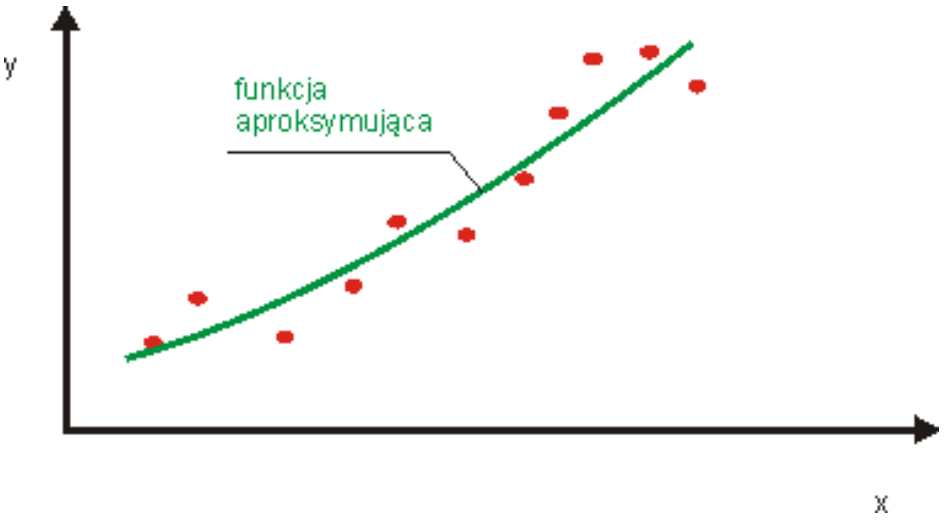
Aproksymacja

opracował dr inż. Robert JAKUBOWSKI, Politechnika Rzeszowska, KSiSL

Table of Contents

Aproksymacja polega przybliżeniu przebiegu punktów określoną funcją z zapewnieniem najmniejszego błedu pomiędzy funkcją przybliżajacą a zadanymi punktami. W odróznieniu od interpolacji w tym wypadku nie zależy nam na tym aby wyznaczona funkcja dokładnie przechodziła przez zadane punkty. Zagadnienia aproksymacji najczęściej dotyczą wyznaczenia stosunkowo prostej funkcji oddającej ogólny trend rozkładu otrzymanych punktów. Zagadnienia te wykorzystuje się do analizy wyników badań obarczonych błedem pomiarów, które rozkładają się względem krzywej o określonym kształcie np. funkci liniowej, funkcji kwadratowej, trygonometrycznej itd. Przykład aproksymacji przedstawiono na rysunku 1.



Rys.1 Przykład zagadnienia aproksymacji

Funkcje aproksymujące mogą być wielomianam, funkcją trygonometryczną, funkcją logarytmiczną, funkcją wymierną itd.

Poszukiwanie funkcji aproksymujacej można realizować metodą najmniejszych kwadratów, stąd też metoda aproksymacji nosi nazę aproksymacji średniokwadratowej lub najmniejszych kwadratów, co można zapisać:

$$F = \min\left(\sum_{i-1}^{n} y_i - F(x_i)\right)$$

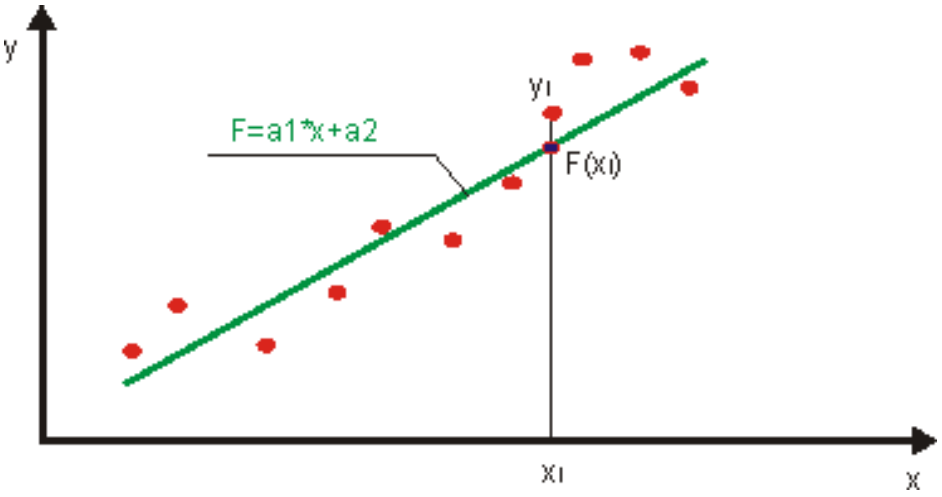
Pokżmy zagadnienie aproksymacji na najprostrzym przykładzie aproksymacji funkcją liniową zwanej też metodą regresji liniowej. Berdziej zaawansowane metody zostaną przedstawione w kolejnych rozdziałach.

Regresja liniowa

Zakładamy, że przybliżenie otrzymanych punktów będzie funkcją liniową :

$$F = a_1 x + a_2$$

Problem aproksymacji przedstawiono na rysunku 2



Rys 2. Aprolsymacja funkcją liniową

Poszukujemy wartości parametrów a_1 i a_2, funkcji $F = a_1 x + a_2$, takich aby spełniony został warunek

$$S(a_1, a_2) = \min\left(\sum_{i=1}^{n} (y_i - F(x_i))^2\right) = \min\left(\sum_{i=1}^{n} (y_i - (a_1 x_i + a_2))^2\right)$$

Zadanie to można rozwiązać różniczkując wyrażenie w nawiasie po a_1 i a_2 i przyrównujac do 0.

$$\frac{\partial}{\partial a_1} S(a_1, a_2) = 0$$

$$\frac{\partial}{\partial a_2} S(a_1, a_2) = 0$$

Po podtsawieniu wyrażenia na F otrzymujemy

$$\frac{\partial}{\partial a_1} \sum_{i=1}^{n} (y_i - (a_1 x_i + a_2))^2 = 0$$

$$\frac{\partial}{\partial a_2} \sum_{i=1}^{n} (y_i - (a_1 x_i + a_2))^2 = 0$$

po zróżniczkowaniu otrzymujemy układ równań w postaci:

$$2\sum_{i-1}^{n} (y_i - (a_1 x_i + a_2))(-x_i) = 0$$

$$2\sum_{i=1}^{n} (y_i - (a_1 x_i + a_2))(-1) = 0$$

po przekształceniu sprowadza się to do następujacego układu równań:

$$\sum_{i-1}^{n} \left(a_1\ x_i^2 + a_2\, x_i\right) = \sum_{i-1}^{n} y_i\, x_i$$

$$\sum_{i-1}^{n} \left(a_1\ x_i + a_2\ \right) = \sum_{i-1}^{n} y_i$$

W postaci macierzowej można ten układ przedstawić:

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i\, x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Sprawdźmy to na konkretnym przykładzie. Zadajmy punkty

```
xi=[2, 4, 6, 8, 10];
yi=[2.5 10 32 40 60];
```

Przedstawmy to graficznie na wykresie

```
plot(xi,yi,'pr','MarkerFaceColor',"r")
```

Przygotujmy układ równań wg wcześniej podanej zależności w postaci:

$$P * a = Y$$

gdzie:

$$P = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix}$$

a= $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

$$Y = \begin{bmatrix} \sum_{i=1}^{n} y_i\, x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Wykorzystajmy mnożenie macieżowe, którym zastąpimy sumowanie. Zgodnie z zapisem macieżowym:

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = a_1 + a_2 + a_2 + a_4,$$

stąd,

```
n=length(xi)
```

```
n = 5
```

```
% tworzymy wektor pomocniczy o jednej kolumnie i n wierszach
mp=ones(n,1)
```

```
mp = 5×1
     1
     1
     1
     1
     1
```

```
% Przygotujmy wektory P i Y o odpowiednich wymiarach
P=ones(2);
Y=ones(2,1);
% Obliczmy wartości P i Y według podanych wyżej zależności
% Skalarnie jest podnoszony wektor xi do kwadratu, a potem wymnażany
% wektorowo z mp
P(1,1)=(xi.^2)*mp;
P(1,2)=xi*mp
```

```
P = 2×2
   220    30
     1     1
```

```
P(2,1)=P(1,2);
P(2,2)=n
```

```
P = 2×2
   220    30
    30     5
```

```
% W tym wypadku wykorzystujemnymnożenie kombinowane skalarne xi i yi z
% wektorowym mp
Y(1,1)=(xi.*yi)*mp;
Y(2,1)=yi*mp
```

```
Y = 2×1
10^3 ×
    1.1570
    0.1445
```

Rozwiązmy układ równań w celu wynzcenia wektora a

```
a=(P^-1)*Y
```

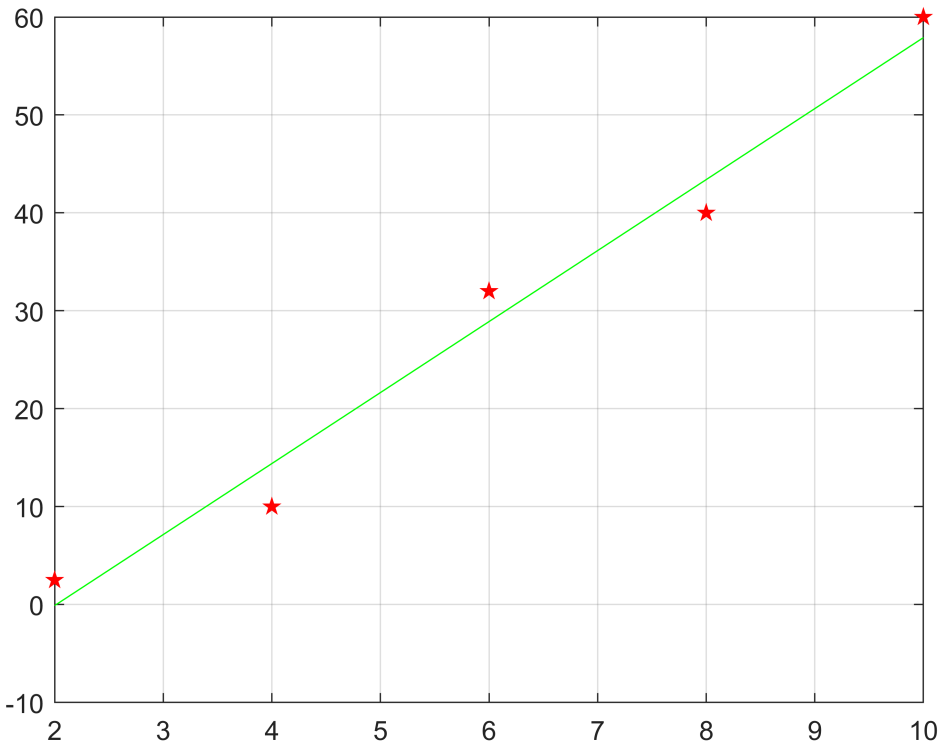
```
a = 2×1
    7.2500
  -14.6000
```

```
% zamieńmy wektor kolumnowy w wektor wierszowy
a=a'
```

```
a = 1×2
    7.2500  -14.6000
```

W ten sposób zostały wyznaczone wartości parametrów wielomianu interpolującego, a właściwie funkcji liniowej. Narysujmy ją z wykorzystaniem poznanej wcześniej funkcji polyval.

```
hold on
grid on
fplot(@(x) polyval(a,x),[xi(1),xi(end)],'g')
hold off
```



Odchylenie standardowe wg [1] wyznacza się z zależności:

$$S = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n} e_i^2}$$

gdzie $e_i = y_i - F(x_i)$, co dla metody regresji liniowej daje $e_i = y_i - (a_1 x_i + a_2)$.

Sumę błędu względnego dzieli się przez n-2 w celu uwzględninia niepewności pomiarów oraz błędów przy wyznaczniu a_1 i a_2. Policzmy zatem odchylenie standradowe

```
e=0;
for i=1:n
    e=e+(yi(i)-(a(1).*xi(i)+a(2))).^2;
```

```
end
S=sqrt(e./(n-2))
```

```
S = 4.1513
```

Wyznaczona w ten sposób wartość jest brana pod uwagę przy określaniu rzeczywistej wartości wyrażaonej na podstawie aproksymacji, stąd:

$y(x_i) = F(x_i) \pm \mathrm{S}$

czyli dla x=5;

```
y_5_max=polyval(a,5)+S
```

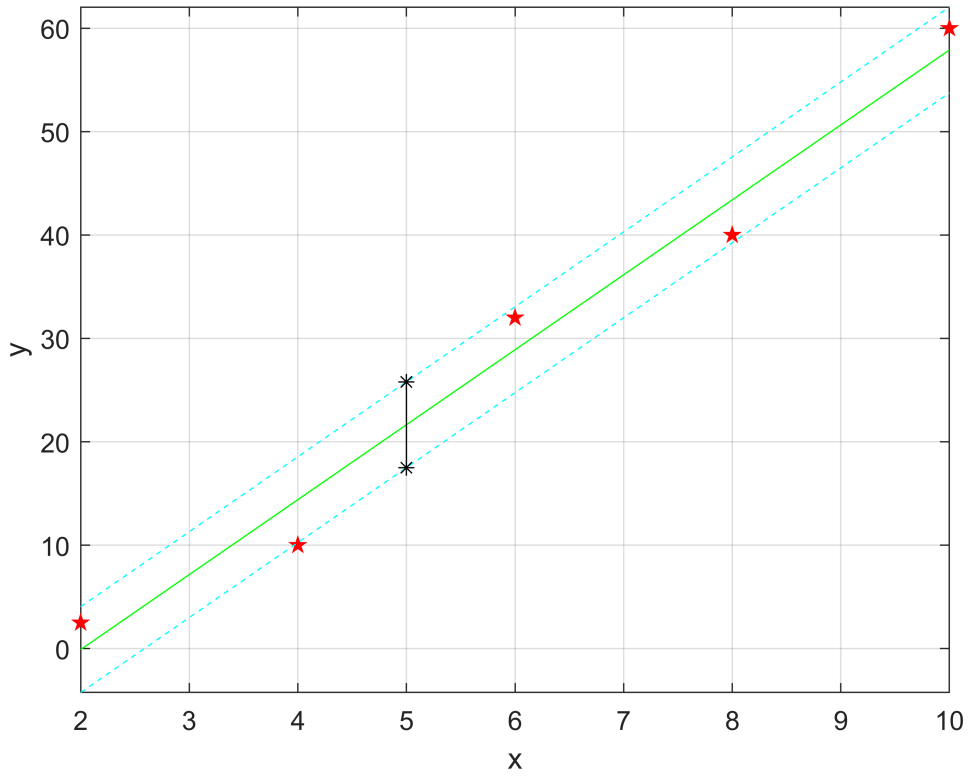
```
y_5_max = 25.8013
```

```
y_5_min=polyval(a,5)-S
```

```
y_5_min = 17.4987
```

Wykres z zaznaczonym przedziałem występowania rozwiązania przedstawiono graficznie liniami przerywanymi na wykresie

```
fplot(@(x) polyval(a,x),[xi(1),xi(end)],'g')
hold on
grid on
plot(xi,yi,'pr','MarkerFaceColor',"r")
fplot(@(x) polyval(a,x)+S,[xi(1),xi(end)],'c--')
fplot(@(x) polyval(a,x)-S,[xi(1),xi(end)],'c--')
xlabel("x")
ylabel('y')
plot([5,5],[y_5_min,y_5_max],'-k*')
hold off
```



Współczynnik korelacji liniowej Pearsona

Współczynnik korelacji określa poziom dopasowania danych do funkcji aproksymującej. W przypadku współczynnika Pearsona określa on skorelowanie punktów z aproksymacją zaleznościa liniową. Wyznacza się go z zalezności:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

gdzie wartości x i y średnie liczy się następująco:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Współczynnik korelacji zawiera się w przedziale od -1 do 1. Gdy jest on bliski -1 ozncza to silną korelację ujemną, a gdy jest on bliski 1 oznacza so silną korelację dodatnią. Gdy jest on bliski 0 oznacza to brak korelacji. W naszym przypadku współczynnik korelacji wyliczymy następująco:

Wartości sredanie x i y

```
x_sr=1/n.*(xi*mp)
```

```
x_sr = 6
```

```
y_sr=1/n.*(yi*mp)
```

```
y_sr = 28.9000
```

Pozostałe składniki zależności

```
licznik=(xi-x_sr).*(yi-y_sr)*mp;
mianownik1=(xi-x_sr).^2*mp;
mianownik2=(yi-y_sr).^2*mp;
r=licznik/sqrt(mianownik1*mianownik2)
```

```
r = 0.9879
```

Współczynnik korelacji wyszedł na poziomie bliskim jedności, czyli mamy do czynienia z silną dodatnią korelacją liniową.

Aproksymacja wielomianem wyższego stopnia

Aproksymację wielomianem wyższego żędu można sprowadzić do rozwiązania podobnego układu równań jak w przypadku aproksymacji liniowej. Przykładowo dla aproksymacji funkcją drugiego stopnia:

$$F = a_1 x^2 + a_2 x + a_3$$

poszukiwali będziemy parametrów a_1, a_2, a_3. Dla takiego przypadku układ równań będzie

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^4 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i & n \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Ogólnie dla aproksymacji wielomianem m-tego stopnia układ równań przedstawia się następująco

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^{2m} & \sum_{i=1}^{n} x_i^{2m-1} & \cdots & \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m} \\ \sum_{i=1}^{n} x_i^{2m-1} & \sum_{i=1}^{n} x_i^{2m-2} & \cdots & \sum_{i=1}^{n} x_i^{m} & \sum_{i=1}^{n} x_i^{m-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m} & \cdots & \sum_{i=1}^{n} x_i^{2} & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^{m} & \sum_{i=1}^{n} x_i^{m-1} & \cdots & \sum_{i=1}^{n} x_i & n \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ a_{m+1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i x_i^{m} \\ \sum_{i=1}^{n} y_i x_i^{m-1} \\ \vdots \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Zadanie to zostało zawarte w funkcji aproksymacja_wielomian

Wypróbujmy jak zadziała funkcja , gdy zastosujemy ją do wcześniej rozwiązywanego zadania. Wywołajmy ją z wcześniej zdefiniowanymi punktami, oraz stopniem wielomianu aproksymacyjnego równym 1.

```
[P,S,R]=aproksymacja_wielomian(xi,yi,1)
```

```
P = 1×2
    7.2500  -14.6000
S = 4.1513
R = 0.9879
```

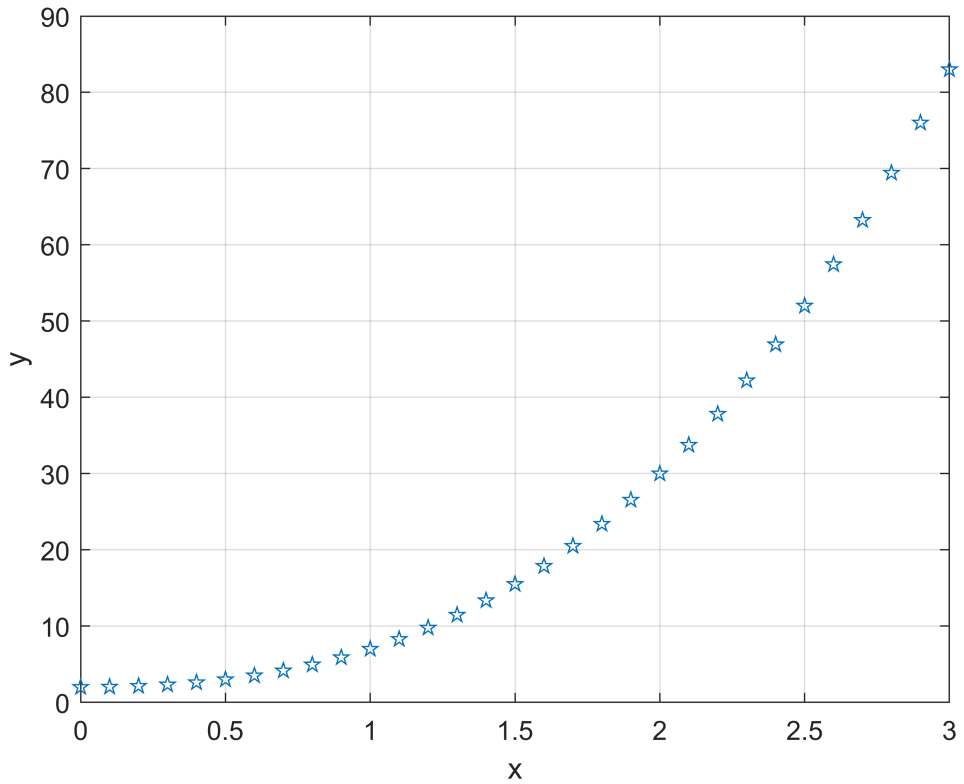
Otrzymany wynik jest dokładnie taki sam jak otrzymano wcześniej

Przygotujmy punkty, które będą układać się w funkcję trzeciego stopnia:

```
xi=0:0.1:3;
yi=2.*xi.^3+3.*xi.^2+2;
```

Przedstawmy to graficznie w postaci punktów

```
plot(xi,yi,"p")
xlabel("x")
ylabel("y")
grid on
```



Zweryfikujmy jakie wartości parametrów funkcji aproksymującej otrzymamy gdy wstawimy wartość wykładnika wielomianu aproksymujacego 2

```
[P2,S2,R2]=aproksymacja_wielomian(xi,yi,2)
```

```
P2 = 1×3
   12.0000  -10.6240    4.4360
S2 = 1.1557
R2 = 0.9345
```

Dla porównania zobaczmy jakie parametry otrzymamy w przypadku aproksymacji wielomianem 3-go stopnia

```
[P3,S3,R3]=aproksymacja_wielomian(xi,yi,3)
```

```
P3 = 1×4
    2.0000    3.0000    0.0000    2.0000
S3 = 2.0651e-12
R3 = 0.9345
```

W przypadku aproksymacji funkcją 3-go stopnia odchylenie standardowe S3 jest równe 0, dla drugiego stopnia wartość ta wyszła nieco większa S2=1.156

Dodajmy jeszce aproksymację funkcją liniową i porónajmy wyniki graficznie

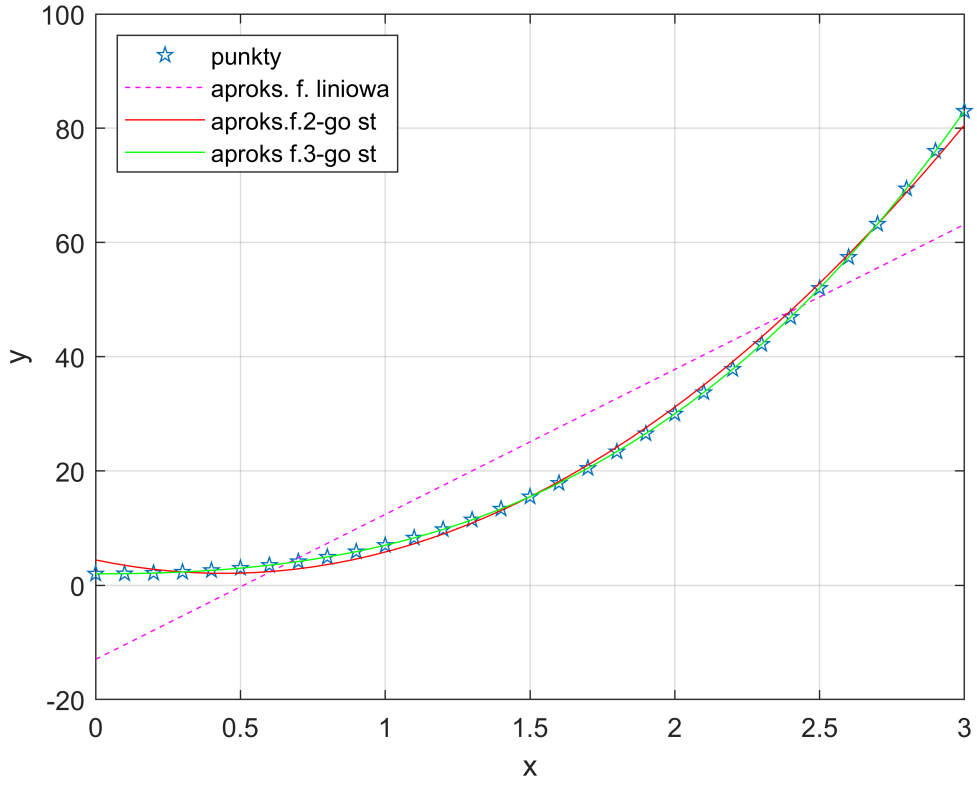
```
[P1,S1,R1]=aproksymacja_wielomian(xi,yi,1)
```

```
P1 = 1×2
   25.3760  -12.9640
S1 = 8.9388
R1 = 0.9345
```

W tym wypadku odchylenie standardowe S1 wyszło na poziomie 8.9. Dla wszystkich obliczeń wartość współczynnika korelacji liniowej jest taki sam, bowiem on zależy od ułożenia punktów i nie wpływa na niego zastosowana funkcja aproksymujaca.

Narysujmy fynkcję punkty wraz z obydwoma funkcjami aproksymującymi

```
plot(xi,yi,"p")
xlabel("x")
ylabel("y")
hold on
fplot(@(x) polyval(P1,x),[xi(1),xi(end)],'m--')
fplot(@(x) polyval(P2,x),[xi(1),xi(end)],'r-')
fplot(@(x) polyval(P3,x),[xi(1),xi(end)],'g-')
legend("punkty", "aproks. f. liniowa" , "aproks.f.2-go st","aproks f.3-go st","Location","north
grid on
hold off
```



Analiza pokazana graficznie, że funkcja liniowa zdecydowanie gorzej oddaje trend pokazanych punktów, niż pozostałe dwie funkcje. Natomiast interpolacja funkcją drugiego i trzeciego stopnia daje dobrą korelację, oczywiście funkcja 3-go stopnia w sposób idealny oddaje trend, odchylenie standardowe jest równe 0.

Aproksymacja z wykorzystaniem funkcji Matlaba

W matlabie do aproksymacji wielomianem możemy wykorzystać funkcję ***polyfit***, którą wykorzystywaliśmy w obliczeniach wielomianu interpolacyjnego. W tym wypadku stopień wielomianu określamy zgodnie z funkcją, którą chcemy aproksymować dane. Dla zadanych wcześniej punktów spróbujemy wyznaczyć wielomian 2-go i 3-go stopnia z wykorzysraniem funkcji ***polyfit*** i porównać wyniki

```
Pol2=polyfit(xi,yi,2)
```

```
Pol2 = 1×3
   12.0000  -10.6240    4.4360
```

```
P2
```

```
P2 = 1×3
   12.0000  -10.6240    4.4360
```

```
Pol3=polyfit(xi,yi,3)
```

```
Pol3 = 1×4
    2.0000    3.0000   -0.0000    2.0000
```

```
P3
```

```
P3 = 1×4
    2.0000    3.0000    0.0000    2.0000
```

Porównanie wyników otrzymanych obydwoma metodami wskazuje, że uzyskaliśmy te same wartości wskaźników wielomianów aproksymujących, co potwierdza poprawność funkcji przygotowanej do wykładu.

Ćwiczenia do tematu

Przygotuj punkty stanowiące zależność wielomianu 4-go stopnia. Przedstaw je graficznie na wykresie. Wykonaj aproksymację wielomianem 2-go, 3-go i 4-go jedną z przedstawionych funkcji. Wyniki dołącz do wykresu

Funkcje przygotowane do wykładu

Funkcja aproksymacja_wielomian w postacie [P,S,R]=aproksymacja_wileomian(xi,yi,m). zmienne wyjściowe to P - parametry wielomianu zgodne z funkcją polyval, S - odchylenie standardowe, R - współczynnik korelacji. Zmienne wejściowe xi - wektor x, yi - wektor y, m - stopień wielomianu aproksymującego.

```
function [P,S,R]=aproksymacja_wielomian(xi,yi,m)
%sprawdzenie wymiaru wektorów w celu weryfikacji
[s1_x,s2_x]=size(xi);
[s1_y,s2_y]=size(yi);
%Zmienna Pom wprowadzona, do kontynuacji obliczeń Pom=0, obliczenia są
%kontynuowane, Pom=1 Obliczenia należy przerwać;
Pom=0;
if s1_x>1
    if s2_x>1
        disp("Zmienna xi powinna być wektorem")
        Pom=1;
    end
    xi=xi';
end

if s1_y>1
    if s2_y>1
        disp("Zmienna yi powinna być wektorem")

        Pom=1;
    end
    yi=yi';
end

if length(xi)~=length(yi)
    disp("Długość wektorów xi i yi powinna być równa")
    Pom=1;
end
% KOD WŁAŚCIWY FUNKCJI
if Pom==0
    n=length(xi);
```

```
    % tworzymy wektor pomocniczy o jednej kolumnie i n wierszach
    mp=ones(n,1);
    P=zeros(m+1);
    %Zawarcie xi w postaci wektora
    for i=1:m+1
       P_pom(i,:)=xi.^(m+1-i);
       Y_pom(i,:)=yi;
    end
    for i=1:m+1
        P(:,i)=(P_pom.*xi.^(m+1-i))*mp;
    end

    Y=zeros(1,m+1);
    Y=Y_pom.*P_pom*mp;
    Y(m+1,1)=yi*mp;

    a=(P^-1)*Y;
    % zamieńmy wektor kolumnowy w wektor wierszowy i przypisanie do
    % zmiennej P
    P=a';
    %ODCHYLENIE STANDARDOWE
    e=0;
    for i=1:n
        e=e+(yi(i)-polyval(P,xi(i))).^2;
    end
    S=sqrt(e./(n-2));
    %WSPÓŁCZYNNIK KORELACJI
    %Wartości sredanie x i y
    x_sr=1/n.*(xi*mp);
    y_sr=1/n.*(yi*mp);
    %Pozostałe składniki zależności
    licznik=(xi-x_sr).*(yi-y_sr)*mp;
    mianownik1=(xi-x_sr).^2*mp;
    mianownik2=(yi-y_sr).^2*mp;
    R=licznik/sqrt(mianownik1*mianownik2);
else
        P=NaN;
        S=NaN;
        R=NaN;
end
end
```