

2015/2016

Metody i analiza danych

Funkcje, pętle i grafika

Laboratorium
komputerowe
3

Anna Kielbus

Zakres tematyczny

1. Funkcje i skrypty
Pętle i instrukcje sterujące
2. Grafika dwuwymiarowa
3. Grafika trójwymiarowa
4. Zadanie.

1. Funkcje i skrypty

Ćwiczenie 1

Napisz skrypt (otwierając z menu File z opcji New plik M-file), który kreśli wykres wybranej przez użytkownika funkcji jednej zmiennej w przedziale $< 0, 4\pi >$

```
% skrypt rysuje wykres wybranej funkcji
x=[0:0.1:4*pi];
wzor=input('Podaj wzór funkcji jednej zmiennej f(x): ','s')
y=eval(wzor);
plot(x,y);      % kreślenie wykresu funkcji y=f(x)
```

Zapisz go pod nazwą wykres.m, a następnie uruchom wpisując w oknie komend jego nazwę:
» wykres

WSKAZÓWKA: Podaj na przykład funkcję: $\sin(x)+2*\cos(2*x)$

a) operatory logiczne w języku MATLAB:

```
= = równe
~ = różne
< = mniejsze
> = większe
< = = mniejsze równe
> = = większe równe
& i
| lub
```

b) instrukcje sterujące

- **Pętla FOR („dla”):**

```
for zmienna_iterowana = macierz_wartości
    ciąg_instrukcji
end
```

Działanie pętli polega na wykonaniu *ciągu_instrukcji* dla kolejnych wartości *zmiennej_iterowanej*. Wartościami tymi są kolejne wektory kolumnowe pobrane z *macierzy_wartości* (jeżeli jest to wektor, to kolejno zostaną wykonane instrukcje dla danych elementów tego wektora).

Ćwiczenie 2

Napisz skrypt, który generuje wektor **A** o wymiarze 1x5, którego elementy spełniają zależność:

$$A_i = \sqrt{1+i}$$

```
% Próba realizacji pętli FOR
for i=1:5
A(i)=sqrt(1+i); % pierwiastek kwadratowy

end
A
```

Zapisz go w pliku petlafor.m i uruchom.

Rozbuduj powyższy skrypt, aby generował macierz **A** o wymiarze 10x5, którego elementy spełniają zależność:

$$A_{ij} = \sqrt{1 + \frac{i}{j}}$$

```
% Próba realizacji pętli FOR for i=1:10 for j=1:5 A(i,j)=sqrt(1+i/j); % pierwiastek
kwadratowy end
end
A
```

- **Pętla WHILE („dopóki”):**

while wyrażenie_warunkowe
ciąg_instrukcji
end

Działanie pętli polega na wykonaniu *ciągu_instrukcji* dopóki wyrażenie_warunkowe jest spełnione.

Ćwiczenie 3

```
% Próba realizacji pętli WHILE
i=0;
while i<100
i=i+1
end
```

Zapisz skrypt w pliku petlawhile.m i uruchom go.

- **Instrukcja warunkowa IF („jeżeli”):**

```
if wyrażenie_warunkowe1
ciąg_instrukcji1
elseif wyrażenie_warunkowe2
ciąg_instrukcji2
else ciąg_instrukcji3
end
```

Działanie instrukcji jest następujące: Jeżeli wyrażenie_warunkowe1 jest spełnione, to wykonywany jest *ciąg_instrukcji1*, w przeciwnym razie sprawdzane jest

wyrażenie `warunkowe2`, jeżeli jest ono spełnione wykonywany jest `ciąg_instrukcji2`, jeżeli nie, wykonywany jest `ciąg_instrukcji3`.

Instrukcję warunkową IF można rozbudować dla większej liczby `wyrażeń_warunkowych` i odpowiadających im `ciągów_instrukcji`.

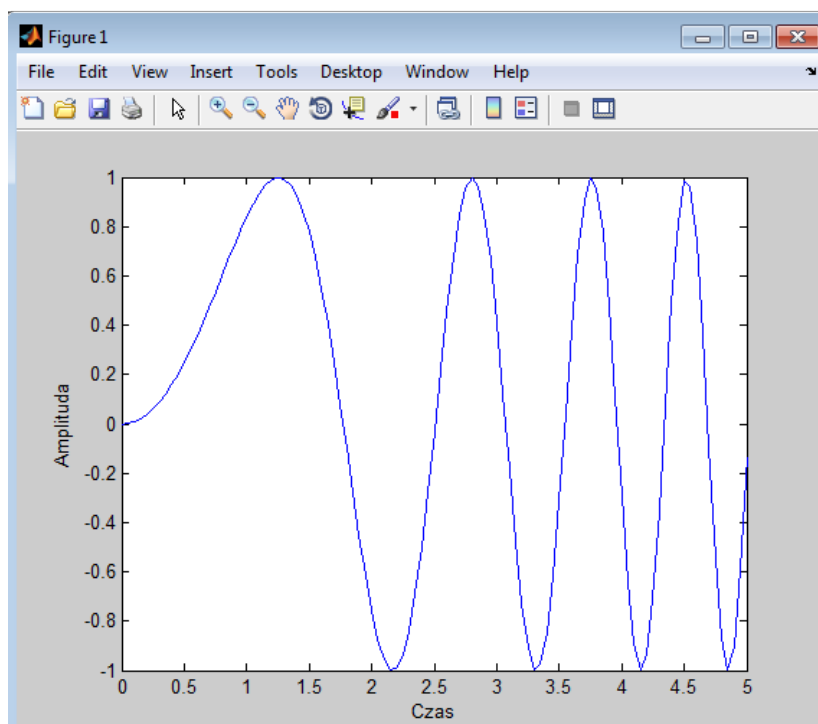
2. Grafika dwuwymiarowa

- Najczęściej spotykanym sposobem graficznej prezentacji danych w języku MATLAB jest wykres funkcji jednej zmiennej. Służy do tego funkcja `plot(x,y)`, gdzie $y=f(x)$;
- Okno graficzne można wyczyścić wywołując funkcję `clf`;
- Zamknięcie okna graficznego odbywa się poprzez wywołanie funkcji `close`;
- Dodatkowe okna można otworzyć przy pomocy funkcji `figure`;
- Otworzyć jak i zamknąć można dowolne okno podając jego numer jako argument;
- W celu uzyskania kilku wykresów w jednym oknie należy wykorzystać funkcję `subplot(m,n,p)`,
gdzie:
m - liczba wykresów w pionie;
n - liczba wykresów w poziomie;
p - kolejny numer wykresu.
- Skala wykresu dobierana jest automatycznie. Chcąc ją zmienić, trzeba wywołać funkcję `axis([xmin xmax ymin ymax])` i jako argument podać wektor określający nowe parametry osi.
- Wykres można opisać podając nazwy zmiennych, tytuł, itp.
`title('tekst')` - tytuł rysunku;
`xlabel('tekst')` - opis osi x;
`ylabel('tekst')` - opis osi y;
`text(x,y,'tekst')` - umieszcza 'tekst' w dowolnym punkcie o współrzędnych (x,y);
`grid` - włącza lub wyłącza siatkę;

Ćwiczenie 4

Narysuj wykres dla funkcji $y=\sin(x.^2)$, gdzie x zawarte jest w przedziale $\langle 0,5 \rangle$, z krokiem 0.05. Oś x oznacz jako czas, oś y oznacz jako amplituda.

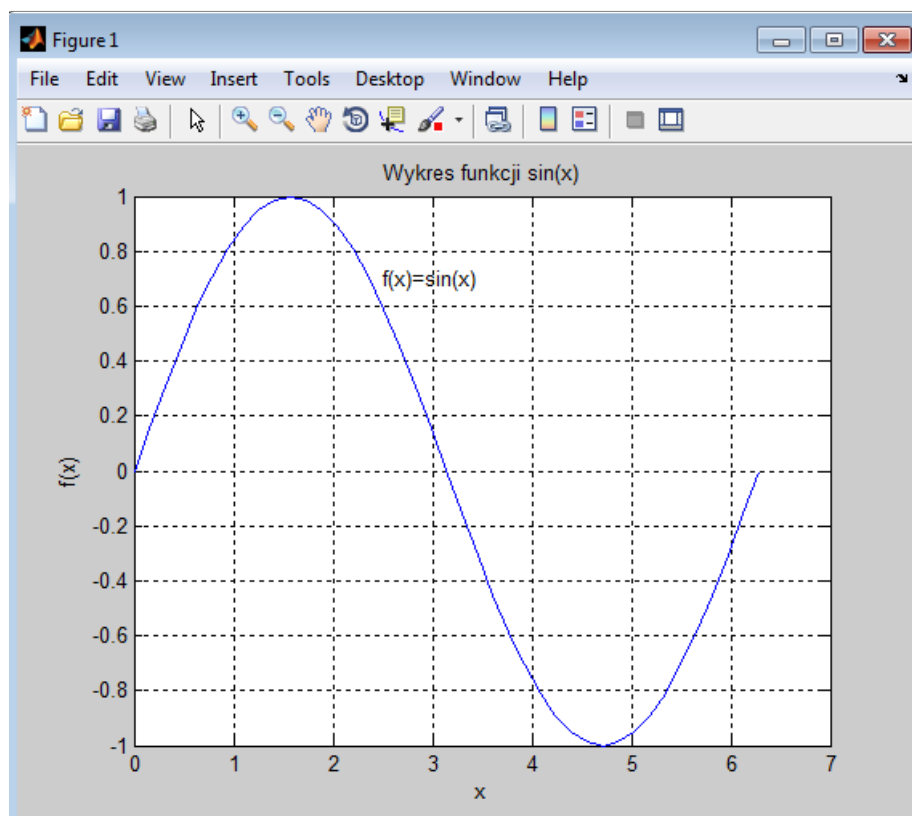
```
>> x = 0:0.05:5;
>> y = sin(x.^2);
>> plot(x,y);
>> xlabel('Czas')
>> ylabel('Amplituda')
```



Ćwiczenie 5

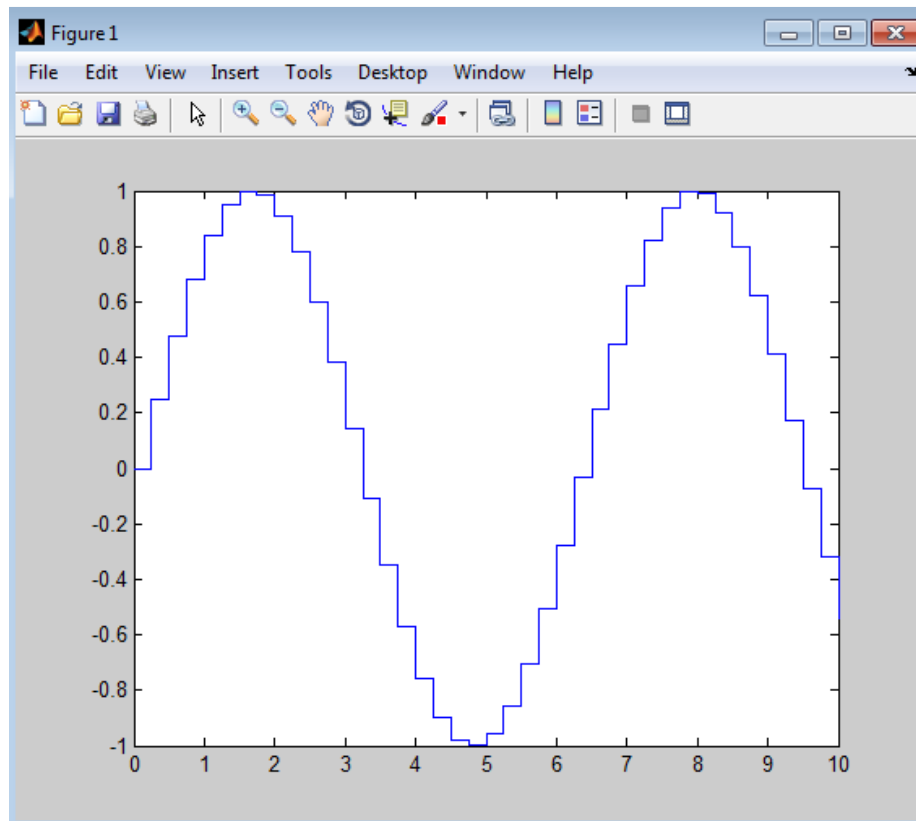
Napisz skrypt kreślący przykładowy wykres wraz z opisem. Zapisz go pod nazwą wykresopis.m i uruchom.

```
x=[0:pi/20:2*pi];  
y=sin(x);  
plot(x,y)  
title('Wykres funkcji sin(x)')  
xlabel('x')  
ylabel('f(x)')  
text(2.5,0.7,'f(x)=sin(x)')  
grid
```



Ćwiczenie 6

Narysuj wykres schodkowy dla funkcji $\sin(x)$.



```
>> x = 0:0.25:10;  
>> stairs(x,sin(x));
```

a) Rysowanie

Istnieją funkcje pozwalające na tworzenie dowolnych rysunków z linii i wielokątów.

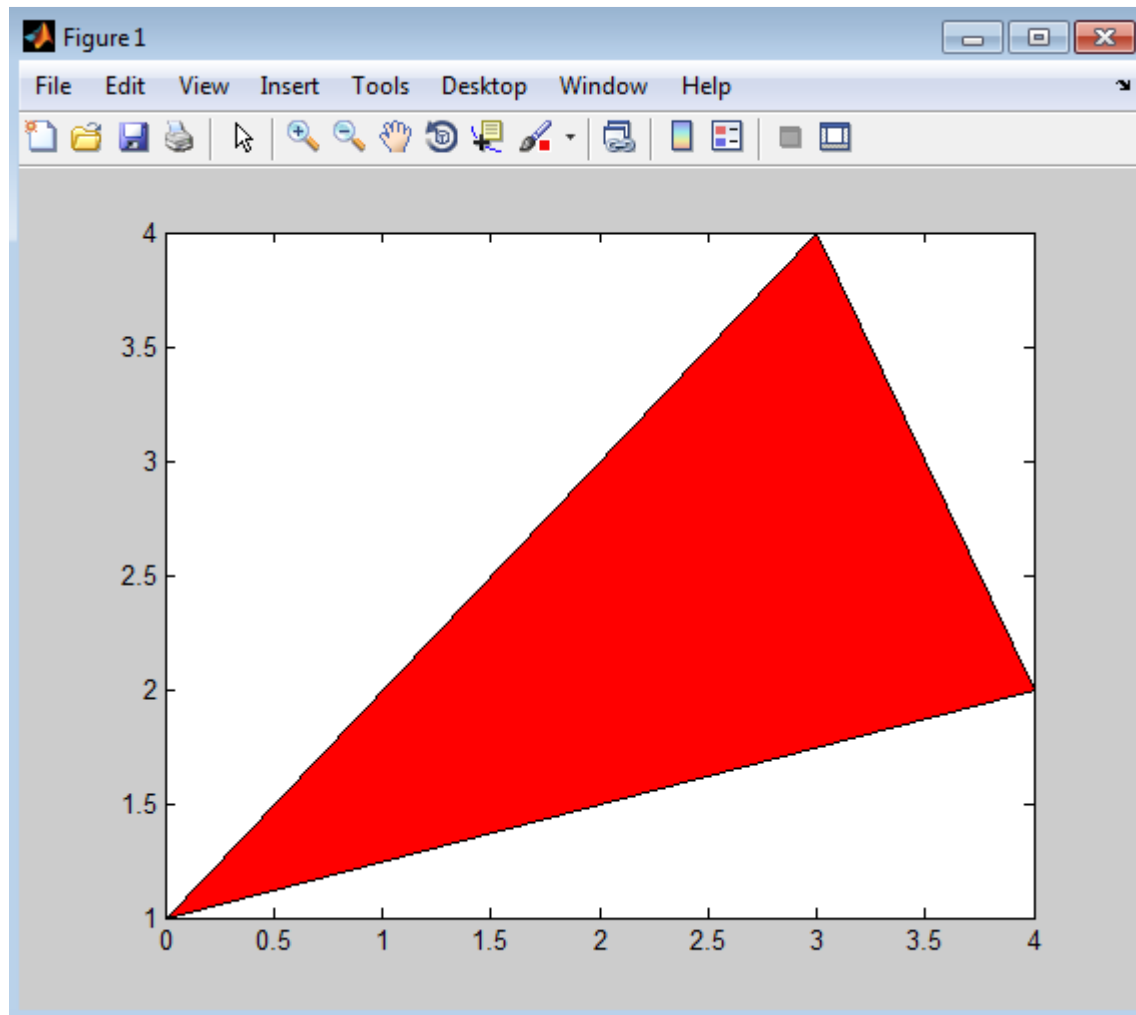
`line(x,y)` - rysuje linię łamaną łącząc wierzchołki punktów wyznaczonych przez elementy wektorów x i y ;

`fill(x,y,'c')` - rysuje wielokąt o wierzchołkach w punktach wyznaczonych przez elementy wektorów x i y wypełniony kolorem określonym przez argument c według poniższego opisu kolorów:

y - żółty
m - fioletowy
c - turkusowy
r - czerwony
g - zielony
b - niebieski
w - biały
k - czarny

Ćwiczenie 7

Narysuj trójkąt o wierzchołkach w punktach $(0,1)$, $(3,4)$, $(4,2)$ używając funkcji `line` oraz `fill` z wypełnieniem w kolorze czerwonym.



```
>> line([0 3 4 0],[1 4 2 1])
>> fill([0 3 4],[1 4 2],'r')
```

3. Grafika trójwymiarowa

Większość funkcji języka MATLAB generujących rysunki trójwymiarowe służy do kreślenia powierzchni. W praktyce definiując powierzchnię trzeba się ograniczyć do skończonego zbioru punktów należących do obszaru.

`[x,y]=meshgrid(X,Y)` - tworzy macierze `x` i `y` opisujące położenie węzłów prostokątnej siatki pobierając wartości z wektorów `X` i `Y`.

`mesh(x,y,z)` - rysuje siatkę powierzchni opisaną przez macierze `x`, `y` i `z`.

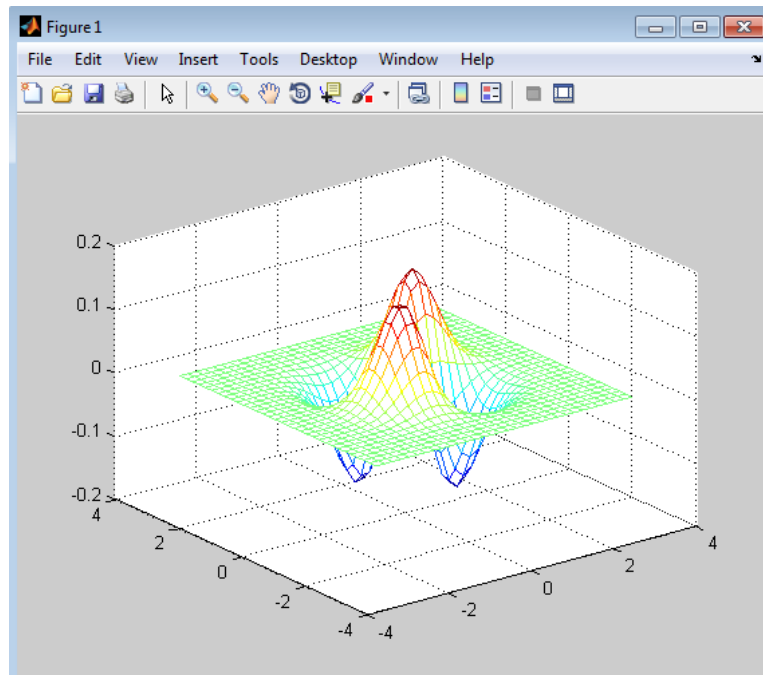
`surf(x,y,z)` - rysuje kolorową powierzchnię opisaną przez macierze `x`, `y` i `z`.

`surfl(x,y,z)` - rysuje kolorową powierzchnię opisaną przez macierze `x`, `y` i `z` uwzględniając na niej odbicie światła.

`plot3(x,y,z)` - rysuje krzywą w przestrzeni opisaną przez wektory `x`, `y` i `z`.

Ćwiczenie 8

Napisz skrypt kreślący siatkę wartości funkcji $f(x, y) = \sin(x) * \sin(y) * \exp(-x^2 - y^2)$ w przedziale $\langle -\pi, \pi \rangle$.



```
clf
```

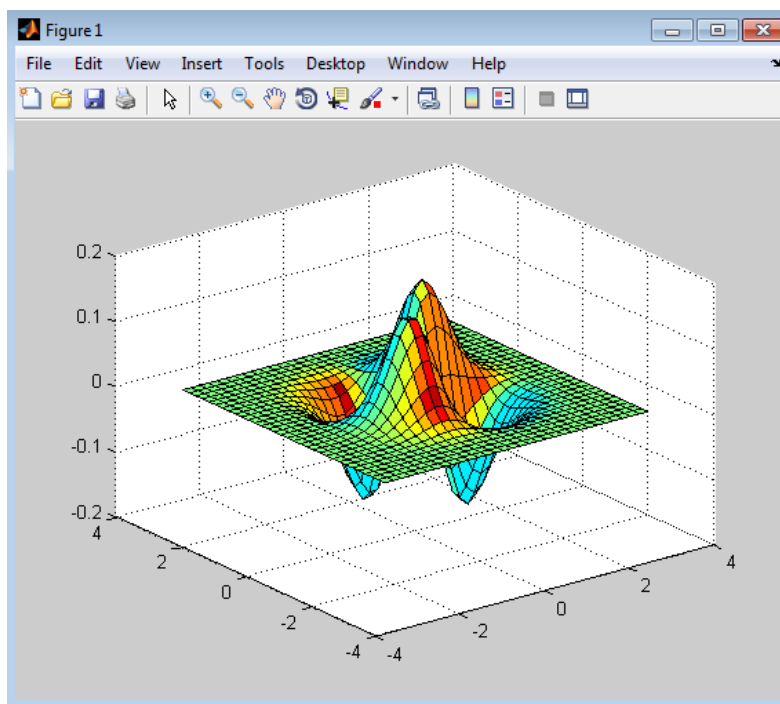
```
[x,y]=meshgrid(-pi:0.2:pi,-pi:0.2:pi)
```

```
z=sin(x).*sin(y).*exp(-x.^2-y.^2)
```

```
mesh(x,y,z)
```

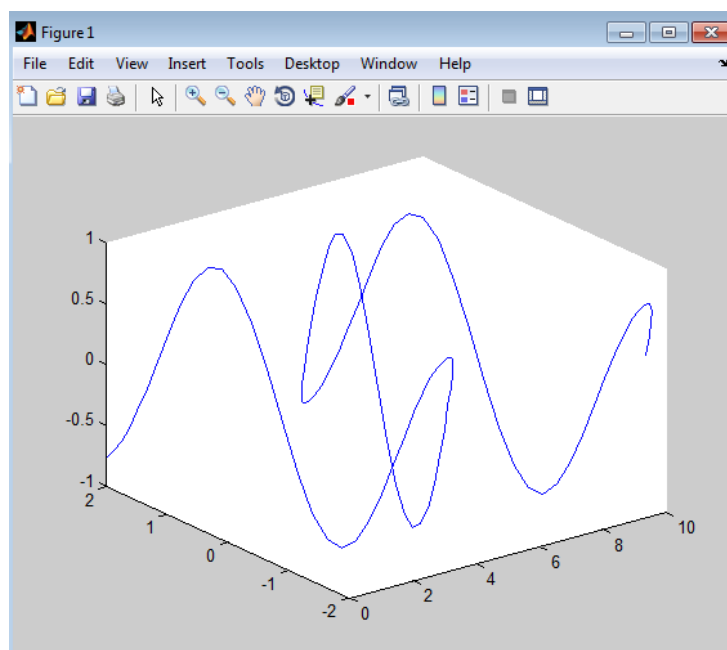
Zapisz go pod nazwą wykres3d.m i uruchom.

Rozbuduj powyższy skrypt o rysowanie kolorowej powierzchni poprzez dodanie na końcu polecenia: `surf(x,y,z)` lub: `surfl(x,y,z)`



Ćwiczenie 9

Napisz skrypt kreślący krzywą w przestrzeni trójwymiarowej.



% Skrypt kreśli krzywą w przestrzeni trójwymiarowej

```
x=[0:0.1:10];
```

```
y=2*cos(x);  
z=sin(2*y);  
plot3(x,y,z)  
grid  
title('Wykres krzywej w przestrzeni trójwymiarowej')  
xlabel('x')  
ylabel('y')  
zlabel('z')
```

Zapisz go pod nazwą krzywa3d.m i uruchom.

- Wykreślone powierzchnie można poddać cieniowaniu używając funkcji:

shading flat

shading interp

shading faceted

Ćwiczenie 10

Napisz skrypt:

% Skrypt rysuje powierzchnie poddane cieniowaniu

```
clf
```

```
[x,y]=meshgrid(-3.5:0.7:3.5);
```

```
z=sin(x).*sin(y)+4*exp(-(x-0.5).^2-(y-0.5).^2);
```

%Wykres w trybie flat

```
subplot(1,3,1)
```

```
surf(x,y,z)
```

shading flat

title('flat')

%Wykres w trybie interp

```
subplot(1,3,2)
surf(x,y,z)
shading interp
title('interp')
%Wykres w trybie faceted
subplot(1,3,3)
surf(x,y,z)
shading faceted
title('faceted')
```

Zapisz go pod nazwą powierzchnie.m i uruchom.

- Inne elementy rysunków, takie jak: opisy, etykiety, linie pomocnicze wykonuje się podobnie, jak w grafice dwuwymiarowej. Dodatkowo jednak należy zdefiniować elementy dotyczące trzeciego wymiaru, np.: `text(x,y,z,'tekst');`

5.Zadanie

Zadanie 1.Przykład programowania w języku MATLAB

```
%Zad 1
clear all
close all
clc

global a k c d
a=1;
k=2;
c=3;
```

```
d=4;
```

```
wart_posz={'a','k','c','d'};
```

```
tytul='Dane poszukiwane';
```

```
ilosc_lini=1;
```

```
wartosc={'1','2','3','4'};
```

```
rezultat=inputdlg(wart_posz,tytul,ilosc_lini,wartosc);
```

```
a=eval(rezultat{1})
```

```
k=eval(rezultat{2})
```

```
c=eval(rezultat{3})
```

```
d=eval(rezultat{4})
```

```
x=[0:10];
```

```
y1=sin(x)+a;
```

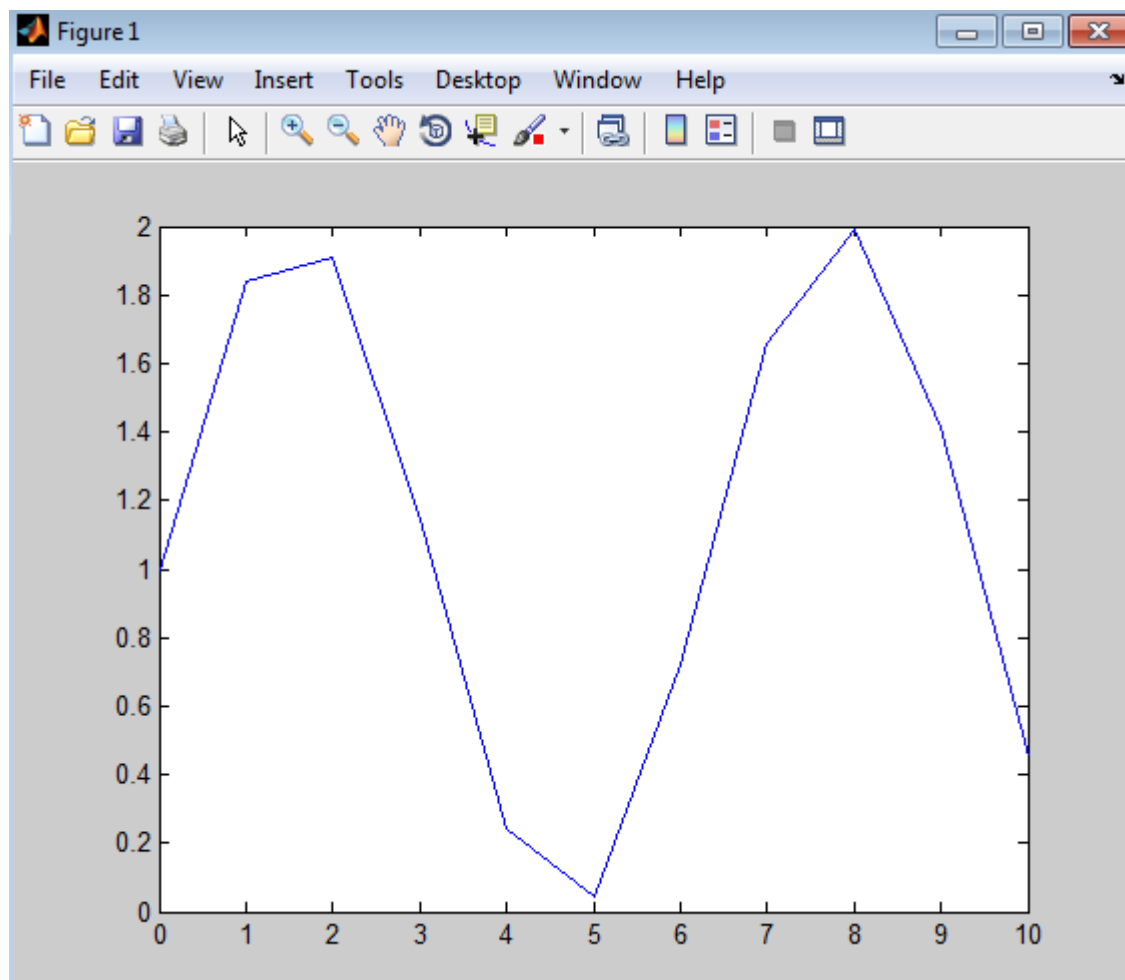
```
y2=cos(x)+c;
```

```
y3=tan(x)+d;
```

```
y4=exp(x)+k;
```

```
disp('Rysowanie charakterystyk')
```

```
disp('Naciśnij Enter aby narysować charakterystykę y1')
```

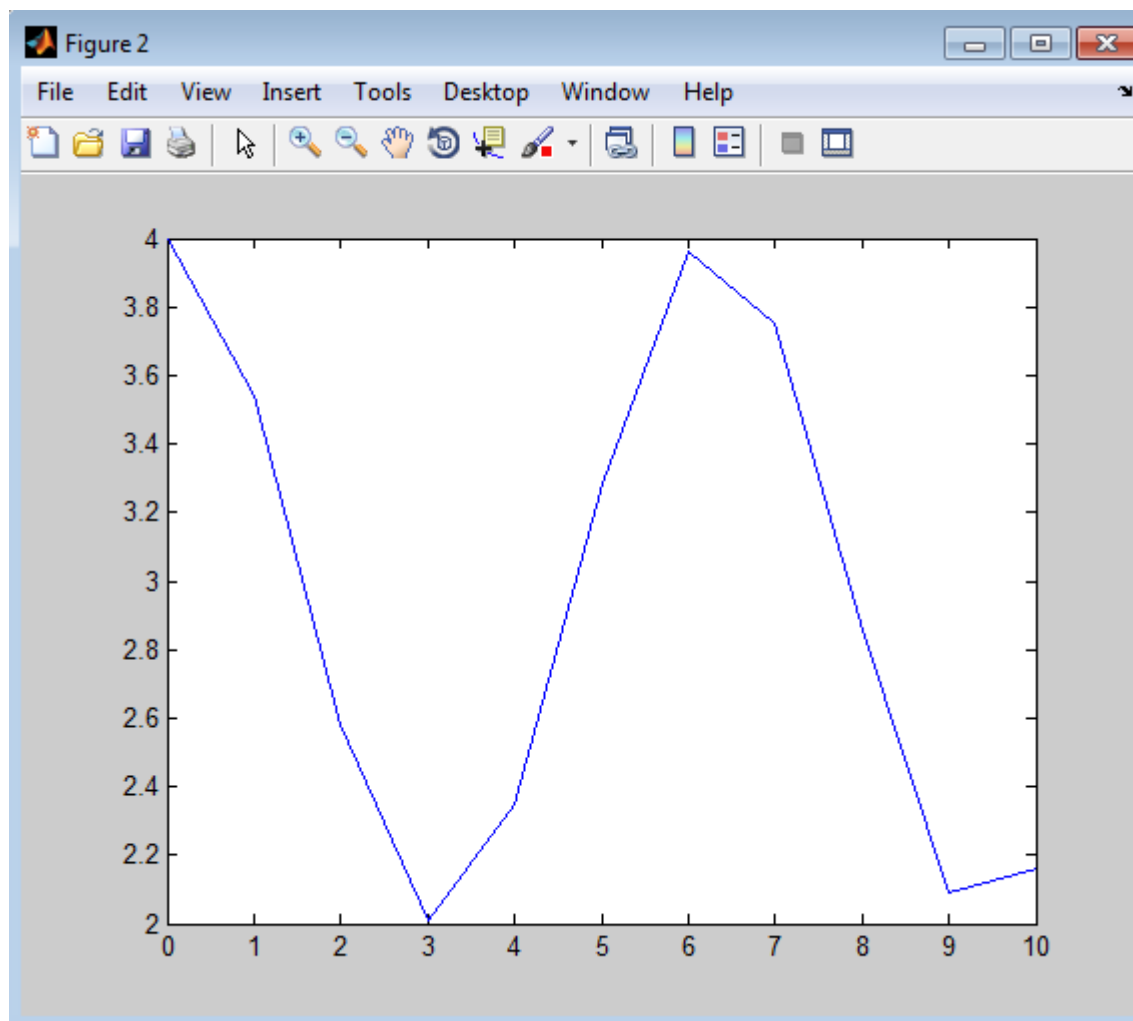


pause

figure(1)

plot(x,y1)

disp('Naciśnij Enter aby narysować charakterystykę y2')

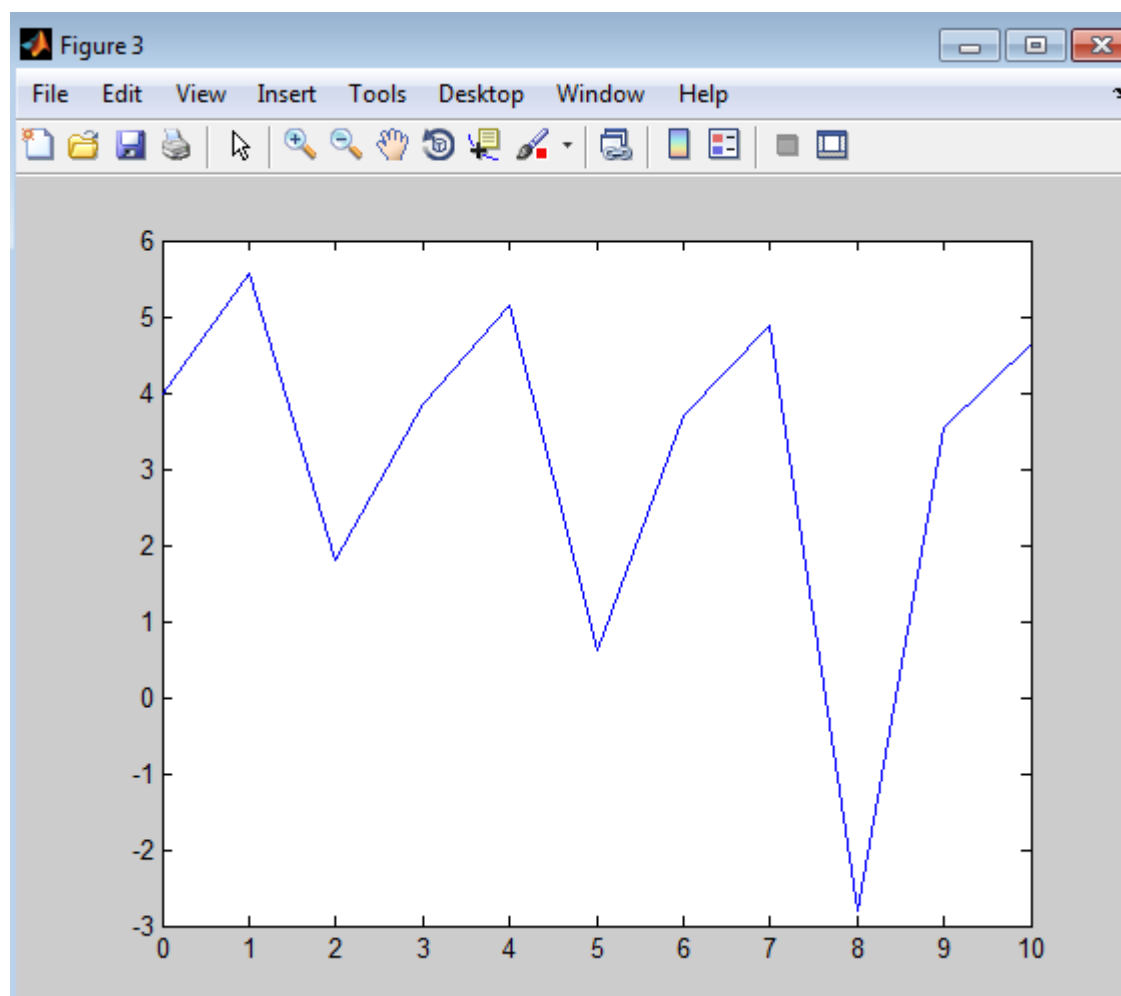


pause

figure(2)

plot(x,y2)

disp('Naciśnij Enter aby narysować charakterystykę y3')



```
pause
```

```
figure(3)
```

```
plot(x,y3)
```

```
disp('Naciśnij Enter aby narysować charakterystykę y4')
```

```
pause
```

```
figure(4)
```

```
plot(x,y4)
```

```
disp('Naciśnij Enter aby kontynuować')
```

```
pause
```

```
close all
```



```
m1=menu('Wybierz 1 charakterystykę do wyświetlenia:', 'y1', 'y2', 'y3', 'y4');
```

```
m2=menu('Wybierz 2 charakterystykę do wyświetlenia:', 'y1', 'y2', 'y3', 'y4');
```

```
plot(x, WysWyk(m1), 'r', x, WysWyk(m2), 'b')
```

```
legend('Wykres 1', 'Wykres 2')
```