

Feliks Bańka, 323733

MNUM – Projekt 2

Zadanie 1

Dla podanych w tabeli poniżej danych pomiarowych (próbek) metodą najmniejszych kwadratów należy wyznaczyć funkcję wielomianową $y = f(x)$ najlepiej aproksymującą te dane (proszę przetestować wielomiany stopni: 3, 5, 7, 9, 10). Kod aproksymujący ma mieć postać funkcji Matlaba zwracającej wartość wielomianu w podanym punkcie. W sprawozdaniu proszę przedstawić na rysunku otrzymaną funkcję na tle danych (funkcję aproksymującą proszę próbować przynajmniej 10 razy częściej niż dane). Do rozwiązania zadania najmniejszych kwadratów proszę wykorzystać rozkład SVD.

Rozwiązać ten układ korzystając z solwera Matlaba. Proszę obliczyć błąd aproksymacji w dwóch normach: euklidesowej oraz maksimum (nieskończoność).

x_i	y_i
-10	-18.7370
-8	-8.1583
-6	-1.9146
-4	-0.3887
-2	1.8030
0	1.1890
2	0.4738
4	0.4726
6	0.0941
8	-2.3716
10	-6.6512

Liniowe zadanie najmniejszych kwadratów

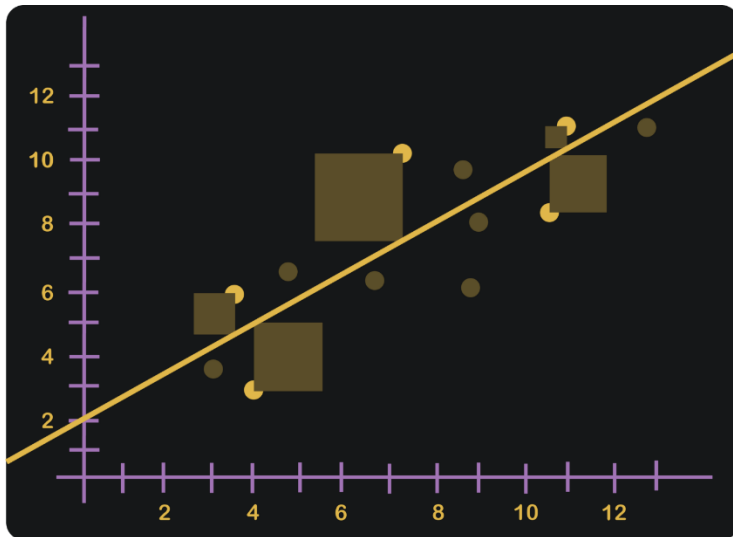
Liniowe zadanie najmniejszych kwadratów polega na rozwiązaniu m równań z n niewiadomymi ($m > n$), w sensie minimalizacji normy drugiego wektora niespełnienia równań.

Dla danej macierzy A i wektora b należy znaleźć wektor u , taki, że:

$$\forall x \in \mathbb{R}^n \quad \|b - Ax\|_2 \leq \|b - Au\|_2$$

W sposób graficzny można to zinterpretować jako funkcję $G(x)$, która będzie leżała jak najbliżej wszystkich wyników – tak aby suma odległości wszystkich punktów od linii jak najmniejsza. Z racji tego, że niektóre wartości tych odległości są ujemne, a niektóre dodatnie, przed zsumowaniem ich podnosi się je do kwadratu.

$$G(x, y) = \sum_{i=1}^n (y_i - f(x_i))^2$$



Rozkład SVD

Dla dowolnej macierzy $A_{m \times n}$ ($A \in R^{m \times n}$, $m \geq n$) istnieją macierze ortonormalne $U_{m \times m}$, $V_{n \times n}$ i macierz $\Sigma_{m \times n}$ takie, że:

$$A = U \Sigma V^T$$

Gdzie:

- $V_{m \times m}$ – macierz, której kolumnami są ortonormalne wektory własne macierzy $A^T A$
- $U_{n \times n}$ – macierz, której kolumnami są ortonormalne wektory własne macierzy $A A^T$
- Σ - macierz diagonalna (przekątniowa), taka, że $\Sigma = \text{diag}(\sigma_i)$, gdzie σ_i – nieujemne wartości szczególne (osobliwe) macierzy A , zwyczajowo uporządkowane nierosnąco.

Rozwiązywanie LZNK za pomocą rozkładu SVD

W naszym przykładzie wartości naszych danych z tabeli są niejako współczynnikami (macierzą A w $Ax=b$). Naszą macierz A tworzą rzędy kolejnych próbek (x_i) kolejnych potęg x_i od 0 do n , gdzie n to stopień wielomianu, dla którego obliczamy aproksymację.

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$Ax = b$$

Do rozwiązania liniowego zadania najmniejszych kwadratów możemy wykorzystać rozkład SVD. Z niego mamy zaś:

$$\|b - Ax\|_2 = \|b - U\Sigma V^T x\|_2 = \|U^T b - \Sigma(V^T x)\|_2 = \|\tilde{b} - \Sigma(\tilde{x})\|_2$$

Gdzie: $\tilde{b} = U^T b$, $\tilde{x} = V^T x$

Jeśli stworzymy macierz: $\Sigma^+ = [D_{n \times n}^+ + 0_{n \times (m-n)}]$, $D^+ = \text{diag}\left\{\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_k}, 0, \dots, 0\right\}$

To prawdziwa jest równość:

$$V^T \hat{x} = \Sigma^+ U^T b$$

Czyli:

$$\hat{x} = V \Sigma^+ U^T b$$

Czyli macierz pseudoodwrotna A:

$$A^+ = V \Sigma^+ U^T$$

Wtedy z:

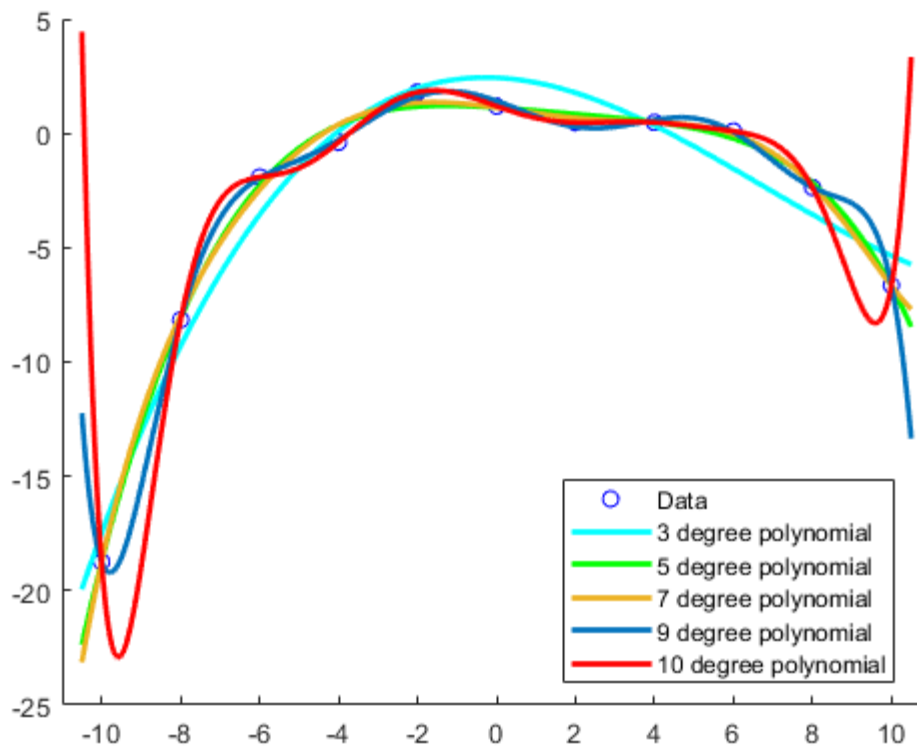
$$Az = b$$

Otrzymujemy:

$$z = V \Sigma^+ U^T b$$

Po otrzymaniu macierzy x możemy obliczyć $y = f(x_n) = z_1 x_n^n + \dots + z_n x_n + z_{n+1}$

Wykresy funkcji wielomianowych dla danych z zadania



Na wykresie znajduje się 5 funkcji wielomianowych różnych stopni, które są zminimalizowane pod względem sumy kwadratów różnic funkcji $f(x)$ i y_i . Co ciekawe funkcje 5 i 7 stopnia są niemalże tak samo dokładnie dopasowane. Wielomian 10 stopnia praktycznie idealnie pokrywa się z punktami z danych, co widać po wyniku normy residuum.

Błąd aproksymacji

Wektor residuum obliczamy wzorem:

$$\partial = A \cdot X - b$$

Normę euklidesową korzystając ze wzoru:

$$\varepsilon = \| A\tilde{x} - b \|_2 = \sum_{i=1}^n |x_i|$$

Natomiast normę maksimum:

$$\varepsilon = \| A\tilde{x} - b \|_\infty = \max_{i=1, \dots, d} |x_i|$$

Poniżej przedstawione są błędy rozwiązania dla obu tych metod:

```
Euklides measurement error for polynomial of degree 3: 3.860805
Euklides measurement error for polynomial of degree 5: 1.183178
Euklides measurement error for polynomial of degree 7: 1.107040
Euklides measurement error for polynomial of degree 9: 0.387175
Euklides measurement error for polynomial of degree 10: 0.000000
```

```
Maximum measurement error for polynomial of degree 3: 1.645689
Maximum measurement error for polynomial of degree 5: 0.757950
Maximum measurement error for polynomial of degree 7: 0.769788
Maximum measurement error for polynomial of degree 9: 0.226991
Maximum measurement error for polynomial of degree 10: 0.000000
```

Jak widać norma euklidesowa jest większa dla każdego przypadku niż norma maksimum. Natomiast błąd dla wielomianu 10 stopnia jest równy praktycznie równy 0 (wynosi dokładnie euklidesowa: $4.242800809369194e-10$, maksimum: $3.057962771890743e-10$).

Efektywność

Możemy również porównać naszą funkcję z już wbudowaną w Matlab funkcję `lsqcurvefit()`, która rozwiązuje liniowe zadanie najmniejszych kwadratów.

Po wyliczeniu norm residuum funkcji `lsqcurvefit()` dla naszych wielomianów otrzymaliśmy następujące wyniki:

```
Euklides measurement error for polynomial of degree 3: 3.860805
Euklides measurement error for polynomial of degree 5: 1.183178
Euklides measurement error for polynomial of degree 7: 1.107040
Euklides measurement error for polynomial of degree 9: 0.541421

Maximum measurement error for polynomial of degree 3: 1.645689
Maximum measurement error for polynomial of degree 5: 0.757950
Maximum measurement error for polynomial of degree 7: 0.769788
Maximum measurement error for polynomial of degree 9: 0.260240
```

Można zauważyć, że dla wielomianów stopnia 7 nasze wyniki są identyczne (do 6 miejsc po przecinku), natomiast dla wielomianu 9 stopnia nasza funkcja jest dokładniejsza w obu normach. (Dla wielomianu 10 stopnia funkcja `lsqcurvefit()` zwraca error "ponieważ względna suma kwadratów (r) zmienia się o mniej niż `options.FunctionTolerance = 1.000000e-06`.”)

Zadanie 2

Dla danych z p. 1. znaleźć wielomian interpolacyjny korzystając z macierzy Vandermonde’a. Ma on mieć postać funkcji Matlaba zwracającej wartość w podanym punkcie. Dodać wykres otrzymanej funkcji do rysunku z p. 1 próbując przynajmniej 10 razy częściej niż dane.

Interpolacja jest rodzajem estymacji, metodą konstruowania (znajdowania) nowych punktów danych w oparciu o zakres dyskretnego zbioru znanych punktów danych. Stosujemy ją, gdy analityczna funkcji $f(x)$ nie jest znana.

Twierdzenie: Istnieje dokładnie jeden wielomian interpolacyjny stopnia co najwyżej n ($n \geq 0$), który w punktach $x_0, x_1, x_2, \dots, x_n$ przyjmuje wartości $y_0, y_1, y_2, \dots, y_n$.

Naszym celem jest znalezienie wielomianu:

$$W_n(x) = a_n x^n + a_{n-1} x_{n-1} + \dots + a_1 x + a_0.$$

Macierz współczynników układu to macierz Vandermonde:

$$X = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}$$

Jest to tzw. macierz Vandermonde'a, której wyznacznik wyraża się wzorem:

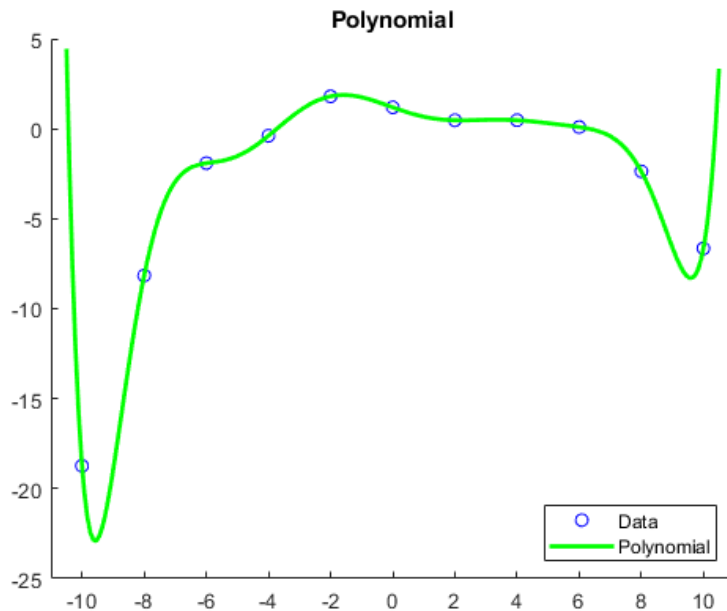
$$\det X = \prod_{1 \leq i < j \leq n} (x_j - x_i)$$

(W programie Matlab mamy już wbudowaną funkcję zwracającą macierz Vandermonde'a (`vander(Xdata)`).)

Widać stąd, że:

$$\begin{aligned} Xa &= y \\ a &= X^{-1}y \end{aligned}$$

Wykresy funkcji dla danych z zadania



Błąd aproksymacji

Błędy były liczone w taki sam sposób jak w zadaniu 1.

Norma euklidesowa: $4.439488314142017e-12$

Norma nieskończoność: $3.151257033096044e-12$

Wnioski

Po pierwsze, łatwo zauważyć, że funkcja aproksymująca dla wielomianu stopnia 10 jest bardzo podobna do funkcji interpolacji. Norma interpolacji jest ok. 100 razy mniejsza, jednak oczywiście obie te wartości są bardzo małe. Różnica pojawia się dla wielomianów niższych stopni, gdzie ta dokładność wyraźnie spada i taka aproksymacja może nam służyć jedynie, gdy z założenia musimy wykorzystać wielomian danego stopnia, licząc się z niedokładnością.