

<p style="text-align: center;">Politechnika Świętokrzyska Wydział Elektrotechniki, Automatyki i Informatyki Katedra Zastosowań Informatyki</p>	
Metody obliczeniowe – laboratorium	
Instrukcja laboratoryjna nr 5B: Interpolacja - matlab	Opracował: <i>dr inż. Andrzej Kułakowski</i> Data: <i>23.10.2013 r.</i>

6. Implementacja w Matlabie

Wzory z instrukcji A pozwalają napisać procedurę (funkcję), która oblicza wartości wielomianu interpolacyjnego stopnia n w dowolnym punkcie x leżącym wewnątrz przedziału $[a, b]$ różnym od punktu węzłowego.

```
function y=lagrange(xj,yj,x,n)
%LAGRANGE wzór interpolacyjny Lagrange'a
%LAGRANGE(xj,yj,x,n) oblicza wartości wielomianu interpolacyjnego stopnia n
% w dowolnym punkcie x dla węzłów xj i odpowiadających im wartości
% funkcji interpolowanej yj.
% np.: y=lagrange([1 2 3],[1 2 3],1.7,2) daje y=1.7

omega_w=1;s=0;
for j=1:n+1
    omega_w=omega_w*(x-xj(j));
    omega_p=1;
    for i=1:n+1
        if i~=j
            omega_p=omega_p*xj(j)-xj(i);
        end
    end
    s=s+yj(j)/(omega_p*(x-xj(j)));
end
y=omega_w*s;
end
```

Listing 1. Funkcja Lagrange wykorzystana w listingu 2.

Zauważmy, że rząd wielomianu może być co najwyżej równy liczbie węzłów pomniejszonej o 1. Oznacza to, że jeżeli wektory x_j oraz y_j mają tak jak w przykładzie trzy elementy (podajemy 3 węzły interpolacji) funkcją interpolującą może być co najwyżej funkcją kwadratowa. Powyższą funkcję zapisaną w pliku `lagrange.m` wykorzystać można do wykreślenia wykresu funkcji interpolującej, np. funkcji: $1 / (1 + 25x^2)$.

```

x=-1:0.01:1; y=1./(1+25*x.^2);
ia=[1:100:201];
ib=[1:50:201];
ic=[1:25:201];

xia=x(ia); yia=y(ia);
xla=x; xla(ia)=[];
for k=1:198
    ya(k)=lagrange(xia,yia,xla(k),2);
end

xib=x(ib); yib=y(ib);
xlb=x; xlb(ib)=[];
for k=1:196
    yb(k)=lagrange(xib,yib,xlb(k),4);
end

xic=x(ic); yic=y(ic);
xlc=x; xlc(ic)=[];
for k=1:192
    yc(k)=lagrange(xic,yic,xlc(k),8);
end

subplot(2,2,1)
plot(x,y)
ylim([-1.5 1.5])
title('funkcja interpolowana:  $(1+25x^2)^{-1}$ ')

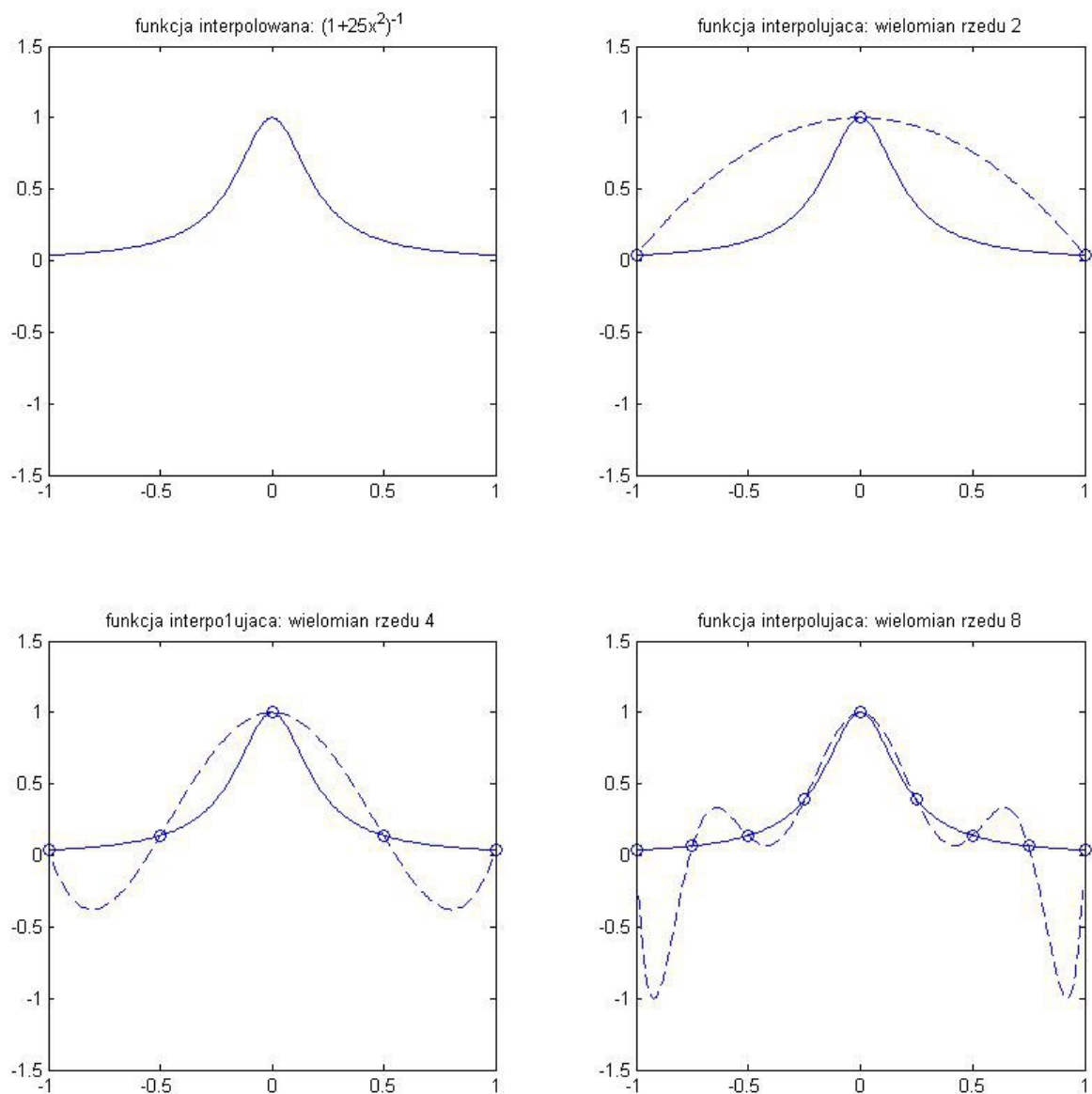
subplot(2,2,2)
plot(xla,ya,'--'), hold on, plot(xia,yia,'o'), plot(x,y)
ylim([-1.5 1.5])
title('funkcja interpolujaca: wielomian rzędu 2')

subplot(2,2,3)
plot(xlb,yb,'--'), hold on, plot(xib,yib,'o'), plot(x,y)
ylim([-1.5 1.5])
title('funkcja interpolujaca: wielomian rzędu 4')

subplot(2,2,4)
plot(xlc,yc,'--'), hold on, plot(xic,yic,'o'), plot(x,y)
ylim([-1.5 1.5])
title('funkcja interpolujaca: wielomian rzędu 8')

```

Listing 2. Skrypt obliczający i wykreślający wykres funkcji interpolacyjnej.



Rys B1. Wynik działania skryptu z listingu 2

7. Metody wbudowane

Interpolacja wykorzystywana jest często podczas wizualizacji danych pomiarowych lub wyników obliczeń. Ponieważ jednym z głównych zastosowań pakietu Matlab jest graficzna prezentacja danych, został on wyposażony w rozbudowane funkcje interpolacyjne. Narzędzia podzielić można na dedykowane interpolacji:

- danych jednowymiarowych,
- danych dwuwymiarowych,
- danych trój- i wielowymiarowych,
- macierzy rzadkich.

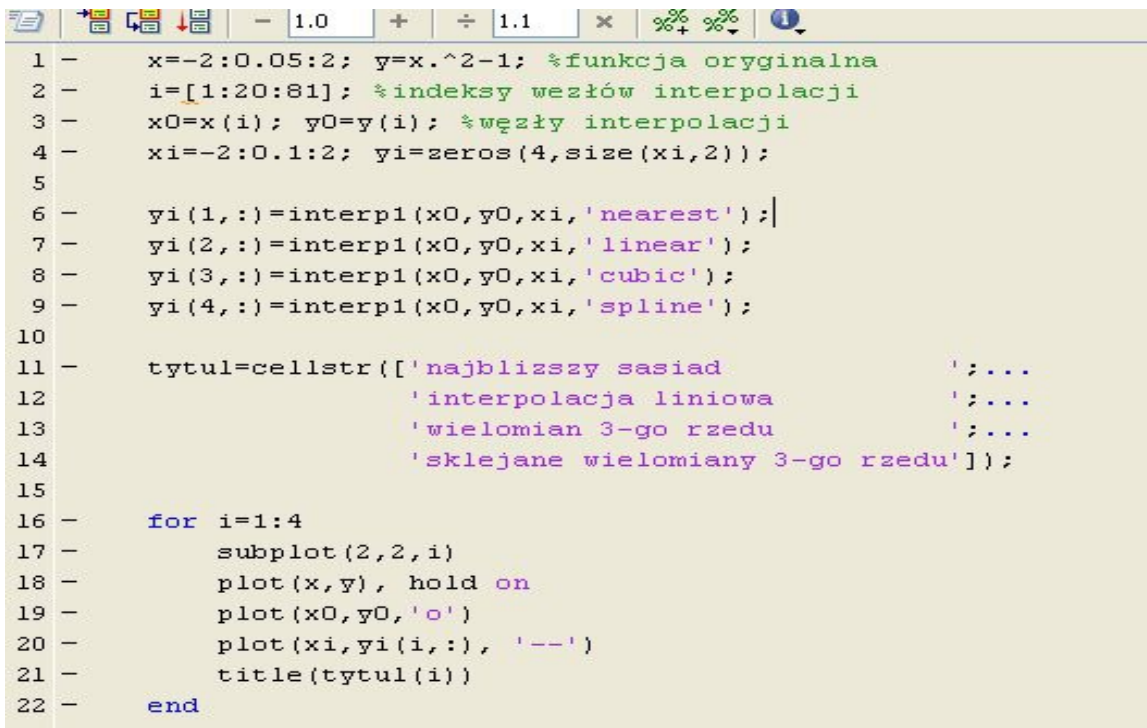
Z metod interpolacji jednowymiarowej dostępna jest interpolacja wielomianowa w postaci polecenia `interp1`. Jego wywołanie ma następującą postać:

$$y_i = \text{interp1}(x, y, x_i, \text{metoda})$$

gdzie:

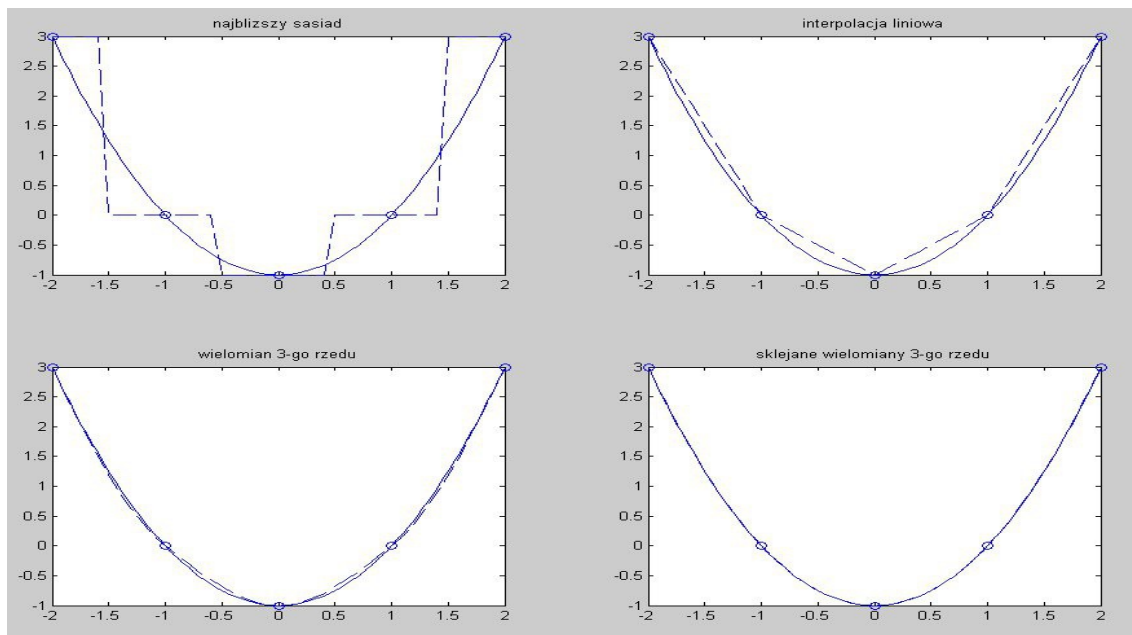
- x oraz y są wektorami reprezentującymi dyskretne wartości interpolowanej funkcji (węzły interpolacji),
- x_i jest wektorem wartości zmiennej niezależnej, dla których wyznaczone mają zostać wartości funkcji interpolującej,
- y_i jest wektorem wartości obliczonych przy pomocy wybranej metody,
- metoda to zmienna tekstowa określająca wybraną metodę; może ona przybierać wartości: 'nearest' (wartość najbliższego punktu); 'linear' (interpolacja liniowa, jest to metoda domyślna), 'pchip' lub 'cubic' (interpolacja wielomianem trzeciego rzędu), 'spline' (interpolacja sklejanymi wielomianami trzeciego stopnia).

Na poniższych skryptach przedstawiono działanie metod wbudowanych w system MATLAB.



```
1 - x=-2:0.05:2; y=x.^2-1; %funkcja oryginalna
2 - i=[1:20:81]; %indeksy węzłów interpolacji
3 - x0=x(i); y0=y(i); %węzły interpolacji
4 - xi=-2:0.1:2; yi=zeros(4,size(xi,2));
5
6 - yi(1,:)=interp1(x0,y0,xi,'nearest');|
7 - yi(2,:)=interp1(x0,y0,xi,'linear');
8 - yi(3,:)=interp1(x0,y0,xi,'cubic');
9 - yi(4,:)=interp1(x0,y0,xi,'spline');
10
11 - tytuł=cellstr(['najbliższy sąsiad'           ';...
12                'interpolacja liniowa'         ';...
13                'wielomian 3-go rzędu'          ';...
14                'sklejane wielomiany 3-go rzędu']);
15
16 - for i=1:4
17 -     subplot(2,2,i)
18 -     plot(x,y), hold on
19 -     plot(x0,y0,'o')
20 -     plot(xi,yi(i,:), '--')
21 -     title(tytuł(i))
22 - end
```

Listing 3. Przykład wykorzystania metod wbudowanych



Rys B2. Wynik działania skryptu z listingu 3

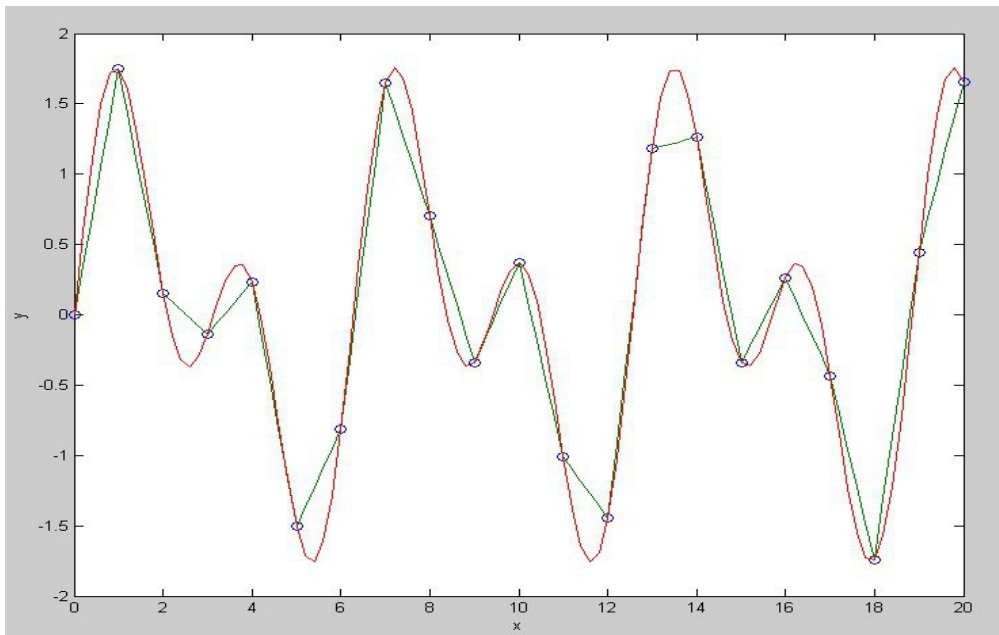
Inny przykład pokazujący dokładność interpolacji przy wykorzystaniu funkcji trygonometrycznej, a raczej punktów węzłowych otrzymanych z takiej funkcji.

```

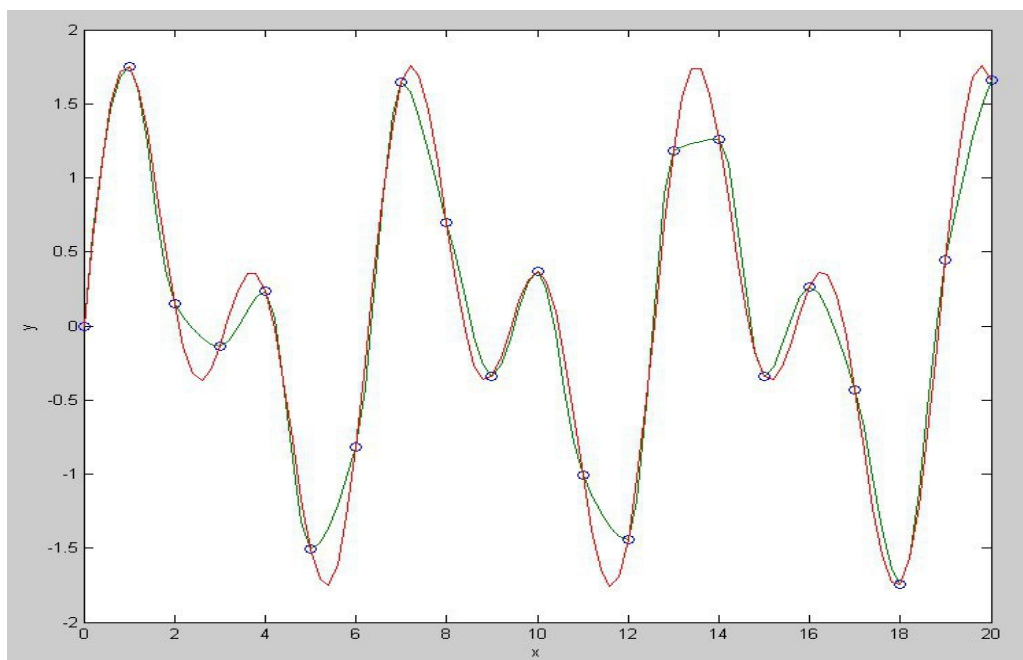
1 - x=0:20;
2 - y=sin(x)+sin(2*x);
3 - xi=0:.2:20;
4 - yi=interp1(x,y,xi,'linear');
5 - plot(x,y,'o',xi,yi,xi,sin(xi)+sin(2*xi))
6 - xlabel('x')
7 - ylabel('y');
8 - figure(2)
9 - yi=interp1(x,y,xi,'cubic');
10 - plot(x,y,'o',xi,yi,xi,sin(xi)+sin(2*xi))
11 - xlabel('x')
12 - ylabel('y')
13 - figure(3)
14 - yi=interp1(x,y,xi,'spline');
15 - plot(x,y,'o',xi,yi,xi,sin(xi)+sin(2*xi))
16 - xlabel('x')
17 - ylabel('y')

```

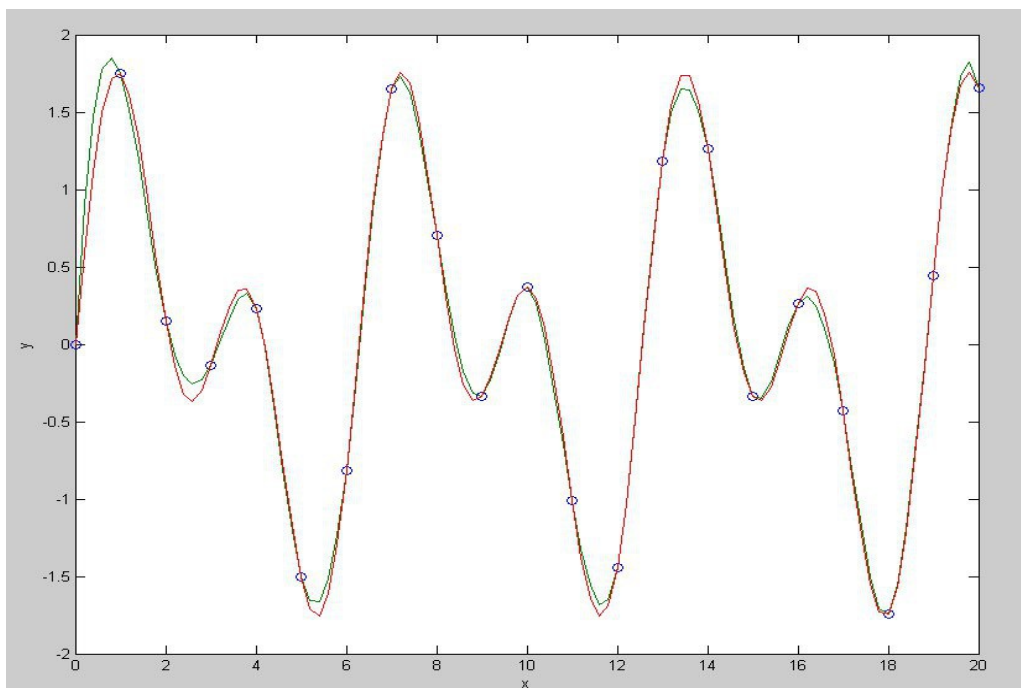
Listing 4. Przykład wykorzystania metod wbudowanych do interpolacji przy wykorzystaniu funkcji trygonometrycznej



Rys B3. Wynik działania metody `linear` ze skryptu z listingu 4



Rys B4. Wynik działania metody `cubic` ze skryptu z listingu 4



Rys B5. Wynik działania metody `spline` ze skryptu z listingu 4

8.Literatura

literatura z instrukcji A,

literatury podana dla przedmiotu Metody Obliczeniowe,

Stachurski M.: Metody numeryczne w programie Matlab, Mikom, Wa-wa 2003.

Zalewski A., Cegiela R.: Matlab – obliczenia numeryczne i ich zastosowania, Nakom, Poznań, 1996.