

zadanie 2.22

1. Dla podanych w tabeli poniżej danych pomiarowych metodą najmniejszych kwadratów należy wyznaczyć funkcję wielomianową $y = f(x)$ najlepiej aproksymującą te dane korzystając z układu QR.

Aproksymacja to metoda polegająca na znajdowaniu przybliżonych rozwiązań jakiegoś problemu, którego nie można przedstawić dokładnie. Najczęściej polega na przybliżaniu pewnej funkcji za pomocą funkcji przybliżającej zwanej dalej aproksymującą tak, aby uzyskać jak najdokładniejsze przybliżenie szukanej wartości. Funkcje aproksymujące najczęściej mają postać wielomianów o pewnych współczynnikach. Podany problem polega na znalezieniu tych współczynników, a następnie obliczeniu wartości funkcji aproksymującej dla danego argumentu oraz porównaniu wyniku z wartością funkcji aproksymowanej (pierwotnej).

Jako dane mamy wektor argumentów funkcji aproksymowanej „x”:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{w-1} \\ x_w \end{bmatrix}$$

oraz wektor wartości „y” jakie dana funkcja przyjmuje dla danych argumentów:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{w-1} \\ y_w \end{bmatrix}$$

gdzie „w” to liczba elementów wektora, która jest taka sama dla obu wektorów.

Aby znaleźć funkcję aproksymującą dane musimy stworzyć macierz A o „w” wierszach i „n” kolumnach, gdzie n oznacza stopień wielomianu aproksymującego dane.

$$A = \begin{bmatrix} x_1^n & x_1^{n-1} & x_1^2 & x_1^1 & 1 \\ x_2^n & x_2^{n-1} & x_2^2 & x_2^1 & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_{w-1}^n & x_{w-1}^{n-1} & x_{w-1}^2 & x_{w-1}^1 & 1 \\ x_w^n & x_w^{n-1} & x_w^2 & x_w^1 & 1 \end{bmatrix}$$

Następnie należy dokonać rozkładu QR macierzy A na ortonormalną macierz Q i górną, trójkątną macierz R.

Pokażemy, jak rozkłada się macierz A na macierz Q i R na przykładzie metody Gramma-Schmidta. Macierz Q będzie takich samych rozmiarów jak macierz A, natomiast macierz R będzie macierzą kwadratową o boku równym ilości kolumn macierzy A. Lecz najpierw przypiszmy macierzy Q macierz A: $Q = A$. W programie MATLAB użyjemy wbudowanej funkcji „qr”, aby uzyskać taki rozkład.

Załóżmy, że mamy i-tą kolumnę macierzy A, którą traktujemy jako wektor. Aby uzyskać i-ty wektor ortonormalny macierzy Q posługujemy się wzorem:

$$Q_n(:, i) = \frac{(A(:, i) - \sum_{j=1}^{i-1} (\frac{A(:, i) \cdot Q^T(:, j)}{Q^T(:, j) \cdot Q(:, j)}) \cdot Q(:, j))}{\|Q(:, i)\|}$$

gdzie $i \in \{1; k\}$, gdzie „k” to liczba kolumn macierzy A. Następnie wyznaczamy macierz R, która będzie wyglądać tak:

$$R = \begin{bmatrix} \langle Q(:, 1), A(:, 1) \rangle & \langle Q(:, 1), A(:, 2) \rangle & \dots & \langle Q(:, 1), A(:, k-1) \rangle & \langle Q(:, 1), A(:, k) \rangle \\ 0 & \langle Q(:, 2), A(:, 2) \rangle & \dots & \langle Q(:, 2), A(:, k-1) \rangle & \langle Q(:, 2), A(:, k) \rangle \\ 0 & 0 & \dots & \langle Q(:, 3), A(:, k-1) \rangle & \langle Q(:, 3), A(:, k) \rangle \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & \langle Q(:, k-1), A(:, k-1) \rangle & \langle Q(:, k-1), A(:, k) \rangle \\ 0 & 0 & \dots & 0 & \langle Q(:, k), A(:, k) \rangle \end{bmatrix}$$

Gdzie zapis $\langle Q(\dots), A(\dots) \rangle$ oznacza iloczyn skalarny wektorów wchodzących w skład macierzy Q i A. Jednak macierz R można wyznaczyć w prostszy sposób.

Macierz A to iloczyn macierzy Q i R:

$$QR = A$$

Pomnożmy równanie stronami przez transponowaną macierz Q: Q^T

$$QR = A \quad / \cdot Q^T$$

Dzięki temu otrzymamy:

$$Q^T QR = Q^T A$$

Jako iż Q jest macierzą ortonormalną działanie $Q^T Q$ daje nam macierz jednostkową I:

$$IR = Q^T A$$

Macierz I możemy wykreślić z równania, gdyż nic ona nie zmienia w naszym równaniu. Dzięki temu otrzymujemy wzór:

$$R = Q^T A$$

Następnie pozostaje nam tylko rozwiązać układ równań:

$$QR \cdot a = y$$

gdzie a to wektor szukanych współczynników wielomianu aproksymującego. Aby wyznaczyć wartość funkcji aproksymującej w żądanym przez nas punkcie wystarczy posłużyć się wzorem:

$$f(x) = a(1, :) \cdot x^n + a(2, :) \cdot x^{n-1} + \dots + a(l-1, :) \cdot x^1 + a(l, :) \cdot x^0$$

gdzie „l” oznacza liczbę elementów wektora, „n” – stopień wielomianu.

Kod aproksymujący napisany w Matlabie:

Generacja macierzy potrzebnej do znalezienia funkcji aproksymującej metodą najmniejszych kwadratów:

```

%Definicja funkcji generującej macierz na podstawie wektora x:
function [A] = ex1_gen_data(x, n)
% x - wektor argumentów funkcji
% n - stopień wielomianu aproksymującego
[1, ~] = size(x);
A = zeros(1, n+1);
% Generowanie macierzy A:
for i=1:n+1
    if i == n+1
        A(:,i) = x.^0;
    else
        A(:,i) = x.^(n+1-i);
    end
end
end

```

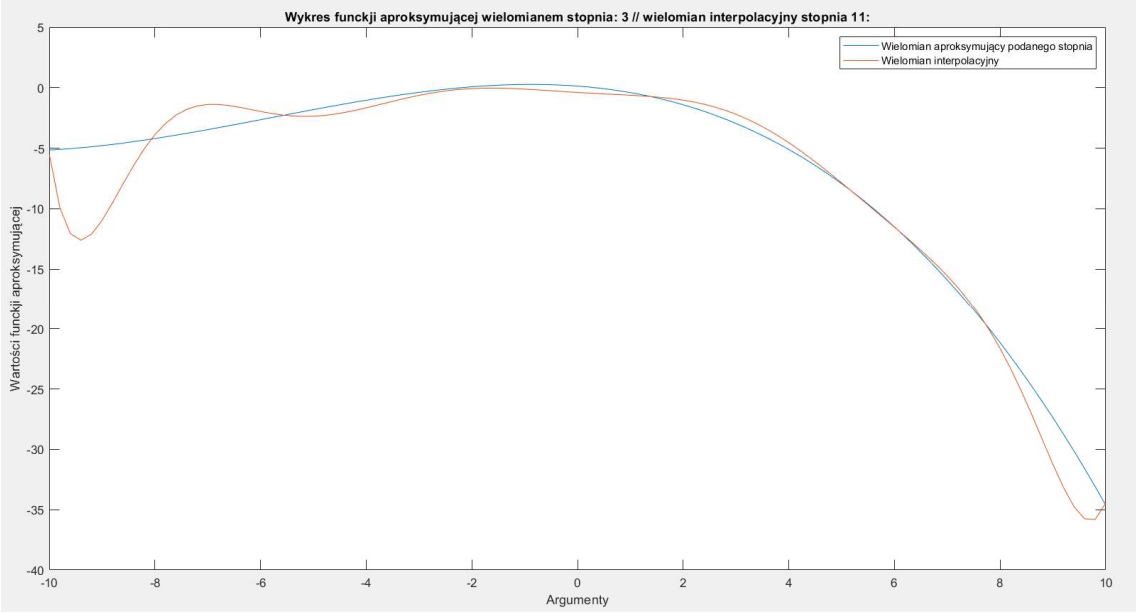
Funkcja aproksymująca:

```

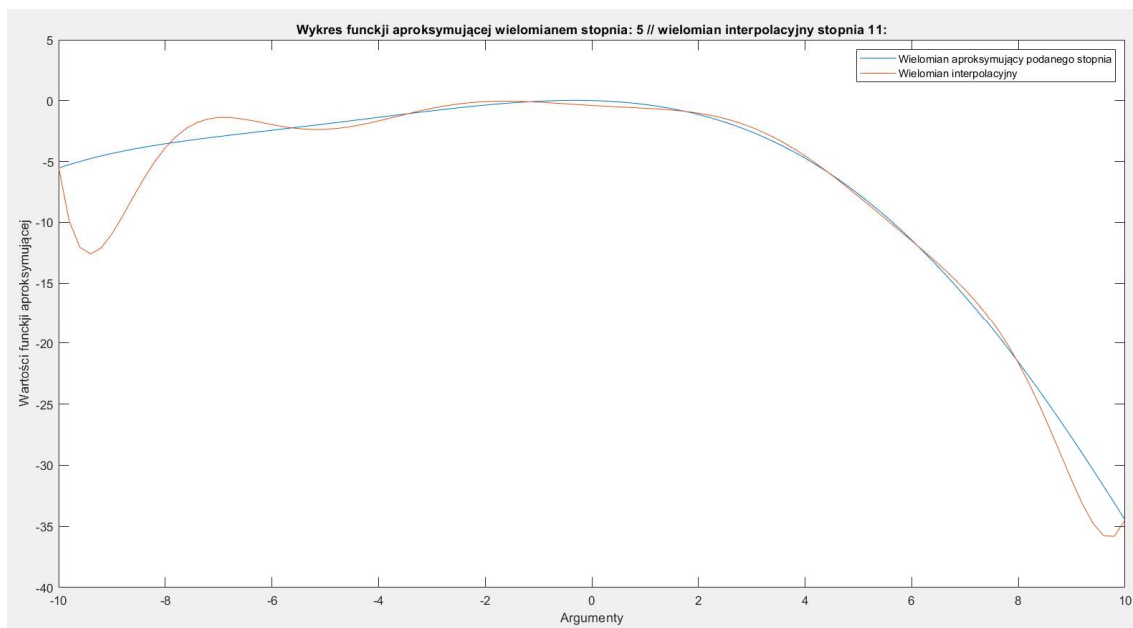
% Definicja procedury, jej parametrów wejściowych i wyjściowych:
function [yw] = qr_approx(C:\Users\Adam\Documents\MATLAB\qr_approximate.m)
% xp - argument, dla którego funkcja ma zwracać wartość
% n - stopień wielomianu aproksymującego
% x - wektor argumentów funkcji
% y - wektor wartości funkcji
% Uzyskanie macierzy A:
A = ex1_gen_data(x,n);
% Rozkład macierzy A na macierz Q i R:
[Q, R] = qr(A);
% Uzyskanie wektora współczynników wielomianu aproksymującego rozwiązując
% równanie  $QR \cdot x = y$ :
a = linsolve(Q * R, y);
yw = 0;
% Obliczenie wartości wielomianu aproksymującego dla zadanego punktu xp:
for i=1:n+1
    yw = yw + a(i,:)*(xp^(n+1-i));
end
end

```

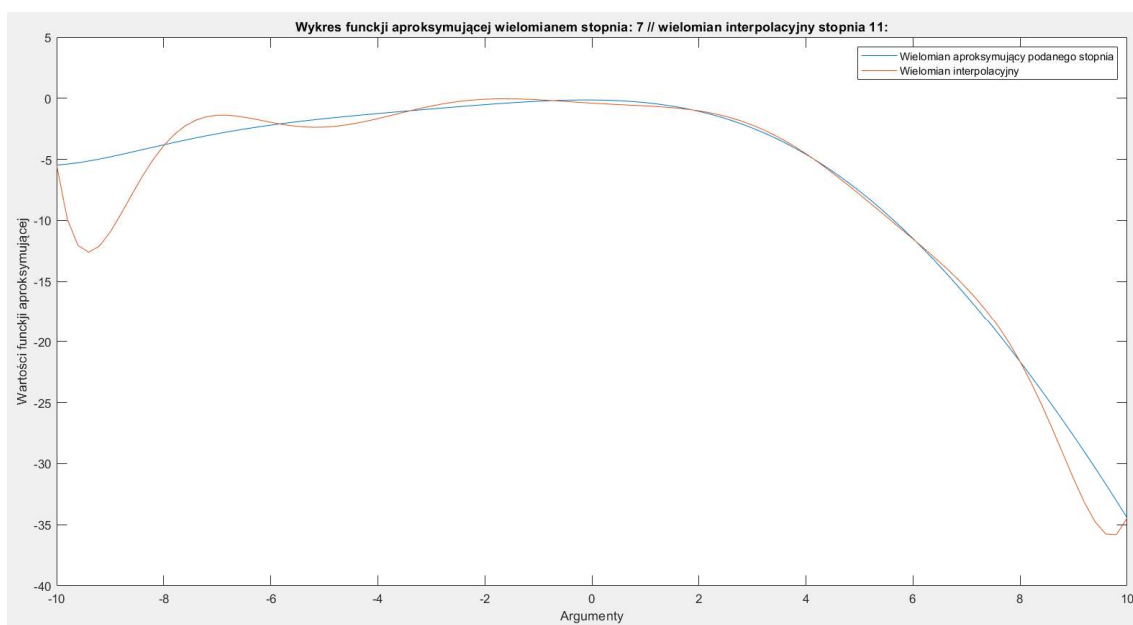
Wykresy poszczególnych funkcji aproksymujących jako wielomiany stopni: 3, 5, 7, 9, 10 (na wykresach umieszczono również wielomian interpolacyjny utworzony za pomocą macierzy Vandermonde'a):



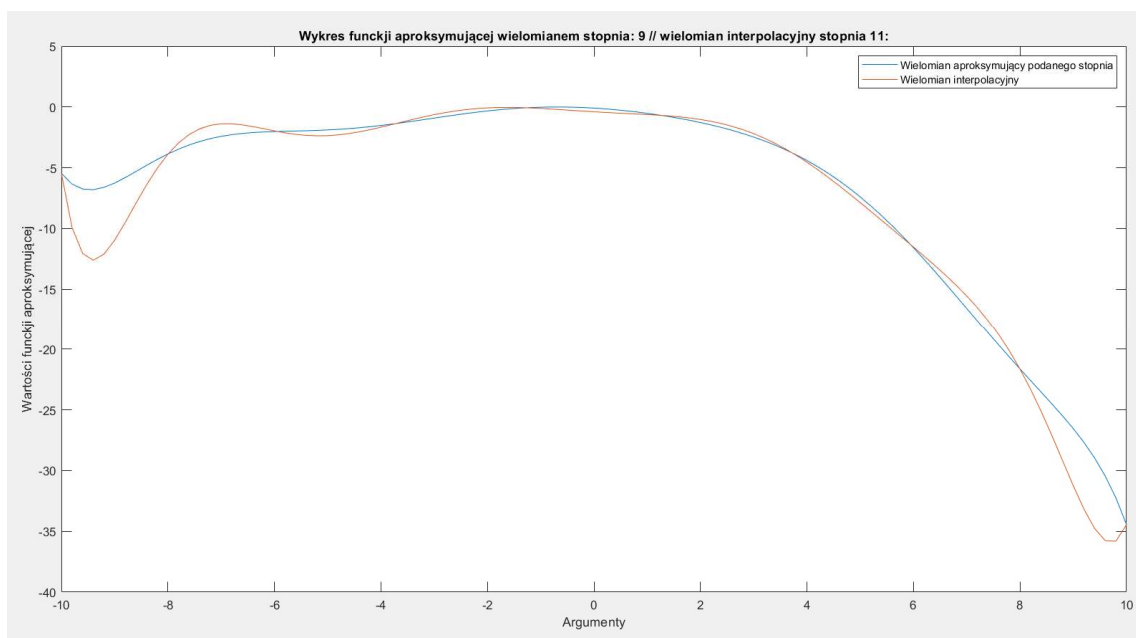
Argumenty	Wartości funkcji aproksymującej (n=3)
-10	-5,1537
-8	-4,2023
-6	-2,6468
-4	-1,0386
-2	0,0708
0	0,1303
2	-1,4117
4	-5,1065
6	-11,5054
8	-21,1599
10	-34,6213



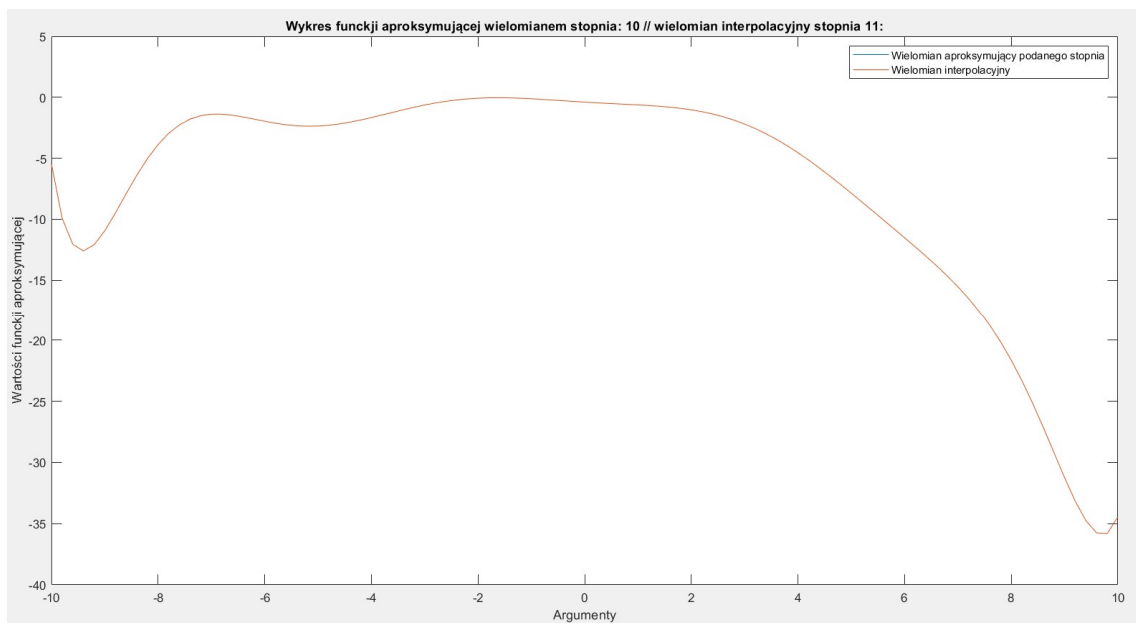
Argumenty	Wartości funkcji aproksymującej (n=5)
-10	-5,5437
-8	-3,5464
-6	-2,4341
-4	-1,3725
-2	-0,3664
0	0,0062
2	-1,1398
4	-4,7313
6	-11,4700
8	-21,5677
10	-34,4794



Argumenty	Wartości funkcji aproksymującej (n=7)
-10	-5,4697
-8	-3,8110
-6	-2,1981
-4	-1,2507
-2	-0,5212
0	-0,1380
2	-1,0716
4	-4,5935
6	-11,4969
8	-21,6492
10	-34,4453

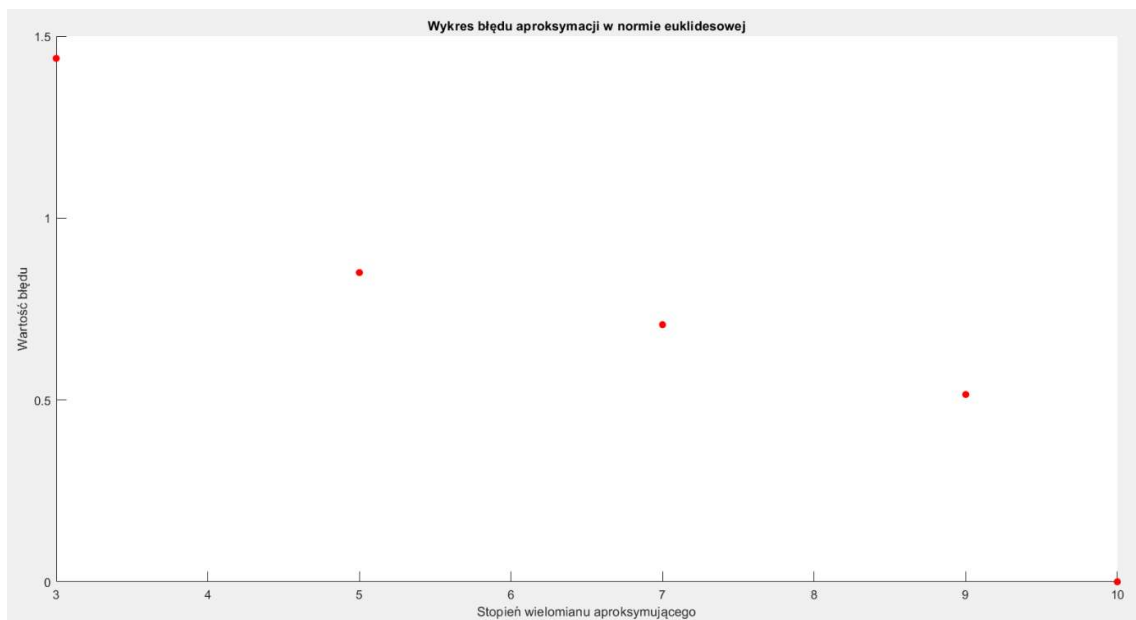


Argumenty	Wartości funkcji aproksymującej (n=9)
-10	-5,4618
-8	-3,8684
-6	-2,0238
-4	-1,5228
-2	-0,3280
0	-0,0952
2	-1,2819
4	-4,4045
6	-11,5819
8	-21,6297
10	-34,4470

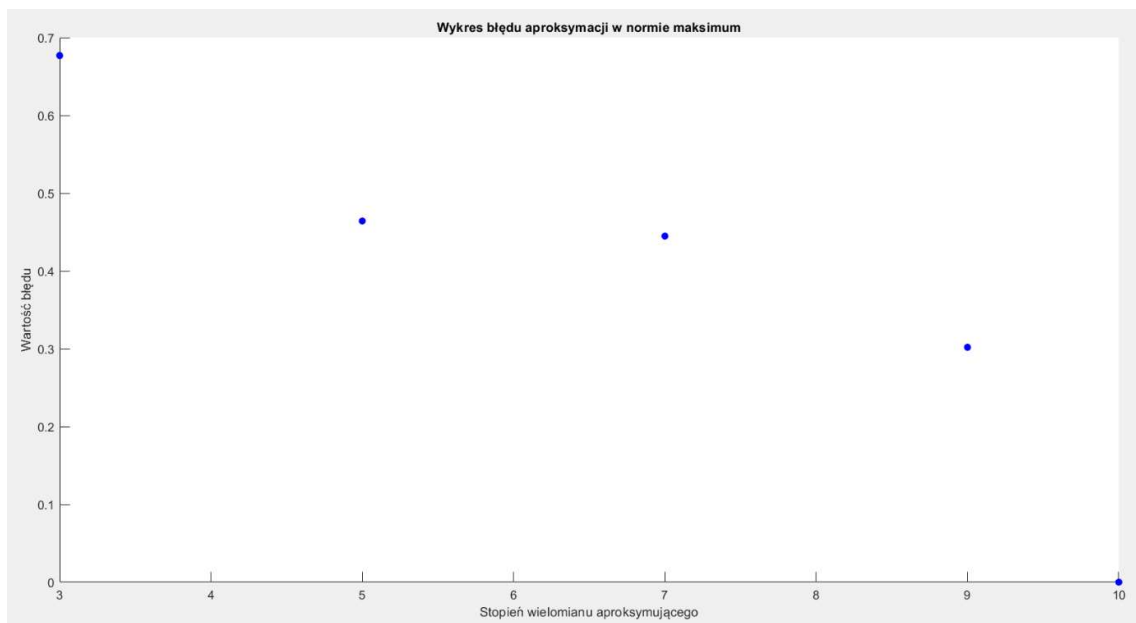


Argumenty	Wartości funkcji aproksymującej (n=10)
-10	-5,4606
-8	-3,8804
-6	-1,9699
-4	-1,6666
-2	-0,0764
0	-0,3971
2	-1,0303
4	-4,5483
6	-11,5280
8	-21,6417
10	-34,4458

Wykresy błędów aproksymacji w poszczególnych normach (kolejno: euklidesowej, maksimum) z różnicy wektorów wartości funkcji aproksymowanej i wartości funkcji aproksymującej:



Stopień wielomianu aproksymującego	Wartości błędu w normie: euklidesowej
3	1,4390
5	0,8501
7	0,7069
9	0,5150
10	3,98E-13



Stopień wielomianu aproksymującego	Wartości błędu w normie: euklidesowej
3	0,6769
5	0,4642
7	0,4448
9	0,3019
10	3,20E-13

Wnioski z zadania 1:

- błędy aproksymacji liczone w normie euklidesowej oraz w normie maksimum maleją wraz ze wzrostem stopnia wielomianu aproksymującego
- im większy stopień wielomianu aproksymującego tym jego wykres jest coraz bliższy wykresowi wielomianu interpolacyjnego
- wykres wielomianu stopnia 10 jest wręcz identyczny z wykresem wielomianu interpolacyjnego, błąd aproksymacji jest rzędu E-13
- dla wielomianów stopnia 5 i 7 wykresy są niemalże identyczne, gdyż różnica pomiędzy wartościami ich błędów jest wtedy najmniejsza

2. Dla danych z p. 1. znaleźć wielomian interpolacyjny korzystając z macierzy Vandermonde'a.

Zadanie 2 ma przebieg analogiczny tak jak zadanie 1 z tym wyjątkiem, że nie korzystamy już z rozkładu QR tylko rozwiązujemy układ równań z macierzą Vandermonde'a.

Tak jak w zadaniu 1 mamy wektor argumentów „x” i wektor wartości jakie przyjmuje ta funkcja dla poszczególnych argumentów:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{w-1} \\ x_w \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{w-1} \\ y_w \end{bmatrix}$$

gdzie w to liczba elementów w wektorze, identyczna dla wektorów „y” i „x”. „V” jest macierzą Vandermonde'a, jest to macierz kwadratowa o wymiarach $w \times w$.

$$V = \begin{bmatrix} 1 & x_1^1 & x_1^{w-2} & x_1^{w-1} & x_1^w \\ 1 & x_2^1 & x_2^{w-2} & x_2^{w-1} & x_2^w \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{w-1}^1 & x_{w-1}^{w-2} & x_{w-1}^{w-1} & x_{w-1}^w \\ 1 & x_w^1 & x_w^{w-2} & x_w^{w-1} & x_w^w \end{bmatrix}$$

Następnie wystarczy rozwiązać równanie:

$$V \cdot a = y$$

Aby znaleźć wielomian i jego wartość w konkretnym punkcie postępujemy tak jak w zadaniu 1.

Kod tworzący wielomian interpolacyjny napisany w Matlabie:

Generowanie macierzy Vandermonde'a:

```

% Definicja funkcji generującej macierzy Vandermonde'a na podstawie wektora
% argumentów funkcji aproksymowanej:
function [V] = ex2_gen_data(x)
[1, ~] = size(x);
V = zeros(1, 1);
% Generowanie macierzy Vandermonde'a:
for i=0:l-1
    if i == 0
        V(:,i+1) = x.^0;
    else
        V(:,i+1) = x.^(i);
    end
end
end

```

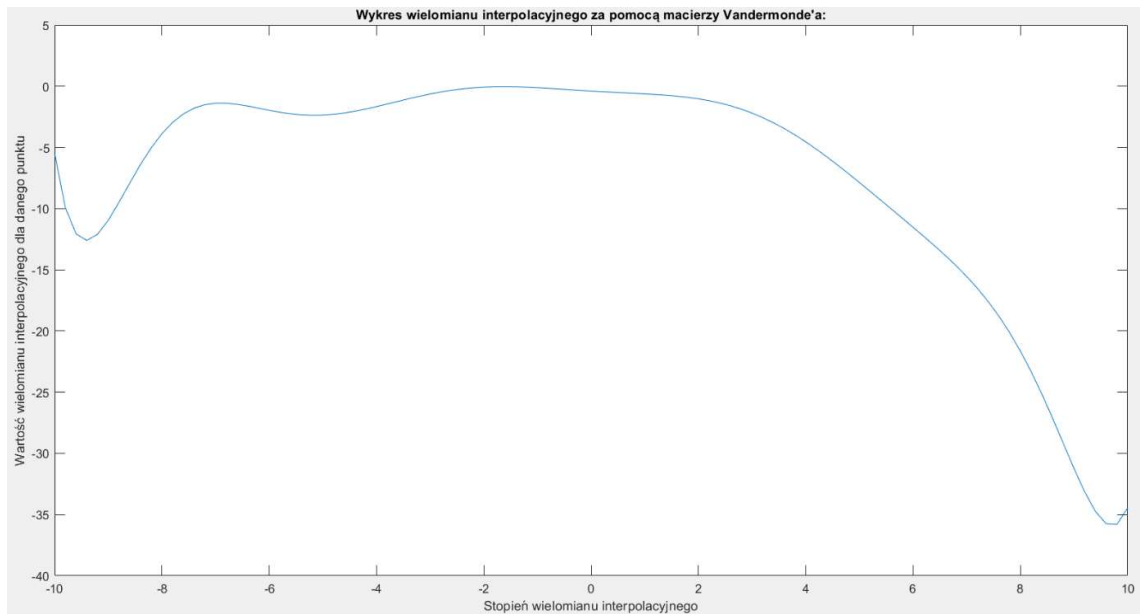
Wielomian interpolacyjny:

```

% Definicja procedury, jej parametrów wejściowych i wyjściowych:
function [yw] = vandermonde(a, x, y)
% Uzyskanie macierzy Vandermonde'a:
[V] = ex2_gen_data(x);
% Rozwiązanie układu równań  $Vx=y$ :
xw = linsolve(V,y);
[~,k] = size(V);
yw = 0;
% Wyznaczenie wartości wielomianu interpolacyjnego dla zadanego argumentu:
for i=1:k
    yw = yw + xw(i,:)*(a^(i-1));
end

```

Wykres wielomianu interpolacyjnego:



Argumenty	Wartości wielomianu interpolacyjnego
-10	-5,4606
-8	-3,8804
-6	-1,9699
-4	-1,6666
-2	-0,0764
0	-0,3971
2	-1,0303
4	-4,5483
6	-11,5280
8	-21,6417
10	-34,4458

Błędy interpolacji w normie odpowiednio: euklidesowej i maksimum wynoszą:

$$2.6380e - 13$$

$$1.6365e - 13$$

Wnioski z zadania 2:

- wielomian interpolacyjny jest wielomianem stopnia $n=11$, stopień wielomianu jest równy liczbie elementów wektora danych
- wielomian interpolacyjny wręcz idealnie przybliża funkcję pierwotną
- błąd wielomianu interpolacyjnego jest rzędu $E-13$

Wnioski ogólne:

Można zatem wywnioskować, że istnieją dwa takie stopnie wielomianu, które aproksymują nasze dane niemalże idealnie z błędem rzędu $E-13$. Dla naszych danych są to wartości $n = \{10, 11\}$. Ogólnie rzecz biorąc im większy stopień wielomianu tym wartości funkcji aproksymującej są coraz bliższe wartościom funkcji pierwotnej. Jeśli chodzi o efektywność metod to obie tak samo dokładnie aproksymują nasze dane pod warunkiem, jeżeli w pierwszej metodzie będziemy przybliżać funkcję wielomianem stopnia 10 lub 11. Jednak metoda interpolacyjna macierzą Vanermonde'a jest bardziej intuicyjna i łatwiejsza w implementacji.