

Krzysztof Wesołowski
Automatyka i Robotyka
Grupa: środa, 14.00 - LD_3
Data: 22 XII 2008

Aproksymacja

Aproksymacja wielomianowa

Dane do obliczeń

Poniżej przykładowy wektor punktów x i wartości funkcji w tych pkt, i macierz przechowująca punkty w wierszach

$$\begin{array}{l} \text{px} := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix} \quad \text{py} := \begin{pmatrix} -5.97 \\ -2.63 \\ 2.7 \\ -6.49 \\ -3.54 \\ -25.1 \\ -24.79 \\ -45.42 \\ -64.17 \\ -79.55 \end{pmatrix} \quad \text{pxy} := \text{augment}(\text{px}, \text{py}) \end{array}$$

funkcja będąca bazą jednomianową, wykorzystywana do obliczania wartości funkcji w punkcie x , dla nr_funkcji funkcji z bazy

$\text{bjm}(x, \text{nr_funkcji}) := x^{\text{nr_funkcji}}$

Funkcja tworząca macierz D , gdzie ilość wierszy zależy od ilości punktów, zaś ilość kolumn zależy od oczekiwanej ilości funkcji z bazy, ostatni argument to nazwa funkcji służących za bazę (tutaj zostanie użyta funkcja bjm)

$$D(\text{ilosc}, \text{punkty}, \text{baza}) := \begin{array}{l} \text{for } w \in 0.. \text{rows}(\text{punkty}) - 1 \\ \quad \text{for } k \in 0.. \text{ilosc} - 1 \\ \quad \quad \text{wynik}_{w,k} \leftarrow \text{baza}(\text{punkty}_{w,0}, k) \\ \text{wynik} \end{array}$$

I prezentacja jej działania dla wielomianu 3 stopnia (czyli 4 wymiarowa baza jednomianowa), i naszych 10 punktów

$$D(4, \text{pxy}, \text{bjm}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \\ 1 & 5 & 25 & 125 \\ 1 & 6 & 36 & 216 \\ 1 & 7 & 49 & 343 \\ 1 & 8 & 64 & 512 \\ 1 & 9 & 81 & 729 \end{pmatrix}$$

Funkcja obliczająca wektor współczynników przy zadanej ilości funkcji z bazy, punktach oraz bazie funkcji:

```
w_wsp(ilosc, punkty, baza) := | D ← D(ilosc, punkty, baza)
                             | dtf ← DT · submatrix(punkty, 0, rows(punkty) - 1, 1, 1)
                             | dtd ← DT D
                             | lsolve(dtd, dtf)
```

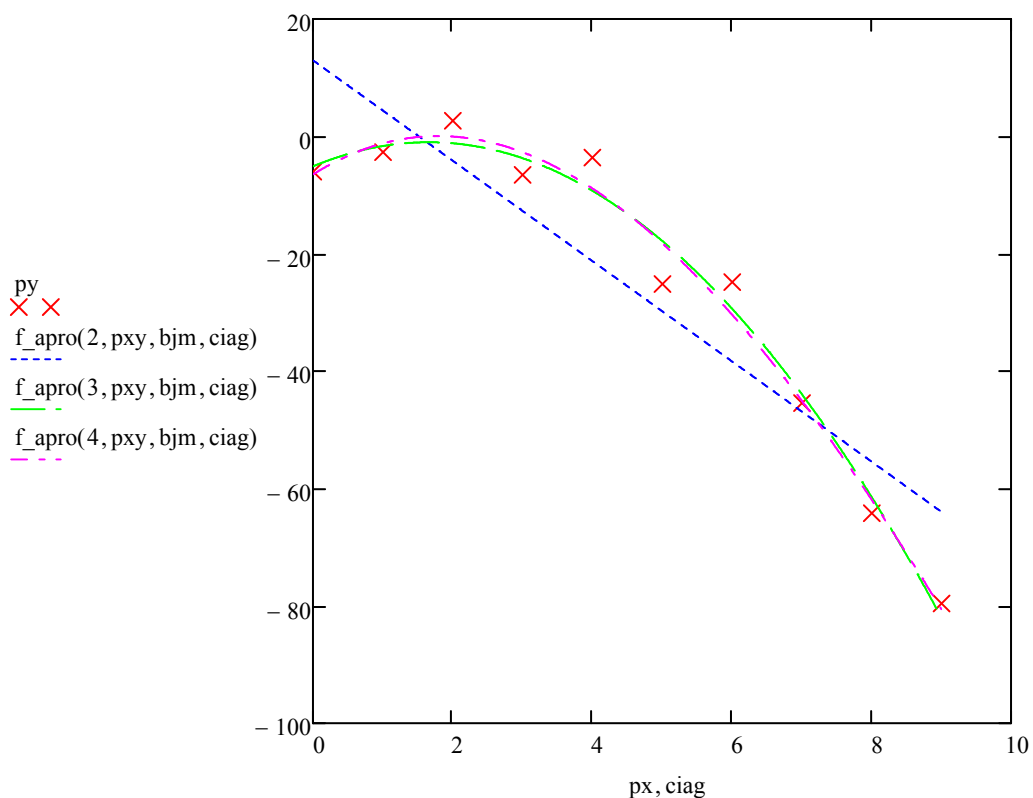
$$w_wsp(4, \text{pxy}, \text{bjm}) = \begin{pmatrix} -6.399 \\ 7.461 \\ -2.236 \\ 0.055 \end{pmatrix}$$

Funkcja obliczająca wartość funkcji aproksymującej przy zadanej bazie i ilości funkcji bazowych (2funkcje bazowe => wielomina 1 stopnia), oraz danych punktach już zadanych. Funkcja na swoje potrzeby wyznacza wektor współczynników.

```
f_apro(ilosc, punkty, baza, x) := | współczynniki ← w_wsp(ilosc, punkty, baza)
                                | ∑i=0ilosc-1 (współczynnikii · baza(x, i))
```

ciag := 0, 0.01 .. 9 *ciąg do rysowania wykresu*

I wykres funkcji aproksymującej dla 1, 2 i 3 stopnia wielomianu.



Poniżej zamieszczam implementacje funkcji bazowych ortogonalnych Grama, które można przekazać jako argument do funkcji aproksymującej. ilość węzłów dla wygody pozostawie wpisana na sztywno.

$$\text{wgr}(\text{argument}, \text{ktory}) := \left| \begin{array}{l} k \leftarrow \text{ktory} \\ x \leftarrow \text{argument} \\ n \leftarrow 9 \\ \sum_{s=0}^k \left[(-1)^s \cdot \frac{k!}{s! \cdot (k-s)!} \cdot \frac{(k+s)!}{s! \cdot k!} \cdot \text{if } s > 0, \frac{\prod_{i=0}^{s-1} (x-i)}{\prod_{i=0}^{s-1} (n-i)}, 1 \right] \end{array} \right|$$

Dla dowolnego argumentu, np 4.5 możemy więc obliczyć wartość funkcji aproksymującej w następujący sposób:

$$\begin{array}{ll} \text{f_apro}(2, \text{pxy}, \text{bjm}, 4.5) = -25.496 & \text{dla bazy jednomianowej, ilość f bazowych to 2 czyli} \\ & \text{wielomian pierwszego stopnia} \\ \text{f_apro}(2, \text{pxy}, \text{wgr}, 4.5) = -25.496 & \text{dla bazy ortogonalnej} \end{array}$$

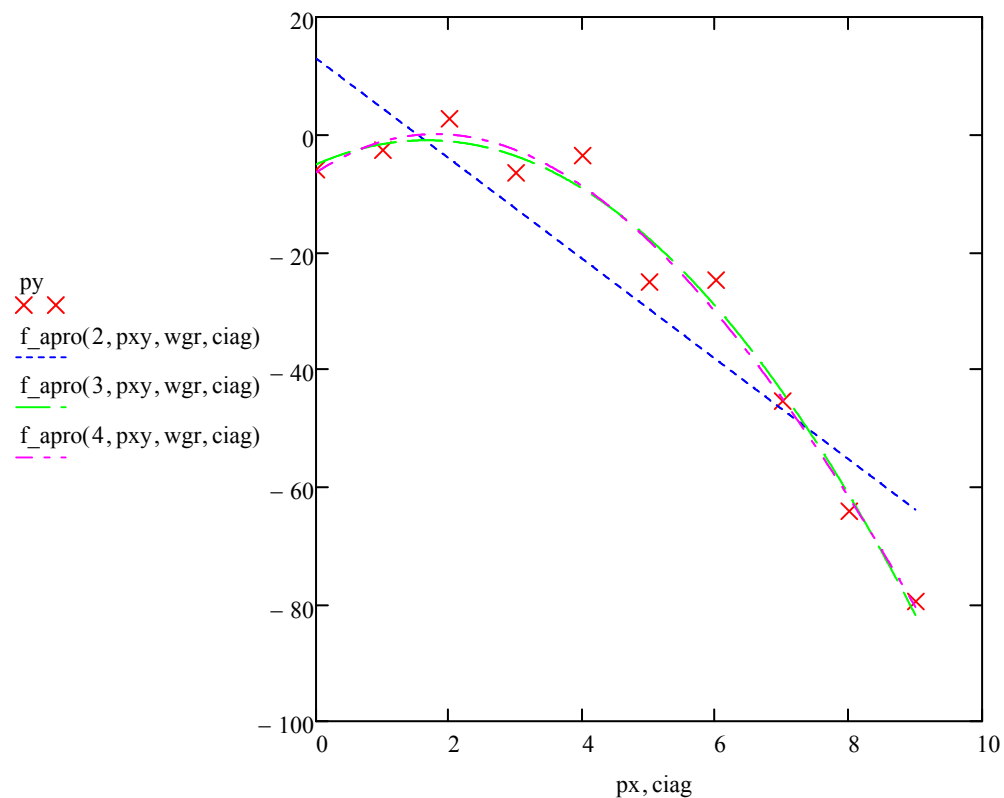
Macierz $D^T D$ zgodnie z oczekiwaniami jest diagonalna:

$$D(3, \text{pxy}, \text{wgr})^T \cdot D(3, \text{pxy}, \text{wgr}) = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 4.074 & 0 \\ 0 & 0 & 3.667 \end{pmatrix}$$

Zaś wektor współczynników jest równy:

$$\text{w_wsp}(3, \text{pxy}, \text{wgr}) = \begin{pmatrix} -25.496 \\ 38.456 \\ -17.982 \end{pmatrix}$$

Poniżej wykres funkcji aproksymujących, zgodnie z oczekiwaniami jest on identyczny, gdyż obie metody używają wielominaów tego samego stopnia, a optymalny wśród nich wybierany jest tą samą metodą.



Zaimplementuj teraz funkcję obliczającą odchylenie standardowe na podstawie 2 wektorów punktów, punktów funkcji aproksymującej i punktów rzeczywistych

$$\text{od_std}(\text{v_apr}, \text{v_real}) := \sqrt{\frac{\sum_{i=0}^{\text{last}(\text{v_apr})} \left[(\text{v_apr}_i - \text{v_real}_i)^2 \right]}{\text{last}(\text{v_apr}) + 1}}$$

Ponizej obliczenia dla tego, konkretnego przypadku, majace na celu optymalne dobranie stopnia wielomianu aproksymujacego na podstawie poprawiania sie dokladnosci aproksymacji, za ktorej miare przyjmuje odchylenie standardowe.

Wektor wartosci dla wielomianu danego stopnia:

```
apr(stopien) :=
  for i ∈ px
    wyniki ← f_apro(stopien + 1, pxy, wgr, i)
  wynik
```

i kolejne odchylenia standardowe:

	0
0	0
1	11.543
2	3.833
3	3.71
4	3.7
5	3.67
6	3.489
7	3.487
8	3.487
9	6.287·10 ⁻¹⁵
10	4.217·10 ⁻¹⁴

Jak widać odchylenie maleje przy przejściu z stopnia 1 na drugi, kolejne poprawy sa pomijalne, aż do momentu osiagnięcia stopnia 9, gdzie to aproksymacja staje sie interpolacja a odchylenie jest wywolane bledem maszynowym

Aproksymacja trygonometryczna

Funkcja ktora zaburzamy i ktora bedziemy przyblizac.

$$\text{tryg_func}(x) := \sin(x) + \sin\left(2x + \frac{\pi}{4}\right)$$

Ilosc argumentow ktore uznamy za dane, z przedziału o 0 do 2π , zakładam ze funkcja ma okres bedacy wielokrotnoscia 2π , przez co takie punkty sa wystarczajace i na tym przedziale funkcj mozna aproksymowac dokladnie

ilosc_argumentow := 8

Wektory argumentow i wektory danych wartosci funkcji:

```
tx :=
  for i ∈ 0..ilosc_argumentow - 1
    wyniki ←  $\frac{2\pi \cdot i}{\text{ilosc\_argumentow}}$ 
  wynik
```

```

ty :=
  for i ∈ 0..ilosc_argumentow - 1
  |
    wyniki ← tryg_func(txi) + rnd(1) - 0.5
  |
    wynik

```

```

tx =
  (
    0
    0.785
    1.571
    2.356
    3.142
    3.927
    4.712
    5.498
  )
ty =
  (
    0.208
    1.108
    0.378
    -0.15
    1.03
    -0.326
    -1.497
    -1.61
  )
txy := augment(tx, ty)

```

Teraz stworze 2 funkcje, ktore dla podanego stopnia (ilosc par funkcji trygonometrycznych ktorych uzyjemy) utworza nam wektory wspolczynnika, dla cos i sin.

```

t_wsp_a(st) :=
  N ← ilosc_argumentow
  for j ∈ 0..st
  |
    wynikj ←  $\frac{2}{N} \cdot \sum_{i=0}^{N-1} \left( ty_i \cdot \cos\left(\frac{2\pi j \cdot i}{N}\right) \right)$ 
  |
    wynik

```

```

t_wsp_b(st) :=
  N ← ilosc_argumentow
  for j ∈ 0..st
  |
    wynikj ←  $\frac{2}{N} \cdot \sum_{i=0}^{N-1} \left( ty_i \cdot \sin\left(\frac{2\pi j \cdot i}{N}\right) \right)$ 
  |
    wynik

```

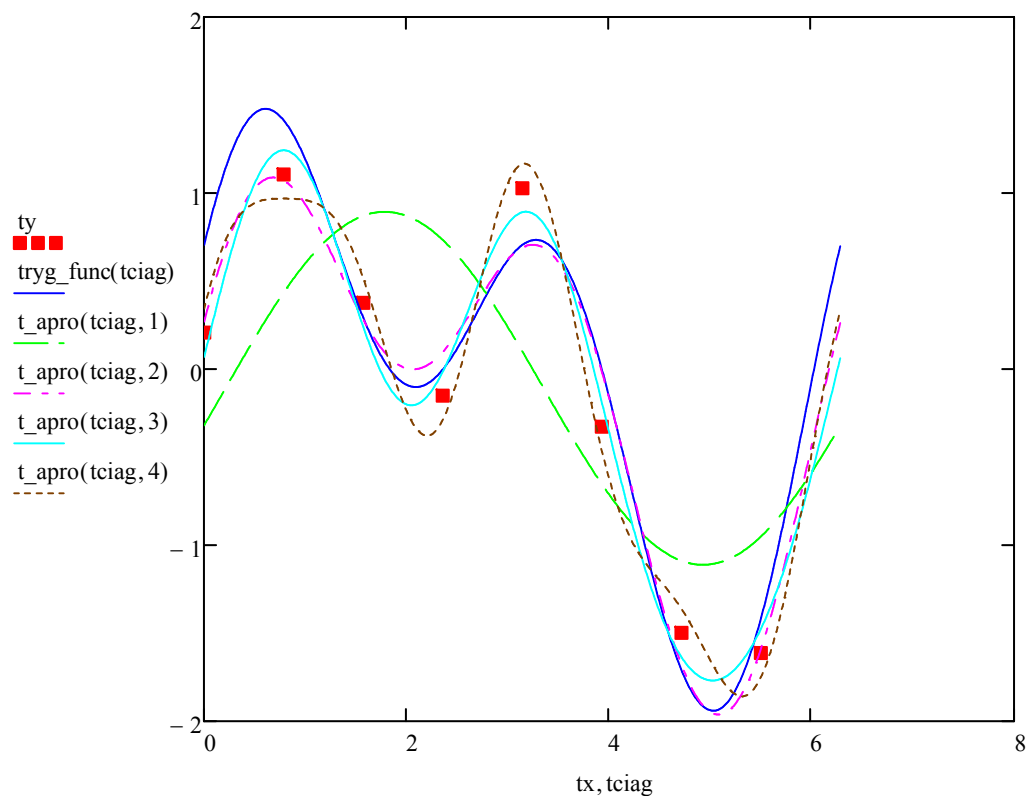
Wlasciwa funkcja aproksymujaca:

$$t_{apro}(x, st) := \frac{1}{2} t_{wsp_a}(st)_0 + \sum_{j=1}^{st} \left(t_{wsp_a}(st)_j \cos(j \cdot x) + t_{wsp_b}(st)_j \sin(x \cdot j) \right)$$

Ciąg argumentów dla wykresu:

```
tciag := 0, 0.01..2π
```

Oraz wykres prezentujący po kolei: punkty ktorych uzywamy w aproksymacji, funkcja ktora na dobra sprawe jest tutaj aproksymowana, oraz kolejne aproksymacje, z rozna iloscia harmonicznch.



Funkcja obliczająca wartości funkcji aproksymującej w punktach węzłowych w zależności od ilości funkcji trygonometrycznych bazowych:

```

tapr(stopien) :=
  for i ∈ 0..ilosc_argumentow - 1
    wyniki ← t_apro(txi, stopien)
  wynik

```

Oraz wektor odchyleń standardowych:

```

  for i ∈ 0..3
    wyniki ← od_std(tapr(i + 1), ty)
  wynik

```

$$= \begin{pmatrix} 0.644 \\ 0.2 \\ 0.137 \\ 0.137 \end{pmatrix}$$

Jak widać największy zysk dokładności otrzymujemy przy zmianie stopnia z 1 na 2. Pomimo że dalej można to odchylenie jeszcze zmniejszyć, to na wykresie widac że zdecydowanie najlepiej wykorzystać wiedzę o tym jakiego "stopnia" jest poszukiwana funkcja. Wynik ten można też delikatnie zmienić, użycie Ctrl+F9 do przeliczenia arkusza wylosuje nowe zaburzenia.