

## 4. Wykresy 2D

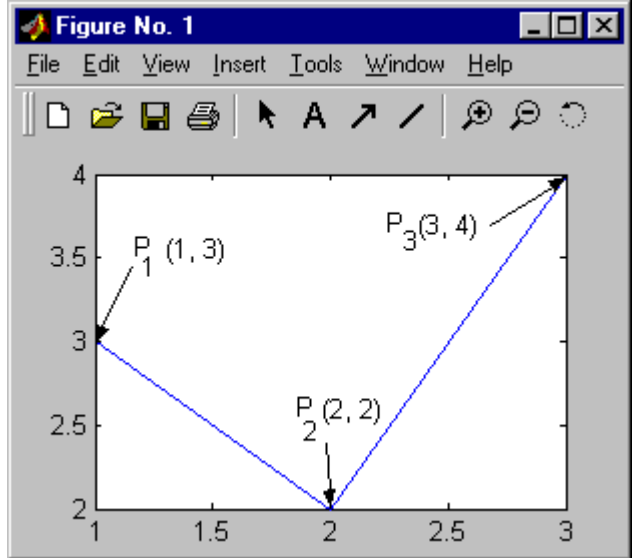
MATLAB oferuje szereg możliwości pozwalających użytkownikowi szybko przedstawić rezultaty wykonanych obliczeń. W punkcie tym przedstawione zostaną jedynie niektóre funkcje przeznaczone do prezentacji danych w postaci wykresów dwuwymiarowych.

### 4.1. Generowanie wykresów – funkcja `plot`

Podstawową funkcją do generowania wykresów jest funkcja: **plot**. Funkcja ta generuje wykresy w specjalnym okienku graficznym, które otwiera się poza środowiskiem MATLAB-a (patrz przykłady). Funkcja **plot** może być wywoływana na kilka różnych sposobów, np.:

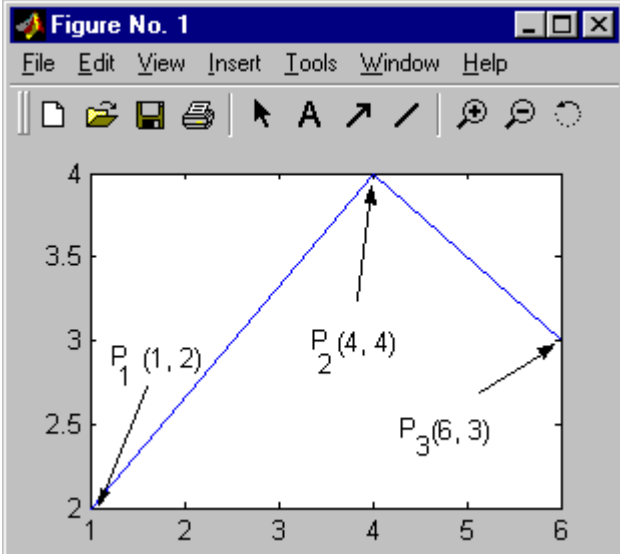
#### **plot(y)**

Funkcja wywołana z jednym parametrem, wektorem **y**, rysuje wykres ciągu elementów wektora względem ich indeksów.

Przykład	Wykres
<p>Jeżeli np. wektor <math>\mathbf{y} = [3 \ 2 \ 4]</math> to rysowany jest wykres łączący trzy punkty:</p> <p><math>P_1 = (x_1, y_1)</math>, <math>P_2 = (x_2, y_2)</math>, <math>P_3 = (x_3, y_3)</math>.</p> <p>Funkcja <code>plot</code> łączy kolejno odcinkami punkty: <math>P_1</math> i <math>P_2</math> oraz <math>P_2</math> i <math>P_3</math>.</p> <p><math>x_1, x_2, x_3</math> – to indeksy elementów wektora <b>y</b>, czyli: <math>x_1=1, x_2=2, x_3=3</math>.</p> <p><math>y_1, y_2, y_3</math> – to kolejne elementy wektora <b>y</b>, czyli: <math>y_1=3, y_2=2, y_3=4</math>;</p>	

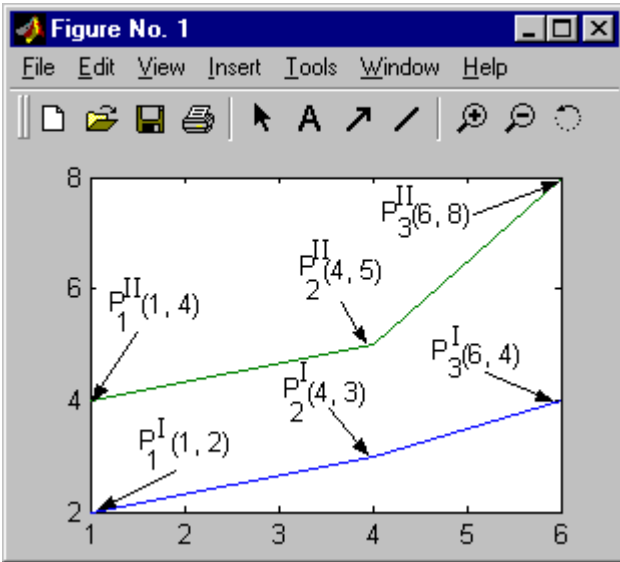
#### **plot(x, y)**

Funkcja wywołana z dwoma parametrami, wektorem **x** i wektorem **y**, rysuje wykres ciągu elementów wektora **y** względem kolejnych elementów wektora **x**.

Przykład	Wykres
<p>Jeżeli np. wektor <math>\mathbf{x} = [1 \ 4 \ 6]</math> a <math>\mathbf{y} = [2 \ 4 \ 3]</math> to rysowany jest wykres łączący trzy kolejne punkty: <math>P_1 = (x_1, y_1)</math>, <math>P_2 = (x_2, y_2)</math>, <math>P_3 = (x_3, y_3)</math>.</p> <p>Funkcja <code>plot</code> łączy kolejno odcinkami punkty: <math>P_1</math> i <math>P_2</math> oraz <math>P_2</math> i <math>P_3</math>.</p> <p><math>x_1, x_2, x_3</math> – to kolejne elementy wektora <math>\mathbf{x}</math>, czyli: <math>x_1=1, x_2=4, x_3=6</math>;</p> <p><math>y_1, y_2, y_3</math> – to kolejne elementy wektora <math>\mathbf{y}</math>, czyli: <math>y_1=2, y_2=4, y_3=3</math>;</p>	

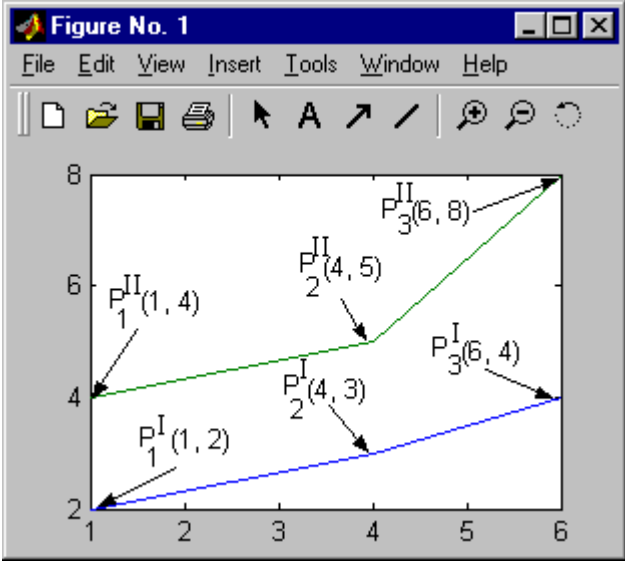
**Uwaga:**

Dopuszczalne jest aby jeden lub oba argumenty funkcji były macierzami. Załóżmy, że pierwszy argument jest wektorem, a drugi macierzą. W takim przypadku rysowane są wykresy kolumn lub wierszy macierzy względem wektora stanowiącego pierwszy argument. Wybór czy należy rysować wykresy kolumn czy wierszy macierzy względem wektora zależy od wymiarów wektora i macierzy.

Przykład	Wykres
<p>Jeżeli np. wektor <math>\mathbf{x} = [1 \ 4 \ 6]</math> a macierz <math>\mathbf{y} = [2 \ 4; 3 \ 5; 4 \ 8]</math> to rysowane są wykresy przedstawiające kolumny macierzy <math>\mathbf{y}</math> względem wektora <math>\mathbf{x}</math> (długość kolumny macierzy <math>\mathbf{y}</math> odpowiada wielkości wektora <math>\mathbf{x}</math>).</p> <p>Rysowane są więc 2 wykresy (są 2 kolumny):          wykres pierwszej kolumny macierzy względem wektora <math>\mathbf{x}</math> i          wykres drugiej kolumny względem wektora <math>\mathbf{x}</math>.</p> <p>Pierwszy wykres łączy punkty:  <math>P_1^I = (x_1, y_1^I)</math>, <math>P_2^I = (x_2, y_2^I)</math>, <math>P_3^I = (x_3, y_3^I)</math>.</p> <p>Drugi wykres łączy punkty:  <math>P_1^{II} = (x_1, y_1^{II})</math>, <math>P_2^{II} = (x_2, y_2^{II})</math>, <math>P_3^{II} = (x_3, y_3^{II})</math>.</p> <p><math>x_1, x_2, x_3</math> – to kolejne elementy wektora <math>\mathbf{x}</math>, czyli: <math>x_1=1, x_2=4, x_3=6</math>;</p> <p><math>y_1^I, y_2^I, y_3^I</math> – to kolejne elementy pierwszej kolumny macierzy <math>\mathbf{y}</math>, czyli: <math>y_1^I=2, y_2^I=3, y_3^I=4</math>;</p> <p><math>y_1^{II}, y_2^{II}, y_3^{II}</math> – to kolejne elementy drugiej kolumny macierzy <math>\mathbf{y}</math>, czyli: <math>y_1^{II}=4, y_2^{II}=5, y_3^{II}=8</math>;</p>	

**plot(x<sub>1</sub>, y<sub>1</sub>, x<sub>2</sub>, y<sub>2</sub>, x<sub>3</sub>, y<sub>3</sub>, ..., x<sub>n</sub>, y<sub>n</sub>)**

Powyżej przedstawiona została jedna z metod rysowania równocześnie kilku wykresów. Wywołanie funkcji plot z kilkoma parami wektorów lub macierzy jest inną metodą na otrzymanie równocześnie wielu wykresów. W takim przypadku, każda para argumentów zostanie zinterpretowana tak jak przy wywołaniu plot(x, y) – generując odpowiedni wykres.

Przykład	Wykres
<p>Jeżeli np. wektor <math>\mathbf{x} = [1 \ 4 \ 6]</math>, <math>\mathbf{y} = [2 \ 3 \ 4]</math>, <math>\mathbf{z} = [4 \ 5 \ 8]</math> to przy wywołaniu <b>plot(x, y, x, z)</b> rysowane są wykresy:</p> <ul style="list-style-type: none"> <li>– wektora <math>\mathbf{y}</math> względem wektora <math>\mathbf{x}</math> oraz</li> <li>– wektora <math>\mathbf{z}</math> względem wektora <math>\mathbf{x}</math>.</li> </ul> <p>Pierwszy wykres łączy punkty:  <math>P^I_1 = (x_1, y_1)</math>, <math>P^I_2 = (x_2, y_2)</math>, <math>P^I_3 = (x_3, y_3)</math>.</p> <p>Drugi wykres łączy punkty:  <math>P^{II}_1 = (x_1, z_1)</math>, <math>P^{II}_2 = (x_2, z_2)</math>, <math>P^{II}_3 = (x_3, z_3)</math>.</p> <p><math>x_1, x_2, x_3</math> – to kolejne elementy wektora <math>\mathbf{x}</math>, czyli:  <math>x_1=1, x_2=4, x_3=6</math>;</p> <p><math>y_1, y_2, y_3</math> – to kolejne elementy wektora <math>\mathbf{y}</math>, czyli:  <math>y_1=2, y_2=3, y_3=4</math>;</p> <p><math>z_1, z_2, z_3</math> – to kolejne elementy wektora <math>\mathbf{z}</math>, czyli:  <math>z_1=4, z_2=5, z_3=8</math>;</p>	

**Uwaga:**

**Jeżeli użytkownik nie zamknie okienka wykresu, wróci do środowiska i wygeneruje następny wykres – wykres ten zostanie narysowany w istniejącym oknie wykresu ale okno to nie zostanie przesunięte na pierwszy plan. W takim przypadku należy przejść do okienka wykresu wykorzystując np. systemowy pasek zadań.**

**Przykład 1.**

Należy wygenerować wykres funkcji  $y = x^2$  w przedziale  $x \in [-5, 5]$ .

**Metoda 1**

```
>> x = -5:5
```

```
x =
```

```
-5 -4 -3 -2 -1 0 1 2 3 4 5
```



```
>> y = x.^2
y =
    25    16     9     4     1     0     1     4     9    16    25
>> plot(x, y)
```

### Metoda 2

Wykres przygotowany pierwszym sposobem jest „kanciasty”. Żeby go wygładzić zwiększono liczbę punktów wektorów  $x$  i  $y$ :

```
>> x = -5:0.1:5
x =
   -5  -4.9  -4.8 ...  4.8  4.9  5
>> y = x.^2
y =
    25  24.01  23.04 ...  23.04  24.01  25
>> plot(x, y)
```

### Metoda 3

Korzystając z funkcji **plot** argumenty można generować wprost w wywołaniu:

```
>> plot(-5:0.1:5, [-5:0.1:5].^2)
```

### Przykład 2.

Należy wygenerować wykres funkcji  $\sin(x)$  w przedziale  $x \in [-2\pi, 2\pi]$ .

Generując wektor  $x$  wykorzystano funkcję MATLAB-a: **pi**. Funkcja ta zwraca wartość liczby  $\pi = 3.1415926535897\dots$

```
>> x = -2*pi : 0.1 : 2*pi
x =
  -6.2832 ...  6.2168
>> y = sin(x)
y =
    0.0000 ... -0.0663
>> plot(x, y)
```



### Przykład 3.

Należy wygenerować wykresy funkcji  $\sin(x)$  i  $\cos(x)$  w przedziale  $x \in [-2\pi, 2\pi]$ .

```
>> x = -2*pi : 0.1 : 2*pi
x =
   -6.2832 ... 6.2168
>> y = sin(x)
y =
   0.0000 ... -0.0663
>> z = cos(x)
z =
   1.0000 ... 0.9978
>> plot(x, y, x, z)
```

## 4.2. Generowanie wykresów – funkcja *fplot*

Funkcja **plot** służyła do generowania dowolnych wykresów. Do szybkiego rysowania zależności funkcyjnych została udostępniona funkcja **fplot**. Istnieje kilka różnych metod korzystania z tej funkcji – zostanie omówiona jedna z nich:

### **fplot(f, granice)**

Funkcja **fplot** rysuje wykres funkcji o nazwie określonej w parametrze **f**. Parametr **granice** to dwuelementowy wektor, którego pierwszym elementem jest jedna z granic przedziału, a drugim – druga granica.

### *Uwaga:*

Parametr **f** powinien zawierać ujętą w apostrofy nazwę rysowanej funkcji. Może to być nazwa jednej z funkcji MATLAB-a (np.: `sin`, `cos`, `log`, `abs` itp.) lub nazwa funkcji zdefiniowanej przez użytkownika.

Jedynym mankamentem tej metody rysowania, jest konieczność napisania odpowiedniej funkcji jeżeli funkcja taka nie istnieje w bibliotece funkcji MATLAB-a. Jeżeli np. użytkownik chciałby narysować wykres  $y=x^2$  musiałby napisać funkcję realizującą podnoszenie do kwadratu (nie ma takiej funkcji w bibliotece funkcji MATLAB-a) i dopiero wtedy mógłby skorzystać z rysowania wykresów przy pomocy **fplot**.



**Przykład 4.**

Należy wygenerować wykres funkcji  $\sin(x)$  w przedziale  $x \in [-2\pi, 2\pi]$ .

```
>> fplot('sin', [-2*pi 2*pi])
```

**4.3. Generowanie wielu wykresów**

Czasami wygodne jest umieszczenie kilku wykresów obok siebie w jednym oknie. Umożliwia to funkcja **subplot**.

**subplot(m, n, nr)**

Funkcja **subplot** dzieli okno wykresu na **m** wierszy i **n** kolumn. Parametr **nr** określa numer wykresu, który zostanie narysowany najbliższym wywołaniem funkcji **plot** lub **fplot**.

Numer wykresu należy podać posługując się zasadą:

- w pierwszym wierszu numery wykresów zmieniają się od 1 do n
- w drugim wierszu numery zmieniają się od n+1 do 2\*n
- itd.

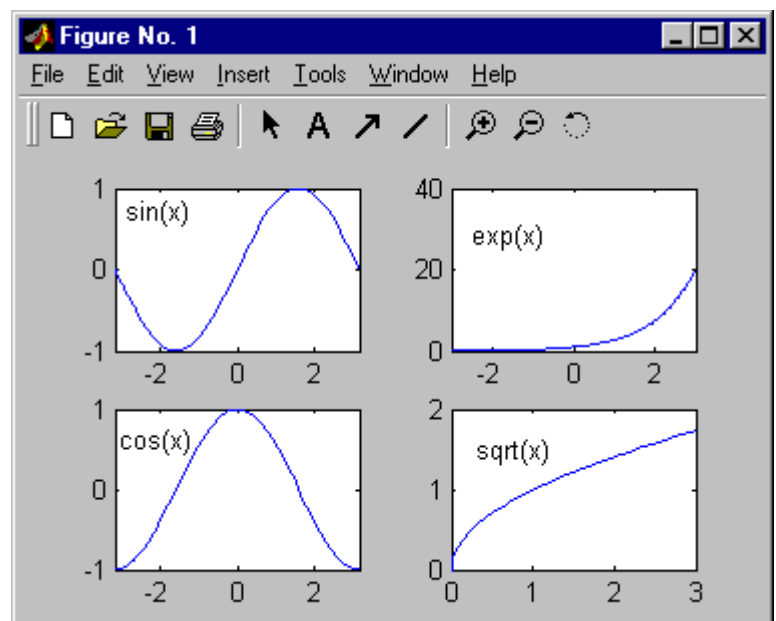
**Przykład:**

Należy wygenerować wykresy funkcji:

- $\sin(x)$ ,  $\cos(x)$  w przedziale  $x \in [-\pi, \pi]$ ,
- $\exp(x)$  w przedziale  $x \in [-3, 3]$ ,
- $\sqrt{x}$  w przedziale  $x \in [0, 3]$ .

Układ funkcji w okienku wykresu powinien wyglądać jak na rysunku obok.

```
>> subplot(2, 2, 1)
>> fplot('sin', [-pi pi])
>> subplot(2, 2, 3)
>> fplot('cos', [-pi pi])
>> subplot(2, 2, 2)
>> fplot('exp', [-3 3])
>> subplot(2, 2, 4)
>> fplot('sqrt', [0 3])
```



**Uwaga:**

**Nie można zamykać okienka wykresu w trakcie wykonywania wymienionych powyżej operacji. Jeżeli okno to zostanie zamknięte np. po wywołaniu `fplot('cos', [-pi pi])` to po wykonaniu pozostałych operacji widoczne będą tylko wykresy funkcji `exp` i `sqrt`.**

**Ćwiczenia**

1. Przy pomocy funkcji `plot` narysuj wykresy funkcji w przedziale  $x \in [-4, 4]$ :

- a)  $x^2 - 4$
- b)  $x^3$
- c)  $x^{-1}$
- d)  $|x|$

**Uwaga:**

Przygotuj do każdego wykresu odpowiednie wektory **ya**, **yb**, **yc**, i **yd** z wartościami funkcji **a)**, **b)**, **c)** oraz **d)** w przedziale  $x \in [-4, 4]$ .

2. Umieść przygotowane wykresy w jednym oknie zgodnie z następującym wzorcem:

<b>a)</b>	<b>b)</b>
<b>c)</b>	<b>d)</b>

**Uwaga!**

Wykorzystaj funkcję `subplot`.

