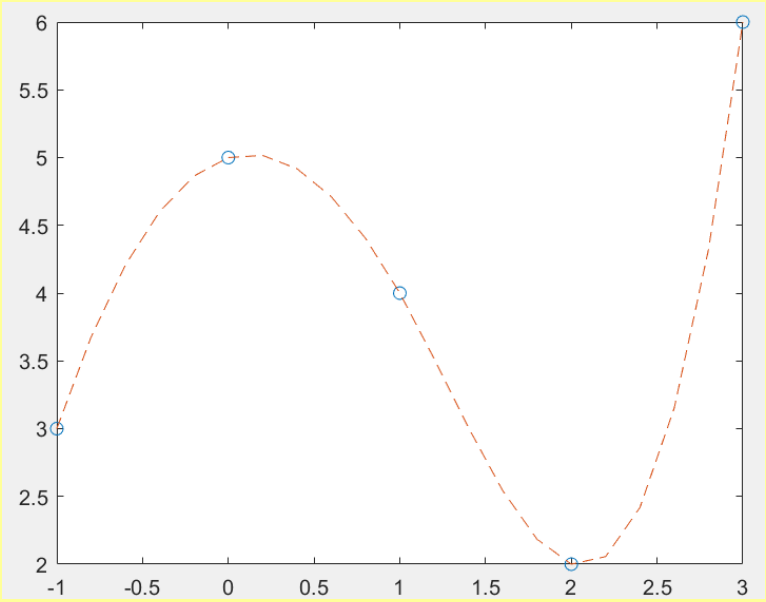
Wykład 6

Interpolacja i aproksymacja

1. Interpolacja

Zagadnienie interpolacji można sformułować następująco: w przedziale <a,b> danych jest n+1 różnych punktów x_1, x_1, ..., x_n, które nazywamy ***węzłami interpolacji***, oraz wartości pewnej funkcji $y=f(x)$ w tych punktach $f(x_0)=y_0$, $f(x_1)=y_1$,..., $f(x_n)=y_n$.

Zadaniem interpolacji jest wyznaczenie przybliżonych wartości funkcji w punktach niebędących węzłami oraz oszacowanie błędu tych przybliżonych wartości. W tym celu należy znaleźć funkcję $F(x)$ zwaną ***funkcją interpolującą***, która w węzłach interpolacji przyjmuje takie same wartości jak funkcja $f(x)$.



Funkcje interpolujące

y_i=interp1(x, y, x_i, metoda) - zwraca wektor y_i, będący wartościami funkcji jednej zmiennej y=*f*(x) w punktach określonych wektorem x_1; węzły interpolacji określają wektory x i y, metoda to łańcuch znaków określający metodę interpolującą.

z_i=interp2(x, y, z, x_i, y_i, metoda) - zwraca macierz z_1 zawierającą wartości funkcji dwóch zmiennych z=*f*(x,y) w punktach określonych wektorami x_i i y_i; węzły interpolacji określają macierze x, y, z.

v_i=interp3(x, y, z, v, x_i, y_i, z_i, metoda) - interpolacja funkcją trzech zmiennych, analogicznie jak w **interp2**.

v_i=interpn(x_1, x_2, x_3,...v, x_i, y_i, z_i, metoda) - interpolacja funkcją n zmiennych, analogicznie jak w **interp2**.

Metody interpolacji

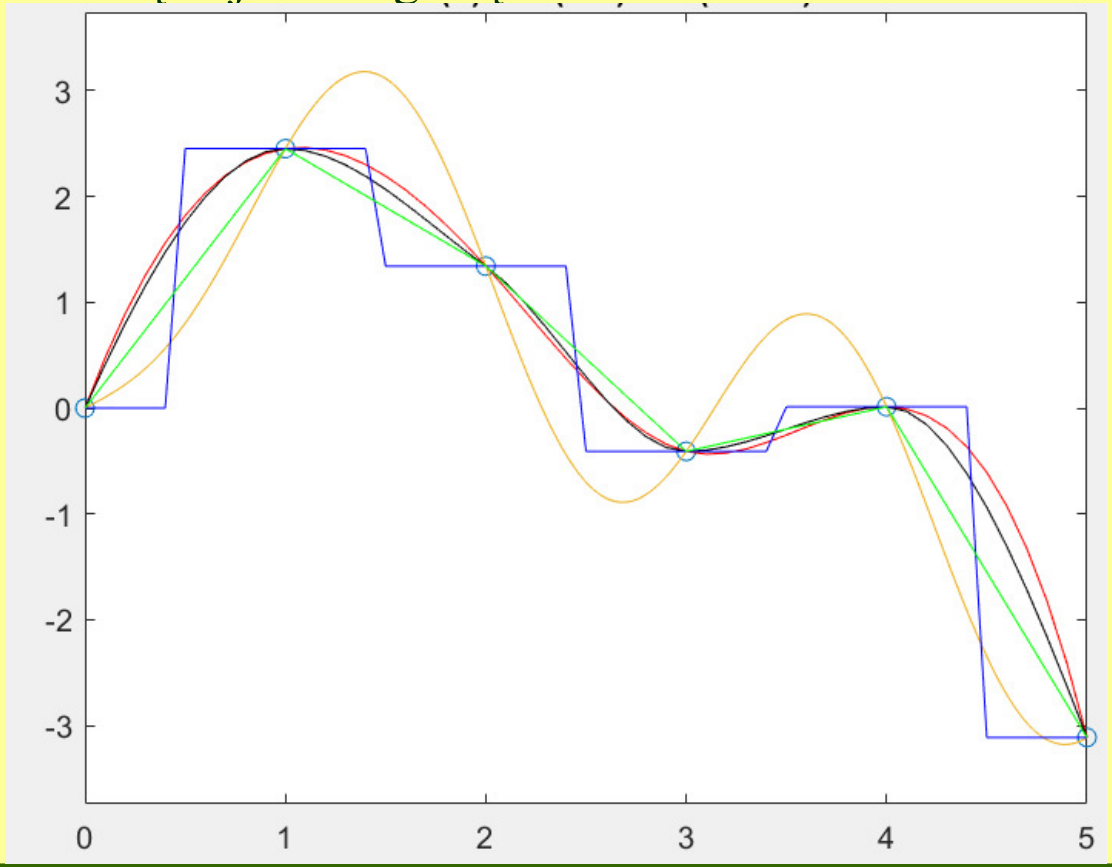
'linear' - interpolacja liniowa,

'spline' - interpolacja funkcjami sklejanymi stopnia trzeciego,

'cubic' lub 'pchip' - interpolacja wielomianami stopnia trzeciego,

'nearest' - interpolacja wartością najbliższego sąsiada.

Wszystkie metody interpolacji wymagają, aby ciąg **x** był monotoniczny.



2. Interpolacja Lagrange'a

Interpolacja za pomocą wielomianu polega na znalezieniu takiego wielomianu $L_n(x)$ stopnia co najwyżej n, aby wartości tego wielomianu i funkcji interpolowanej w węzłach były sobie równe, czyli: $L_n(x) = a_0+a_1x+a_2x^2+...+ a_nx^n$

Zadanie interpolacji wielomianowej ma jednoznaczne rozwiązanie, czyli istnieje tylko jeden wielomian spełniający warunek. Wartości współczynników wielomianu wyliczane są ze wzoru Lagrange'a. Szukany wielomian ma postać:

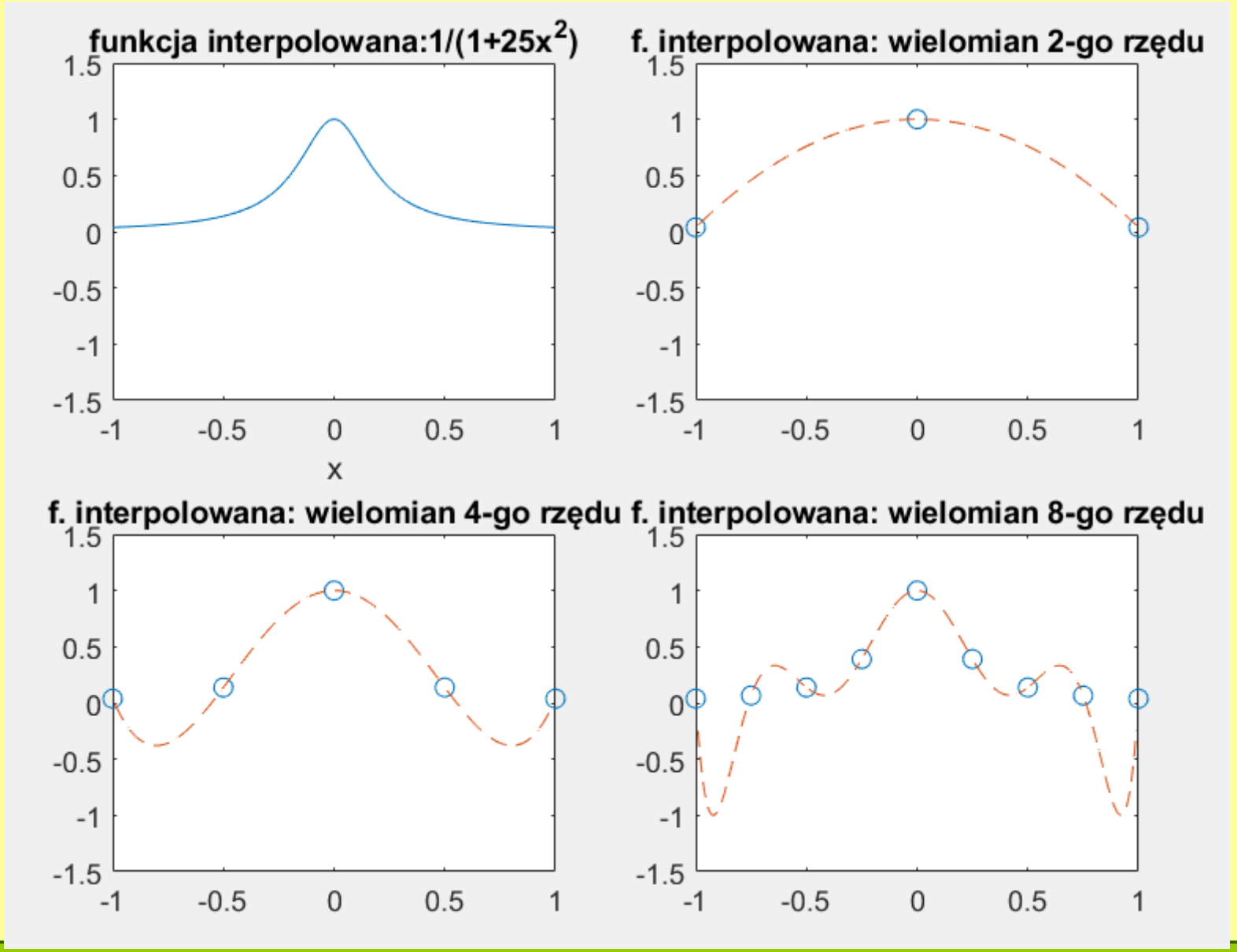
$$L_n(x) = \sum_{j=0}^{n} y_j \frac{(x-x_o)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n)}{(x_j-x_o)(x_j-x_1)...(x_j-x_{j-1})(x_j-x_{j+1})...(x_j-x_n)}$$

przyjmując, że: $\omega_n(x) = (x-x_o)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n)$

wzór Lagrange'a ma wtedy postać: $L_n(x) = \sum_{j=0}^{n} y_j \frac{\omega_n(x)}{(x-x_j)\omega_n(x_j)}$

gdzie $y_j=y(x_j)$, a $\omega_n(x_j)$ jest wartością pochodnej wielomianu $\omega(x)$ w punkcie x_j.

Przykład wielomianu Lagrange'a



3. Aproksymacja

Załóżmy, że w przedziale <a, b> jest n+1 punktów x_o = a, x_1, x_2, ...,x_n=b oraz wartości pewnej funkcji $y=f(x)$ w tych punktach. Poszukiwana jest funkcja F(x) dobrze przybliżająca f(x) w przedziale <a,b>, z tym że w odróżnieniu od interpolacji, obydwie funkcje nie muszą mieć takich samych wartości w podanych punktach.

Pojawia się zatem błąd aproksymacji zdefiniowany jako różnica między wartościami funkcji aproksymującej F i funkcji aproksymowanej f w punktach xi, i=0, 1, ..., n. Zadaniem metody numerycznej jest w tym przypadku minimalizacja błędu aproksymacji.

Aproksymacja jest nazywana również dopasowywaniem (fitowaniem) krzywej do danego zbioru danych. Jeżeli punkty (x_i, y_i) pochodzą z badań eksperymentalnych nad procesami fizycznymi, to zwykle ich wartości są obarczone błędami pomiaru.

Jedną z najpopularniejszych metod aproksymacji jest aproksymacja średniokwadratowa, zwana także metodą najmniejszych kwadratów. Polega ona na minimalizacji błędu aproksymacji zdefiniowanego jako suma kwadratów odchyłek we wszystkich punktach obliczeniowych:

$$\sum_{i=0}^{n} w(x_i)\left[F(x_i) - f(x_i)\right]^2$$

Do poprawy przybliżenia w wybranych węzłach może być wykorzystana funkcja wagowa $w(x_i)$. Zwykle jest ona tożsamościowo równa jedności, zatem wszystkie odchyłki są jednakowo istotne. Podstawową funkcją jaką oferuje Matlab do rozwiązania zadania aproksymacji jest funkcja **polyfit**.

a = polyfit(x, y, n) - znajduje współczynniki wielomianu n-tego stopnia, który w sensie najmniejszych kwadratów najlepiej pasuje do serii danych pomiarowych (x, y).

Uwaga: do obliczania wartości wielomianu $W(x, a)=a_1x^n+ a_2x^{n-1}+...+ a_nx+ a_{n+1}$ o współczynnikach **a** w punkcie x można wykorzystać funkcję polyval(a, x), gdzie a(1) jest współczynnikiem stojącym przy x^n, a(2) jest współczynnikiem stojącym przy x^{n-1}, ..., a(n+1)jest wyrazem wolnym.

Do poprawy przybliżenia w wybranych węzłach może być wykorzystana funkcja wagowa $w(x_i)$. Zwykle jest ona tożsamościowo równa jedności, zatem wszystkie odchyłki są jednakowo istotne. Podstawową funkcją jaką oferuje Matlab do rozwiązania zadania aproksymacji jest funkcja **polyfit**.

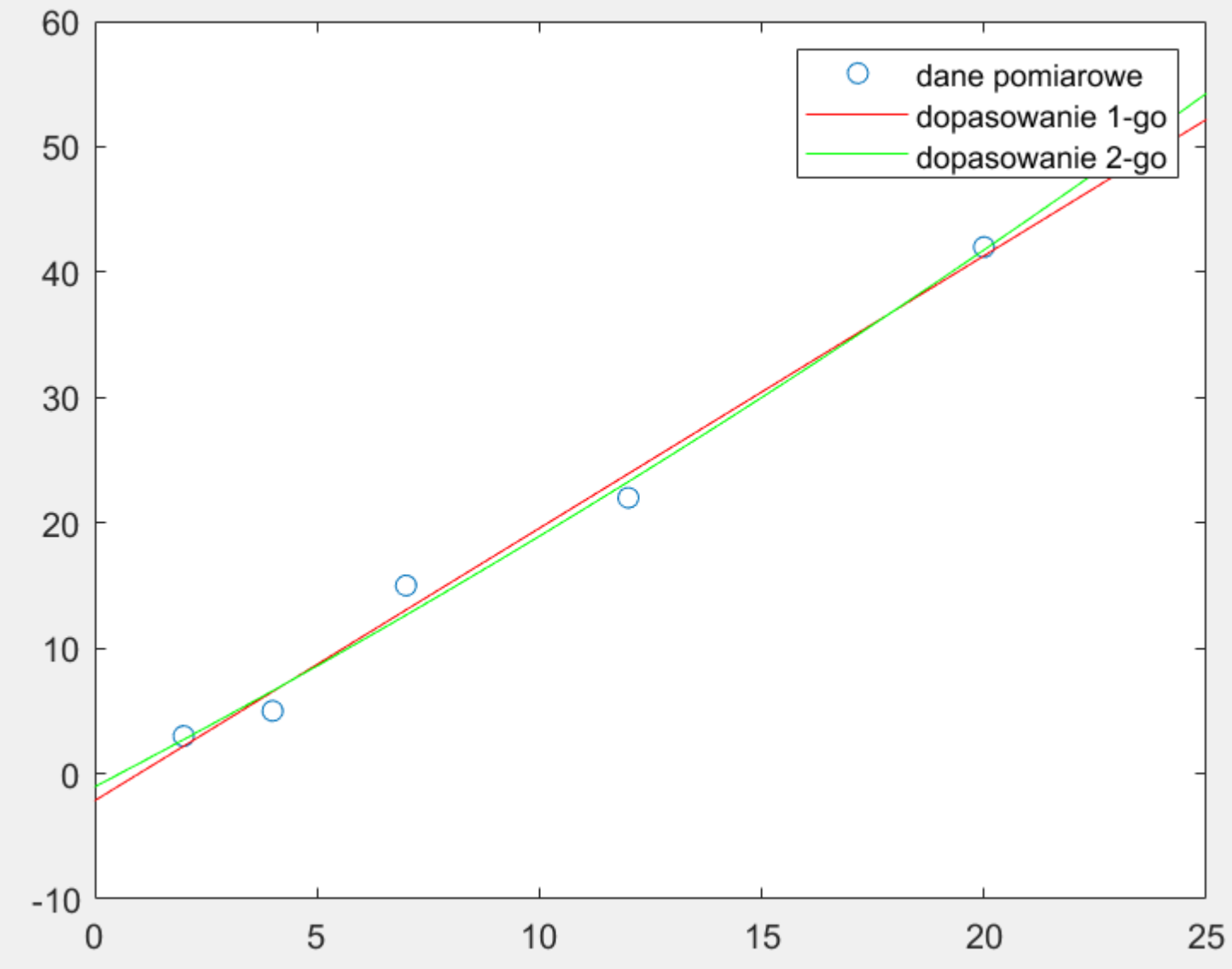
a = polyfit(x, y, n) - znajduje współczynniki wielomianu n-tego stopnia, który w sensie najmniejszych kwadratów najlepiej pasuje do serii danych pomiarowych (x, y).

Uwaga: do obliczania wartości wielomianu $W(x, a)=a_1x^n+ a_2x^{n-1}+...+ a_nx+ a_{n+1}$

o współczynnikach **a** w punkcie x można wykorzystać funkcję polyval(a, x), gdzie a(1) jest współczynnikiem stojącym przy x^n, a(2) jest współczynnikiem stojącym przy x^{n-1}, ..., a(n+1)jest wyrazem wolnym.

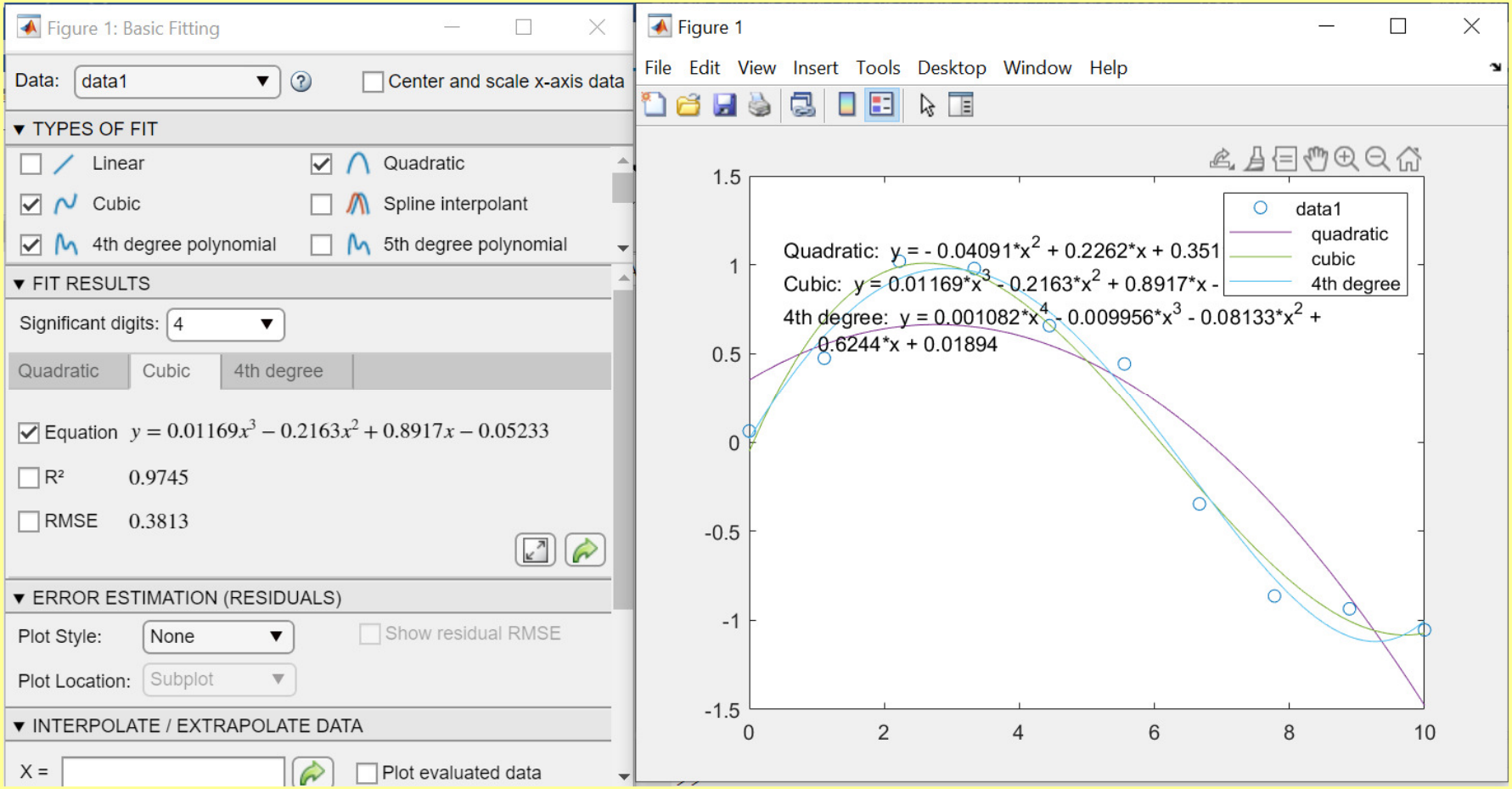
Przykład:

```
x=[2 4 7 12 20]; y=[3 5 15 22 42];
figure, plot(x, y,'o');
a=polyfit(x,y,1);  b=polyfit(x,y,2);
xt=0:25;
yt=polyval(a,xt);
yt1=polyval(b,xt);
```

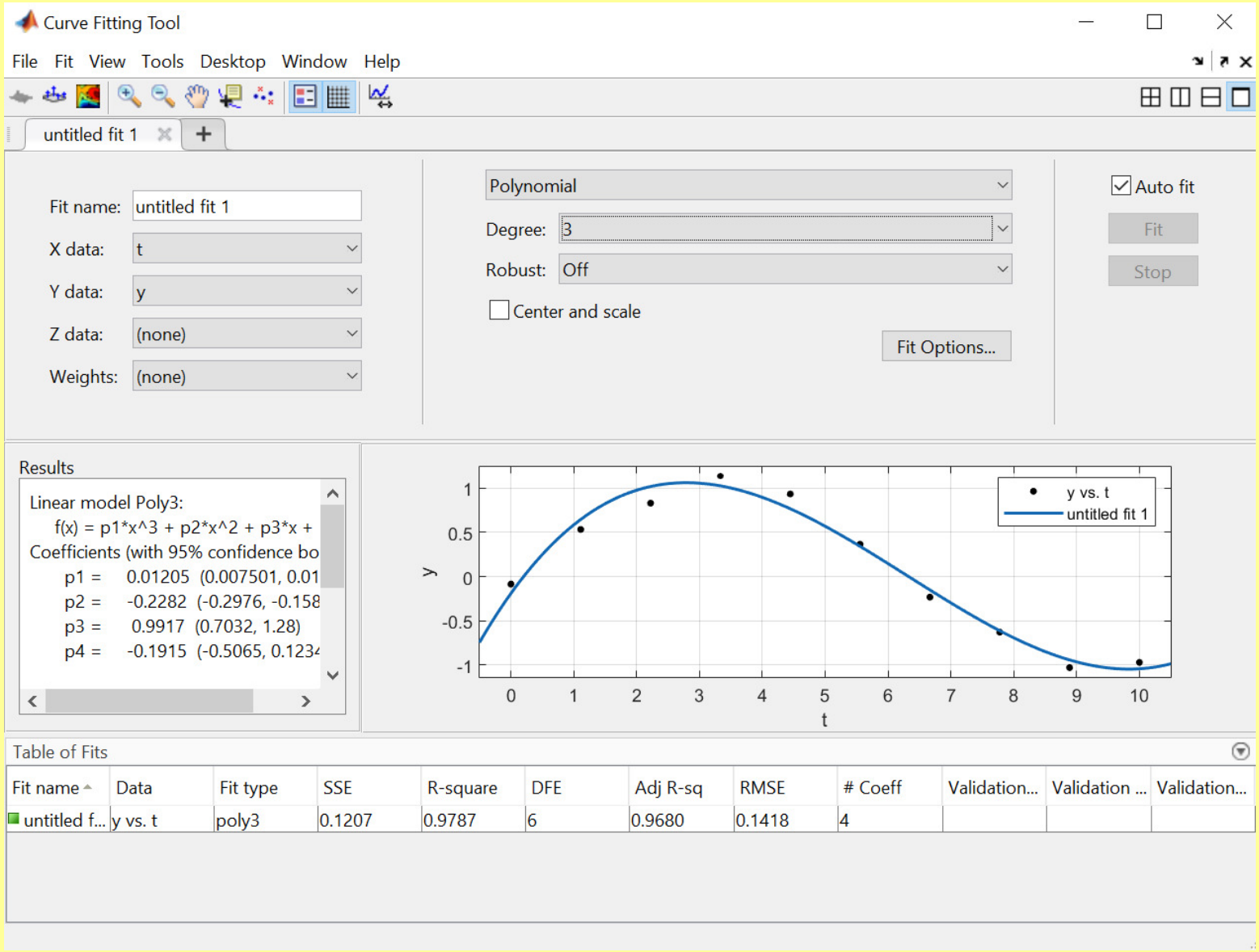

dane pomiarowe
dopasowanie 1-go
dopasowanie 2-go
60
50
40
30
20
10
0
-10
0
5
10
15
20
25

Poza funkcją polyfit Matlab daje użytkownikowi narzędzie **BasicFitting**, uruchamiane w menu *Tools* w oknie graficznym.

Z menu *Tools* wybieramy *Basic Fitting*. Zaznaczamy rodzaje dopasowania, np. q*uadratic, qubic, itp.*, a następnie zaznaczamy opcje: *Show equations, Plot residuals* oraz *Show norm of residuals*.



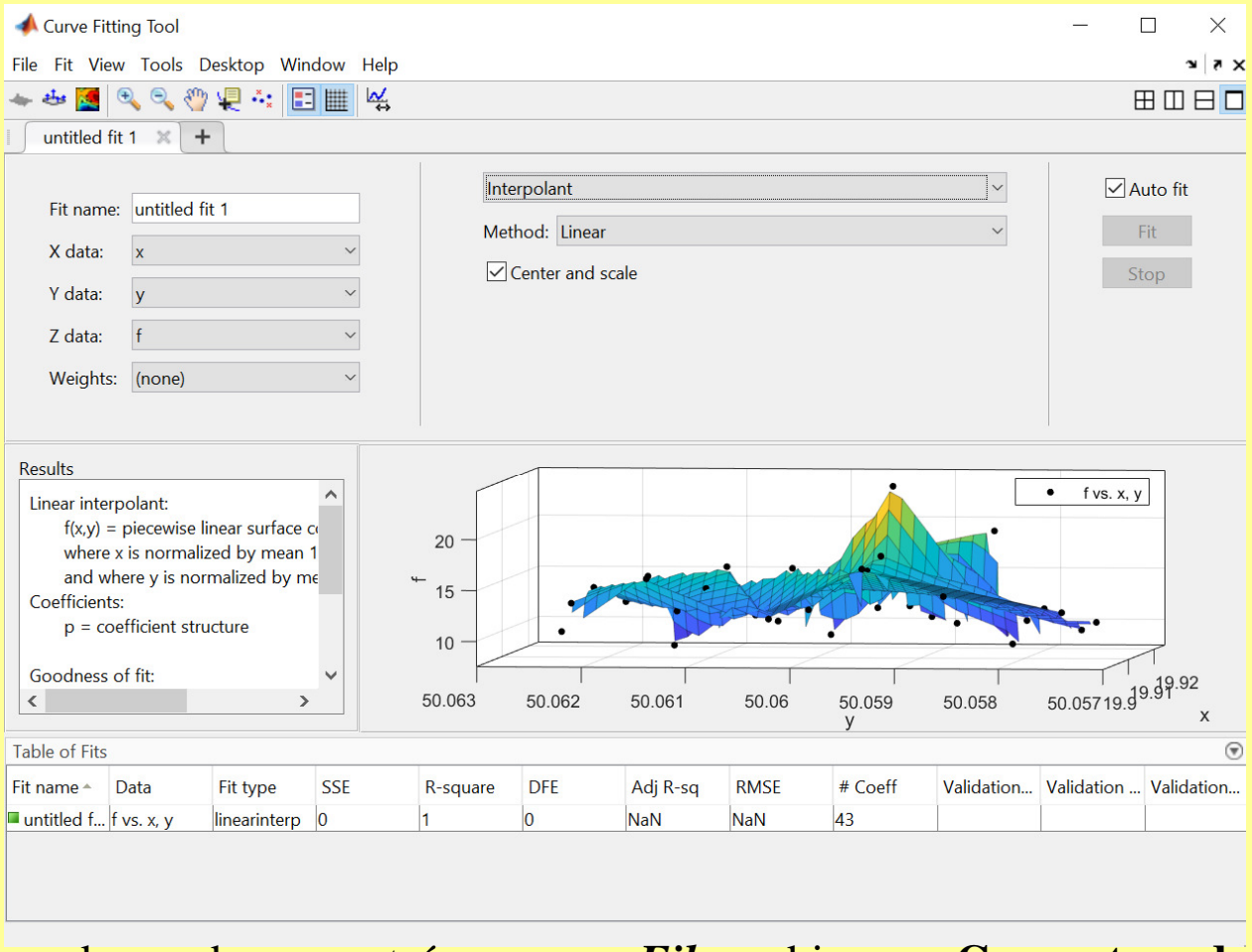
Kolejny sposób to wpisanie z linii komend polecenia ***cftool***.



Fitowanie powierzchni

Wczytujemy z pliku dane Z=f(X, Y) i próbujemy dopasować powierzchnię.

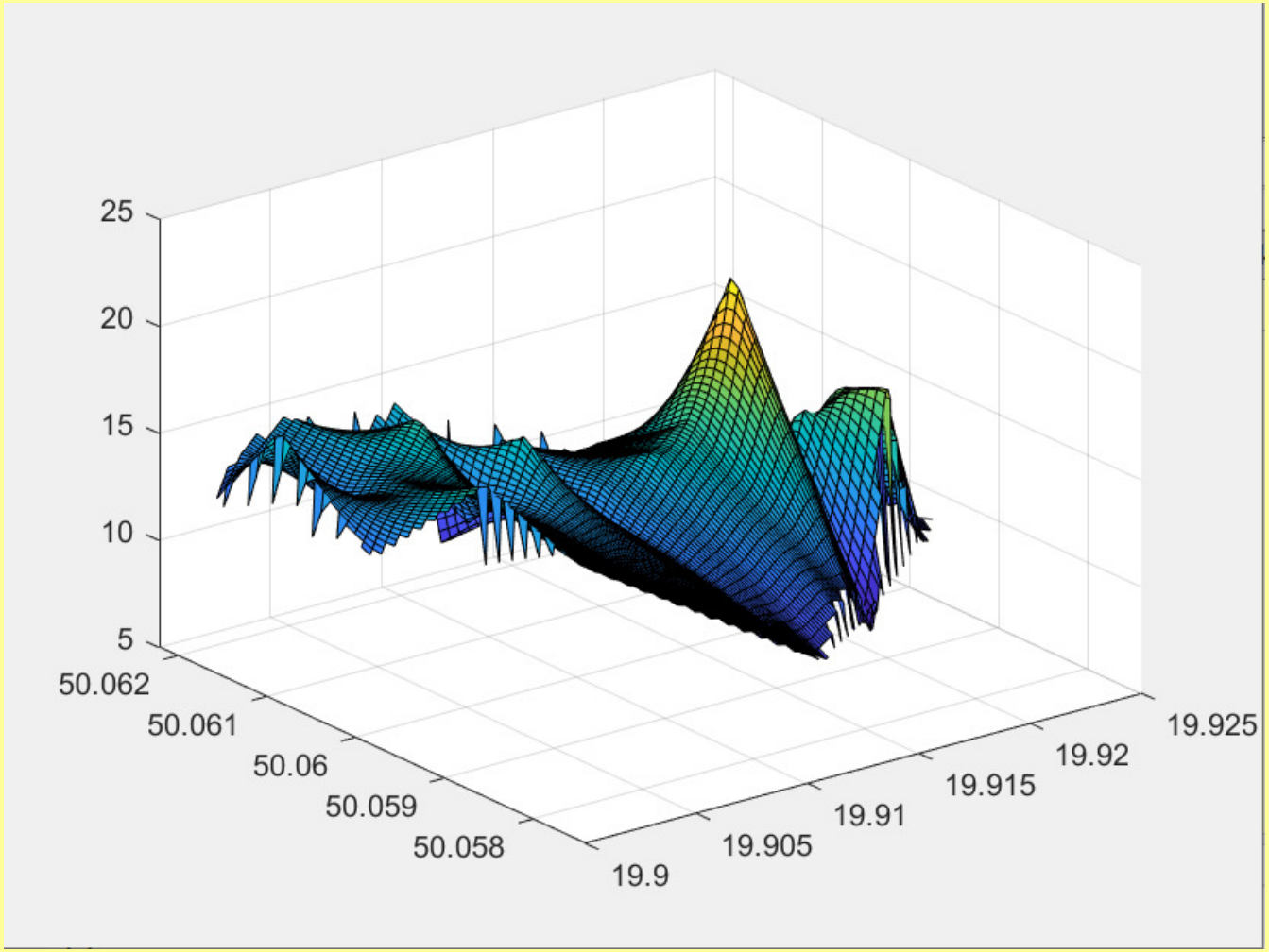
1. Wykorzystanie narzędzia *sftool*



Po ustawieniu wybranych parametrów z menu ***File*** wybieramy **Generate code –** powstaje funkcja (m-plik) **createSurtaceFit**, którą można uruchamiać z nowym zestawem danych: **createSurtaceFit(x, y, f)**

2. Polecenie TriScatteredInterp

```
[X Y]=meshgrid(19.9:0.0001:19.925,50.057:0.0001:50.063);
A=TriScatteredInterp(x,y,f,'natural');
Z=A(X,Y);
surf(X,Y,Z);
```



3.Polecenie griddata

```
[XX YY]=meshgrid(19.9:0.00001:19.925,50.057:0.00001:50.063);
ZZ=griddata(x,y,f,XX,YY,'cubic');% lub linear lub nearest
surf(XX,YY,ZZ);
```