

MATLAB

Përmbajtje

Pjesa e parë	2
Përdorimi i MATLAB si makinë llogaritëse	3
Ndryshoret në MATLAB.....	4
Mbivendosja e ndryshores	4
Mesazhet e gabimeve	4
Kryerja e korigjimeve	4
Kontrolli i hierarkisë së operatorëve ose përparësisë.....	5
Kontrolli i paraqitjes të numrave real.....	5
Menaxhimi i hapësirës së punës	5
Ruajtja e sesionit tuaj të punës	6
Futja e deklaratave të shumëfishta për rresht.....	6
Komanda të ndryshme	6
Funksionet Matematikore	6
Ndërtimi i grafikëve.....	7
Strukturat e kontrollit dhe përsëritëse.....	9
a) if ... elseif ... else ... end.....	9
b) switch op case vlera 1	9
c) for ... end	9
d) while ... end.....	9
Matricat	11
Zgjidhja e ekuacioneve lineare	17
Ushtrimi 1	17
Ushtrimi 2	18
Ushtrimi 3	18
Ushtrimi 4	18
Ushtrimi 5	18
Ushtrimi 6	19
Ushtrimi 7	19
Ushtrimi 8	19

Objekti kryesor i këtij materiali është t'ju ndihmojë të mësoni shpejt duke punuar me shembuj për të parë se si funksionon MATLAB. Në këtë material do të përshkruajmë se si MATLAB trajton shprehje numerike të thjeshta dhe formula matematikore.

Emri MATLAB është shkurtim i **MAT**rix **LAB**oratory. MATLAB u shkrua fillimisht për të siguruar qasje të lehtë në softuerin e matricës të zhvilluar nga projektet LINPACK (paketa lineare të sistemit) dhe EISPACK (Paketa e Sistemit Eigen).

MATLAB është një gjuhë me performancë të lartë, e cila ka të integruar mjedisin e llogaritjes, vizualizimit dhe të programimit. Për më tepër, MATLAB është një mjedis modern i gjuhës programuese: ka struktura të sofistikuar të të dhënave, përmban mjete të brendshme të redaktimit dhe debugging dhe mbështet programimin e orientuar drejt objekteve. Këta faktorë e bëjnë MATLAB një mjet të shkëlqyeshëm për mësimdhënie dhe kërkime.

MATLAB ka shumë përparësi krahasuar me gjuhët konvencionale kompjuterike (p.sh. C, FORTRAN) për zgjidhjen e problemeve teknike. MATLAB është një sistem interaktiv, element bazë e të dhënave të së cilit është një matricë që nuk kërkon dimensionin. Paketa e softuerit është në dispozicion në treg që nga 1984 dhe tani konsiderohet si një mjet standard në universitetet dhe industrinë më të shumta në mbarë botën.

Ka rutina të fuqishme të inkorporuar që mundësojnë një shumëllojshmëri të gjerë të llogaritjeve. Ajo gjithashtu përdor lehtësisht komandat grafike që e bëjnë vizualizimin e rezultateve menjëherë të disponueshme. Aplikacionet specifike mbledhen në pako të referuara si toolbox. Ka toolboxe për përpunimin e sinjalit, llogaritjen simbolike, teorinë e kontrollit, simulimin, optimizimin dhe disa fusha të tjera të shkencës dhe inxhinierisë së aplikuar.

Përveç dokumentacionit MATLAB, i cili është më së shumti i disponueshëm on-line, do të rekomandojmë librat në vijim pasi ato janë të shkëlqyera në aplikimet e tyre specifike.

1. S. J. Chapman. MATLAB Programming for Engineers. Thomson, 2004.
2. C. B. Moler. Numerical Computing with MATLAB. Siam, 2004.
3. C. F. Van Loan. Introduction to Scientific Computing. Prentice Hall, 1997.
4. D. J. Higham and N. J. Higham. MATLAB Guide. Siam, second edition edition, 2005.
5. K. R. Coombes, B. R. Hunt, R. L. Lipsman, J. E. Osborn, and G. J. Stuck. Differential Equations with MATLAB. John Wiley and Sons, 2000.
6. Gilat. MATLAB: An introduction with Applications. John Wiley and Sons, 2004.
7. J. Cooper. A MATLAB Companion for Multivariable Calculus. Academic Press, 2001.
8. J. C. Polking and D. Arnold. ODE using MATLAB. Prentice Hall, 2004.

Pjesa e parë

Qëllimi i kësaj pjese është që të mësojë hapat e parë: se si të hyri në MATLAB; si të kërkoni në MATLAB; të bëni disa llogaritje të thjeshta dhe si të dilni nga MATLAB-i.

Pas hyrjes në llogarinë tuaj, mund të futeni në MATLAB duke klikuar dy herë në ikonën MATLAB të vendosur në desktopin tuaj të Windows. Kur fillon punon në MATLAB, shfaqet dritarja e veçantë e quajtur MATLAB desktop. Desktop është një dritare që përmban dritare të tjera. Pjesët kryesore janë:

- Command Window (Dritarja e komandës)
- Command History (Historia e komandës)
- Workspace (Hapësira e punës)
- Current Folder (Udhëzuesi aktual)
- Help Browser (Shfletuesi i ndihmës)
- Start (Butoni Start)

Me poshtë paraqitet ndërfaqja grafike në hapësirën e punës MATLAB:

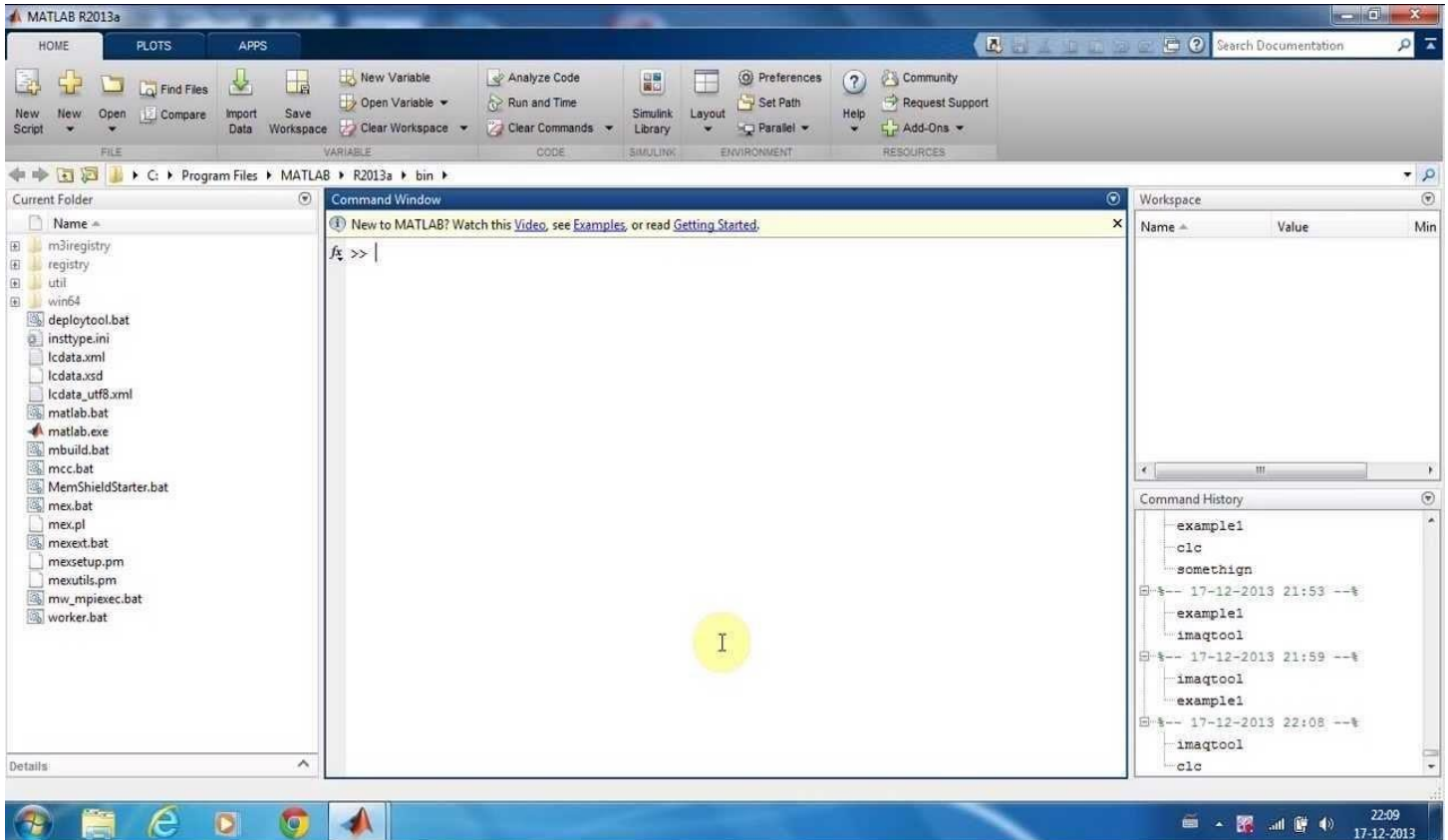


Figura e mësipërme tregon konfigurimin e parazgjedhur të punës në MATLAB. Mund të rregulloni mjetet dhe dokumentet sipas nevojave tuaja. Fillimisht do të bëjmë disa llogaritje të thjeshta.

Në desktop-in e MATLAB-it, mund të shikoni simbolin `>>` në dritaren e komandës, i cili është prompt-i. Zakonisht, ekzistojnë 2 lloje të shpejtë:

1. `>>` për versionin e plotë
2. `EDU>` për version edukimi

Ne do të përdorim këtë: `>>`, si shenjë standarde të prompt-it, edhe pse versioni ynë MATLAB është për qëllime edukimi. Për t'i dhënë fund sesionit **MATLAB**, shtypni butonin **QUIT** në dritaren e komandës, ose përzgjidhni **File** → **Exit MATLAB** në menu-në kryesore të desktop-it.

Përdorimi i MATLAB si makinë llogaritëse

Si shembull i një llogaritjeje të thjeshtë interaktive, shkruani shprehjen që dëshironi të vlerësoni. Për shembull, le të supozojmë që dëshironi të llogaritni shprehjen $1 + 2 * 3$. Ju shkruani atë në komandën prompt (`>>`) si më poshtë: `>> 1+2*3`

ans =

7

Vëmë re se, nëse nuk specifikojmë një variabël të output-it, MATLAB përdor një **ndryshore** të parazgjedhur të emëruar **ans**, shkurt për përgjigje, për të ruajtur rezultatet e llogaritjes aktuale.

Gjithashtu ndryshorja **ans** është krijuar (ose mbishkruar, nëse ekziston tashmë). Për të shmangur këtë, mund të caktoni një vlerë tek një ndryshore tjetër të emëruar **x**. Për shembull:

```
>> x = 1+2*3
```

```
x =
7
```

x do të marrë vlerën $1 + 2 * 3 = 7$. Kjo ndryshore mund përdoret për t'iu referuar rezultateve të llogaritjeve të mëposhtme. Për shembull: do të rezultojë në llogaritjen e $4x$

```
>> 4*x ans =
28.0000
```

Operatorët aritmetikë në Matlab janë:

Operatori	Veprimi	Shembull
+	Mbledhja	2+3
-	Diferenca	2-3
*	Prodhimi	2*3
/	Herësi	2/3

Ndryshoret në MATLAB

Ndryshoret në MATLAB krijohen **me një deklarin vlerë dhënie**. Sintaksa e deklaramit të një ndryshoreje është:

emër ndryshoreje = një vlerë (ose një shprehje)

ku shprehje është një kombinim i vlerave numerike, operatorëve matematikore, variablave dhe thirrjeve funksionale.

Me fjalë të tjera, shprehja mund të përfshijë:

1. hyrje manuale
2. funksione të integruara
3. funksione të përcaktuara nga përdoruesit

Mbivendosja e ndryshores

Pasi të krijohet një ndryshore, ajo mund të ripërcaktohet. Përveç kësaj, **nëse nuk dëshironi të shihni rezultatet e ndërmjetme, mund të shtypni daljen numerike duke vendosur një pikëpresje (;) në fund të vijës**. Për shembull:

```
>> t = 5; >> t = t+1
t =
6
```

Mesazhet e gabimeve

Nëse shprehim gabimisht një shprehje, MATLAB do të kthejë një mesazh gabimi.

Për shembull, në vijim, kemi lënë shenjën e shumëzimit * në shprehjen e mëposhtme:

```
>> x = 10;
>> 5x
??? 5x
|
```

Error: Unexpected MATLAB expression.

Kryerja e korigjimeve

Për të bërë korigjime, mundemi, natyrisht, t'i përsërisim shprehjet. Por, nëse shprehja është e gjatë, bëjmë më shumë gabime duke shtypur një herë të dytë. Një komandë e shtypur më parë mund të rikthehet me tastën up-arrow ↑. Kur komanda shfaqet në komandën e shpejtë, mund të modifikohet nëse kërkohet dhe ekzekutohet.

Kontrolli i hierarkisë së operatorëve ose përparësisë

Le të marrim parasysh veprimin e mëparshëm aritmetik, por tani do të përfshijë kllapat (). Për shembull:

```
1 + 2 * 3 do të bëhet (1 + 2) * 3
>> (1+2)*3 ans =
    9
```

Duke shtuar kllapat, këto dy shprehje japin rezultate të ndryshme: 9 dhe 7.

Rendi në të cilin MATLAB kryen operacione aritmetike është pikërisht ajo që mësohet në kurset e algjebërës së shkollës së mesme. Ngritja në fuqi bëhet e para, pasuar nga shumëzimet dhe pjesëtimet, dhe më në fund nga mbledhja dhe zbritja. Megjithatë, rendi standard i përparësisë së operacioneve aritmetike mund të ndryshohet duke futur kllapat.

Për shembull, rezultati i $1 + 2 * 3$ është mjaft i ndryshëm nga shprehja e ngjashme me kllapa $(1 + 2) * 3$.

Rezultatet janë përkatësisht 7 dhe 9. Kllapat gjithmonë mund të përdoren **për të anashkaluar prioritetin**, dhe përdorimi i tyre rekomandohet në disa shprehje komplekse për të shmangur paqartësinë.

Prandaj, për të bërë vlerësimin e shprehjeve MATLAB ka krijuar një sërë rregullash. Rendi në të cilin vlerësohen veprimet aritmetike jepet më poshtë. Operatorët aritmetikë të MATLAB-it i binden rregullave të njëjta precedence si ato në shumicën e programeve kompjuterike.

Precedenca Operacionet matematikore

- 1) Përmbajtja e të gjitha kllapave vlerësohet së pari, duke filluar nga kllapat më të thella dhe duke u drejtuar jashtë.
- 2) Vlerësohen të gjithë ngritjet në fuqi, duke filluar nga e majta në të djathtë.
- 3) Vlerësohen të gjitha shumëzimet dhe pjesëtimet, duke filluar nga e majta në të djathtë.
- 4) Vlerësohen të gjitha shtesat dhe zbritjet, duke filluar nga e majta në të djathtë.

Për operatorët me përparësi të barabartë, vlerësimi është nga e majta në të djathtë. Për shembull:

```
>> 1/(2+3^2)+4/5*6/7 ans =
    0.7766 ose, nëse mungojnë
```

```
kllapat: >> 1/2+3^2+4/5*6/7 ans =
    10.1857
```

Kontrolli i paraqitjes të numrave real

MATLAB sipas parazgjedhjes shfaq vetëm 4 shifra dhjetore në rezultatin e llogaritjeve, për shembull: 163.6667, ose siç tregohet në shembujt e mësipërm. Megjithatë, MATLAB bën llogaritjet numerike me saktësi të dyfishtë, që është deri në 15 shifra dhjetore. Format i komandës kontrollon se si shfaqen rezultatet e llogaritjeve. Më poshtë janë dhënë disa shembuj të formateve të ndryshme së bashku me rezultatet rezultuese.

```
>> format short
>> x=-163.6667
```

```
>> format long
>> x= -1.636666666666667e+002
```

Për t'u kthyer në formatin standard, fut formatin e shkurtër (format short), ose thjesht formatin (format). Ka edhe disa formate të tjera. Për më shumë detaje, shikoni dokumentacionin MATLAB ose shkruani formatin e ndihmës (help format).

Menaxhimi i hapësirës së punës

Përmbajtja e hapësirës së punës vazhdon midis ekzekutimeve të komandave të veçanta. Prandaj, është e mundur që rezultatet e një problemi të kenë një efekt në tjetrin. Për të shmangur këtë mundësi, është mirë të përdoret një komandë e qartë në fillim të çdo llogarie të re të pavarur. >> **clear**

Komanda **clear** ose **clear all** heq të gjithë variablat nga hapësira e punës. Kjo liron kujtesën e sistemit.

Për të shfaqur **një listë të ndryshoreve** që janë aktualisht në kujtesë shtypim komandën who:

```
>> who
```

E cila do të japë më shumë detaje, të cilat përfshijnë madhësinë, shpërndarjen e hapësirës dhe klasën e variablave.

Ruajtja e sesionit tuaj të punës

Është e mundur të mbash gjurmët e gjithçkaje të bërë gjatë një sesioni MATLAB me komandën **diary**.

```
>> diary ose të japë një emër në një skedar të krijuar:
```

```
>> diary FileName ku FileName mund të jetë ndonjë emër skedari që ju zgjidhni.
```

Funksioni **Diary** është i dobishëm nëse dëshironi të ruani një sesion të plotë MATLAB. Ata ruajnë të gjitha hyrjet dhe daljet kur ato shfaqen në dritaren MATLAB. Kur doni ta ndalni regjistrimin, futni komandën **diary off**. Nëse dëshironi të filloni sërish regjistrimin, futni komandën **diary on**. Skedari që krijohet është një skedar i thjeshtë tekst. Mund të hapet nga një redaktor ose një program i përpunimit të tekstit dhe të redaktohet për të hequr materialet e panjohura ose për të shtuar komentet tuaja. Ju mund të përdorni llojin e funksionit për të parë skedarin e ditarit ose mund të redaktoni në një redaktues teksti ose të printoni. Kjo komandë është e dobishme, për shembull në procesin e përgatitjes së detyrave të shtëpisë ose dorëzimit të laboratorit.

Futja e deklaratave të shumëfishta për rresht

Është e mundur të futësh deklaratat të shumëfishta për rresht. Përdorni presjen (,) ose pikëpresjen (;) për të futur më shumë se një deklaratë në të njëjtën kohë. Komanda (,) lejon deklaratat të shumëfishta për rresht duke afishuar edhe rezultatin. >> a=7; b=cos(a), c=cosh(a) b =

```
0.6570 c = 548.3170
```

Komanda të ndryshme

Disa komanda shtesë të dobishme:

- Për të pastruar dritaren e komandës, shkruani **clc**
- Për të ndërprerë një llogaritje MATLAB, shtypni **ctrl-c**
- Për të vazhduar një linjë shkruani . . .

Komanda **help** - Për të parë dokumentacionin në internet, zgjidhni MATLAB Help nga menyuja Ndhimë ose MATLAB Help direkt në dritaren e komandës. Metoda e preferuar është përdorimi i Browser-it të Ndhimës. Shfletuesi i ndihmës mund të fillohet duke përzgjedhur ? ikonë nga shiriti i toolbox-it në desktop. Për shembull: >> help sqrt Ose >>lookfor emër funksioni

Funksionet Matematikore

MATLAB ofron shumë funksione matematikore të paracaktuara. Shkrimi i funksioneve **help elfun** dhe **help** bën thirrjen për listën e plotë të funksioneve elementare dhe të veçanta.

Ka një listë të gjatë të funksioneve matematikore që janë ndërtuar në MATLAB, si p.sh.: funksionet matematikore sin (x), cos (x), tan (x), ex, ln (x), vlerësohen nga funksionet sin, cos, tan, exp dhe log përkatësisht në MATLAB.

Më poshtë po listojmë disa funksione që përdoren më zakonisht, ku ndryshoret x dhe y mund të jenë numra, vektorë ose matrica.

Funksione trigonometrike

Funksione numerike

Funksione numerike

cos(x) - kosinusi	abs(x) – Vlera Absolute	exp(x) Exponenciali
sin(x) – Sinusi	sign(x) –shenja e numrit	sqrt(x) rrënja katrore
tan(x) Tangenti	max(x) – Vlera Maksimale	rem(x) Mbetja e pjesëtimit
acos(x) Arc Kosinusi	min(x) Minimum value	log(x) Logaritmi natyror
asin(x) Arc Sinusi	ceil(x) rrumbullakimi drejt $+\infty$	log10(x) logaritmi me bazë 10
atan(x) Arc Tangenti	floor(x) rrumbullakimi drejt $+\infty$	round(x) rrumbullakimi drejt int

Përveç funksioneve elementare, MATLAB përfshin një numër vlerash konstante të paracaktuara.

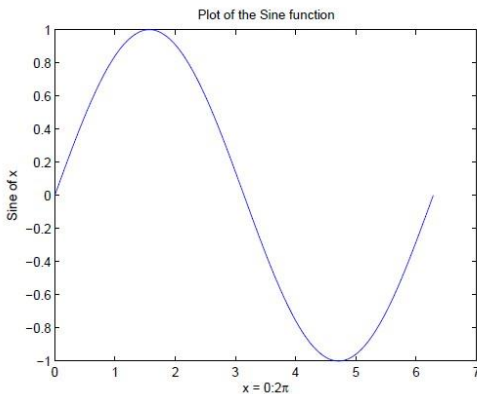
pi numri $\pi = 3:14159 \dots$ i,j njësitë imagjinare Inf vlera infinit

NaN nuk është një numër

Më poshtë po ilustronim disa shembuj tipikë që lidhen me funksionet elementare të përcaktuara më parë.

Si shembull i parë, vlerësojmë vlerën e shprehjes: $y = e^{-a}\sin(x) + 10\sqrt{y}$, për vlera fillestare $a = 5$, $x = 2$, dhe $y = 8$, i cili llogaritet në Matlab nga komandat:

```
>> a = 5; x = 2; y = 8;
>> y = exp(-a)*sin(x)+10*sqrt(y)    y =
    28.2904
Shembuj të tjerë:
>> log(142)                    ans = 4.9558
>> log10(142)                ans =2.1523
>> sin(pi/4)                 ans =0.7071
>> exp(10)                    ans =2.2026e+004
```



Përdorni vetëm funksione të ndërtuara në anën e djathtë të një shprehjeje. Ri-përcaktimi i vlerës në një funksion të integruar mund të krijojë probleme.

Ka disa përjashtime. **Numrat imagjinare i dhe j përcaktohen me vlerë fillestare rrënja katrore e -1.** Në ciklin **for** përdoren si **indekse** në vend të i dhe j, emërtimet ii ose jj.

Ndërtimi i grafikëve

MATLAB ka një grup të shkëlqyeshëm të mjeteve grafike.

Plotësimi i një grupi të dhënash të caktuar ose rezultatet e

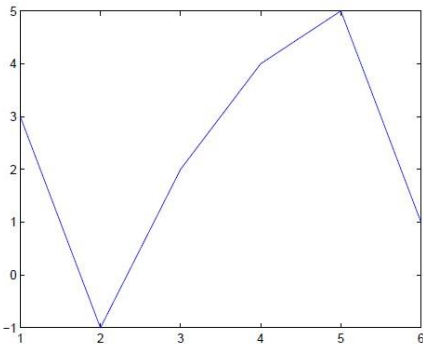
llogaritjes mundësohen me pak komanda. Duke u përpjekur për të kuptuar ekuacionet matematikore me grafikë është një mënyrë e kënaqshme dhe shumë efiçente për të mësuar matematikën.

Procedura bazë e grafik-imit MATLAB, për shembull në 2D, është të marrë një vektor të x-koordinatave, $x = (x_1; \dots; x_N)$, dhe një vektor i y-koordinatave, $y = (y_1; \dots; y_N)$, gjeni pikë (x_i, y_i) , me $i = 1; 2; \dots; n$ dhe pastaj bashkohen me ato me linja të drejta. Ju duhet të përgatisni x dhe y në një formë të njëjtë array; d.m.th., x dhe y janë të dy vektorë me gjatësi të njëjtë. Komanda MATLAB për të ndërtuar një grafik është plot(x,y). Për shembull: Jepen vektorët $x = (1; 2; 3; 4; 5; 6)$ dhe $y = (3; -1; 2; 4; 5; 1)$ të ndërtohen komandat për të ndërtuar grafikun në Matlab:

```
>> x = [1 2 3 4 5 6];
>> y = [3 -1 2 4 5 1];
```

```
>> plot(x,y)
```

Funksionet për ndërtimin e grafikëve kanë forma të ndryshme në varësi të argumenteve të hyrjes. Nëse y është një vektor atëherë plot(y) prodhon një grafik linear të elementeve të y kundrejt indeksit të elementit të tij. Nëse specifikojmë dy vektorë, siç u përmend më lart, plot(x,y) prodhon një grafik të y kundrejt x. Për shembull, për të ndërtuar grafikun e



funksionit $\sin(x)$ në intervalin $[0; 2\pi]$, së pari krijojmë një vektor të vlerave x duke filluar nga 0 në 2π , pastaj llogarisim sin e e këtyre vlerave, dhe më në fund grafiku i tij dhe komandat në Matlab:

```
>> x = 0:pi/100:2*pi;
>> y = sin(x);
>> plot(x,y)
```

MATLAB ju mundëson të shtoni etiketat dhe titujt e boshteve. Për shembull, duke përdorur grafikun nga shembulli i mëparshëm, shtoni etiketat e boshtit x dhe y. Tani emërtoni akset dhe shtoni një titull. Karakteri π krijon simbolin π . Një shembull i grafikut 2D është paraqitur më poshtë:

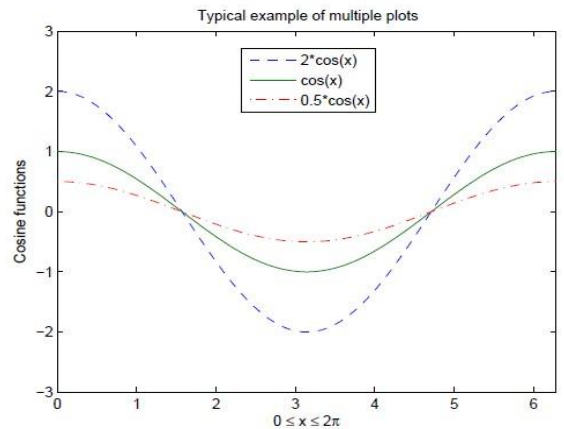
Komandat në Matlab për emërtimin e boshteve:

```
>> xlabel('x = 0:2\pi')
>> ylabel('Sine of x')
>> title('Plot of the Sine function')
```

Ngjyra e një kurbe të vetme është, sipas parazgjedhur, ngjyrë blu. Ngjyra e dëshiruar tregohet nga një argument i tretë ne funksionin plot. Për shembull, e kuqe përzgjidhet nga plott(x,y,'r').

Argumentet e shumëfishta (x; y) krijojnë grafik të shumëfishta me një thirrje të vetme për grafik nga komanda plot. Për shembull komandat e mëposhtme paraqesin grafikun me tre vija:

```
>> x = 0:pi/100:2*pi;
>> y1 = 2*cos(x);
>> y2 = cos(x);
>> y3 = 0.5*cos(x);
>> plot(x,y1,'--',x,y2,'-',x,y3,':')
>> xlabel('0 \leq x \leq 2\pi')
>> ylabel('Cosine functions')
>> legend('2*cos(x)','cos(x)','0.5*cos(x)')
>> title('Typical example of multiple plots')
>> axis([0 2*pi -3 3])
```



MATLAB përdor stilin e linjës dhe ngjyrën për të dalluar grupet e të dhënave në grafik. Megjithatë, mund të ndryshoni pamjen e këtyre komponentëve grafik ose të shtoni shënime në grafik për të ndihmuar në shpjegimin e të dhënave tuaja.

SYMBOL	COLOR	SYMBOL	LINE STYLE	SYMBOL	MARKER
k	Black	—	Solid	+	Plus sign
r	Red	--	Dashed	o	Circle
b	Blue	:	Dotted	*	Asterisk
g	Green	-.	Dash-dot	.	Point
c	Cyan	none	No line	×	Cross
m	Magenta			s	Square
y	Yellow			d	Diamond

Është e mundur që të specifikohen stilet e linjave, ngjyrat dhe shënuesit (p.sh., qarqet, shenjat shtesë ...) duke përdorur komandën plot: plot(x,y,'style_color_marker'). Për të gjetur informacione shtesë, shkruani help plot ose doc plot.

Strukturat e kontrollit dhe përsëritëse

Më kryesoret janë: stuktura if; ciklet for dhe while:

a) if ... elseif ... else ... end

Struktura e tij:

Inicializimi i variablit të kontrollit bëhet para IF

if<Kushti llogjik1>

Veprim1;

Elseif <Kushti llogjik2> Veprim2; **end**

Ndërtoni një strukturë IF që gjen më të madhin midis dy numrave të dhënë A dhe B. **if** A > B

‘A është më i madhi’

elseif A < B

‘B është më i madhi’

elseif A = B

‘A është e barabartë me B’

End

b) switch op case vlera 1

veprim 1

... case vlera n veprim n

otherwise

veprim **end**

c) for ... end

Struktura e tij:

Inicializimi i variablit të kontrollit, vlerësimi në kusht dhe ndryshimi bëhen në deklarin e ciklit FOR **for** variabël kontrolli = vlerë fillestare : vlerë përfundimtare

Veprim **End**

Ndërtoni një strukturë FOR që formon matricën H me elemente 1/(Shumën e indekseve):

for i = 1:m

for j = 1:n H(i,j) =

1/(i+j)

end

d) while ... end

Struktura e tij:

Inicializimi i variabëlit të kontrollit:

while <Kushti llogjik>

Veprime;

End

Shembull: Ndërtoni një strukture WHILE që gjen shumën e 1000 numrave të plotë.

i = 0; while (i<1000)

 s = s + i;

 i = i + 1;

end

Ushtrimi 1 Gjenerimi i një numri të rastit dhe kontrolli nëse është çift.

```
a = randi(100, 1); if rem(a, 2) == 0
    disp('a is even') end
```

Ushtrimi 2 Gjenerimi i një numri të rastit dhe kontrolli nëse është më i vogël se 30 afishon small, nëse është midis 30 dhe 80 afishon medium dhe nëse është më i madh se 80 afishon large.

```
a = randi(100, 1);
if a < 30    disp('small') elseif a <
80    disp('medium') else
disp('large')
end
```

Ushtrimi 3 Ndërtoni një strukturë switch, e cila për një ditë të caktuar të javës paraqet mesazhin e mëposhtëm:

Monday	Start of the work week
Tuesday	Day 2
Wednesday	Day 3
Thursday	Day 4
Friday	Last day of the work week
Ne te kundert	Weekend!

```
[dayNum, dayString] = weekday(date, 'long', 'en_US');
```

```
switch dayString case 'Monday'
    disp('Start of the work week')
case 'Tuesday'    disp('Day 2')
case 'Wednesday' disp('Day 3')
case 'Thursday'  disp('Day 4')
case 'Friday'    disp('Last day of the work week')
otherwise       disp('Weekend!')
end
```

Ushtrimi 4 Ndërtoni një strukturë kontrolli që për një numër të dhënë nga përdoruesi kontrollon nëse është negativ, pozitiv apo zero.

```
yourNumber = input('Shkruaj nje numer: ');
if yourNumber < 0
    disp('Negativ')
elseif yourNumber > 0
    disp('Pozitiv')
else disp('Zero') end
```

Ushtrimi 5 Ndërtoni një strukturë, që kontrollon nëse një karakter i dhënë nga përdoruesi është një zanore ose jo duke përdorur Strukturën switch.

Ushtrimi 6 Ndërtoni një strukturë, që gjen maksimumin midis dy numrave real te dhene nga perdoruesi.

Ushtrimi 7 Ndërtoni një strukturë, që gjen heresin dhe mbetjen midis dy numrave te plote te dhene nga perdoruesi.

Ushtrimi 8 Ndërtoni një strukturë, që lexon tre numra A, B, C dhe i afishon ata të renditur në rendin zbritës.

Ushtrimi 9 Të ndërtohet një cikël për të përcaktuar vlerat e funksionit:

$y = x + 3$ për vlera të x midis x_0 dhe x_1 , duke e ndryshuar atë me hapin dx .

Ushtrimi 10 Jepet funksioni $y = x^2 + 6x + 2$ i përcaktuar në segmentin (a_1, a_2) dhe një numër b .

Të ndërtohet një cikël që segmentin (a_1, a_2) e ndan në b pjesë të barabarta dhe llogarit vlerat e funksionit në pikat e ndarjes.

Ushtrimi 11 Jepet funksioni $y = x^2 - 3x + 1$.

Të gjenden vlerat e këtij funksioni në pikat x_1, x_2, \dots, x_n , të cilat lexohen nga tastiera njëra pas tjetrës.

Ushtrimi 12 Të ndërtohet një cikël që gjen shumën e n numrave të parë të secilës nga seritë e mëposhtme:

a) $S = 1 + 2 + 3 + \dots$

b) $S = 1 + 3 + 5 + \dots$

c) $S = -1 + 7 - 13 + 19 - \dots$

d) $S = -2 + 4 - 6 + 8 - 10 + \dots$

Matricat

Matricat janë elementet bazë të mjedisit MATLAB. Një matricë është një grup dy-dimensional, i përbërë nga m rreshta dhe n kolona. Raste të veçanta janë vektorët e kolonës ($n = 1$) dhe rresht vektorë ($m = 1$). Në këtë seksion do të ilustrim se si të aplikojmë operacione të ndryshme në matrica.

Një vektor është një rast i veçantë i një matrice. Qëllimi i këtij seksioni është të tregojë se si të krijohen vektorë dhe matrica në MATLAB. Siç u diskutua më herët, një sërë dimensionesh:

$1 \times n$ quhet një **vektor rresht**, ndërsa një sferë e dimensionit **$m \times 1$** quhet **një vektor i kolonës**. Elementet e vektorëve në MATLAB janë mbyllur me kllapa katrorë dhe janë të ndara me **hapësirë** ose me **presje**. Për shembull, për të futur një **vektor rresht**, **v**, shkruani: $\gg v = [1\ 4\ 7\ 10\ 13]$ $v =$

```
1 4 7 10 13
```

Vektori shtyllë krijohet në mënyrë të ngjashme, megjithatë, pikëpresje (;) duhet të ndajë elementet e saj:

```
\gg w = [1;4;7;10;13]
```

```
w =
```

```
1
```

```
4
```

```
7
```

```
10
```

```
13
```

Nga ana tjetër, një vektor i rreshtit **konvertohet** në një vektor të kolonës duke përdorur operatorin e **trans-pozimit**, i cili është shënuar me një apostrof (').

```
>> w = v'
w =
    1
    4
    7
   10
   13
```

Kështu, v (1) është elementi i parë i vektorit v, v (2) elementi i tij i dytë, dhe kështu me radhë. Për më tepër, për të hyrë në blloqe të elementeve, përdorim **simbolin e zorrës së trashë** të MATLAB (:).

Për shembull, për të hyrë në tre elementët e parë të v, shkruajmë:

```
>> v(1:3) ans =
    1 4 7
```

Ose, të gjithë elementët nga e treta deri tek elementët e fundit:

```
>> v(3,end) ans =
    7 10 13
```

Nëse v është një vektor, shkruaj:

```
>> v(:) prodhon një vektor të kolonës, ndërsa kur shkruan:
>> v(1:end) prodhon një vektor rresht.
```

Një matricë është një grup numrash. Për të futur një matricë në MATLAB ju duhet:

- Të filloni me një kllapë katrore, [
- elementet ndahen në një rresht me hapësira ose presje (,)
- përdorni një pikëpresje (;) për ndarjen e rreshtave
- përfundoni matricën me një kllapë tjetër,].

Për të krijuar matricën A në matlab shkruajmë komandën:

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

MATLAB më pas afishon matricën 3 x 3, si më poshtë:

```
A =
    1    2    3
    4    5    6
    7    8    9
```

Vëmë re se përdorimi i pikëpresjeve (;) këtu është ndryshe nga përdorimi i tyre i përmendur më herët për të shtypur prodhimin ose për të shkruar komanda të shumta në një linjë të vetme.

Sapo të kemi hyrë në matricë, ai ruhet dhe ruhet automatikisht në hapësirën e punës. Mund t'i referohemi thjesht si matricë A. Mund ta shohim një element të veçantë në një matricë, duke specifikuar vendndodhjen e tij.

Ne shkruajmë:

```
>> A(2,1) ans =
    4
```

A (2,1) është një element i vendosur në rreshtin e dytë dhe kolonën e parë. Vlera e tij është 4.

Ne zgjedhim elemente në një matricë, ashtu siç bëjmë për vektorët, por tani kemi nevojë për dy indekse. Elementi i rreshtit i dhe kolona j e matricës A është e shënuar me A (i, j). Kështu, A (i, j) në MATLAB i referohet elementit A_{ij} të matricës A. Indeksi i parë është numri i rreshtit dhe indeksi i dytë është numri i kolonës. Për shembull, A (1,3) është një element i rreshtit të parë dhe kolonës së tretë. Tek matrica e dhënë vlera është: $A (1,3) = 3$.

Korrigjimi i çdo hyrjeje është i lehtë përmes indeksimit.

Kështu zëvendësojmë $A(3,3) = 9$ me $A(3,3) = 0$. Rezultati është:

```
>> A(3,3) = 0
```

```
A =
```

```
 1   2   3
 4   5   6
 7   8   0
```

Elementet e përgjithshëm të matricës janë si $A(i, j)$, ku $i \geq 1$ dhe $j \geq 1$. Indekset me vlerë ≤ 0 nuk pranohen në MATLAB.

Operatori (:) Është një operator i dobishëm, efikas dhe i përshtatshëm në MATLAB kur punojmë me matricat ose vektorët që janë me **përmasa shumë të mëdha**. Për shembull, supozojmë se duam të krijojmë një vektor x të përbërë nga pikat $(0, 0.1, 0.2, 0.3, \dots, 5)$. Ne mund të përdorim komandën në Matlab për të krijuar vektorin rresht me 51 elemente:

```
>> x = 0:0.1:5;
```

Nga ana tjetër, ekziston një komandë për të gjeneruar vektorë: `linspace`. Është e ngjashme me operatorin (:), por jep kontroll të drejtpërdrejtë **mbi numrin e elementeve**. Për shembull:

```
>> y = linspace(a,b) gjeneron një vektor të rreshtit y me 100 pika të ndara linearisht të barabarta midis tyre nga a në b.
```

```
>> y = linspace(a,b,n) gjeneron një vektor të rreshtit y të n pikave të ndara në mënyrë lineare midis a dhe b.
```

Kjo është e dobishme kur duam të ndajmë një interval në një numër nën-intervalesh të së njëjtës gjatësi. Për shembull:

```
>> theta = linspace(0,2*pi,101) ndan intervalin në 100 nën-intervale të barabarta, duke krijuar një vektor prej 101
```

elementësh.

Operatori (:) mund të përdoret gjithashtu për **të zgjedhur një rresht ose kolonë** të caktuar.

Për shembull, deklarata $A(m:n, k:l)$ specifikon rreshtat m në n dhe kolonën k deri në l . Shprehjet e indeksit i referohen një pjese të një matrice.

Për shembull:

```
>> A(2,:) ans =
    4 5 6
```

Janë të gjithë elementët e rreshtit të dytë të matricës A . Operatori (:) mund të përdoret gjithashtu për të nxjerrë një nën-matricë nga një matricë A .

```
>> A(:,2:3) ans =
    2 3
    5 6
    8 0
```

$A(:,2:3)$ është një nën-matricë me dy kolonat e fundit të A . Një rresht ose një kolonë e një matrice mund të fshihet duke vendosur atë në një vektor null [].

```
>> A(:,2)=[]
ans =
    1 3
    4 6
    7 0
```

Për të nxjerrë një nën-matrice B që përbëhet nga rreshtat 2 dhe 3 dhe kolonat 1 dhe 2 të matricës A :

```
>> B = A([2 3],[1 2])
B =
    4 5
    7 8
```

Për të shkëmbyer rreshtat 1 dhe 2 të A , përdorni vektorin e indekseve të rreshtit së bashku me operatorin (:).

```
>> C = A([2 1 3],:)
```

```
C =
    4 5 6
    1 2 3
    7 8 0
```

Është e rëndësishme të theksohet se operatori (:) qëndron për të gjitha kolonat ose të gjitha rreshtat.

Për të krijuar një version vektorial të matricës A:

```
>> A(:)
ans = 1 2 3 4 5 6 7 8 0
```

Për të fshirë një rresht ose kolonë të një matrice, përdorni **operatorin bosh** të vektorit, [].

```
>> A(3,:) = []    A =
    1 2 3
    4 5 6
```

Rreshti i tretë i matricës A është fshirë tani. Për të rivendosur rreshtin e tretë, përdorim teknikën për krijimin e një matrice.

```
>> A = [A(1,:);A(2,:);[7 8 0]]
A =
    1 2 3
    4 5 6
    7 8 0
```

Matrica A tani është kthyer në formën e saj origjinale.

Për të përcaktuar dimensionet e një matrice ose vektori, përdorni komandën size. Për shembull:

```
>> size(A)    ans =
    3 3
```

do të thotë 3 rreshta dhe 3 kolona. Ose >> [m,n]=size(A).

Operacioni i trans-pozimit është shënuar me një apostrof ('). Është një matricë rreth diagonales së saj kryesore dhe e kthen një vektor rresht në një vektor kolonë. Jepet matrica:

```
A = 1 2 3
    4 5 6
    7 8 9
>> A' ans =
    1 4 7
    2 5 8
    3 6 0
```

Duke përdorur simbol linear algjebër, trans-pozimi i matricës reale $m \times n$ është matrica e matricës $n \times m$ që rezulton nga ndërthurja e rreshtave dhe shtyllave të A. Matrica e trans-pozuar shënohet A^T .

Bashkimi i matricave - Matricat mund të përbëhen nga nën-matricat. Jepet matrica:

```
A = 1 2 3
    4 5 6
    7 8 9
```

Matrica e re B do të jetë:

```
>> B = [A 10*A; -A [1 0 0; 0 1 0; 0 0 1]]
B =     1     2     3    10    20    30
      4     5     6    40    50    60
      7     8     9    70    80    90
     -1    -2    -3     1     0     0
     -4    -5    -6     0     1     0
     -7    -8    -9     0     0     1
```

MATLAB siguron funksione që gjenerojnë matricat elementare. Matrica zeros, matrica ones, dhe matrica identike janë krijuar nga funksionet zeros, ones dhe eye, respektivisht.

eye(m,n) Kthen një matricë m-nga-n me 1 në diagonalen kryesore

eye(n) Kthen një matricë identike katrore n-nga-n

zeros(m,n) Kthen një matricë m-nga-n me vlera 0

ones(m,n) Kthen një matricë m-nga-n me vlera 1

diag(A) Ekstrakton diagonalin e matricës A

rand(m,n) Kthen një matricë m-nga-n të numrave të rastit

Për një listë të plotë të matricave elementare dhe manipulimeve të matricës, type help elmat ose doc elmat.

Më poshtë janë dhënë disa shembuj:

```
>> b=ones(3,1) b =
      1
      1
      1
```

Është e njëjtë si të shkruajmë komandën: >> b=[1;1;1]

```
>> eye(3) ans =
      1  0  0
      0  1  0
      0  0  1
>> c=zeros(2,3)
c =
      0  0  0
      0  0  0
```

Dy funksione të tjera të rëndësishme të krijimit të matricës janë rand dhe randn, të cilat gjenerojnë matricat e numrave të rastësishëm duke përdorur sintaksën e njëjtë si eye.

Përveç kësaj, është e rëndësishme të kujtojmë se tre operacionet elementare të shtimit (+), zbritjes (-) dhe shumëzimit (*) vlejnë edhe për matricat sa herë që dimensionet janë të pajtueshme.

Dy funksione të tjera të rëndësishme të krijimit të matricës janë rand dhe randn, të cilat gjenerojnë matricat e numrave të rastësishëm duke përdorur sintaksën e njëjtë si eye.

Përveç kësaj, matricat mund të ndërtohen në një formë bllok. Psh. Jepet matrica C e përcaktuar nga $C = [1 \ 2; 3 \ 4]$, të krijohej një matricë D si më poshtë:

```
>> D = [C zeros(2); ones(2) eye(2)]
D =
      1  2  0  0
      3  4  0  0
      1  1  1  0
```

1 1 0 1

MATLAB siguron funksione për krijimin e matricave të veçanta. Këto matrica kanë veçori interesante që i bëjnë ato të dobishme për ndërtimin e shembujve dhe për algoritmet e testimit.

hilb	matrica Hilbert
invhilb	inversi i matricës Hilbert
magic	katrori magjik
pascal	matrica Pascal
toeplitz	matrica Toeplitz
vander	matrica Vandermonde
wilkinson	Matrica e testit të ekuivalencës së Wilkinson

MATLAB ka **dy** lloje të ndryshme të **operatorëve aritmetikë**:

1. operatorët matematikorë aritmetikë
2. operatorët aritmetikë të grupeve.

MATLAB lejon operatorët aritmetikë: +, -, * dhe ^ të kryhen në matrica.

- $A+B$ ose $B+A$ është e vlefshme nëse A dhe B janë të së njëjtës madhësi
- $A*B$ është e vlefshme nëse numri i kolonave A është i barabartë me numrin e rreshtave të B
- A^2 e vlefshme nëse A është katror dhe i barabartë $A*A$
- $\alpha*A$ or $A*\alpha$ shumëfishon çdo element të A me α

Nga ana tjetër, operatorët aritmetikë të grupeve kryhen element-nga-element. Karakteri **pikë** dallon operatorët e grupeve nga operatorët e matricës. Lista e operatorëve të grupeve është treguar më poshtë, nëse A dhe B janë dy matrica të së njëjtës madhësi me elementë $A = [a_{ij}]$ dhe $B = [b_{ij}]$, atëherë komanda:

- `.*` Shumëzim element-për-element
- `./` pjesëtim element-për-element
- `.^` ngritje ne fuqi element-për-element

Komanda `>> C = A.*B` prodhon një matricë tjetër C të së njëjtës madhësi me A dhe B , ku çdo element përftohet si prodhim $[a_{ij}] * [b_{ij}]$. Për shembull:

```
A =           B =
    1  2  3      10 20 30
    4  5  6      40 50 60
    7  8  9      70 80 90
```

```
>> C = A.*B
C =
    10    40    90
   160   250   360
   490   640   810
```

```
>> A.^2
ans =
    1    4    9
   16   25   36
   49   64   81
```



```
>> A=[1 2;3 4];
>> B=[5 6;7 8];
>> A*B
    19    22
    43    50
```

Ndërsa

```
>> A.*B
     5    12
    21    32
```

Ushtrimi 2

Elementi në rreshtin e i -të dhe shtyllën e j -të për një matricë A shënohet $A(i, j)$, i cili është edhe elementi i përgjithshëm i tij.

Per shembull:

```
A = zeros(2,2);
A(1,1) + A(1,2) + A(2,2)
```

Ushtrimi 3

Përcaktimi i një matrice duke dhënë elementet e tij:

```
>> A = [16 3; 5 10]
A = 16     3
     5    10
>> B = [3 4 5
        6 7 8]
B = 3 4 5
     6 7 8
```

Ushtrimi 4

Krijimi i një vektori dhe matrice trans-posë:

Vektori

```
>> a=[1 2 3];
>> a'
     1
     2
     3
```

Matrica

```
>> A=[1 2; 3 4]; >> A' ans =
     1     3
     2     4
```

Ushtrimi 5

- Krijë një vektor me intervale të njëpasnjëshme $h=0.5$ në segmentin $0:\pi$: `>> x=0:0.5:pi` x =
0 0.5000 1.0000 1.5000 2.0000 2.5000 3.0000
- Krijë një vektor me n elemente ($n=7$) në segmentin $0:\pi$: `>> x=linspace(0, pi, 7)`

```
x =
0 0.5236 1.0472 1.5708 2.0944 2.6180 3.1416
```

c) Krijo një vektor në segmentin 10:100, ku hapi llogaritet në hapësira logaritmike >>

```
x=logspace(1,2,7)
x =
10.0000 14.6780 21.5443 ... 68.1292 100.0000
```

Ushtrimi 6

Shtimi i elementeve në një Vektor:

```
>> A=1:3
A=
1 2 3
>> A(4:6)=5:2:9
A=
1 2 3 5 7 9
>> B=1:2
B=
1 2
>> B(5)=7; B=
1 2 0 0 7
```

Ushtrimi 7

Shtimi i elementeve në një Matricë:

```
>> C=[1 2; 3 4]
C=
1 2
3 4
>> C(3,:)=5 6;
C=
1 2
3 4
5 6
>> D=linspace(4,12,3);
>> E=[C D']
E=
1 2 4
3 4 8
5 6 12
```

Ushtrimi 8

Operatori dy pika (:) është një nga operatorët më të rëndësishëm të MATLAB-it dhe ka shumë përdorime.

3:-2:-11 është një vektor që përmban numra të plotë nga 3 deri në -11, me zvogëlim prej -2.

Shprehjet e indeksit që përfshijnë dy pika referohen pjesëve të një matrice. $A(1:3, 2)$ është e para për elementët e tretë të kolonës së dytë të A.