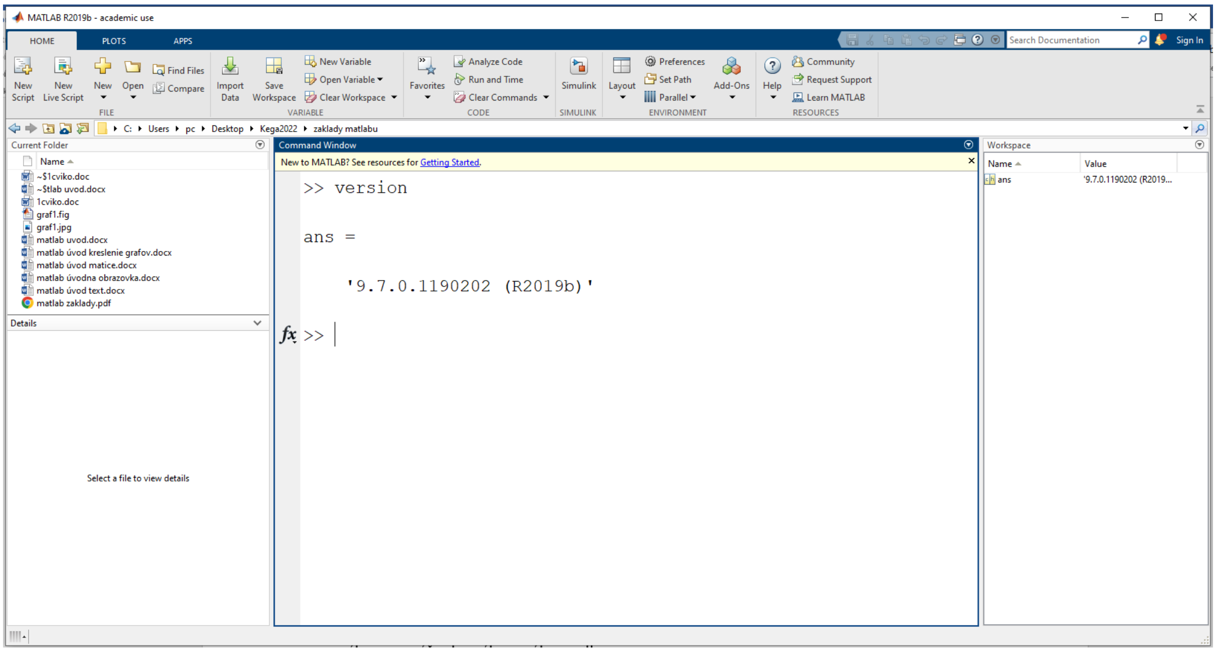
Úvod do MATLABu



Na obrázku vidíme obrazovku výpočtového systému MATLAB.

Uprostred je okno `Command Window`. Do tohto okna zadávame príkazy.
Znak >> na začiatku riadku znamená, že MATLAB očakáva príkaz, resp. dokončil predchádzajúci príkaz. Zapísaný príkaz sa vykoná po stlačení klávesy ENTER. V histórii príkazov sa pohybujeme pomocou znakov ↑ a ↓ na klávesnici .
Na obrázku vidíme výstup po zadaní príkazu `version`, teda vidíme verziu MATLABu, ktorú používame.

Vpravo je okno s aktuálne použitými premennými `Workspace`.

Vľavo vidíme zoznam súborov, ktoré sú v aktuálne otvorenom adresári `Current Folder`, pričom nad tým vidíme, ktorý to je adresár.

Základné matematické operácie a premenné

Pre súčet, rozdiel, súčin, podiel používame v MATLABe znaky + , - , * , / , pre umocňovanie znak ^ .
Pri zadávaní desatinného čísla ako desatinnú čiarku v MATLABe dávame bodku.

Vypočítajme: $2{,}13 - 8\left(4{,}75 + \frac{7}{8}\right)$

```
>> 2.13-8*(4.75+7/8)

ans =
```

```
 -42.869999999999997
```

Znak pre násobenie * treba v MATLABe vždy napísať, aj keď v matematike ho nepíšeme.

Vypočítajme: $11\,.\,7 - \frac{20}{3} + 2^3$

```
>> 11*7-20/3+2^3

ans =

  78.333333333333329
```

Výsledok sa uloží do premennej `ans` (ako answer – odpoveď). Keď chceme s výsledkom ďalej pracovať, je potrebné ho uložiť do nejakej premennej.

```
>> a=11*7-20/3+2^3

a =

  78.333333333333329
```

S takto vytvorenou premennou `a` môžeme ďalej pracovať.

Vypočítajme: 2 a , výsledok uložme do premennej b

```
>> b=2*a

b =

     1.566666666666667e+02
```

Keď za príkazom použijeme znak **;** (bodkočiarka), príkaz sa vykoná, ale výstup sa nezobrazí na obrazovku.

Vypočítajme: $\frac{3\,b}{a} - 10$, výsledok uložme do premennej: vysledok

```
>> vysledok=(3*b)/a-10;
>>
```

Ak chceme zobraziť obsah premennej, stačí do príkazového riadku napísať meno premennej.

```
>> vysledok

vysledok =

    -4
```

MATLAB rozlišuje malé a veľké písmena. Preto premenná `a` a premenná `A` sú dve rôzne premenné.

Napríklad:

```
>> a=2.18*5

a =

  10.900000000000000

>> A=9/4

A =

   2.250000000000000

>> r=a-A

r =

   8.650000000000000

>>
```

Meno premennej vždy začína písmenom. A môže obsahovať písmená a číslice.

Niekedy sa môže stať, že nám MATLAB ako odpoveď vypíše:

`Inf (Infinity)` – v zmysle nekonečno
`NaN (Not - a - Number)` – v zmysle nedefinovaný výraz, žiadna numerická hodnota

Napríklad:

```
>> 2^1000

ans =

    1.071508607186267e+301

>> 2^10000

ans =

   Inf
```

Je to teda také veľké číslo, že prekračuje maximálnu hodnotu, s ktorou MATLAB pracuje.
Túto hodnotu získame takto:

```
>> realmax

ans =
```

```
    1.797693134862316e+308
```

Najmenšie použiteľné kladné reálne číslo je:

```
>> realmin

ans =

    2.225073858507201e-308
```

Uveďme príklady, kedy ako výstup dostaneme `NaN`.

Napríklad:

```
>> realmax*2/(realmax*2)

ans =

   NaN

>> 0/0

ans =

   NaN
```

Formát výstupu na obrazovku

V premennej `pi` má MATLAB uložené číslo π.

```
>> pi

ans =

   3.141592653589793
```

Obsah premennej je zobrazený na 15 desatinných miest. Niekedy je tento výstup neprehľadný a stačí nám zobraziť výsledok na 4 desatinné miesta. To urobíme príkazom `format short` alebo stačí len `format`.

```
>> format short
>>
```

Príkaz je prepínač, teda nemá žiaden výstup na obrazovku. Len sa MATLAB nastaví tak, že každý ďalší výstup na obrazovku urobí na 4 desatinné miesta.

```
>> pi

ans =
```

```
    3.1416
```

Zobrazenie výstupov na 15 desatinných miest dosiahneme príkazom `format long`.

```
>> format long
>> pi

ans =

   3.141592653589793
```

Elementárne funkcie

Prehľad elementárnych funkcií v MATLABe môže nájsť použitím príkazu:

```
>> help elfun
```

prípadne tiež:

```
>> help specfun
```

Na príkladoch uveďme niektoré funkcie.

Vypočítajme: $\sqrt{144}$

```
>> sqrt(144)

ans =

    12
```

Vypočítajme: $\sin\left(\frac{\pi}{4}\right)$

```
>> sin(pi/4)

ans =

   0.707106781186547
```

Vypočítajme: $e^{-0,5}$

```
>> exp(-0.5)

ans =

   0.606530659712633
```

Vypočítajme: ln 7 (prirodzený logaritmus)

```
>> log(7)
```

```
ans =

   1.945910149055313
```

Vypočítajme: $\log 20$ (dekadický logaritmus)

```
>> log10(20)

ans =

   1.301029995663981
```

Video k tejto téme je možné si pozrieť tu:
https://www.youtube.com/watch?v=FA4RgiAXxi0

Matice

Základným objektom, s ktorým MATLAB pracuje, sú matice.

Úloha: Zadajme matice: $A = \begin{pmatrix} 1 & 3 \\ -0{,}6 & 2{,}7 \end{pmatrix}$ a $B = \begin{pmatrix} 2{,}5 & 0 \\ 4 & -1 \end{pmatrix}$.

Prvky matice zadávame do hranatých zátvoriek, zadávame ich po riadkoch. Prvky v riadku oddeľujeme čiarkou alebo medzerou, riadky navzájom oddeľujeme bodkočiarkou.

```
>> A=[1,3;-0.6,2.7]

A =

    1.0000    3.0000
   -0.6000    2.7000

>> B=[2.5,0;4,-1]

B =

    2.5000         0
    4.0000   -1.0000
```

Pre súčet, rozdiel, súčin matíc používame v MATLABe znaky + , - , * .

Vypočítajme: $5\,A + 3B$, $A\,.\,B$, $B\,.\,A$, A^2 , $A\,.\,A$

```
>> 5*A+3*B

ans =

   12.5000   15.0000
    9.0000   10.5000
```

```
>> A*B

ans =

   14.5000   -3.0000
    9.3000   -2.7000

>> B*A

ans =

    2.5000    7.5000
    4.6000    9.3000

>> A^2

ans =

   -0.8000   11.1000
   -2.2200    5.4900

>> A*A

ans =

   -0.8000   11.1000
   -2.2200    5.4900
```

Je potrebné dôsledne rozlišovať medzi už uvedenými maticovými operáciami a operáciami po prvkoch, ktoré môžeme v MATLABe vhodne využiť. Operácie po prvkoch sa označujú bodkou pred znakom operácie.

Napríklad:

.* je násobenie po prvkoch. Matice A, B musia byť rovnakého typu a vynásobia sa prvky na rovnakých pozíciách.

```
>> A.*B

ans =

    2.5000         0
   -2.4000   -2.7000
```

.^ je umocňovanie po prvkoch. Každý prvok matice A sa umocní na druhú.

```
>> A.^2

ans =
```

```
    1.0000    9.0000
    0.3600    7.2900
```

Úloha: Zadajme vektor $\vec{x} = (1, 3, 2, -4)$

```
>> x=[1 3 2 -4]

x =

     1     3     2    -4
```

S vektorom budeme v MATLABe pracovať ako s maticou, ktorá má jeden riadok, resp. jeden stĺpec.

Napríklad:

```
>> x*x
Error using  *
Incorrect dimensions for matrix multiplication. Check that the
number of columns in
the first matrix matches the number of rows in the second
matrix. To perform
elementwise multiplication, use '.*'.

>> x.*x

ans =

     1     9     4    16

>> x^3
Error using  ^  (line 51)
Incorrect dimensions for raising a matrix to a power. Check
that the matrix is square
and the power is a scalar. To perform elementwise matrix
powers, use '.^'.

>> x.^3

ans =

     1    27     8   -64
```

Úloha: Zadajme vektor $\vec{x} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$
V tomto prípade nemusíme vypisovať všetky prvky vektora. Stačí zadať prvý prvok vektora a, potom krok h, s akým sú vytvorené prvky vektora, a posledný prvok vektora b ($a : h : b$).

```
>> x=1:1:10

x =
```

```
     1     2     3     4     5     6     7     8     9    10
```

Ak je krok 1, tak ho nemusíme písať.

```
>> x=1:10

x =

     1     2     3     4     5     6     7     8     9    10
```

Napríklad:

```
>> x=1:2:10

x =

     1     3     5     7     9

>> x=5:-0.5:2

x =

  Columns 1 through 6

    5.0000    4.5000    4.0000    3.5000    3.0000    2.5000

  Column 7

    2.0000
```

Práca s prvkami matice

Úloha: Zadajme maticu: $A = \begin{pmatrix} 3 & 2 & 7 & -1 \\ -4 & 0 & 4 & 1 \\ 1 & -3 & 2 & 0 \end{pmatrix}$

```
>> A=[3 2 7 -1;-4 0 4 1;1 -3 2 0]

A =

     3     2     7    -1
    -4     0     4     1
     1    -3     2     0
```

Úloha: Do premennej r uložme prvky v druhom riadku matice A. Do premennej s uložme prvky v treťom stĺpci matice A. Do premennej k uložme prvok v prvom riadku a štvrtom stĺpci matice A.

```
>> r=A(2,:)
```

```
r =

    -4     0     4     1

>> s=A(:,3)

s =

     7
     4
     2

>> k=A(1,4)

k =

    -1
```

Úloha: V matici A prvok v treťom riadku a druhom stĺpci nahraďme číslom 10.

```
>> A(3,2)=10

A =

     3     2     7    -1
    -4     0     4     1
     1    10     2     0
```

To, čo je v príkazovom riadku vpravo od znaku = sa vždy najskôr vykoná, vypočíta, a potom uloží do premennej vľavo od znaku = . Preto tá istá premenná môže v príkazovom riadku vystupovať vpravo aj vľavo od znaku = , dokonca aj súčasne v jednom príkazovom riadku.

Úloha: K prvému riadku matice A pripočítajme (-3)-násobok tretieho riadku.

```
>> A(1,:)=A(1,:)+(-3)*A(3,:)

A =

     0   -28     1    -1
    -4     0     4     1
     1    10     2     0
```

Úloha: Do premennej $p3$ uložme tretí prvok vektora $\vec{x} = (2, -8, 7, -4, 10)$.

```
>> x=[2 -8 7 -4 10]

x =

     2    -8     7    -4    10

>> p3=x(1,3)
```

```
p3 =

     7
```

Keďže je to matica, ktorá má iba jeden riadok, stačí to zapísať takto:

```
>> p3=x(3)

p3 =

     7
```

Príkazy pre prácu s maticami

Úloha: Zadajme maticu: $a = \begin{pmatrix} 2 & -4 & -2 \\ -1 & 3 & 1 \\ 0 & -2 & 0 \end{pmatrix}$

```
>> a=[2 -4 -2;-1 3 1;0 -2 0]

a =

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

Uveďme niektoré príkazy pre prácu s maticami.

Napríklad:

Keď chceme zistiť, koľko má matica riadkov a stĺpcov, použijeme príkaz `size`.

```
>> size(a)

ans =

     3     3
```

Prvé číslo určuje počet riadkov a druhé určuje počet stĺpcov. Keď tieto čísla chceme samostatne uložiť do premenných, zapíšeme to takto:

```
>> [m,n]=size(a)

m =

     3


n =
```

```
     3
```

Keď chceme zistiť, aké prvky sú na diagonále matice a a uložiť ich do premennej d, použijeme príkaz `diag`.

```
>> d=diag(a)

d =

     2
     3
     0
```

Podobne si vieme vypísať prvky aj na iných diagonálach matice a.

```
>> diag(a,1)

ans =

    -4
     1
```

Výstupom je vektor prvkov matice a na prvej diagonále nad hlavnou diagonálou.

```
>> diag(a,-1)
ans =

    -1
    -2
```

Výstupom je vektor prvkov matice a na prvej diagonále pod hlavnou diagonálou.

Uveďme niektoré ďalšie príkazy.

```
>> triu(a)

ans =

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

Výstupom je matica, v ktorej prvky matice a pod hlavnou diagonálou sú nahradené nulou, ostatné prvky matice a ostávajú rovnaké.

```
>> tril(a)
ans =

     2     0     0
    -1     3     0
     0    -2     0
```

Výstupom je matica, v ktorej prvky matice a nad hlavnou diagonálou sú nahradené nulou, ostatné prvky matice a ostávajú rovnaké.

```
>> sum(a)

ans =

     1    -3    -1
```

Výstupom je vektor, ktorého prvky určujú postupne súčet prvkov matice a v stĺpcoch, to znamená, že súčet prvkov v prvom stĺpci matice a je 1, v druhom stĺpci je -3 a v treťom stĺpci je -1.

```
>> sum(sum(a))

ans =

    -3
```

Týmto príkazom teda vypočítame súčet všetkých prvkom matice a.

```
>> max(a)

ans =

     2     3     1
```

Výstupom je vektor, ktorého prvky určujú postupne maximálne prvky matice a v stĺpcoch, to znamená, že maximálny prvok v prvom stĺpci matice a je 2, v druhom stĺpci je 3 a v treťom stĺpci je 1.

```
>> [p,k]=max(a)

p =

     2     3     1


k =

     1     2     2
```

Týmto spôsobom zistíme nielen maximálne prvky v jednotlivých stĺpcoch (tie sú uložené v premennej `p`), ale tiež ich poradie v danom stĺpci (tieto poradia sú uložené v premennej `k`). To znamená, že maximálny prvok v prvom stĺpci je 2 a je to prvý prvok v tom stĺpci. Maximálny prvok v druhom stĺpci je 3 a je to druhý prvok v tom stĺpci. Maximálny prvok v treťom stĺpci je 1 a je to druhý prvok v tom stĺpci.

```
>> min(a)
```

```
ans =

    -1    -4    -2
```

Výstupom je vektor, ktorého prvky určujú postupne minimálne prvky matice a v stĺpcoch, to znamená, že minimálny prvok v prvom stĺpci matice a je -1, v druhom stĺpci je -4 a v treťom stĺpci je -2.

```
>> [p,k]=min(a)

p =

    -1    -4    -2


k =

     2     1     1
```

Podobne, ako pri príkaze `max`, týmto spôsobom zistíme nielen minimálne prvky v jednotlivých stĺpcoch (tie sú uložené v premennej `p`), ale tiež ich poradie v danom stĺpci (tieto poradia sú uložené v premennej `k`).

```
>> det(a)

ans =

     0
```

Výstupom tohto príkazu je hodnota determinantu matice a (matica a musí byť štvorcová).

```
>> rank(a)

ans =

     2
```

Tento príkaz určí hodnosť matice a.

```
>> eig(a)

ans =

    4.0000
    0.0000
    1.0000
```

Výstupom tohto príkazu sú vlastné čísla matice a (matica a musí byť štvorcová).

```
>> [v,c]=eig(a)

v =

    0.8018    0.7071    0.0000
   -0.5345         0   -0.4472
    0.2673    0.7071    0.8944


c =

    4.0000         0         0
         0    0.0000         0
         0         0    1.0000
```

Týmto príkazom získame nielen vlastné čísla matice a, ale aj k nim prislúchajúce vlastné vektory. V premennej `c` je diagonálna matica s vlastnými číslami matice a na hlavnej diagonále. V premennej `v` sú vlastné vektory. To znamená, že prvý stĺpec matice `v` je vlastný vektor prislúchajúci vlastnému číslu, ktoré je v matici `c` v prvom stĺpci (a prvom riadku). Druhý stĺpec matice `v` je vlastný vektor prislúchajúci vlastnému číslu, ktoré je v matici `c` v druhom stĺpci (a druhom riadku). V treťom stĺpci matice `v` je vlastný vektor prislúchajúci vlastnému číslu, ktoré je v matici `c` v treťom stĺpci (a v treťom riadku).

Úloha: Zadajme maticu: $b = (5 \quad -7 \quad 2 \quad 3 \quad -2)$

```
>> b=[5 -7 2 3 -2]

b =

     5    -7     2     3    -2
```

Na maticu b sa teraz môžeme pozerať aj ako na riadkový vektor. Poďme sa pozrieť na príkazy `size`, `diag`, `sum`, `max`, `min`.

```
>> [m,n]=size(b)

m =

     1


n =

     5
```

Prvé číslo určuje počet riadkov a druhé počet stĺpcom matice b.

```
>> diag(b)

ans =
```

```
     5     0     0     0     0
     0    -7     0     0     0
     0     0     2     0     0
     0     0     0     3     0
     0     0     0     0    -2
```

Výstupom príkazu `diag` je teraz matica, ktorej prvky na diagonále sú prvky vektora *b* a ostatné prvky sú nuly.

```
>> diag(b,1)

ans =

     0     5     0     0     0     0
     0     0    -7     0     0     0
     0     0     0     2     0     0
     0     0     0     0     3     0
     0     0     0     0     0    -2
     0     0     0     0     0     0
```

Teraz sú prvky vektora *b* umiestnené na prvej diagonále nad hlavnou diagonálou a ostatné prvky sú nuly.

```
>> diag(b,-1)

ans =

     0     0     0     0     0     0
     5     0     0     0     0     0
     0    -7     0     0     0     0
     0     0     2     0     0     0
     0     0     0     3     0     0
     0     0     0     0    -2     0
```

Teraz sú prvky vektora *b* umiestnené na prvej diagonále pod hlavnou diagonálou a ostatné prvky sú nuly.

```
>> sum(b)

ans =

     1
```

Výstupom je súčet prvkom matice *b*.

```
>> max(b)

ans =

     5
```

Výstupom je maximálny prvok matice b.

```
>> [p,k]=max(b)

p =

     5


k =

     1
```

Týmto príkazom sme zistili, že maximálny prvok vektora b, čo je v našom príklad číslo 5, je prvým prvkom vektora.

```
>> min(b)

ans =

    -7
```

Výstupom je minimálny prvok matice b.

```
>> [p,k]=min(b)

p =

    -7


k =

     2
```

Týmto príkazom sme zistili, že minimálny prvok vektora b, čo je v našom príklad číslo -7, je druhým prvkom vektora.

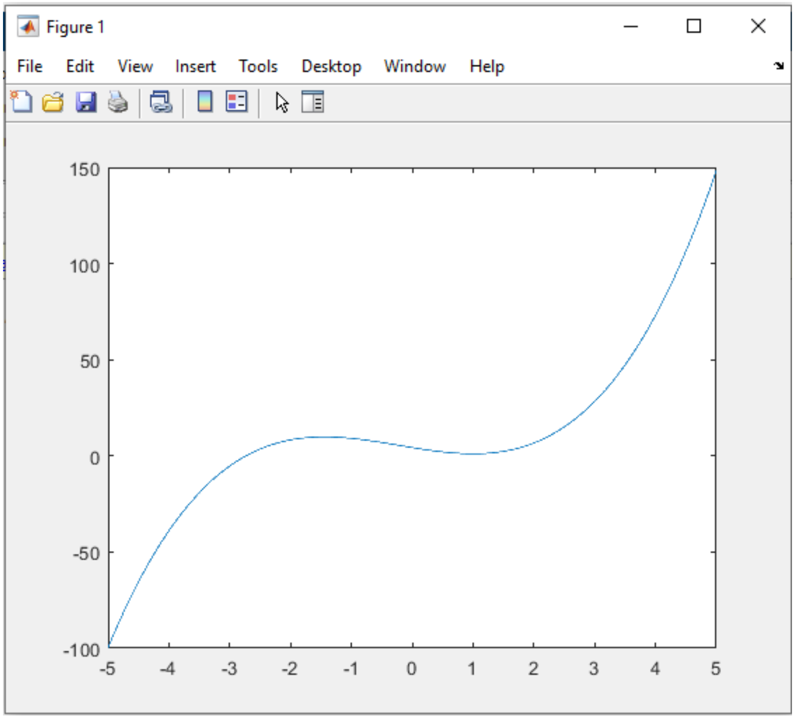
Videá pre prácu s maticami sú tu:
https://www.youtube.com/watch?v=DFvqqHQvj-E
https://www.youtube.com/watch?v=ov-onR-6aoQ
https://www.youtube.com/watch?v=Pz9F3Po-zvA

Kreslenie grafov funkcií

Úloha: Nakreslime graf funkcie $y = 1{,}2\,x^3 + 0{,}8\,x^2 - 5{,}2\,x + 4{,}3$ na intervale $\langle -5, 5 \rangle$.

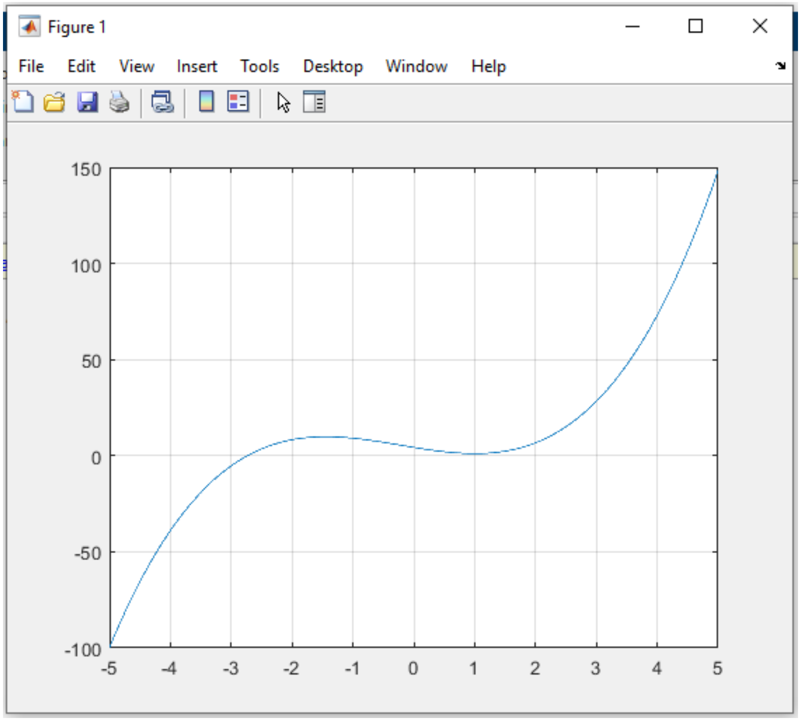
Do premennej x vytvoríme vektor čísel od -5 po 5 s krokom 0,001. Do premennej y vypočítame funkčné hodnoty zadanej funkcie v každom bode vektora x, využijeme operácie po prvkoch. Na nakreslenie grafu použijeme príkaz `plot` – pospája body, ktorých x-ové súradnice sú v premennej x a y-ové súradnice v premennej y.

```
>> x=-5:0.001:5;
>> y=1.2*x.^3+0.8*x.^2-5.2*x+4.3;
>> plot(x,y)
```



Keď necháme grafické okno otvorené, príkazom `grid` do obrázka dokreslíme mriežku.

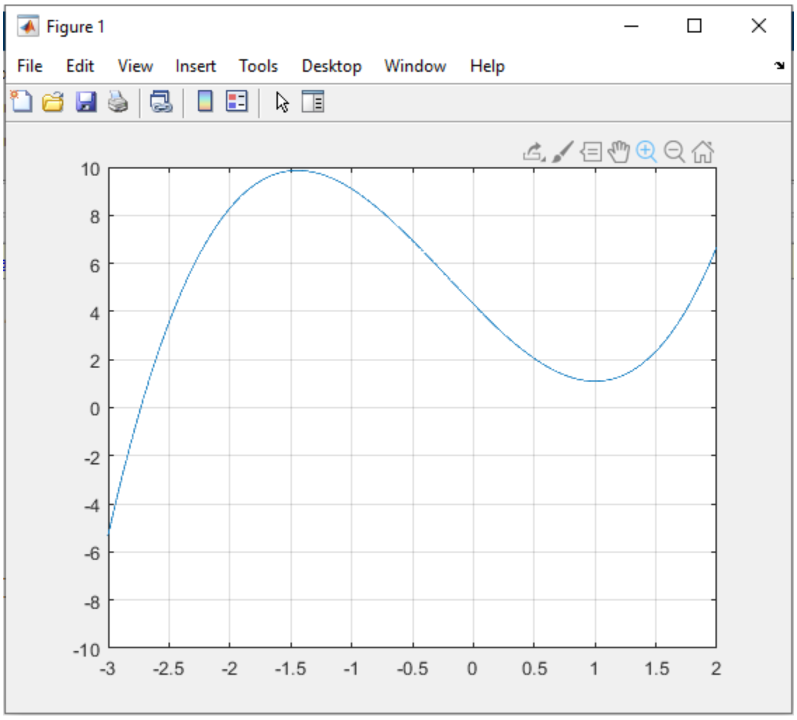
```
>> grid
```



Keď chceme urobiť výrez z grafu, použijem príkaz `axis`, napríklad:

```
>> axis([-3,2,-10,10])
```

Prvé dve čísla určujú rozsah na osi x-ovej, druhé dve čísla rozsah na osi y-ovej.

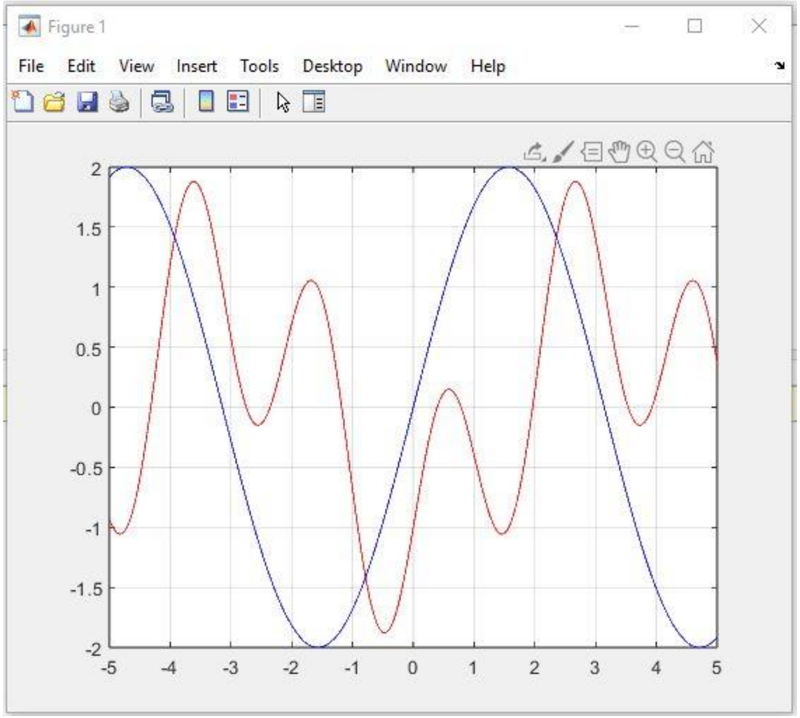


Výrez z grafu môžeme urobiť aj pomocou funkcií samotného grafického okna (vpravo hore lupa so znamienkom +). To je možné si pozrieť vo videu:
https://www.youtube.com/watch?v=hbyTB44uUGI

Úloha: Nakreslime grafy funkcií $f_1(x) = \sin(3x) - \cos x$ a $f_2(x) = 2\sin x$ na intervale $\langle -5, 5 \rangle$ do jedného obrázka.

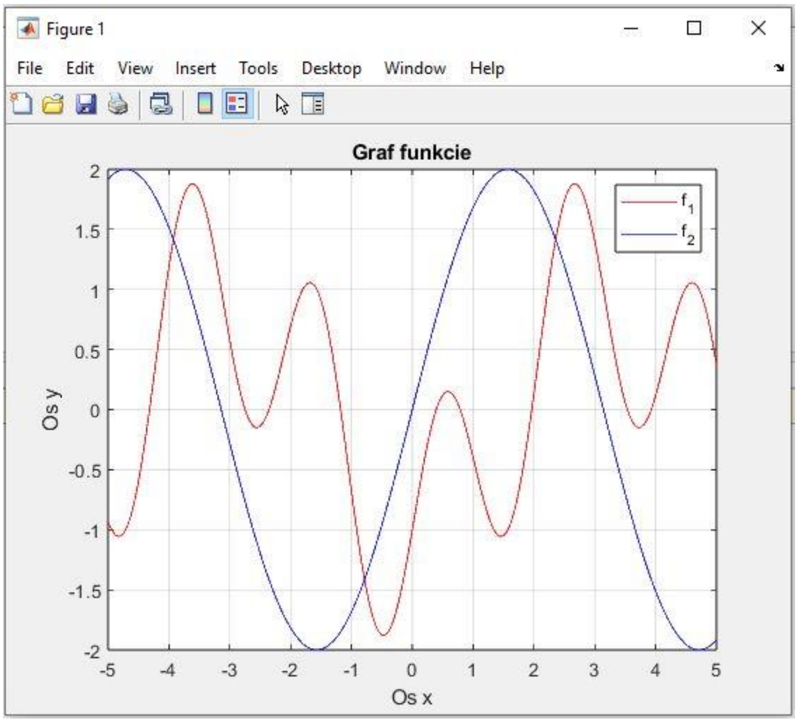
Do premennej x vytvoríme vektor čísel od -5 po 5 s krokom $0{,}001$. Do premennej $y1$ vypočítame funkčné hodnoty funkcie f_1 a do premennej $y2$ funkčné hodnoty funkcie f_2 v každom bode vektora x. Na nakreslenie grafu použijeme príkaz `plot`. V príkaze `plot` môžeme určiť, akou farbou má byť graf vykreslený. Prvý graf, teda graf funkcie f_1 vykreslíme červenou farbou, v príkaze `plot` použijeme `'r'`. Druhý graf, teda graf funkcie f_2 vykreslíme modrou farbou, v príkaze `plot` použijeme `'b'`.

```
>> x=-5:0.001:5;
>> y1=sin(3*x)-cos(x);
>> y2=2*sin(x);
>> plot(x,y1,'r',x,y2,'b')
>> grid
```



Obrázok nechajme otvorený a urobme popis obrázka. Popis súradnicových osí urobíme príkazmi `xlabel` a `ylabel`. Názov doplníme pomocou príkazu `title` a legendu príkazom `legend`.

```
>> xlabel('Os x')
>> ylabel('Os y')
>> title('Graf funkcie')
>> legend('f_1','f_2')
```

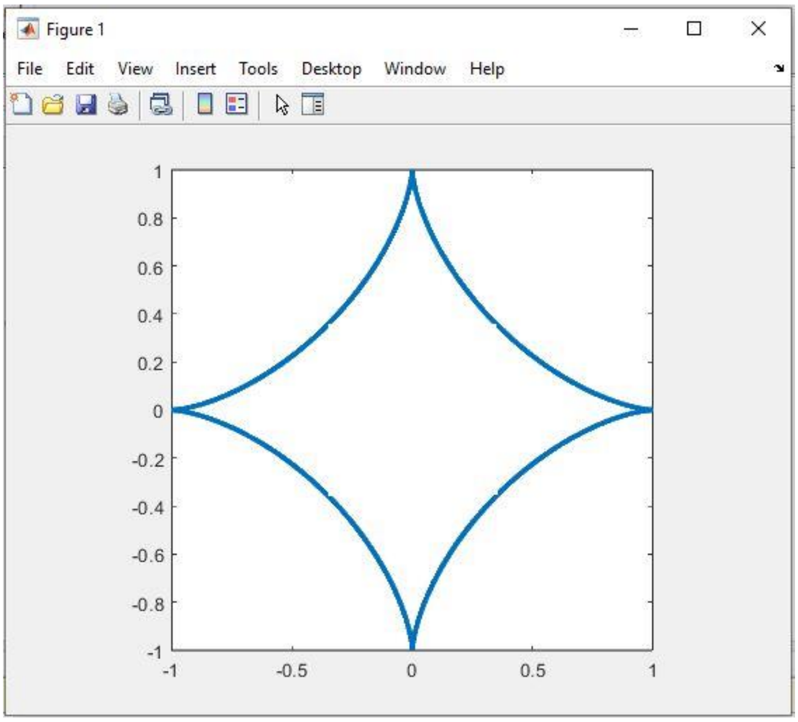


Video k danej úlohe je tu:
https://www.youtube.com/watch?v=WPLWHLNQnFs

Úloha: Nakreslime krivku, ktorej parametrické rovnice sú $x = \cos^3 t$, $y = \sin^3 t$, pre $t \in \langle 0; 2\,\pi \rangle$. Túto krivku nazývame asteroida.

Použijeme nasledujúcu postupnosť príkazov. V príkaze `plot` vieme nastaviť hrúbku čiary. Príkazom `axis('square')` nastavíme, aby krivka bola nakreslená do štvorca, teda v tomto prípade máme rovnakú jednotku dĺžky na obidvoch osiach.

```
>> t=0:0.001:2*pi;
>> x=(cos(t)).^3;
>> y=(sin(t)).^3;
>> plot(x,y,'LineWidth',3)
>> axis('square')
```



Video k danej úlohe je tu:
https://www.youtube.com/watch?v=-eVsAxRaINo