text{lsolve}(T_{cz}, V_{cz}), m_n)$

Funkcja interpolująca, stosunkowa krótka bo większość obliczeń znalazła się powyżej. Obliczanie rozwiązania układu równań dla każdego punktu było zbyt nieefektywne nawet na cele demonstracyjne

```
Spline3(x, w, m) :=
  for i ∈ 0 .. last(w)
    (p ← i) if x ≥ wi
    wynik ← (x - wp)3 ·  $\frac{m_{p+1} - m_p}{6 \cdot h(p, w)}$  + (x - wp)2 ·  $\frac{m_p}{2}$ 
    wynik ← wynik +  $\left[ (x - w_p) \cdot \left[ d(p, w) - \frac{h(p, w)(2m_p + m_{p+1})}{6} \right] + f(w_p) \right]$ 
```

