12 stycznia 2018

Jakub Zbrzezny
Grupa e6

# 32. Aproksymacja średniokwadratowa dyskretna funkcji $y = f(x)$ w bazie funkcji $\{1, x, sinx, e^{-x}\}$ w przedziale $[a, b]$ - porównanie wyników uzyskanych dla węzłów podawanych przez użytkownika z wynikami uzyskanymi dla węzłów Czebyszewa.

## 1 Opis metody.

Mamy $n+1$ różnych punktów $x_0, x_1, ..., x_n \in [a, b]$ i wartości $f_0, f_1, ..., f_n$, gdzie $f_i = f(x_i)$
$f(x) \approx \phi(x) = c_0\phi_0(x)+c_1\phi_1(x)+...+c_m\phi_m(x)$, gdzie $f$ jest funkcją aproksymowaną oraz $\phi$ funkcją aproksymującą. $\{\phi_i(x) : i = 0, 1, ..., m\}$ jest układem funkcji bazowych.
Szukamy $c_0, c_1, ..., c_m \in \mathbb{R}$, aby zminimalizować wartość funkcji $g(c_0, c_1, ..., c_m) = \sum_{i=1}^{n} [f_i - \phi(x_i)]^2$.

Bierzemy $f = \begin{bmatrix} f_0 \\ f_1 \\ . \\ . \\ . \\ f_n \end{bmatrix}$ i $x = \begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix}$. Tworzymy macierz $A = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & ... & \phi_m(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & ... & \phi_m(x_1) \\ ... & ... & ... & ... \\ \phi_0(x_n) & \phi_1(x_n) & ... & \phi_m(x_n) \end{bmatrix}$.

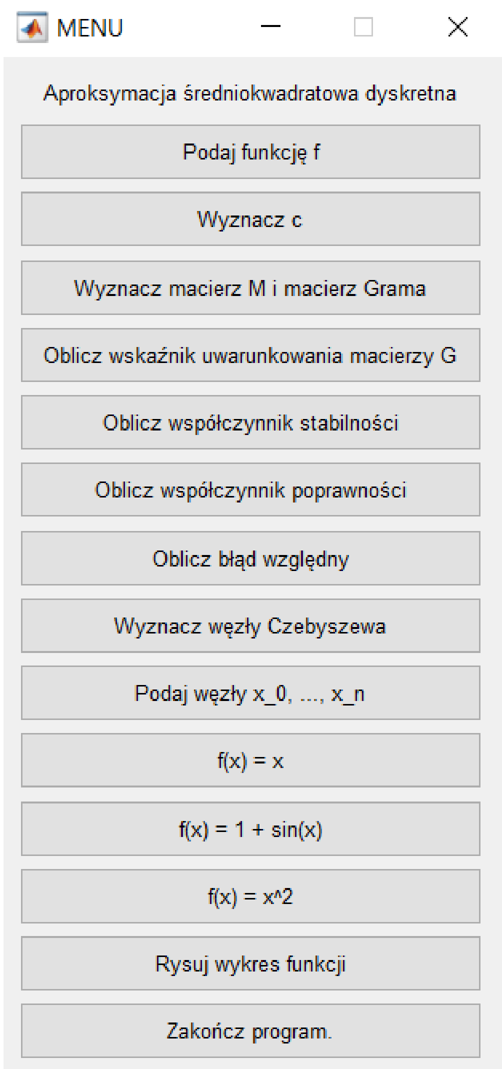
Dla jednoznaczności rozwiązania zakładamy, że kolumny macierzy A są liniowo niezależne.

Tworzymy macierz Grama $G = A^T A$ oraz $g = A^T f$.
Następnie rozwiązujemy układ równań normalnych Gauss $Gc = g$.
W moim projekcie będę szukać funkcji aproksymującej dla m = 3.

Przedstawię również twierdzenie Gaussa o układzie równań normalnych.
Niech $X = \{x \in \mathbb{R}^n : \|b - Ax\|_2 = \|b - Az\|_2\}$, gdzie $A \in \mathbb{R}^{mxn}, b \in \mathbb{R}^n$.
Wówczas $x \in X$ (tj. x jest rozwiązaniem uogólnionym) $\iff A^T Ax = A^T b$.

## 2 Opis programu obliczeniowego.

Po uruchomieniu programu pojawi się poniższe okienko:



Program wyznacza $c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$, rozwiązując równanie $Gc = G$ metodą Crouta, wykorzystując macierz $M = A^T A$, funkcję $f$, $x = \begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ . \\ x_n \end{bmatrix}$, gdzie $x_0, x_1, ..., x_n$ są węzłami ($n \in \mathbb{N}$).
Sprawdza, czy liczba wierszy macierzy $G$ jest równa 4. Jest równość nie zachodzi, to program wyświetla komunikat: "Liczba wierszy macierzy G nie jest równa 4!", a następnie kończy działanie.
Sprawdza także, czy rząd macierzy $G$ jest równy 4. Jeżeli rząd macierzy G nie jest równy 4, to program wyświetla komunikat "Rząd macierzy G nie jest równy 4!. Rozwiązanie nie jest jednoznaczne.".
Następnie program dokonuje rozkładu macierzy $G$ na macierze $L$ i $U$ takie, że $G = LU$.
Potem program rozwiązuje następujący układ równań:
$$\begin{cases} GY = g \\ Uc = Y \end{cases}.$$
Wtedy już mamy otrzymaną wartość $c$.

Program też liczy wskaźnik uwarunkowania macierzy Grama metodą Gaussa (GE). Najpierw rozwiązuje równanie $AA^{-1} = I_4$ metodą GE, żeby wyznaczyć macierz odwrotną. Później liczy wartość $\|A^{-1}\| * \|A\|$.

Program również liczy współczynnik stabilności oraz współczynnik poprawności.
Program liczy także błąd względny, bezwzględny obliczenia.
W programie można wybrać 3 funkcje ($f(x) = x, f(x) = 1+sinx, f(x) = x^2$), ale również można podać swoją.

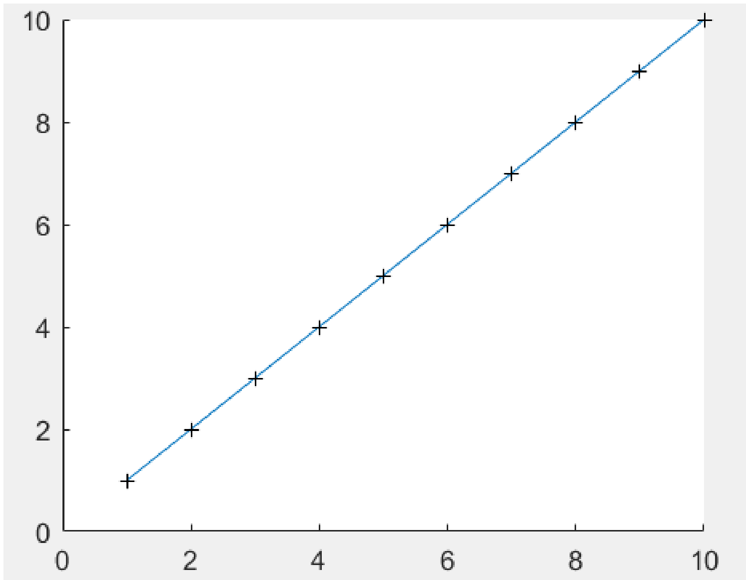
Jeszcze program rysuje wykres funkcji aproksymowanej i funkcji aproksymującej. Przed narysowaniem wykresu użytkownik musi podać, ile chce wprowadzić węzłów.

# 3 Przykłady obliczeniowe.

Teraz podam parę różnych przykładów.

## 3.1 Przykład 1

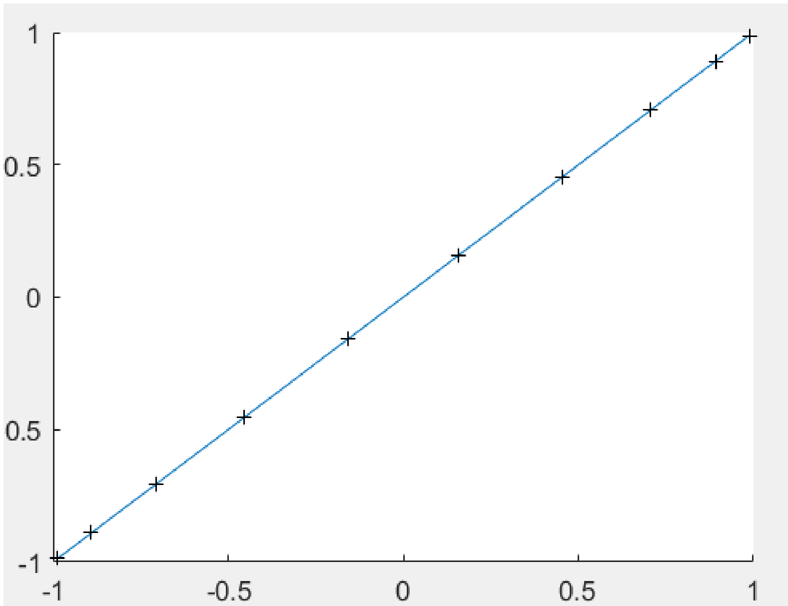
Rozważmy funkcję $f(x) = x$. Najpierw bierzemy węzły $x_0 = 1, x_1 = 2, ..., x_9 = 10$.
Wskaźnik uwarunkowania macierzy G wynosi 2.5370e+03.
Otrzymanym rozwiązaniem jest $c = \begin{bmatrix} 0 \\ 1 \\ -1.904869165421959e-16 \\ 0 \end{bmatrix}$. Natomiast rozwiązaniem poprawnym jest $\alpha = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$.
Na następnej stronie jest wykres funkcji f i otrzymanej funkcji aproksymującej f.


10
8
6
4
2
0
0
2
4
6
8
10

Teraz bierzemy 10 węzłów Czebyszewa. Wskaźnik uwarunkowania macierzy G wynosi 1.9902e+01.

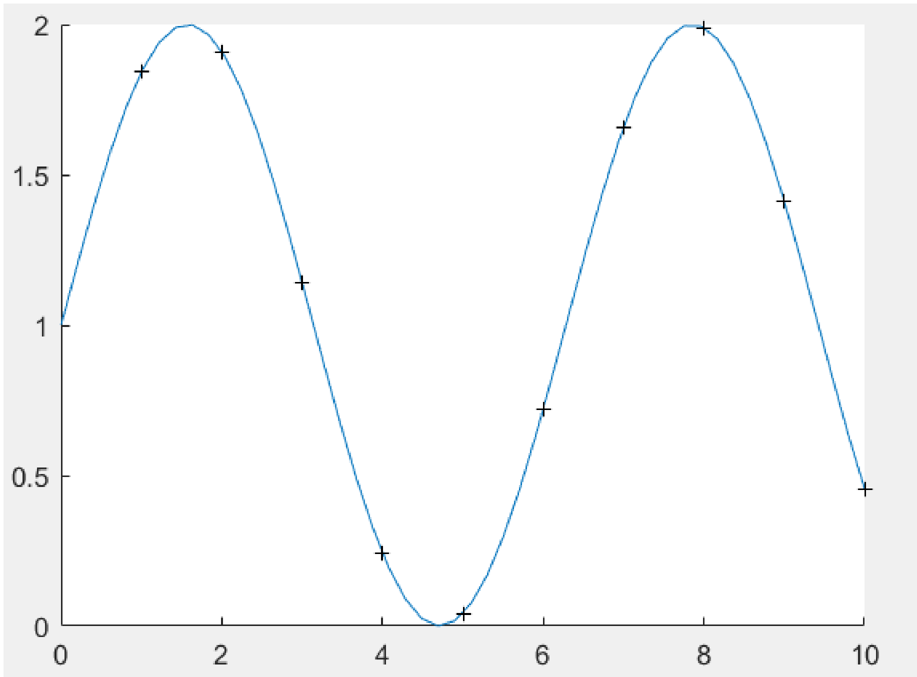
Otrzymanym rozwiązaniem jest $c = \begin{bmatrix} -3.0498e-15 \\ 1.0000e+00 \\ -1.1874e-13 \\ 2.4089e-15 \end{bmatrix}$. Natomiast rozwiązaniem poprawnym jest $\alpha = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$.

Na następnej stronie jest wykres funkcji f i otrzymanej funkcji aproksymującej f.


1
0.5
0
0.5
-1
-1
-0.5
0
0.5
1

## 3.2 Przykład 2

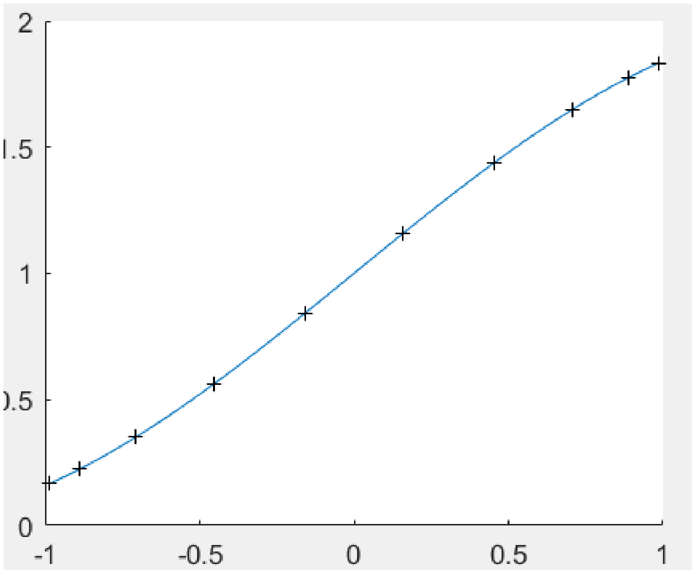
Teraz rozważmy funkcję $f(x) = 1 + sinx$. Bierzemy węzły $x_0 = 1, x_1 = 2, ..., x_9 = 10$.
Wskaźnik uwarunkowania macierzy G wynosi 2.5370e+03.
Otrzymanym rozwiązaniem jest $c = \begin{bmatrix} 1.0000e+00 \\ -4.5797e-16 \\ 1.0000e+00 \\ -9.4714e-15 \end{bmatrix}$. Natomiast rozwiązaniem poprawnym jest $\alpha = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$.
Na następnej stronie jest wykres funkcji f i otrzymanej funkcji aproksymującej f.


2
1.5
1
0.5
0
0
2
4
6
8
10

Teraz bierzemy 10 węzłów Czebyszewa. Wskaźnik uwarunkowania macierzy G wynosi 1.9902e+01.

Otrzymanym rozwiązaniem jest $c = \begin{bmatrix} 1.0000e+00 \\ -5.6399e-14 \\ 1.0000e+00 \\ -2.4089e-15 \end{bmatrix}$. Natomiast rozwiązaniem poprawnym jest $\alpha = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$.

Na następnej stronie jest wykres funkcji f i otrzymanej funkcji aproksymującej f.

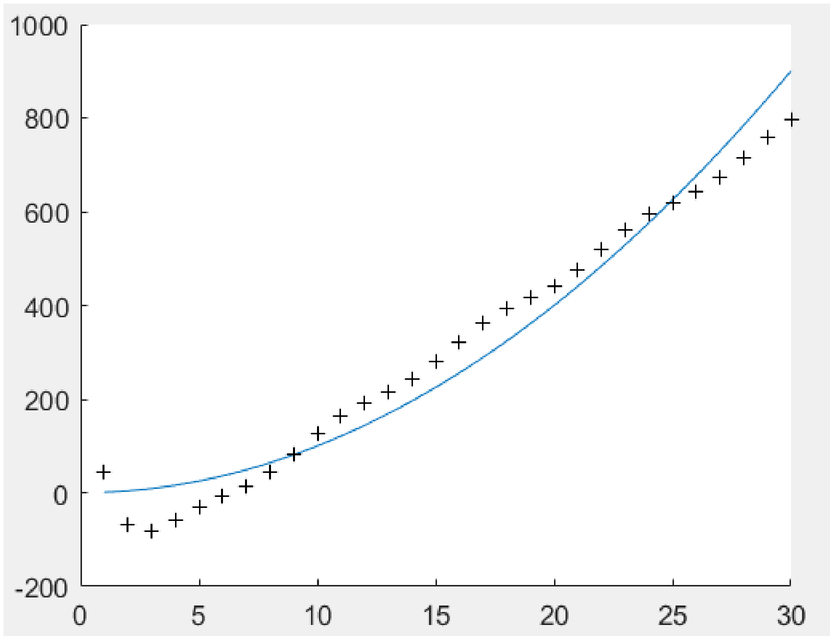

2
.5
1
.5
0
-1
-0.5
0
0.5
1

### 3.3 Przykład 3

Rozważmy teraz funkcję $f(x) = x^2$.
Bierzemy węzły $x_0 = 1, x_1 = 2, ..., x_29 = 30$.
Wskaźnik uwarunkowania macierzy G wynosi 6.0655e+04

Otrzymanym rozwiązaniem jest $c = \begin{bmatrix} -210.9013 \\ 33.1553 \\ -11.3708 \\ 632.3733 \end{bmatrix}$. Natomiast rozwiązaniem poprawnym jest $\alpha = \begin{bmatrix} -210.9013 \\ 33.1553 \\ -11.3708 \\ 632.3733 \end{bmatrix}$.

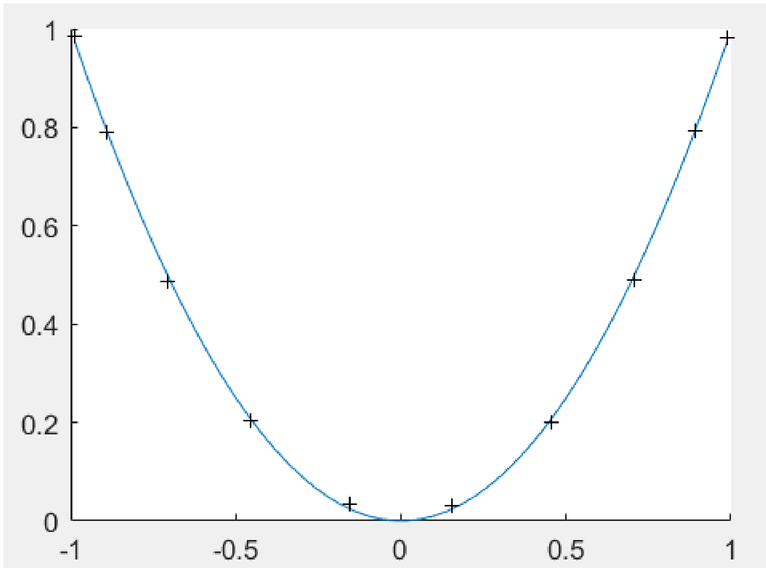
Na następnej stronie jest wykres funkcji f i otrzymanej funkcji aproksymującej f.


1000
800
600
400
200
0
-200
0
5
10
15
20
25
30

Teraz bierzemy 10 węzłów Czebyszewa.
Wskaźnik uwarunkowania macierzy G wynosi 1.9902e+01

Otrzymanym rozwiązaniem jest $c = \begin{bmatrix} -1.8307e+00 \\ 3.9161e+00 \\ -2.0853e+00 \\ 1.8409e+00 \end{bmatrix}$. Natomiast rozwiązaniem poprawnym jest $\alpha = \begin{bmatrix} -1.8307e+00 \\ 3.9161e+00 \\ -2.0853e+00 \\ 1.8409e+00 \end{bmatrix}$.

Na następnej stronie jest wykres funkcji f i otrzymanej funkcji aproksymującej f.


1
0.8
0.6
0.4
0.2
0
-1
-0.5
0
0.5
1

# 4 Analiza wyników.

## 4.1 Przykład 1.

a) Węzły $x_0 = 1, x_1 = 2, ..., x_9 = 10$
Wskaźnik uwarunkowania macierzy G = 2.5370e+03
Błąd względny = 1.9049e-16
Współczynnik poprawności = 2.8253e-19
Współczynnik stabilności = 8.4973e-18

b) 10 węzłów Czebyszewa
Wskaźnik uwarunkowania macierzy G = 1.9902e+01
Błąd względny = 1.0893e-15
Współczynnik poprawności = 2.7438e-17
Współczynnik stabilności = 1.3667e-19

## 4.2 Przykład 2.

a) Węzły $x_0 = 1, x_1 = 2, ..., x_9 = 10$
Wskaźnik uwarunkowania macierzy G = 2.5370e+03
Błąd względny = 1.9049e-16
Współczynnik poprawności = 3.1965e-18
Współczynnik stabilności = 5.5379e-19

b) 10 węzłów Czebyszewa
Wskaźnik uwarunkowania macierzy G = 1.9902e+01
Błąd względny = 1.0893e-15
Współczynnik poprawności = 2.7438e-17
Współczynnik stabilności = 1.0907e-17

## 4.3 Przykład 3.

a) Węzły $x_0 = 1, x_1 = 2, ..., x_29 = 30$
Wskaźnik uwarunkowania macierzy G = 6.0655e+04

Błąd względny = 1.2196e-15
Współczynnik poprawności = 4.6090e-18
Współczynnik stabilności = 1.3956e-20

b) 10 węzłów Czebyszewa
Wskaźnik uwarunkowania macierzy G = 1.9902e+01
Błąd względny = 6.2552e-14
Współczynnik poprawności = 5.1787e-17
Współczynnik stabilności = 7.8484e-18

N podstawie rysunków można więc wywnioskować, że aproksymacja daje lepsze wyniki, gdy aproksymuje się funkcję, która jest sumą elementów z bazy.
Natomiast metoda gorzej się sprawdza, gdy próbuje się aproksymować jakąś funkcję, która nie należy do bazy.
Ponadto wskaźnik uwarunkowania macierzy Grama jest mniejszy, gdy aproksymuje się funkcję dla węzłów Czebyszewa, niż w przypadku węzłow $x_0 = 1, x_1 = 2, ..., x_9 = 10$

# 5 Kody źródłowe wszystkich procedur.

## 5.1 Program.

```
% MENU
clear
clc
zakoncz_program = 14;
kontrol = 1;
while kontrol ~= zakoncz_program
    kontrol = menu('Aproksymacja średniokwadratowa dyskretna', 'Podaj funkcję f',
     'Wyznacz c', 'Wyznacz macierz M i macierz Grama', 'Oblicz wskaźnik
      uwarunkowania macierzy G', 'Oblicz współczynnik stabilności',
       'Oblicz współczynnik poprawności', 'Oblicz błąd względny',
        'Wyznacz węzły Czebyszewa', 'Podaj węzły x_0, ..., x_n', 'f(x) = x',
         'f(x) = 1 + sin(x)', 'f(x) = x^2', 'Rysuj wykres funkcji',
          'Zakończ program.');
    switch kontrol
        case 1
            f = input('Podaj funkcję f.');
        case 2
            c = Najmniejszy_blad_sredniokwadratowy(M,f, x, n);
        case 3
            q_0 = @(t) 1;
            q_1 = @(t) t;
            q_2 = @(t) sin(t);
            q_3 = @(t) exp(-t);
            q = @(t) [q_0(t), q_1(t), q_2(t), q_3(t)]';
            M = zeros(n+1,4);
            for i = 1:n+1
                for j = 1:4
                    a = q(x(i,1));
                    M(i,j) = a(j,1);
                end
            end
            G=M'*M;
        case 4
            % Obliczanie wskaźnika uwarunkowania macierzy G metodą Gaussa.
```

```
    I = eye(4);
    GI = [G I];
    for i = 1:4
        for j = (i + 1):4
            GI(j, (1:8)) = GI(j, (1:8)) - (GI(j,i) / G(i,i))
             * GI(i, (1:8));
        end
    end
    for i = 1:4
        GI(i,(1:8)) = (1/G(i,i))*GI(i,(1:8));
    end
    for j = 2:4
        for i = 1: (j - 1)
            GI(i,(1:8)) = GI(i, (1:8)) - GI(j, (1:8)) * GI(i, j);
        end
    end
    Ginv = GI((1:4),(5:8));
    wsk_uwarunkowania = norm(Ginv) * norm(G);
case 5
    wspolczynnik_stabilnosci(c, rozw_pop, G);
case 6
    wspolczynnik_poprawnosci(G,g,c);
case 7
    blad_wzgledny(c, rozw_pop)
case 8
     n = input('Ile chcesz węzłów ') - 1;
     x = zeros(n, 1);
    for i = 0:n
        x(i + 1, 1) = cos(((2 * i + 1) * pi) / (2*n+2));
    end
case 9
    n = input('Ile chcesz węzłów ') - 1;
    disp('Podaj x');
    x = input('x=      '); % linspace(-pi,pi,n)
case 10
    f = @(t) t;
case 11
    f = @(t) 1 + sin(t);
```

```
        case 12
            f = @(t) t.^2;
        case 13
            fun1 = @(t) c(1) + c(2).*t + c(3).*sin(t) + c(4).*exp(-t);
            p = linspace(x(1), x(n + 1), 100);
            p = p(:);
            y1 = zeros(n + 1, 1); % Funkcja aproksymująca
            y2 = zeros(100, 1); % Funkcja aproksymowana
            for i = 1:50
                z = p(i);
                y2(i) = f(z);
            end
            for i=1:n + 1
                z = x(i);
                y1(i) = fun1(z);
            end
            y1 = y1(:);
            y2 = y2(:);
            hold on
            plot(p, y2);

            ph1 = plot(x,y1);
            set(ph1,'MarkerEdgeColor', 'black', 'MarkerFaceColor', 'black');
            set(ph1,'Marker', '+', 'MarkerSize', 5);
            set(ph1,'LineStyle', 'none');

            hold off
        case 14
            disp('Koniec programu.');
            return;
    end
end
```

### 5.2 Funkcja wyznaczająca c.

```
function c = Najmniejszy_blad_sredniokwadratowy(M, f, x, n)
% Funkcja wyznacza c = (c_0, c_1, c_2, ..., c_n)' tak, aby znimalizować
% wartość wyrażenia g(c_0, c_1, c_2, ..., c_n) = (f_0 - q(x_0))^2 + .... +
```

```
% (f_n - q(x_n))^2, gdzie f jest funkcją aproksymowaną, natomiast q funkjcą
% aproksymującą.
p = zeros(n + 1, 1);
f1 = zeros(n + 1, 1);
for i = 1:n+1
   % p(i,1) = f(x(i,1)); %x(i,1) = x_0
   f1(i)=f(x(i));
end
G = M'*M;
f1 = f1(:);
g=M'*f1
%
disp('Macierz M');
disp(M);
[a_1, b_1] = size(G);
            if (a_1 ~= 4)
                disp('Liczba wierszy macierzy G nie jest równa 4!');
                return;
            end
            if(rank(G) ~= 4)
                disp('Rząd macierzy G nie jest równy 4!. Rozwiązanie nie jest
                jednoznaczne.');
                return;
            end
            L = zeros(4, 4);
            U = eye(4, 4);

            for k = 1 : 4
                for i = k : 4
                    L(i,k) = G(i, k) - MojaSuma(i, k, 1, k - 1, L, U);
                end
                for j = k + 1 : 4
                    U(k, j) = (G(k, j) - MojaSuma(k, j, 1, k - 1, L, U)) /
                     L(k, k);
                end
            end
            y = Rozwiaz_uklad_LYB(L, g); % Moja funkcja -tutaj L jest
            trojkatna dolna
```

```
            c = Rozwiaz_uklad_UXY(U, y);  % Moja funkcja -tutaj U jest
             trojkatna gorna
end
```