

[MNUM] Metody Numeryczne
Semestr 2021L
Projekt 2
Sprawozdanie

Treść.

Dla następujących danych pomiarowych (próbek):

i	x_i	y_i
0	-5	-79,1639
1	-4	-40,7900
2	-3	-18,7814
3	-2	-6,3530
4	-1	-0,4392
5	0	0,8270
6	1	0,0585
7	2	-1,7477
8	3	-3,4384
9	4	-6,3580
10	5	-9,3875

Tabela 1. Dane pomiarowe

Metodą najmniejszych kwadratów należy wyznaczyć funkcję wielomianową $y=f(x)$ najlepiej aproksymującą te dane (proszę przetestować wielomiany różnych stopni). W sprawozdaniu proszę przedstawić na rysunku otrzymaną funkcję na tle danych. Do rozwiązania zadania najmniejszych kwadratów proszę wykorzystać:

- układ równań normalnych
- układ równań liniowych z macierzą R wynikającą z rozkładu QR

macierzy układu równań problemu. Proszę obliczyć błąd aproksymacji w dwóch normach: euklidesowej oraz Czebyszewa (maksimum).

Uwagi:

- rysowaną funkcję proszę próbować 10 razy częściej niż dane
- dane są obarczone pewnym błędem (szumem pomiarowym).

Rozwiązanie.

W zadaniu mamy do czynienia z aproksymacją skończonego zbioru punktów (dyskretną). Z faktu, iż dane pomiarowe obarczone są pewnymi błędami (szumem pomiarowym), wynika, że będziemy nasze dane aproksymować, a nie interpolować.

a) układ równań normalnych

Nasz zbiór danych będziemy aproksymować wielomianem $W(x) = \sum_{i=0}^n a_i \cdot x^i$ (stopnia $n \geq 0$), przy pomocy liniowo niezależnych funkcji bazowych $\phi_i(x) = x^i$ ($i = 0, 1, \dots, n$). Oczywiście, poszukujemy współczynników a_i ($i = 0, 1, \dots, n$).

$$\text{Jeśli przyjmiemy } \mathbf{A} = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \dots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_n(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_n(x_N) \end{bmatrix} \text{ (w naszym przypadku } N = 10),$$

$$\mathbf{a} = [a_0 \quad a_1 \quad \dots \quad a_n]^T,$$

$$\mathbf{y} = [y_0 \quad y_1 \quad \dots \quad y_N]^T,$$

to układ równań normalnych przybiera postać $\mathbf{A}^T \mathbf{A} \mathbf{a} = \mathbf{A}^T \mathbf{y}$ [1].

Warto dodać, że z założenia o liniowej niezależności funkcji bazowych wynika pełen rząd macierzy \mathbf{A} . Z kolei pełen rząd macierzy \mathbf{A} implikuje nieosobliwość macierzy Grama $\mathbf{A}^T \mathbf{A}$ [1].

W naszym konkretnym przypadku (naturalna baza wielomianów), możemy zapisać powyższy układ równań inaczej.

$$\text{Zdefiniujmy } \mathbf{G} = \begin{bmatrix} g_{00} & g_{10} & \dots & g_{n0} \\ g_{01} & g_{11} & \dots & g_{n1} \\ g_{02} & g_{12} & \dots & g_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{0n} & g_{1n} & \dots & g_{nn} \end{bmatrix} \text{ oraz } \boldsymbol{\rho} = [\rho_0 \quad \rho_1 \quad \dots \quad \rho_n]^T,$$

gdzie:

$$g_{ik} \stackrel{\text{def}}{=} \sum_{j=0}^N (x_j)^{i+k} \quad (i, k = 0, 1, \dots, n),$$

$$\rho_k \stackrel{\text{def}}{=} \sum_{j=0}^N y_j \cdot (x_j)^k \quad (k = 0, 1, \dots, n).$$

Wówczas układ równań normalnych przybiera postać $\mathbf{Ga} = \boldsymbol{\rho}$ [1]. (1)

Przekształcając układ równań (1), otrzymujemy $\mathbf{a} = \mathbf{G}^{-1}\boldsymbol{\rho}$. (2)

Moja implementacja uwzględnia wzór (2) oraz definicję macierzy \mathbf{G} i wektora $\boldsymbol{\rho}$.

Listing solwera (solver1.m):

```
function a = solver1(n, x, y, N)
% Znajdowanie współczynników wielomianu
% aproksymującego stopnia n: W(t) = a_1 + a_2*t + ... + a_(n+1) * t^n,
% za pomocą metody najmniejszych kwadratów (układ równań normalnych)

% n - stopień poszukiwanego wielomianu aproksymującego
% a - poszukiwane współczynniki wielomianu aproksymującego
% x, y - wektory danych pomiarowych (probek)
% N - liczba punktów pomiarowych (probek)

% Generacja macierzy G
G = zeros( n+1, n+1 );

for i=1:n+1
    % Elementy poniżej diagonalii
    for k=1:i-1
        G(i,k) = G(k,i);
    end

    % Pozostałe elementy
    for k=i:n+1
        for j=1:N
            G(i,k) = G(i,k) + x(j)^(i+k-2);
        end
    end
end

% Generacja wektora ro
ro = zeros(n+1,1);

for k=1:n+1
    for j=1:N
        ro(k) = ro(k) + y(j)*x(j)^(k-1);
    end
end

% Obliczenie wektora a:
% nieznanymi współczynnikami wielomianu
% aproksymującego (współczynniki od wyrazu
% wolnego do wyrazu przy najwyższej
% potędze zmiennej)
a = inv(G)*ro;

end
```

Listing skryptu wywołującego (finalScript1.m):

```
% a) Układ rownan normalnych

clear;
clc;

% Dane pomiarowe (probki)
x = [-5:5]';
y = [-79.1639 -40.7900 -18.7814 -6.3530 -0.4392 ...
     0.8270 0.0585 -1.7477 -3.4384 -6.3580 -9.3875]';

% Liczba danych pomiarowych (probek)
N = length(x);

% Argumenty do narysowania wykresow wielomianow aproksymujacych
args = [-5:0.1:5]';

% Stopnie wielomianow aproksymujacych
n = 0:N-1;

% czasy dzialania solwera dla roznych stopni wielomianow aproksymujacych
T = zeros( length(n), 1 );

% Bledy aproksymacji dla normy maksimum
error_max = zeros(length(n),1);

% Bledy aproksymacji dla normy euklidesowej
error_Euclid = zeros(length(n),1);

for k=1:length(n) % dla kazdego stopnia wielomianu aproksymujacego

    % Wyznaczenie wspolczynnikow
    tic;
    a = solver1( n(k), x, y, N );
    T(k) = toc;

    % Wypisanie wspolczynnikow wielomianu aproksymujacego
    fprintf( '\nWspolczynniki wielomianu stopnia %s:', num2str( n(k) ) );
    disp(a');

    % Odwrocenie kolejnosci wspolczynnikow
    a = flip(a);

    % Obliczenie wektora F:
    % wartosci wielomianu aproksymujacego
    F = polyval(a,x);

    % Bledy dla wielomianu stopnia k
    error_max(k) = norm(F-y,Inf);
    error_Euclid(k) = norm(F-y,2);
```

```

% Wykres wielomianu aproksymujacego stopnia k
figure(k);
hold on;
plot( args, polyval(a,args) );

for i=1:N
    plot( x(i),y(i), 'r.', 'MarkerSize', 15, 'Color', 'Red' );
end

title(['Wielomian stopnia ', num2str( n(k) ), ' na tle danych']);
xlabel('x');
ylabel('y');
legend(['Wielomian stopnia ', num2str( n(k) )], 'Probka danych',
'Location', 'South');

grid on;
hold off;

end

% Przedstawienie czasow w ms
T = T .* 1000;

% Wykres czasow dzialania solwera
hold on;
figure( length(n)+1 );
plot( n, T, '-r.', 'MarkerSize', 25 );
grid on;

title('Czasy dzialania solwera w zaleznosci od stopnia wielomianu
aproksymujacego');
xlabel('Stopien wielomianu aproksymujacego n');
ylabel('Czas dzialania solwera [ms]');

hold off;

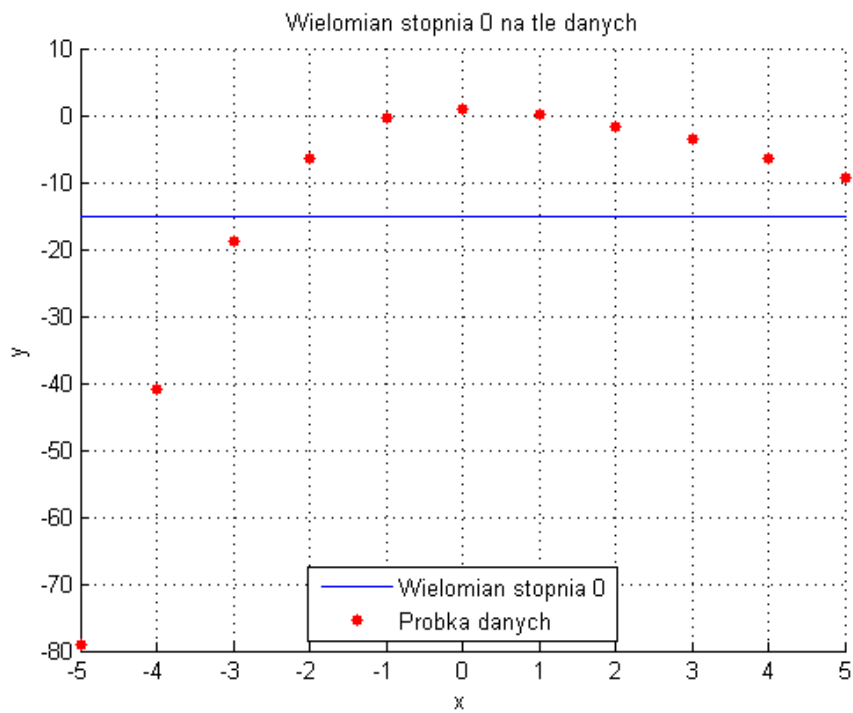
% Wykres bledow aproksymacji
hold on;
figure( length(n)+2 );
plot( n, error_max, '-o', n ,error_Euclid, '-o' );
grid on;

title('Normy bledow aproksymacji w zaleznosci od stopnia wielomianu
aproksymujacego');
xlabel('Stopien wielomianu aproksymujacego n');
ylabel('Norma bledu aproksymacji');

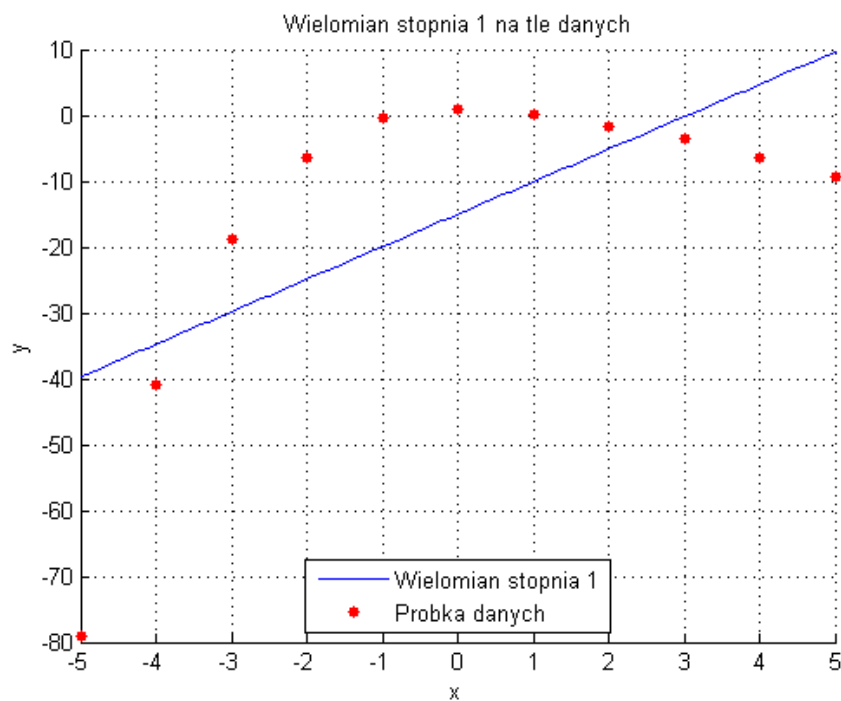
legend('Blad aproksymacji w normie maksimum','Blad aproksymacji w normie
euklidesowej', 'Location', 'North');
hold off;

```

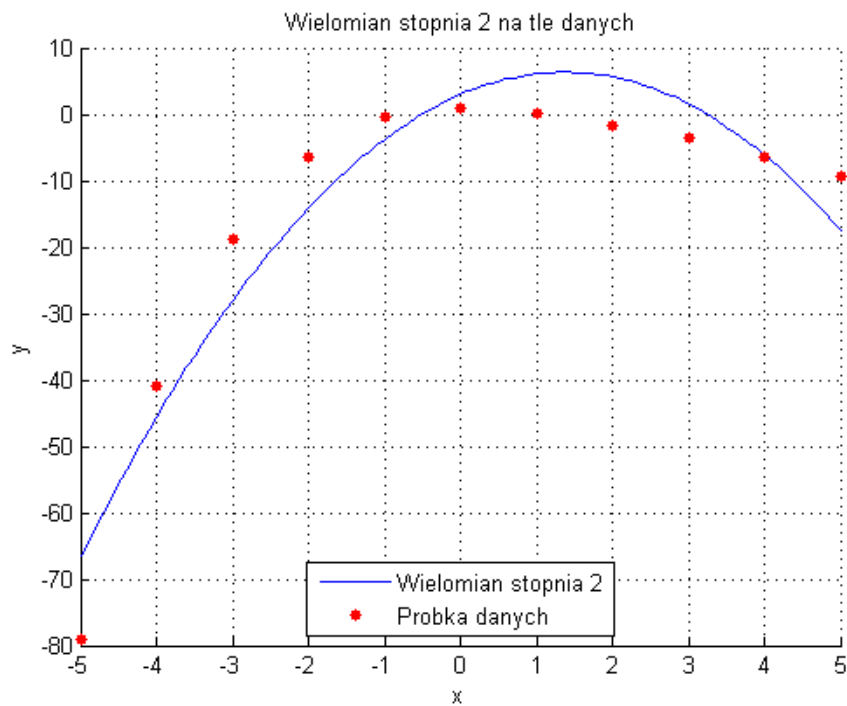
Wykresy:



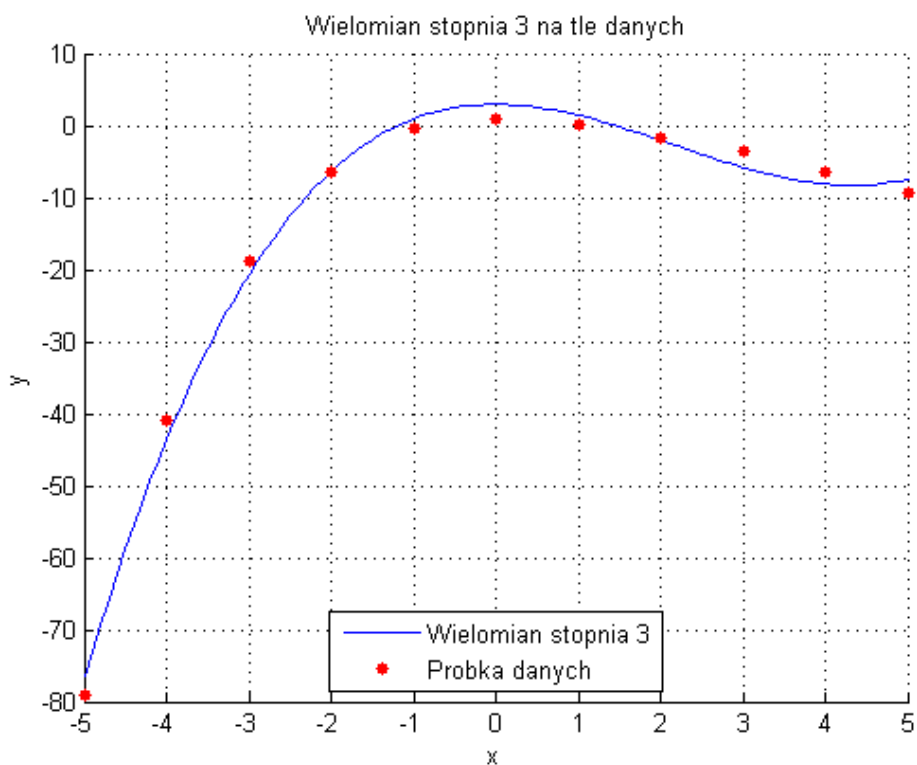
Wykres 1. Wielomian aproksymujący stopnia 0 oraz dane pomiarowe ($\mathbf{a} = [-15.0521]^T$)



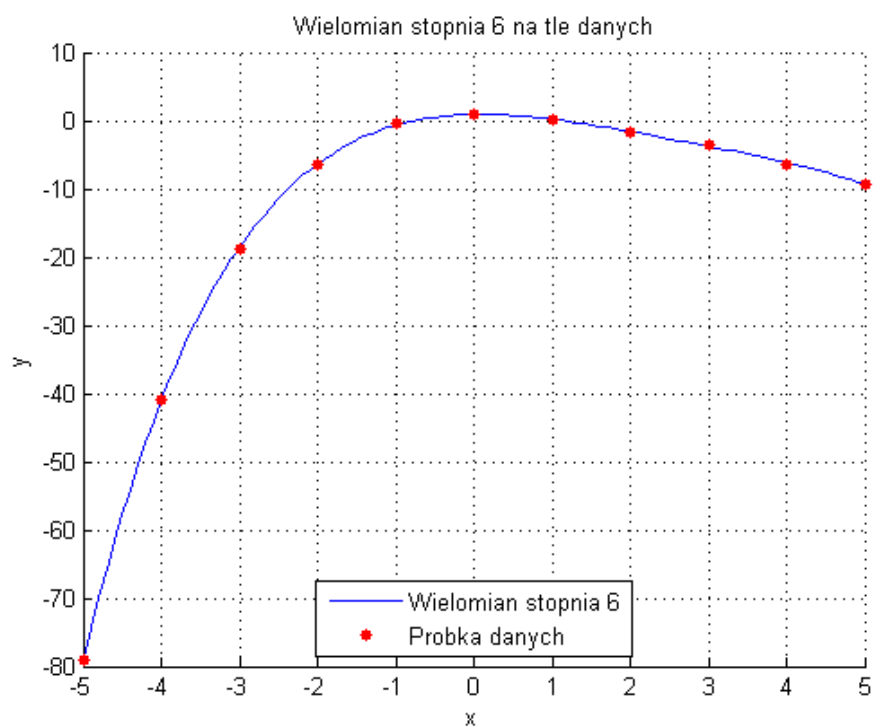
Wykres 2. Wielomian aproksymujący stopnia 1 oraz dane pomiarowe ($\mathbf{a} = [-15.0521 \ 4.9304]^T$)



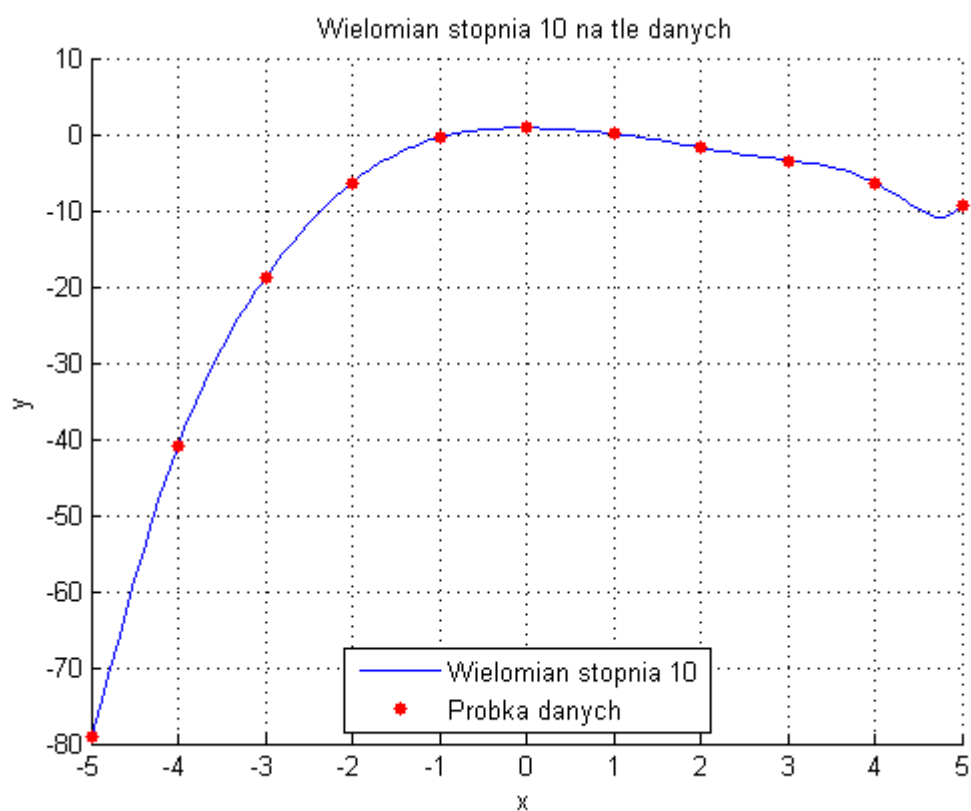
Wykres 3. Wielomian aproksymujący stopnia 2 oraz dane pomiarowe
 $(\mathbf{a} = [2.9569 \quad 4.9304 \quad -1.8009]^T)$



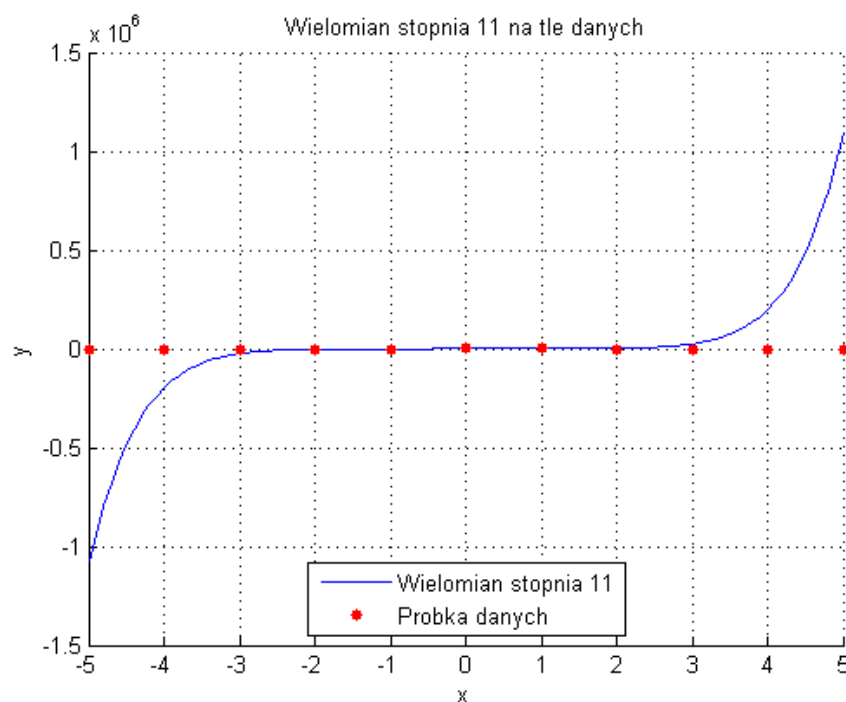
Wykres 4. Wielomian aproksymujący stopnia 3 oraz dane pomiarowe
 $(\mathbf{a} = [2.9569 \quad -0.0357 \quad -1.8009 \quad 0.2790]^T)$



Wykres 5. Wielomian aproksymujący stopnia 6 oraz dane pomiarowe
 $(\mathbf{a} = [0.8901 \ 0.2028 \ -1.1295 \ 0.2394 \ -0.0210 \ 0.0013 \ -0.0002]^T)$

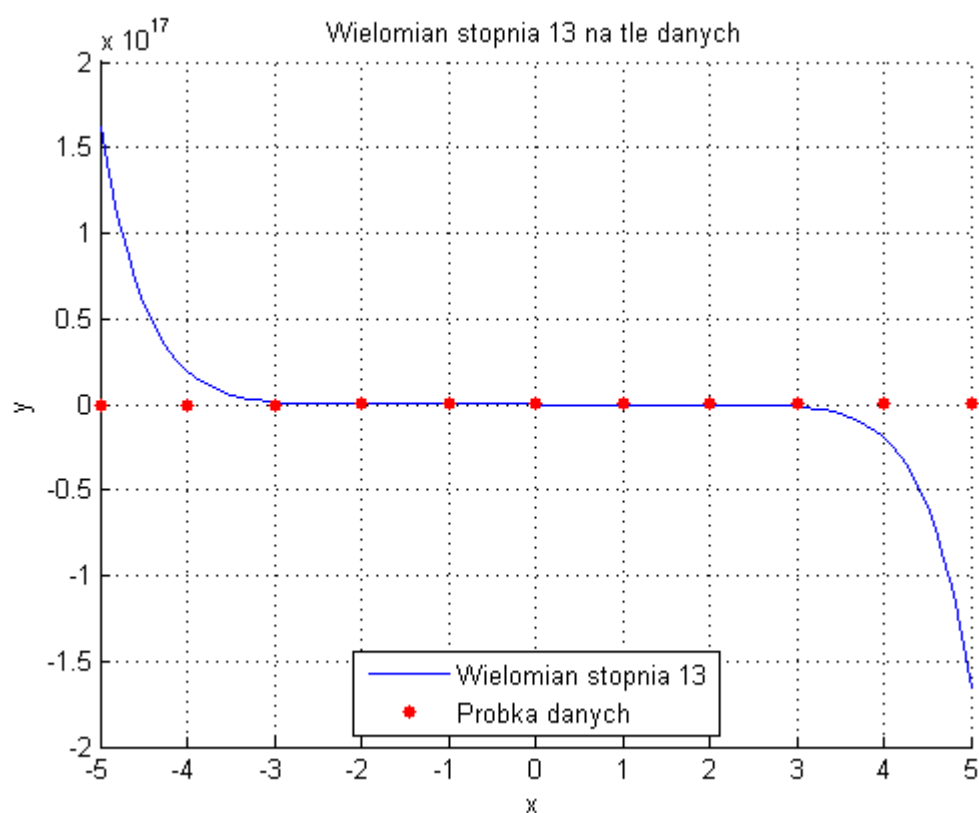


Wykres 6. Wielomian aproksymujący stopnia 10 oraz dane pomiarowe
 $(\mathbf{a} = [0.8270 \ -0.0547 \ -0.9013 \ 0.3030 \ -0.1318 \ 0.0009 \ 0.0167 \ -0.0004 \ -0.0010 \ 0.0000]^T)$



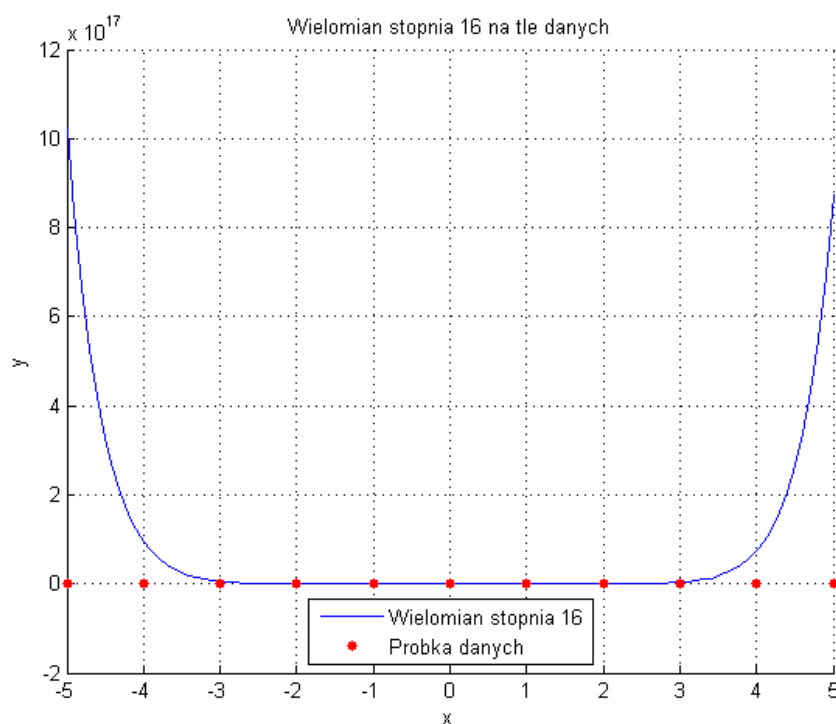
Wykres 7. Wielomian aproksymujący stopnia 11 oraz dane pomiarowe

($a = [0.8270 \quad 0 \quad -0.9013 \quad 5.9219 \quad -0.1318 \quad 19.3828 \quad 0.0167 \quad 6.5908 \quad -0.0010 \quad 0.2660 \quad 0.0000 \quad 0.0000]^T$)

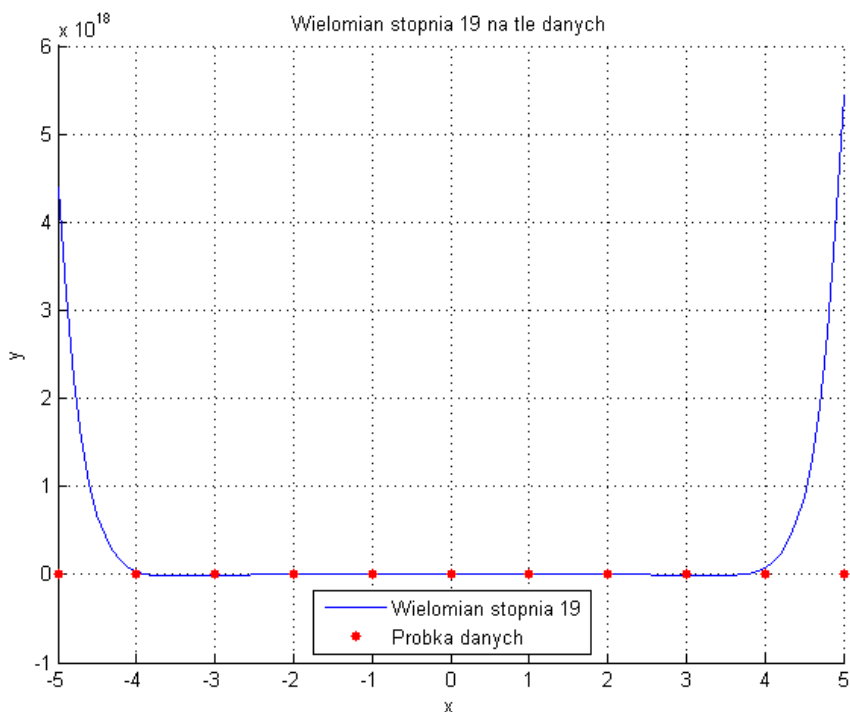


Wykres 8. Wielomian aproksymujący stopnia 13 oraz dane pomiarowe

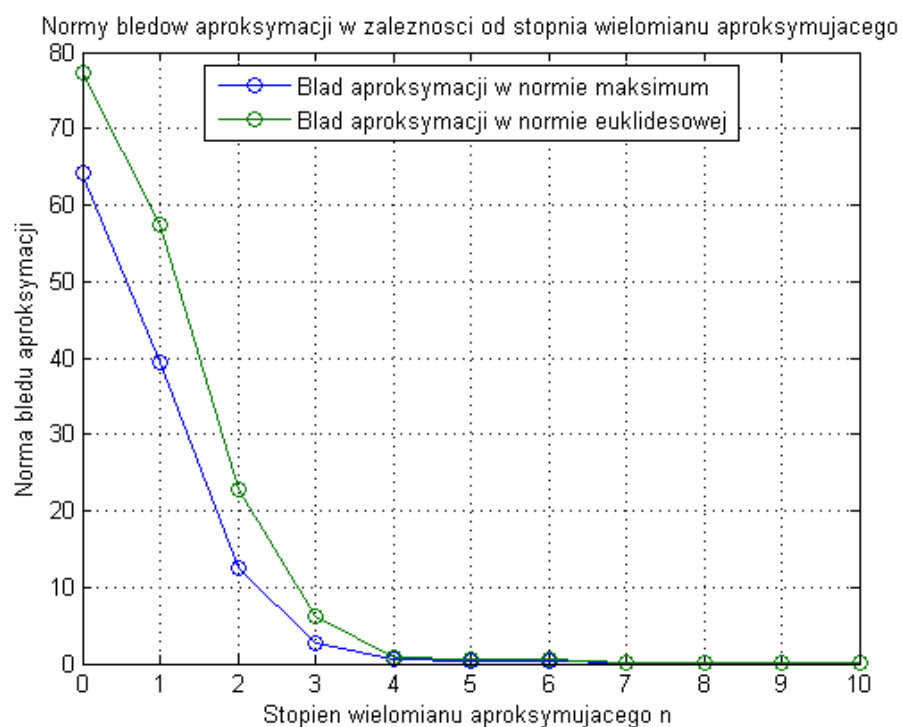
($a = 1e11 \cdot [0.0000 \quad 0.0000 \quad -0.0000 \quad -0.0198 \quad -0.0000 \quad -0.6394 \quad -0.0000 \quad -1.3856 \quad -0.0000 \quad -0.4048 \quad -0.0000 \quad -0.0154 \quad 0.0000 \quad 0]^T$)



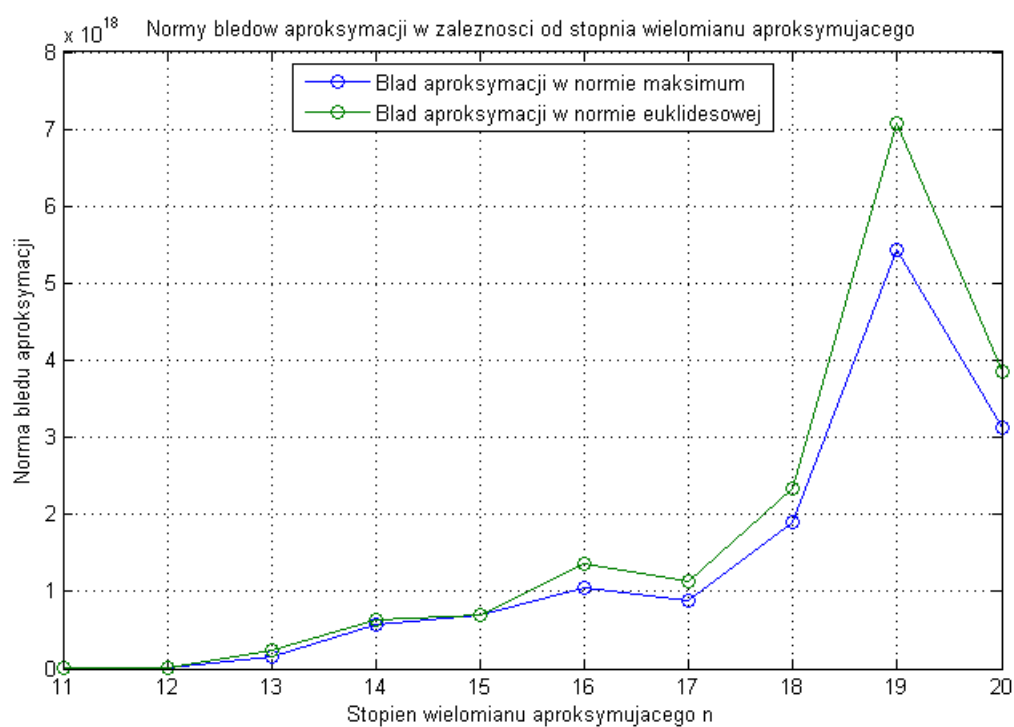
Wykres 9. Wielomian aproksymujący stopnia 16 oraz dane pomiarowe
 $(a = 1e11 \cdot [0.0000 \quad -0.0000 \quad -0.0000 \quad -0.0242 \quad 0.0269 \quad -0.7534 \quad 0.8752 \quad -1.4167 \quad 1.7601 \quad -0.2731 \quad 0.4164 \quad -0.0003 \quad 0.0116 \quad -0.0002 \quad 0.0003 \quad -0.0000 \quad 0.0000]^T)$



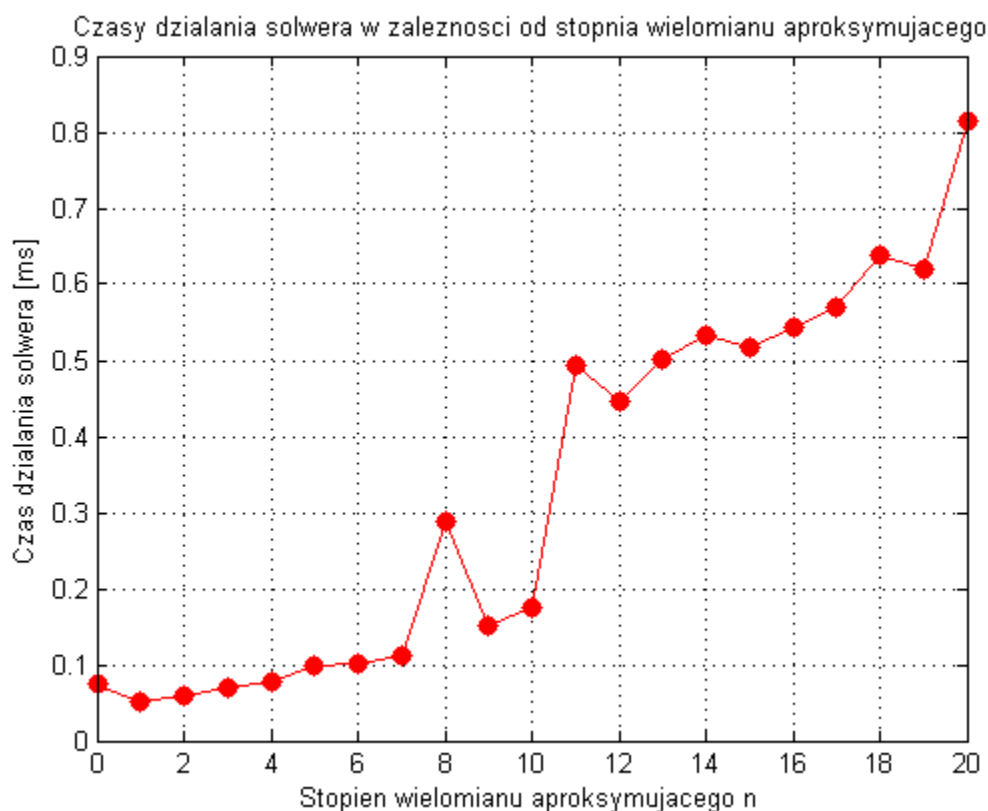
Wykres 10. Wielomian aproksymujący stopnia 19 oraz dane pomiarowe
 $(a = 1e11 \cdot [0.0000 \quad -0.0000 \quad 0.0000 \quad 0.0223 \quad -0.0729 \quad 0.5926 \quad -2.4874 \quad 0.5005 \quad 5.6254 \quad 0.0655 \quad 1.3744 \quad 0.0184 \quad 0.0335 \quad 0.0001 \quad 0.0022 \quad 0.0001 \quad 0.0003 \quad 0.0000 \quad -0.0000 \quad 0.0000]^T)$



Wykres 11. Błędy aproksymacji dla różnych wielomianów aproksymujących



Wykres 12. Błędy aproksymacji dla różnych wielomianów aproksymujących



Wykres 13. Czasy działania solwera dla różnych wielomianów aproksymujących

Komentarz:

Na podstawie analizy wykresów 1. - 6. możemy wnioskować, że wielomian aproksymujący coraz lepiej „dopasowuje” się do próbek danych (sytuacja ta obowiązuje dla stopni wielomianów $n \leq 10$).

Z kolei, na podstawie analizy wykresów 7. - 10. możemy wnioskować, że wielomian aproksymujący jest „niedopasowany” się do próbek danych (sytuacja ta obowiązuje dla stopni wielomianów $n \geq 11$). Sytuację tę można tłumaczyć coraz gorszym uwarunkowaniem macierzy Grama układu równań normalnych wraz ze wzrostem stopnia wielomianu aproksymującego n .

Na podstawie wykresu 11. możemy wnioskować, iż błąd aproksymacji (niezależnie od tego, czy mamy na myśli normę maksimum, czy też normę euklidesową) maleje wraz ze wzrostem stopnia wielomianu aproksymującego – dzieje się tak aż do stopnia wielomianu $n = 10$.

Z kolei, na podstawie wykresu 12. możemy wnioskować, iż błąd aproksymacji (niezależnie od tego, czy mamy na myśli normę maksimum, czy też normę euklidesową) osiąga bardzo duże wartości dla wielomianów stopnia $n \geq 11$ (niemal w całym zakresie stopni wielomianów obserwujemy wzrost błędu aproksymacji).

Analizując wykres 13., możemy zaobserwować tendencję wzrostową czasu działania solwera w zależności od stopnia n wielomianu aproksymującego. Warto nadmienić, iż maksymalny czas nie przekracza 1 ms, zatem możemy uznać, że wyniki dostajemy szybko.

b) układ równań liniowych z macierzą R wynikającą z rozkładu QR macierzy układu równań problemu

Przypomnijmy, że układ równań normalnych ma następującą postać: $\mathbf{A}^T \mathbf{A} \mathbf{a} = \mathbf{A}^T \mathbf{y}$,

gdzie:

$$\mathbf{A} = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \dots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_n(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_n(x_N) \end{bmatrix} \text{ (w naszym przypadku } N = 10),$$

$$\mathbf{a} = [a_0 \quad a_1 \quad \dots \quad a_n]^T,$$

$$\mathbf{y} = [y_0 \quad y_1 \quad \dots \quad y_N]^T.$$

Może się zdarzyć, że macierz Grama $\mathbf{A}^T \mathbf{A}$ jest źle uwarunkowana – wówczas stosuje się rozkład QR macierzy \mathbf{A} [1].

W języku MATLAB istnieje komenda $[Q, R] = \text{qr}(A, 0)$, która dla macierzy $\mathbf{A}_{m \times n}$ zwraca macierze Q oraz R , wynikające z rozkładu QR macierzy A .

1) Jeśli $m > n$, wówczas otrzymujemy rozkład „wąski” QR, tj. macierz ortogonalną $\mathbf{Q}_{m \times n}$ oraz macierz trójkątną górną (o niezerowych elementach na diagonalu) $\mathbf{R}_{n \times n}$.

2) Jeśli $m \leq n$, wówczas otrzymujemy rozkład pełny QR, tj. macierz ortogonalną $Q_{m \times m}$ oraz macierz trójkątną górną $R_{m \times n}$.
 Więcej informacji można znaleźć w dokumentacji MATLAB-a, wpisując hasło 'qr'.

Wyprowadźmy wzór na układ równań liniowych w przypadku 1).

Układ równań normalnych ma postać $A^T A a = A^T y$.

Pamiętając, że $A_{(N+1) \times (n+1)} = Q_{(N+1) \times (n+1)} \cdot R_{(n+1) \times (n+1)}$,
 zapisujemy układ równań następująco: $R^T Q^T Q R a = R^T Q^T y$.

Dalej, uwzględniając ortogonalność macierzy Q (tj. $Q^T \cdot Q = I$)
 oraz nieosobliwość macierzy R , mamy poniższy układ równań: $R a = Q^T y$.
 Stąd $a = R^{-1} Q^T y$. (3)

Wyprowadźmy wzór na układ równań liniowych w przypadku 2).

Układ równań normalnych ma postać $A^T A a = A^T y$.

Pamiętając, że $A_{(N+1) \times (n+1)} = Q_{(N+1) \times (N+1)} \cdot R_{(N+1) \times (n+1)}$,
 zapisujemy układ równań następująco: $R^T Q^T Q R a = R^T Q^T y$.

Dalej, uwzględniając ortogonalność macierzy Q (tj. $Q^T \cdot Q = I$),
 mamy poniższy układ równań: $R^T R a = R^T Q^T y$.
 Stąd $a = (R^T R)^{-1} R^T Q^T y$ (z nieosobliwości macierzy Grama $A^T A$
 wynika nieosobliwość macierzy $R^T R$).
 Zauważmy, że skoro $A = QR$, to $A^T = R^T Q^T$.
 Powyższa obserwacja pozwala nam uprościć rozwiązanie: $a = (R^T R)^{-1} A^T y$. (4)

Zaimplementowałem przypadki opisane zależnościami (3) oraz (4).

Listing solwera (solver2.m):

```
function a = solver2(n, x, y, N)
% Znajdowanie współczynników wielomianu
% aproksymującego stopnia n:  $W(t) = a_1 + a_2 t + \dots + a_{n+1} t^n$ ,
% za pomocą metody najmniejszych kwadratów (układ równań liniowych z
% macierzą R)

% n - stopień poszukiwanego wielomianu aproksymującego
% a - poszukiwane współczynniki wielomianu aproksymującego
% x, y - wektory danych pomiarowych (probek)
% N - liczba punktów pomiarowych (probek)

% Generacja macierzy A
A = zeros( N, n+1 );

for i=1:N
    for j=1:n+1
        A(i,j) = x(i)^(j-1);
    end
end

% Wyznaczenie macierzy Q, R (rozkład QR macierzy A)
[Q, R] = qr(A,0);

% Obliczenie wektora a:
% nieznanych współczynników wielomianu
% aproksymującego (współczynniki od wyrazu
% wolnego do wyrazu przy najwyższej
% potędze zmiennej x)
if n+1 < N % rozkład wąski QR
    a = inv(R) * Q.' * y;
else % rozkład pełny QR
    a = inv(R.' * R) * A.' * y;
end

end
```

Listing skryptu wywołującego (finalScript2.m):

```
% b) Układ rownan liniowych z macierza R

clear;
clc;

% Dane pomiarowe (probki)
x = [-5:5]';
y = [-79.1639 -40.7900 -18.7814 -6.3530 -0.4392 ...
     0.8270 0.0585 -1.7477 -3.4384 -6.3580 -9.3875]';

% Liczba danych pomiarowych (probek)
N = length(x);

% Argumenty do narysowania wykresow wielomianow aproksymujacych
args = [-5:0.1:5]';

% Stopnie wielomianow aproksymujacych
n = 0:N-1;

% Czasy dzialania solwera dla roznych stopni wielomianow aproksymujacych
T = zeros( length(n), 1 );

% Bledy aproksymacji dla normy maksimum
error_max = zeros(length(n),1);

% Bledy aproksymacji dla normy euklidesowej
error_Euclid = zeros(length(n),1);

for k=1:length(n)

    % Wyznaczenie wspolczynnikow
    tic;
    a = solver2( n(k), x, y, N );
    T(k) = toc;

    % Wypisanie wspolczynnikow wielomianu aproksymujacego
    fprintf( '\nWspolczynniki wielomianu stopnia %s:', num2str( n(k) ) );
    disp(a');

    % Odwrocenie kolejnosci wspolczynnikow
    a = flip(a);

    % Obliczenie wektora F:
    % wartosci wielomianu aproksymujacego
    F = polyval(a,x);

    % Bledy dla wielomianu stopnia k
    error_max(k) = norm(F-y,Inf);
```



```

error_Euclid(k) = norm(F-y,2);

% Wykres wielomianu aproksymujacego stopnia k
figure(k);
hold on;
plot( args, polyval(a,args) );

for i=1:N
    plot( x(i),y(i), 'r.', 'MarkerSize', 15, 'Color', 'Red' );
end

title(['Wielomian stopnia ', num2str( n(k) ), ' na tle danych']);
xlabel('x');
ylabel('y');
legend(['Wielomian stopnia ', num2str( n(k) )], 'Probka danych',
'Location', 'South');

grid on;
hold off;
end

% Przedstawienie czasow w ms
T = T .* 1000;

% Wykres czasow
hold on;
figure( length(n)+1 );
plot( n, T, '-r.', 'MarkerSize', 25 );
grid on;

title('Czasy dzialania solwera w zaleznosci od stopnia wielomianu
aproksymujacego');
xlabel('Stopien wielomianu aproksymujacego n');
ylabel('Czas dzialania solwera [ms]');

hold off;

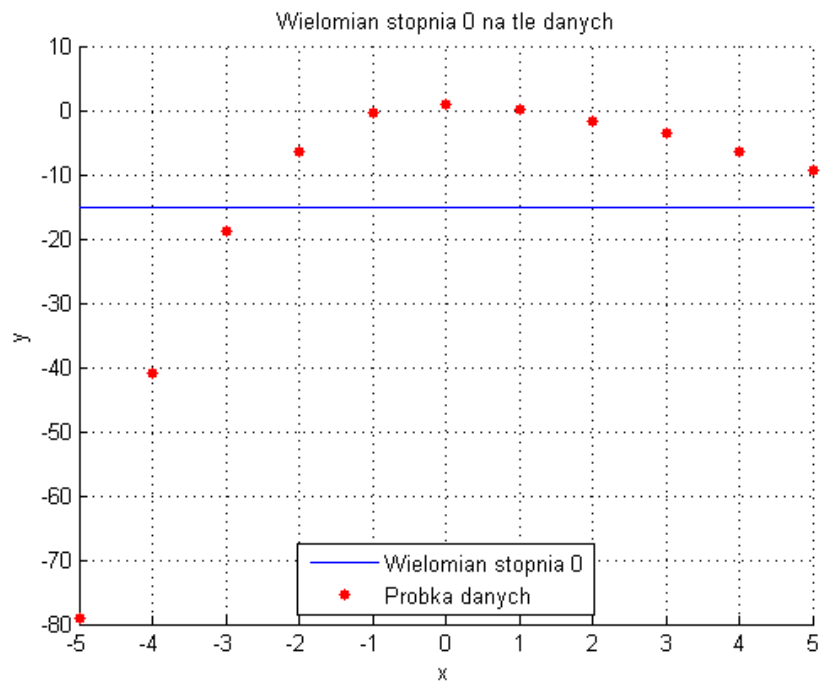
% Wykres bledow
hold on;
figure( length(n)+2 );
plot( n, error_max, '-o', n, error_Euclid, '-o' );
grid on;

title('Normy bledow aproksymacji w zaleznosci od stopnia wielomianu
aproksymujacego');
xlabel('Stopien wielomianu aproksymujacego n');
ylabel('Norma bledu aproksymacji');

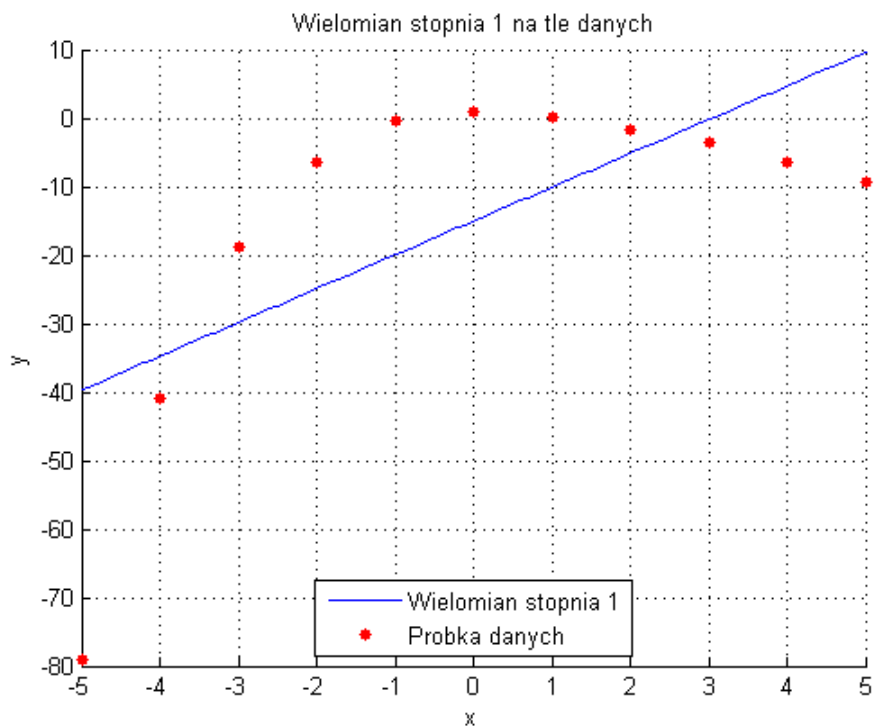
legend('Blad aproksymacji w normie maksimum','Blad aproksymacji w normie
euklidesowej', 'Location', 'North');
hold off;

```

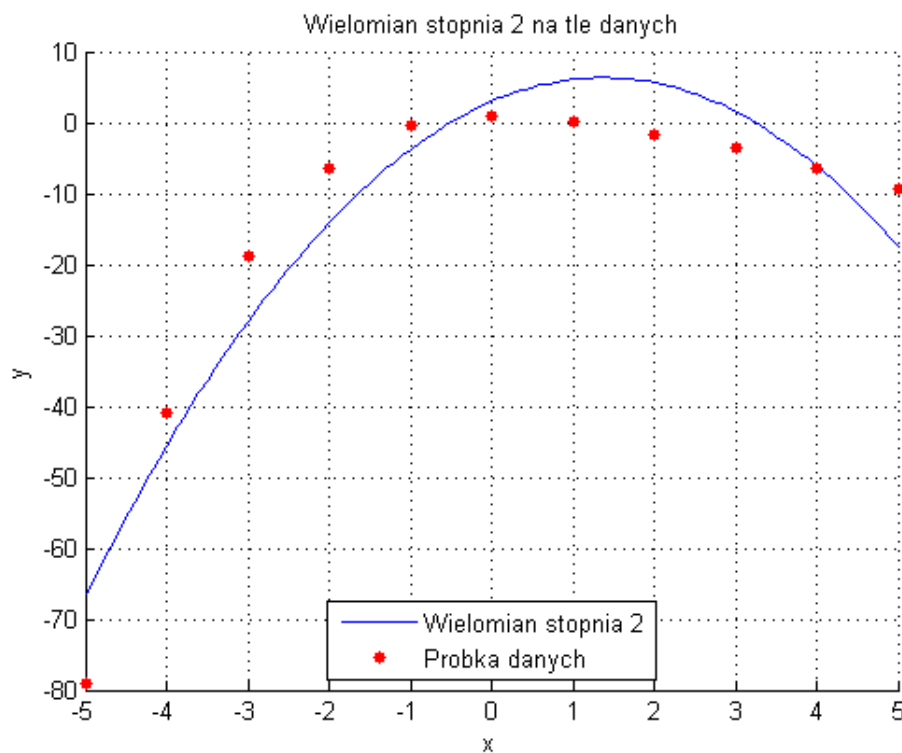
Wykresy:



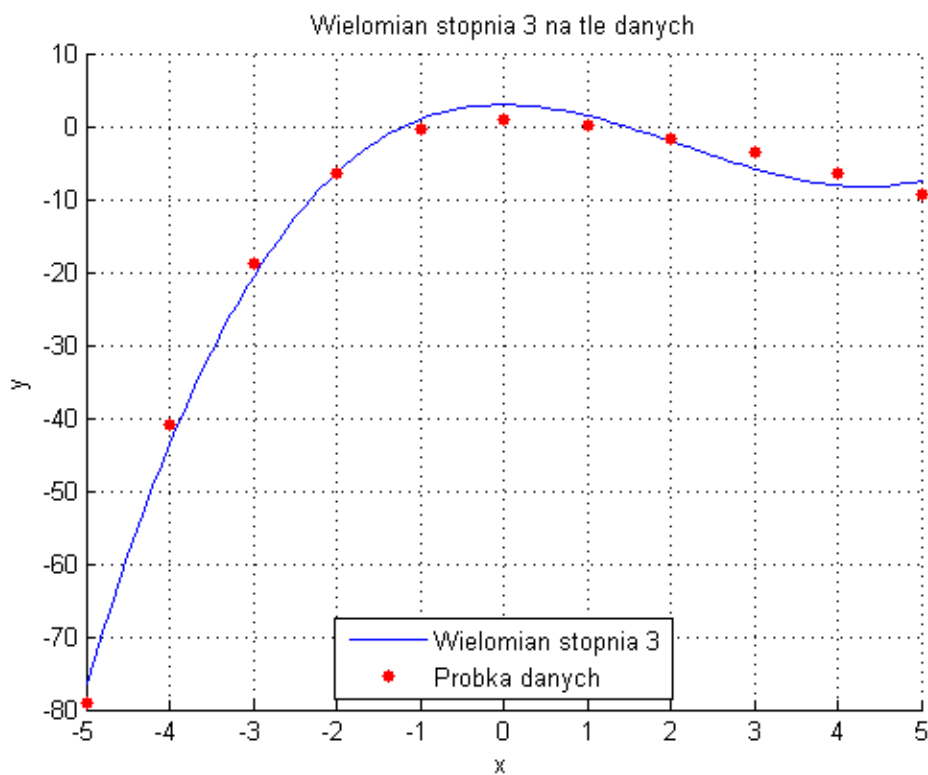
Wykres 14. Wielomian aproksymujący stopnia 0 oraz dane pomiarowe ($\mathbf{a} = [-15.0521]^T$)



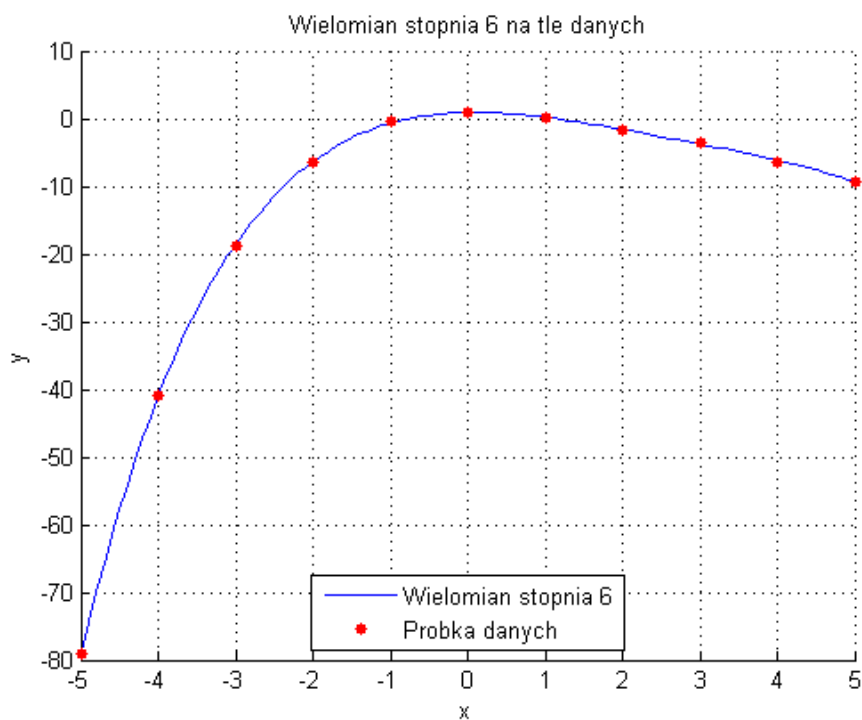
Wykres 15. Wielomian aproksymujący stopnia 1 oraz dane pomiarowe ($\mathbf{a} = [-15.0521 \quad 4.9304]^T$)



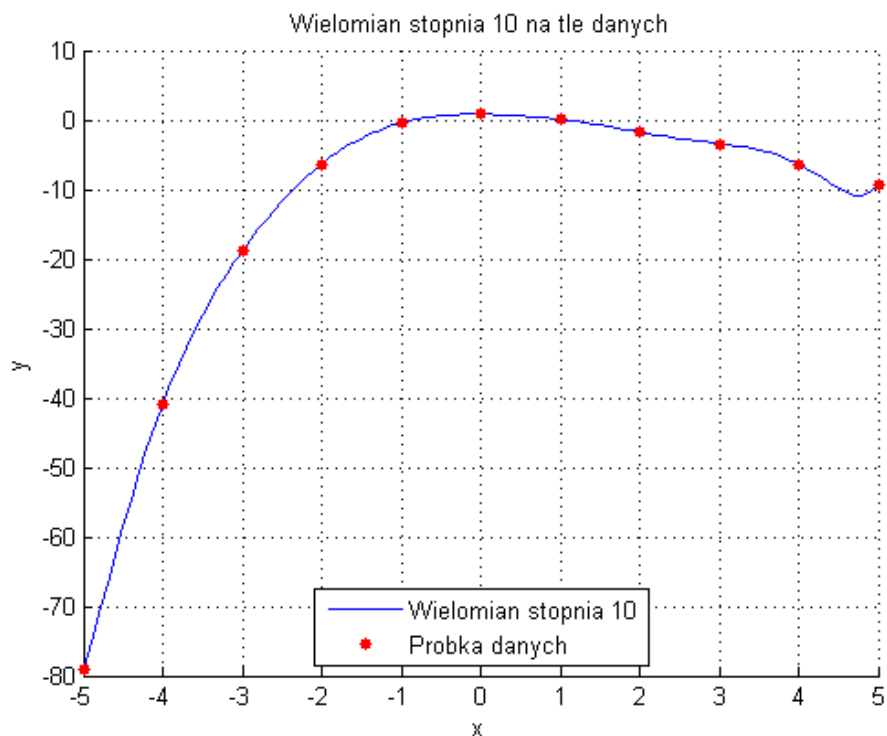
Wykres 16. Wielomian aproksymujący stopnia 2 oraz dane pomiarowe
 $(a = [2.9569 \ 4.9304 \ -1.8009]^T)$



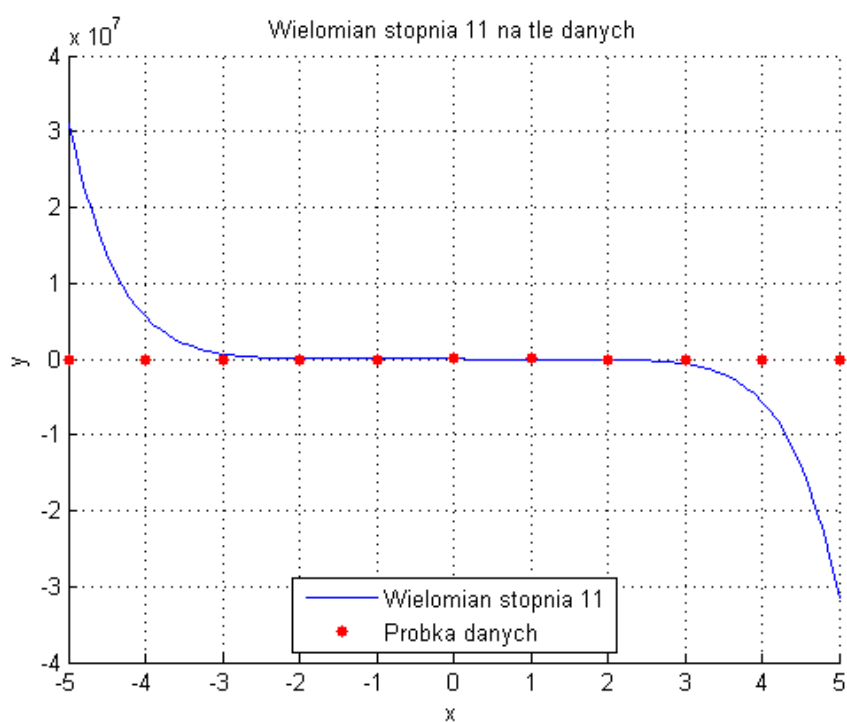
Wykres 17. Wielomian aproksymujący stopnia 3 oraz dane pomiarowe
 $(a = [2.9569 \ -0.0357 \ -1.8009 \ 0.2790]^T)$



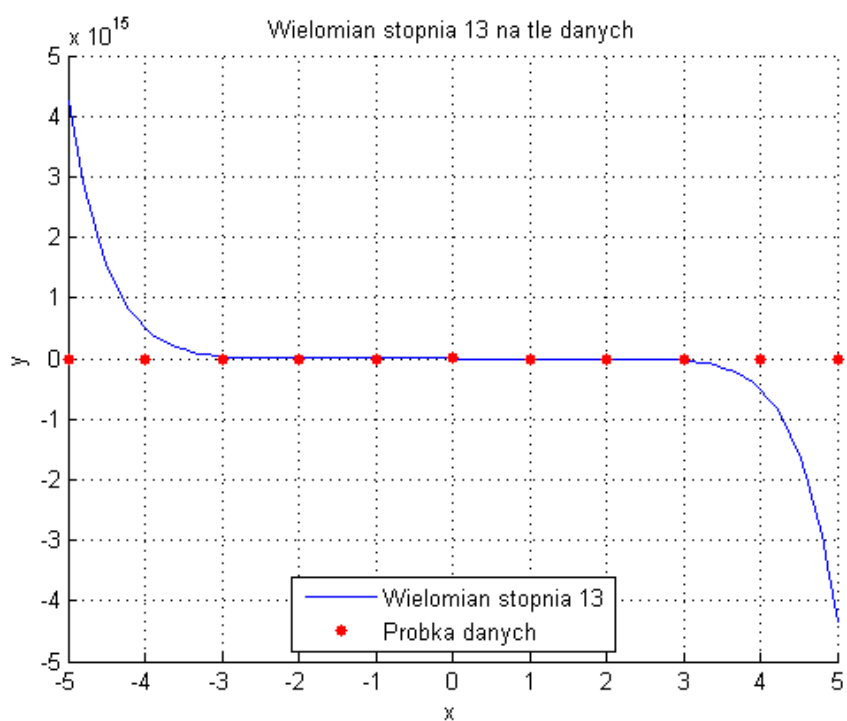
Wykres 18. Wielomian aproksymujący stopnia 6 oraz dane pomiarowe
 $(\mathbf{a} = [0.8901 \ 0.2028 \ -1.1295 \ 0.2394 \ -0.0210 \ 0.0013 \ -0.0002]^T)$



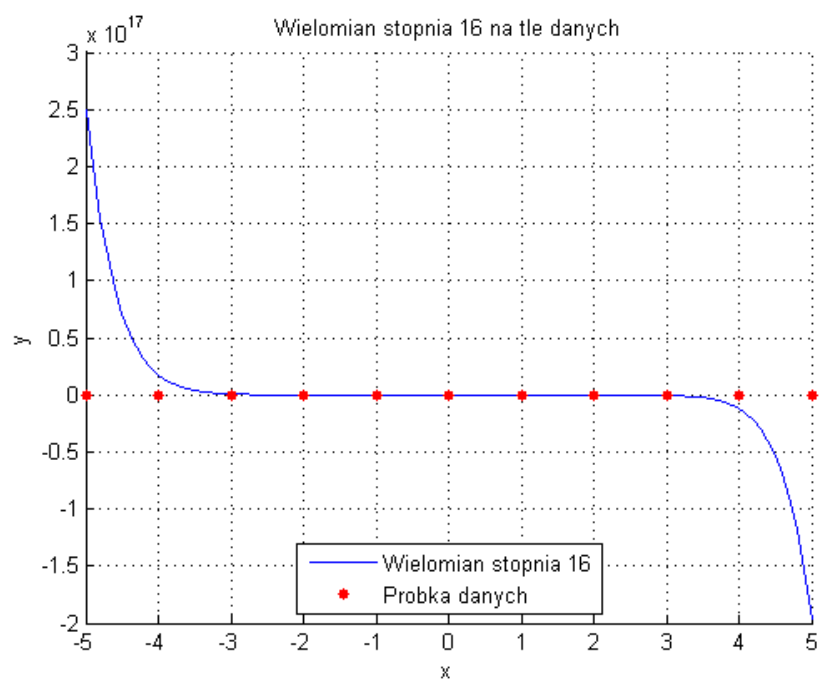
Wykres 19. Wielomian aproksymujący stopnia 10 oraz dane pomiarowe
 $(\mathbf{a} = [0.8270 \ -0.0547 \ -0.9013 \ 0.3030 \ -0.1318 \ 0.0009 \ 0.0167 \ -0.0004 \ -0.0010 \ 0.0000 \ 0.0000]^T)$



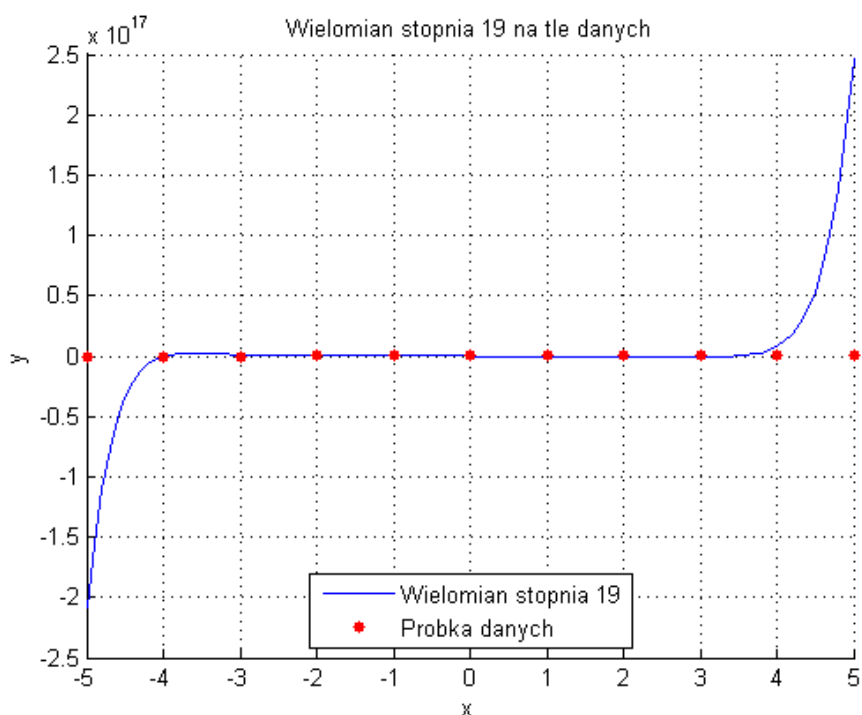
Wykres 20. Wielomian aproksymujący stopnia 11 oraz dane pomiarowe
 $(\mathbf{a} = [0.8270 \quad -0.1300 \quad -0.9013 \quad -121.0889 \quad -0.1318 \quad -487.6761 \quad 0.0167 \quad -184.4216 \quad -0.0010 \quad -7.9910 \quad 0.0000 \quad 0.0000]^T)$



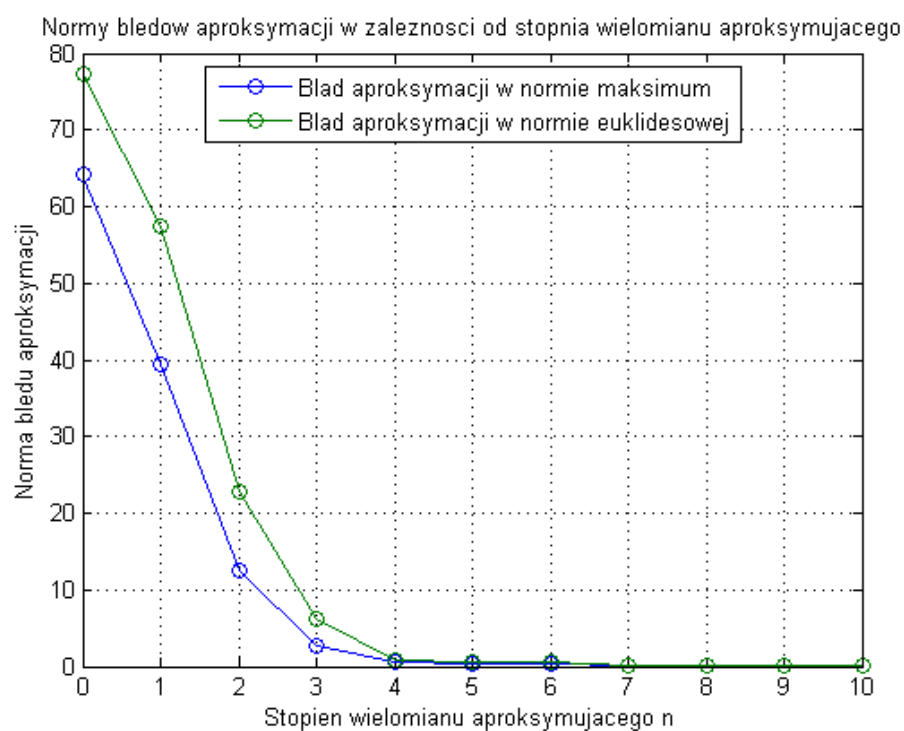
Wykres 21. Wielomian aproksymujący stopnia 13 oraz dane pomiarowe
 $(\mathbf{a} = 1e9 \cdot [0.0000 \quad -0.0000 \quad 0.0000 \quad -0.0521 \quad 0.0000 \quad -1.6832 \quad 0.0000 \quad -3.6475 \quad 0.0000 \quad -1.0656 \quad 0.0000 \quad -0.0405 \quad 0.0000 \quad 0.0000]^T)$



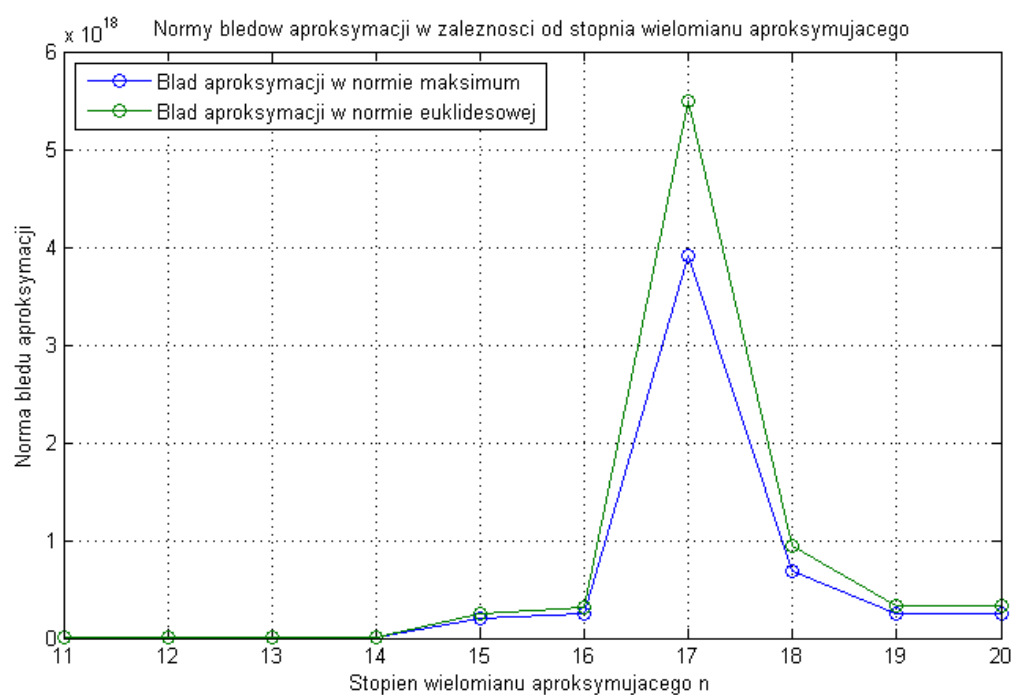
Wykres 22. Wielomian aproksymujący stopnia 16 oraz dane pomiarowe
 $(a = 1e10 \cdot [0.0000 \quad -0.0000 \quad -0.0000 \quad -0.1503 \quad 0.0058 \quad -4.3849 \quad 0.1919 \quad -6.2462 \quad 0.4207 \quad -0.2760 \quad 0.1245 \quad -0.0900 \quad 0.0055 \quad -0.0140 \quad 0.0000 \quad 0.0000 \quad 0.0000]^T)$



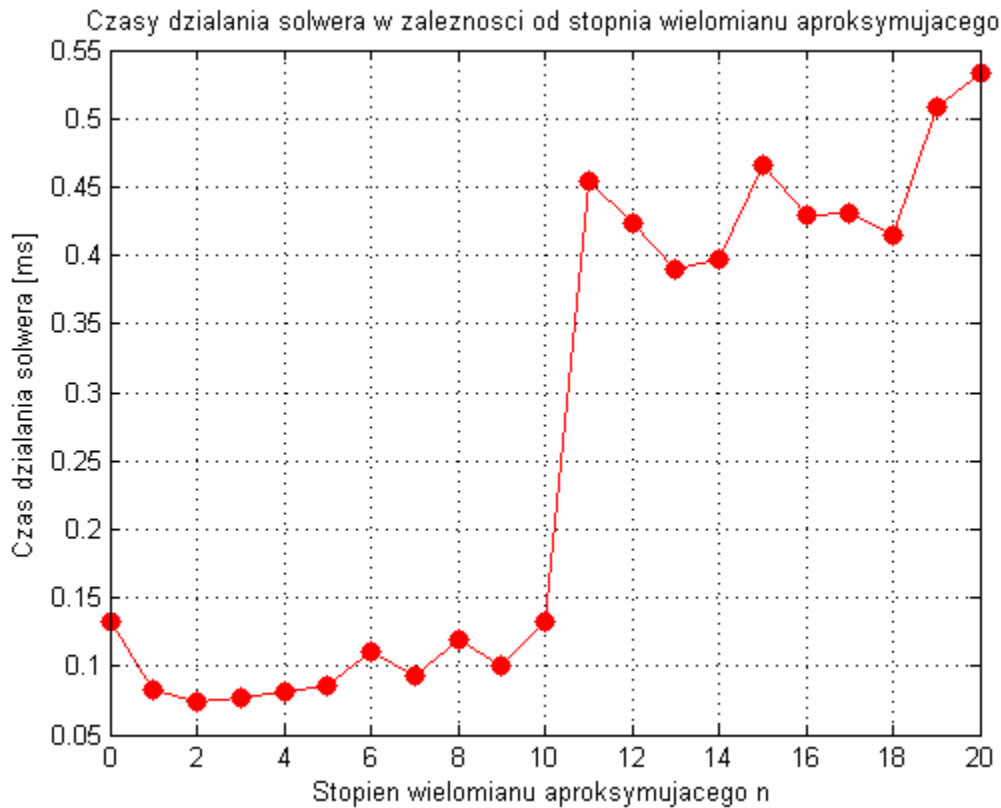
Wykres 23. Wielomian aproksymujący stopnia 19 oraz dane pomiarowe
 $(a = 1e11 \cdot [0.0000 \quad -0.0000 \quad 0.0000 \quad -0.0107 \quad 0.0013 \quad -0.4078 \quad 0.0456 \quad -1.2561 \quad 0.1165 \quad -0.4571 \quad 0.0386 \quad 0.0108 \quad 0.0003 \quad 0.0012 \quad -0.0000 \quad 0.0001 \quad -0.0000 \quad -0.0000 \quad -0.0000 \quad 0.0000 \quad 0.0000]^T)$



Wykres 24. Błędy aproksymacji dla różnych wielomianów aproksymujących



Wykres 25. Błędy aproksymacji dla różnych wielomianów aproksymujących



Wykres 26. Czasy działania solwera dla różnych stopni wielomianów aproksymujących

Komentarz:

Na podstawie analizy wykresów 14. - 19. możemy wnioskować, że wielomian aproksymujący coraz lepiej „dopasowuje” się do próbek danych (sytuacja ta obowiązuje dla stopni wielomianów $n \leq 10$).

Z kolei, na podstawie analizy wykresów 20. - 23. możemy wnioskować, że wielomian aproksymujący jest „niedopasowany” się do próbek danych (sytuacja ta obowiązuje dla stopni wielomianów $n \geq 11$). Sytuację tę można tłumaczyć coraz gorszym uwarunkowaniem macierzy Grama układu równań normalnych wraz ze wzrostem stopnia wielomianu aproksymującego n .

Na podstawie wykresu 24. możemy wnioskować, iż błąd aproksymacji (niezależnie od tego, czy mamy na myśli normę maksimum, czy też normę euklidesową) maleje wraz ze wzrostem stopnia wielomianu aproksymującego – dzieje się tak aż do stopnia wielomianu $n = 10$. Porównując wykresy 11. oraz 24., widzimy, że błędy kształtują się na podobnym poziomie.

Z kolei, na podstawie wykresu 25. możemy wnioskować, iż błąd aproksymacji (niezależnie od tego, czy mamy na myśli normę maksimum, czy też normę euklidesową) osiąga bardzo duże wartości dla wielomianów stopnia $n \geq 11$ (w dużym zakresie stopni wielomianów obserwujemy wzrost błędu aproksymacji).

Warto podkreślić, iż w porównaniu do wykresu 12., uzyskane błędy są mniejsze.

Analizując wykres 26., możemy zaobserwować tendencję wzrostową czasu działania solwera w zależności od stopnia n wielomianu aproksymującego. Interesujący jest duży wzrost czasu działania solwera pomiędzy $n = 10$ oraz $n = 11$ – wynika to z faktu, że dla $n \geq 11$ obliczamy wektor α ze wzoru (4), kosztowniejszego obliczeniowo niż wzór (3). Warto nadmienić, iż maksymalny czas działania solwera nie przekracza 0.55 ms, zatem możemy uznać, że wyniki dostajemy szybko. W porównaniu do wykresu 13., widzimy, że stosując rozkład QR, uzyskujemy rezultaty szybciej niż korzystając bezpośrednio z układu równań normalnych.

Bibliografia:

[1] Piotr Tatjewski „Metody Numeryczne”, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2013