ლექცია 1

შესავალი

MATLAB-ერთდროულად წარმოადგენს ოპერაციულ სისტემასაც და დაპროგრამების ენასაც. სახელი MATLAB წარმოსდგება მათემატიკისა და ლაბორატორიისაგან. ამ სისტემის პირველი ვერსია დაიწერა 70-იან წლებში ნიუ-მექსიკოსა და სტანფორდის უნივერსიტეტებში და მისი დანიშნულება იყო მატრიცათა თეორიის, წრფივი ალგებრისა და რიცხვითი ანალიზის შესწავლა. დღევანდელი MATLAB-ის შესაძლებლობები გაცილებით უფრო ფართოა, მისი გამოყენების სფეროებია: მათემატიკა და გამოთვლები, ალგორითმების შემუშავება, მოდელირება, მონაცემთა ანალიზი, სამეცნიერო და საინჟინრო გამოთვლები.

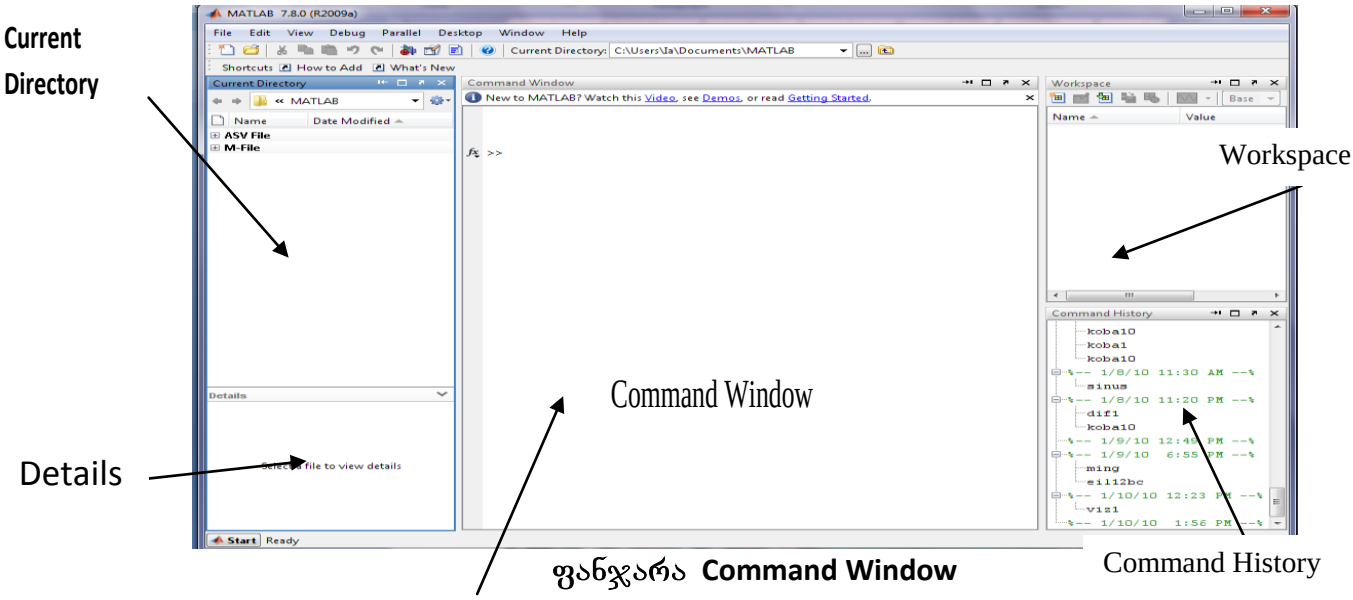
MATLAB-ეს არის ინტერაქტიური სისტემა, რომლის ძირითადი ობიექტია მასივი, რომლისთვისაც არ მოითხოვება განზომილების ცხადი სახით მითითება. რაც საშუალებას გვაძლევს ამოვხსნათ ვექტორულ-მატრიცული ტიპის ამოცანები უფრო ნაკლებ დროში ვიდრე სხვა დაპროგრამების ენებზე (ისეთი როგორიცაა C, C++ , Fortran).

ფაილებს, რომლებიც შეიცავენ MATLAB ენის კოდებს უწოდებენ M-ფაილებს. M-ფაილის შესაქმნელად გამოიყენება ტექსტური რედაქტორი. შესაძლებელია M-ფაილი შევქმნათ ნებისმიერი ტექსტური რედაქტორში(მაგ. Word-ში) ASCII კოდებით, მაგრამ MATLAB-ს გააჩნია საკუთარი ტექსტურ-გამმართველი: built რედაქტორი.

MATLAB-ის გაშვება

პროგრამა MATLAB-ის გაშვება ისევე ხდება როგორც ნებისმიერი პროგრამის გაშვება: Windows -ის ოპერაციულ სისტემაში მთავარი მენიუდან Start→All Programs→MATLAB→**R2009a →Matlab R2009a**, ან შესაძლებელია აგრეთვე ვისარგებლოთ სამუშაო მაგიდაზე განთავსებული პროგრამის იარლიყით. MATLAB-ის გაშვების შემდეგ გამოჩნდება ფანჯარა, რომელსაც MATLAB-ის სამუშაო მაგიდა ეწოდება და ის გაჩუმების პრინციპით შემდეგნაირად გამოიყურება (ნახ.1). სამუშაო მაგიდა გაყოფილია 5 ნაწილად: 'Command Window','Command History', 'Workspace, 'Current Directory' ,'Details'.

შესაძლებელია ამ ფანჯრების ცალკე გამოტანაც. ამისათვის უნდა გამოვიყენოთ Desktop მენიუს მაგიდის Desktop Layout-ის ქვემენიუ. თუ რომელიმე ფანჯრის მარჯვენა ზედა კუთხეში გაღუნულ ისარზე დააწკაპუნებთ შეგიძლიათ გააქროთ ან ცალკე გამოიტანოთ შესაბამისი ფანჯარა. გარდა ამისა შესაძლებელია ამ ფანჯრების სათაურის ზოლში მაუსის ჩავლება და გადათრევა, ასევე მაუსით ზომების შეცვლა. განვიხილოთ ეს ფანჯრები.



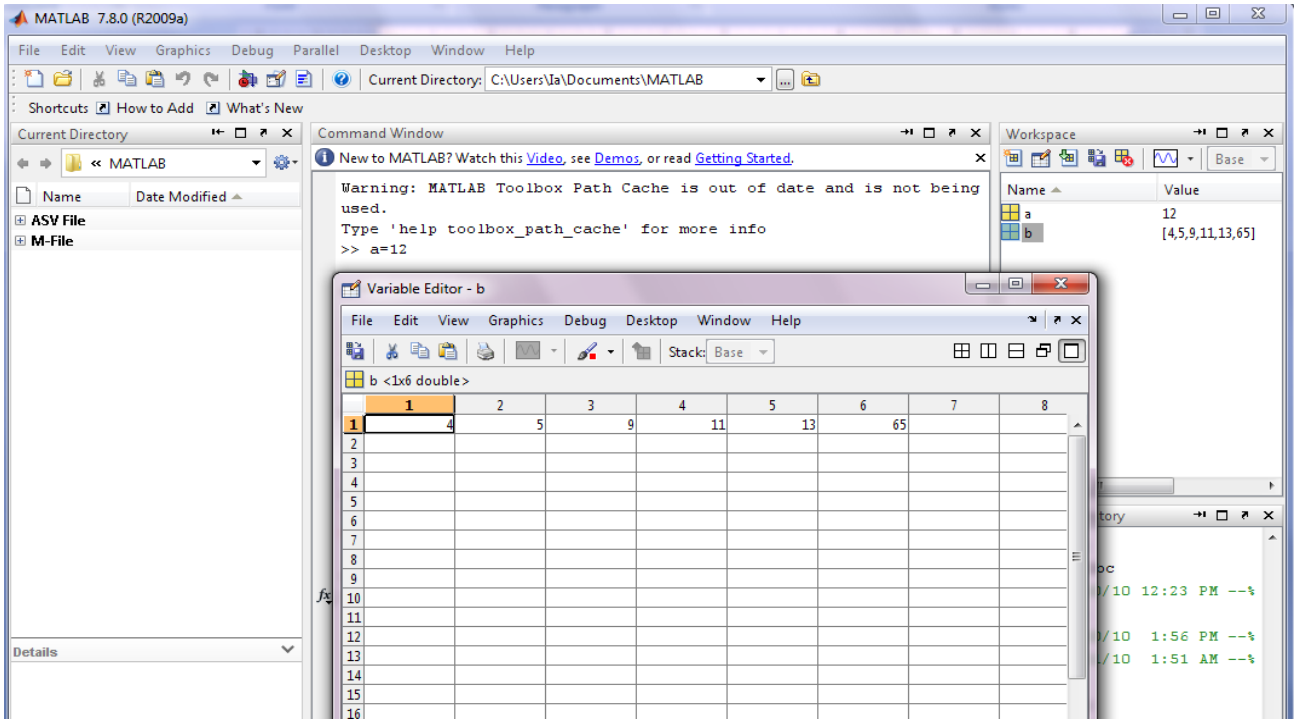
ფანჯარა **Command Window**

Command Window - ბრძანებათა ფანჯარა, ფანჯრის ზედა ნაწილში შესაძლოა შემოთავაზებული იყოს ინფორმაცია **MATLAB –** თან დაკავშირებით, ბრძანებათა ფანჯარაში განთავსებულია >> ან EDU >> (სტუდენტური ვერსია) ნიშანი. ამ ნიშნის შემდეგ შეგვაქვს შესასრულებელი ბრძანება და ვაჭერთ **ENTER** ღილაკზე. ფანჯარა **Command Window** გაყოფილია ორ ზონად: დათვალიერებისა და რედაქტირების ზონებად. დათვალიერების ზონიდან შესაძლებელია ნებისმიერი ფრაგმენტის კოპირება.  და  ღილაკებით შესაძლებელია უკვე შესრულებული ბრძანების გამოტანა ბრძანებათა სტრიქონში, მათი ხელახლა შესრულების მიზნით. **clc** ბრძანება გაასუფთავებს ბრძანებათა ფანჯარის შიგთავსს, (თუმცა **MATLAB –** ს კვლავ ახსოვს შესრულებულ ბრძანებათა მიდევრობა).

ფანჯარა **Workspace**

Workspace -სამუშაო არე. **MATLAB** –თან მუშაობის დროს სპეციალურად გამოიყოფა ოპერატიული მეხსიერება, რომელშიც სეანსის დროს გამოყენებული ცვლადები და მათი მნიშვნელობები ინახება. მას **MATLAB** -ის სამუშაო არე **–Workspace** ეწოდება.

Workspace ფანჯარაში ჩანს გამოყენებული ცვლადებიც და მათი მნიშვნელობები. თუ ორჯერ დავაკაკუნებთ მაუსის მარცხენა ღილაკს ცვლადის სახელზე, გაიხსნება **Variable Editor** ფანჯარა (ცვლადების რედაქტორი), რომელშიც შეიძლება შევცვალოთ ცვლადების მნიშვნელობები. ეს ფანჯარა იხსნება აგრეთვე **openvar** <ცვლადის სახელი> ბრძანებითაც.



ნებისმიერი ცვლადის მნიშვნელობის ნახვა და შეცვლა შესაძლებელია ასევე ბრძანებათა სტრიქონიდანაც. სანახავად საკმარისია ბრძანებათა სტრიქონში ავკრიფოთ ცვლადის სახელი და ვიმოქმედოთ **ENTER** ღილაკზე. შესაცვლელად თავიდან ვანიჭებთ საჭირო მნიშვნელობას.

თუ მუშაობის პროცესში დიდი რაოდენობით ცვლადებს ვიყენებთ, იზრდება სამუშაო სივრცე და მუშაობის ეფექტურობის მცირდება საჭიროა პერიოდულად არასაჭირო ცვლადების წაშლა მეხსიერებიდან.

clear < ცვლადის სახელი1> **<** ცვლადის სახელი2>...

ან **clear ('<** ცვლადის სახელი1>', **'<** ცვლადის სახელი2>')ბრძანება წაშლის მითითებულ ცვლადებს,

clear წაშლის ყველა ცვლადს ერთდროულად.

ბრძანებათა სტრიქონში **who** ბრძანებით **Command Window** ფანჯარაში აისახება **Workspace** ფანჯარის ცვლადთა სია.

ბრძანება **whos** გვაძლევს დეტალურ ცნობებს ცვლადებზე: სახელს, ზომას, ტიპს.

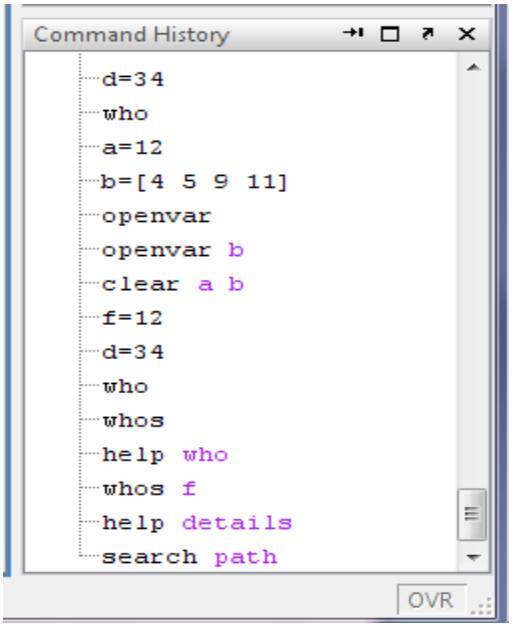
სამუშაოს დამთავრების შემდეგ შესაძლებელია ყველა გამოთვლილი ცვლადის შენახვა ფაილში შემდეგი ბრძანებების მიმდევრობით:

File-Save Workspace as..., ან **save path_to_file\ name_ MAT-file**

ასეთ ფაილებს აქვთ mat გაფართოება და მათ უწოდებენ **'MAT- ფაილებს'.**

შენახული ცვლადების მნიშვნელობების აღსადგენად ვიყენებთ **load** ბრძანებას: **load_file\ name_ MAT-file.**

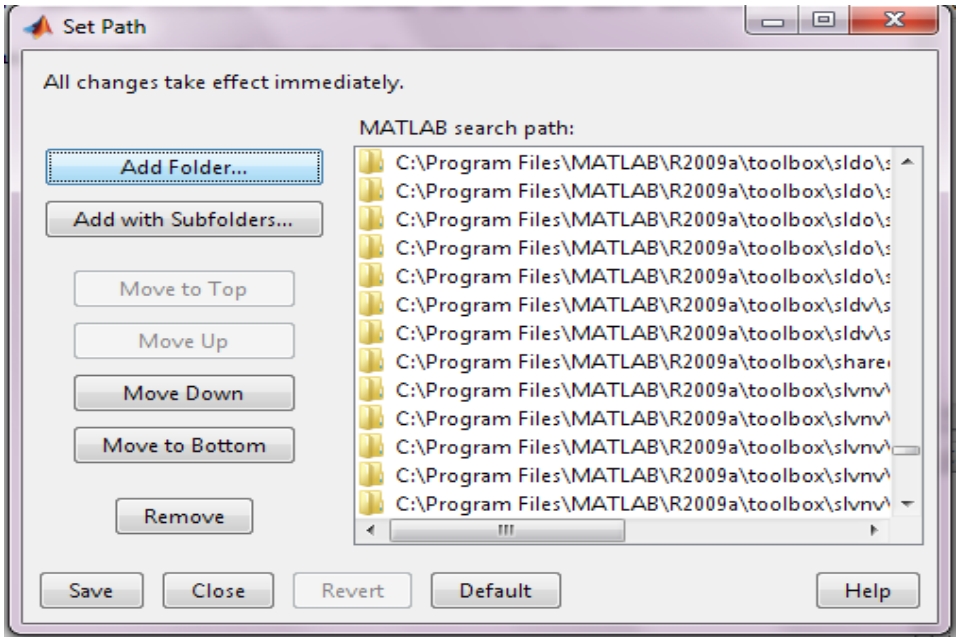
ფანჯარა Command History



Command History - ბრძანებათა ისტორია, ამ ფანჯარაში ინახება ყველა ის ინფორმაცია რასაც **Command Window** ფანჯარაში შევიტანთ, ჩვენ შეგიძლია დავათვალიეროთ ყველა ის ბრძანება რაც შევიტანეთ, ამოვირჩიოთ რომელიმე მათგანი და მასზე ორჯერ დაწკაპუნებით შევასრულოთ **Command Window** ფანჯარაში. ასევე შეგვიძლია მოვნიშნოთ ბრძანებები **Command History** ფანჯარაში და მოვახდინოთ მათი კოპირება **Command Window** ფანჯარაში, ან გამოვთვალოთ F9 ღილაკზე დაჭერით ან შევინახოთ **M**-ფაილად (კონტექსტური მენიუდან შევასრულებთ (Create M-File ბრძანებას).

ფანჯარა Curent Directory

Curent Directory- მიმდინარე დირექტორია, MATLAB მუშაობს მხოლოდ თავის მიმდინარე დირექტორიაში განთავსებულ ფაილებთან ან ეძებს მათ წინასწარ განსაზღვრულ საძიებო არეში, ე.წ. **M**-ფაილები ავტომატურად განთავსდებიან **MATLAB** დირექტორიაში. ფაილების საძიებო ფანჯარას გამოვიტანთ ბრძანებით: File →Set Path...



Add Folder დაამატებს ახალ დირექტორიას.

ფანჯარა Details

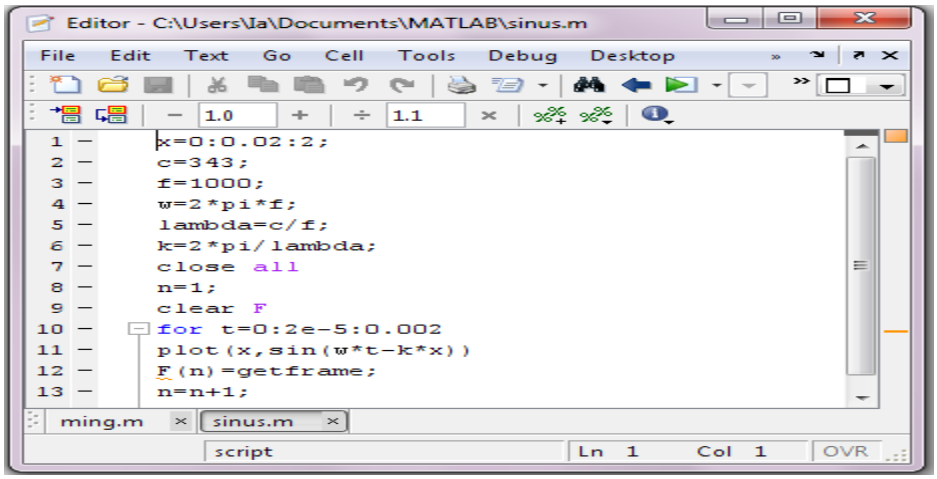
Details – დეტალები, ამ ფანჯარაში გამოიტანება ის პირველი კომენტარი, რომლიც **Curent Directory**-აში მონიშნულ ფაილშია ჩაწერილი.

ფანჯარა Help

Help – დახმარება, ამ ფანჯარაში შესაძლებელია საცნობარო მასალის ნახვა.

ფანჯარა Editor

Editor – პროგრამების რედაქტორი, MATLAB- ის პროგრამული კოდი (ე. წ. **M-**ფაილები) შეიძლება შეიქმნეს ნებისმიერ ტექსტურ რედაქტორში, მაგ (Word-ში) მაგრამ უმჯობესია თუ ამ მიზნისთვის გამოვიყენებთ **MATLAB-** ის **Editor –** რედაქტორს.



იტერაქტიური გამოთვლები MATLAB – ში

როგორც ზემოთ აღვნიშნეთ MATLAB – ის Command Window - ბრძანებათა ფანჯარაში იტერაქტიურ რეჟიმში შესაძლებელია ბრძანებების შესრულება, გარკვეული მიზნით ჩვენ შეგვიძლია შევასრულოთ ცალკეული ბრძანებები ან ბრძანებათა მიმდევრობა –ანუ პროგრამული კოდის ბრძანებების რიგი თანმიმდევრობით. კერძოდ შეგვიძლია გამოვითვალოთ არითმეტიკული გამოსახულებები. მაგალითად ვიპოვოთ შემდეგი გამოსახულების მნიშვნელობა:

$$\frac{(7-6,35):6,5+9,9}{(1,2:36+1,2:0,25-1\frac{5}{16}):\frac{169}{24}}$$

MATLAB – ის Command Window – ში ავკრიფოთ შემდეგი ამოსახულება:

```
>> ((7-6.35)/6.5+9.9)/((1.2/36+1.2/0.25-(1+5/16))/(169/24))
```

და ვიმოქმედოთ **ENTER** ღილაკზე, მივიღებთ:

```
ans =
  20.000000000000004
```

ans ცვლადი MATLAB – ის რეზერვირებული ცვლადია და თუ გამოთვლების დროს საშედეგო სიდიდისათვის არ გვაქვს მინიჭების ოპერატორი ჩაწერილი, მაშინ მიღებულ შედეგს MATLAB – თვითონ მიანიჭებს ans ცვლადს. ანუ თუ ჩავწერდით:

```
>> a=((7-6.35)/6.5+9.9)/((1.2/36+1.2/0.25-(1+5/16))/(169/24))
```

მაშინ მივიღებდით:

```
a =
  20.000000000000004
```

ანუ პირველ შემთხვევაშიც შესრულდა მინიჭების ოპერაცია, მაგრამ რადგან ჩაწერილი გვქონდა მხოლოდ გამოსახულება, ამიტომ შედეგი მიანიჭა ans ცვლადს. ans ცვლადის მნიშვნელობა MATLAB – თან სეანსის განმავლობაში შეიძლება ბევრჯერ შეიცვალოს, ყველა ეტაპზე ის უდრის ბოლოს მინიჭებულ სიდიდეს.

თუ ბრძანებათა სტრიქონში გამოსათვლელი გამოსახულების ბოლოს ჩავწერთ ";"-ს, მაშინ MATLAB – ი გამოთვლებს შეასრულებს ჩვეულებრივი წესით, მხოლოდ შედეგის დაბეჭდვა ფანჯარაში აღარ მოხდება. მაგალითად, შეადარეთ:

```
>> a=12;
>> b=-5
b =
   -5
```

რომელიმე ცვლადის მნიშვნელობის ნახვა შესაძლებელია ან workspace-ში ან ცვლადის სახელის აკრეფით ბრძანებათა სტრიქონში, ჩვენს შემთხვევაში:

```
>> a
a =
   12
```

მაგალითები:

გამოთვალეთ შემდეგი გამოსახულების მნიშვნელობა:

1) $\left(\left(\frac{7}{9}-\frac{47}{72}\right):1,25+\left(\frac{6}{7}-\frac{17}{28}\right):(0,358-0,108)\right)\cdot 1,6-\frac{19}{25}$ (1)

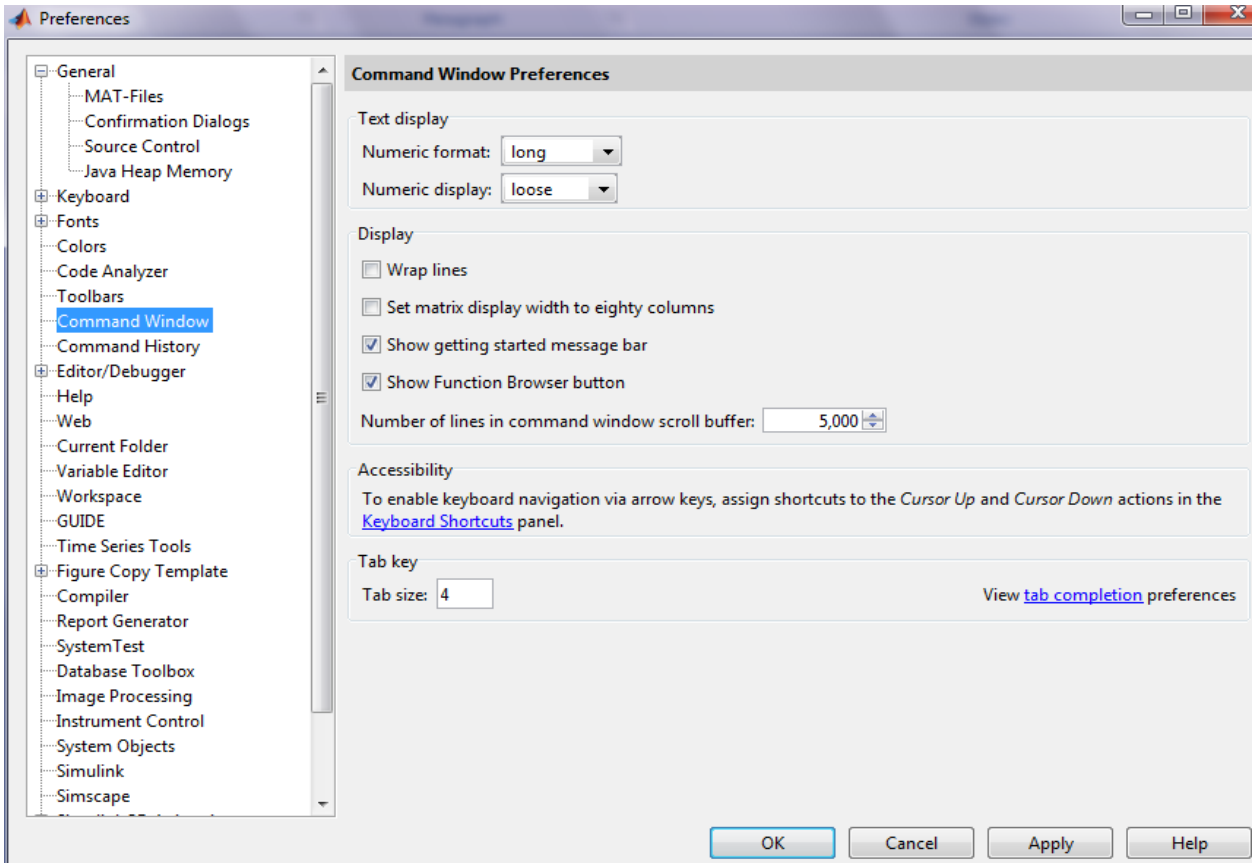
2) $\left(\frac{(2,7-0,8)\cdot 2\frac{1}{3}}{(5,2-1,4):\frac{3}{70})}+0,125\right):2\frac{1}{2}+0,43$ (0,5)

3) $\frac{\left(5\frac{4}{45}-4\frac{1}{6}\right):5\frac{8}{15}}{\left(4\frac{2}{3}+0,75\right)\cdot 3\frac{9}{13}}\cdot 34\frac{2}{7}+\frac{0,3:0,01}{70}+\frac{2}{7}$ (1)

რიცხვითი ფორმატები:

File მენიუში ავირჩიოთ Preferences გამოვა ფანჯარა:

გავააქტიუროთ Command Window – პუნქტი, Text Display განყოფილებაში მოცემულია ფორმატების ასარჩევი სია:



გაჩუმებით Numeric Format-ში წერია long, Numeric display-ში წერია loose.

```
>> 1/2+1/7
ans =
  0.642857142857143
```

ფორმატის შეცვლა შესაძლებელია ბრძანებათა სტრიქონიდანაც, ამისათვის შეგვიძლია ჩავწეროთ მაგალითად:

```
>> format short
>> 1/2+1/7
ans =
   0.6429
```

რიცხვები MATLAB – ში შეგვიძლია ავიღოთ დიაპაზონში:

```
 >> realmin
ans =
 2.2251e-308
>> realmax
ans =
 1.7977e+308
```

MATLAB – ს გააჩნია სპეციალური ცვლადები inf- რომელიც აღნიშნავს ∞-ს, NaN სპეციალური ცვლადია, რომელიც აღნიშნავს 0/0 ან inf/inf.

```
>> 2*realmax
```

```
ans =
  Inf
>> 0/0
ans =
  NaN
>> inf/inf
ans =
  NaN
>> -2*realmax
ans =
 -Inf
```

ლექცია 2

MATLAB – ს გააჩნია სპეციალური პაკეტი Simbolic Mathematics Toolbox

შესაძლებელია მოვახდინოთ სიმბოლური გამოთვლები. იმისათვის, რომ დავრწმუნდეთ რომ დაყენებული გაქვთ კომპიუტერში Simbolic Mathematics Toolbox , ბრძანების სტრიქონში ჩავწეროთ:

```
>>help symbolic
```

გამოიტანს იმ ფუნქციების ჩამონათვალს რომელიც ამ პაკეტს გააჩნია.

სიმბოლური გამოთვლებისათვის ჯერ ცვლადები უნდა გამოვაცხადოთ როგორც სიმბოლურები syms ბრძანებით. მაგ.

```
>> syms x y
>> 9*x^2-(3*x+y)^2
 ans =
 9*x^2 - (3*x + y)^2
```

იმისათვის რომ გავამარტივოთ მიღებული გამოსახულება უნდა გამოვიყენოთ ფუნქცია simple:

```
>>simple (ans)
```

simple განახორციელებს გარდაქმნებს, რომელსაც გვიჩვენებს Command Window – ში და შედეგად მივიღებთ:

```
-y*(6*x + y)
```

ანალოგიურად შეგვიძლია გამოვიყენოთ ფუნქცია simplify, მაგ:

```
>> syms x
>> simplify(x^2+7*x+12)
 ans =
 (x + 3)*(x + 4)
```

MATLAB -ს გააჩნია აგრეთვე ბრძანება **simple** რომელიც საზოგადოდ უკეთესად ასრულებს გარდაქმნებს ვიდრე **simplify** . მაგალითად გავამარტივოთ გამოსახულება: sin(x)*cos(y)+sin(y)*cos(x)

```
>> syms x y
>>simple(sin(x)*cos(y)+sin(y)*cos(x))
 simplify:
 sin(x + y)
 radsimp:
 cos(x)*sin(y) + cos(y)*sin(x)
simplify(100):
 sin(x + y)
 combine(sincos):
 sin(x + y)
 combine(sinhcosh):
 cos(x)*sin(y) + cos(y)*sin(x)
 combine(ln):
 cos(x)*sin(y) + cos(y)*sin(x)
 factor:
 cos(x)*sin(y) + cos(y)*sin(x)
 expand:
 cos(x)*sin(y) + cos(y)*sin(x)
 combine:
 cos(x)*sin(y) + cos(y)*sin(x)
 rewrite(exp):
 ((1/exp(x*i))/2 + exp(x*i)/2)*(((1/exp(y*i))*i)/2 - (exp(y*i)*i)/2) + (((1/exp(x*i))*i)/2 - (exp(x*i)*i)/2)*((1/exp(y*i))/2 + exp(y*i)/2)
 rewrite(sincos):
 cos(x)*sin(y) + cos(y)*sin(x)
 rewrite(sinhcosh):
 cosh(-x*i)*sinh(-y*i)*i + cosh(-y*i)*sinh(-x*i)*i
 rewrite(tan):
 - (2*tan(x/2)*(tan(y/2)^2 - 1))/((tan(x/2)^2 + 1)*(tan(y/2)^2 + 1)) - (2*tan(y/2)*(tan(x/2)^2 - 1))/((tan(x/2)^2 + 1)*(tan(y/2)^2 + 1))
 mwcos2sin:
 - sin(x)*(2*sin(y/2)^2 - 1) - sin(y)*(2*sin(x/2)^2 - 1)
 collect(x):
 cos(x)*sin(y) + cos(y)*sin(x)
ans =
 sin(x + y)
```

ის რაც მუქი შრიფტითაა, MATLAB-ს “ეკუთვნის”, ახლა ვნახოთ იგივე simplify-ით

```
>>simplify(sin(x)*cos(y)+sin(y)*cos(x))
```

ans =

sin(x)*cos(y)+sin(y)*cos(x)

collect ბრძანება ხდება ერთი და იგივე ხარისხის მქონე ცვლადებთან მდგომი კოეფიციენტების შეკრება. მაგალითები:

გავამარტივეთ გამოსახულება: $a\log x - x\log x - x$

```
>> syms a x
>> f=a*log(x)-log(x)*x-x
f =
a*log(x)-log(x)*x-x
>> collect(f,log(x))
ans =
(a-x)*log(x)-x
```

გავამარტივეთ გამოსახულება: x*y+a*x*y+y*x^2-a*y*x^2+x+a*x

```
>> syms x y a
>> p=x*y+a*x*y+y*x^2-a*y*x^2+x+a*x
p =
x*y+a*x*y+y*x^2-a*y*x^2+x+a*x
>> collect(p,x)
ans =
(y-a*y)*x^2+(y+a*y+1+a)*x
>> collect(p,y)
ans =
    (x^2-a*x^2+x+a*x)*y+x+a*x
```

MATLAB -ს ასევე გააჩნია ორი ფუნქცია expand-რომელიც შეასრულებს გამრავლებას და factor-პირიქით ჩაწერს მამრავლების სახით:

```
>> expand((x + 3)*(x + 4))
 ans =
 x^2 + 7*x + 12
```

და პირიქით:

```
>> factor(x^2+7*x+12)
 ans =
 (x + 4)*(x + 3)
```

მოვიყვანოთ ელემენტარულ ფუნქციათა ცხრილი:

ფუნქცია	ფუნქციის ჩაწერა MATLAB -ში
ln x	log(x)
lg x	log10(x)
$\log_2 x$	log2(x)
e^x	exp(x)

|x|	abs(x)
sin x	sin(x)
cos x	cos(x)
tg x	tan(x)
atg x	atan(x)

მაგალითები:

1) $\frac{10}{x^2-25y^2}+\frac{y}{5y^2-xy}-\frac{5}{x^2+5xy}$

2) $\left[\left(\frac{x^2}{y^3}+\frac{1}{x}\right):\left(\frac{x}{y^2}-\frac{1}{y}+\frac{1}{x}\right)\right]:\frac{(x-y)^2+4xy}{1+\frac{y}{x}}$

3) $\left(6a^2+5a-1+\frac{a+4}{a+1}\right):\left(3a-2+\frac{3}{a+1}\right)$

4) $\frac{\frac{1}{\sqrt{x}}+\sqrt{y}}{\sqrt{x}+\left(\sqrt{y}+1\right)^{-1}}-\frac{1}{x\sqrt{y}+x+\sqrt{x}}$

5) $\left(\frac{a^{\frac{1}{2}}+2}{a+2a^{\frac{1}{2}}+1}-\frac{a^{\frac{1}{2}}-2}{a-1}\right)\cdot\frac{a^{\frac{1}{2}}+1}{a^{\frac{1}{2}}}$

6) $\left(\frac{9-4a^{-2}}{3a^{-\frac{1}{2}}+2a^{\frac{-3}{2}}}-\frac{1+a^{-1}-6a^{-2}}{a^{-\frac{1}{2}}+3a^{-\frac{3}{2}}}\right)^4$

7) $W=\left(1+\frac{\ln y}{x+\tan y}\right)^{1+\frac{x+\tan y}{\ln y}}, x=1,5; y=9,82$

8) $W=\frac{1+\sin^{-1}(\cos(2y))}{2^x+3^{-x}}$

9) $B=\frac{2^x+3^{-x}-1}{1+\sin^{-1}\left(\cos 2y\right)}, x=1,5; y=9,82$

10) $A=\sqrt[3]{\frac{x\left(x+1\right)^2\left(1+2x\right)^3}{\ln\left|\cot y\right|}}, x=1,5; y=9,82$

11) $T=\sqrt{1-\frac{\sin y+\sin 2y+\sin 3y}{1+e^x}}, x=1,5; y=9,82$

მნიშვნელობების ჩასმა სიმბოლურ გამოსახულებებში:

ფუნქცია **subs** –ით შესაძლებელია სიმბოლურ გამოსახულებაში შევცვალოთ სიმბოლური ცვლადი სხვა ცვლადით, რიცხვით ან გამოსახულებით**. subs** –სინტაქსი ასეთია:

subs(s)

subs(s,new)

subs(s,old,new)

subs(s) თავისუფალ სიმბოლურ ცვლადებს შეცვლის მათი შესაბამისი რიცხვითი მნიშვნელობებით,

მაგალითად, ავიღოთ სიმბოლური გამოსახულება $s = 2 \cdot x + y^2$

```
>> syms x y
>> s=2*x+y^2
 s =
 y^2 + 2*x
>> subs(s,1)
 ans =
 y^2 + 2
```

აქ თავისუფალი ცვლადი x შეიცვალა მითითებული მნიშვნელობით 1-ით. თავისუფალი ცვლადი შესაძლებელია შევცვალოთ სხვა ცვლადით ან გამოსახულებით:

```
>> syms u v
>> subs(s,x,(u-v)^2)
 ans =
 2*(u - v)^2 + y^2
```

თუ გვსურს შევცვალოთ რამდენიმე ცვლადი ამისათვის ფიგურულ ფრჩხილებში უნდა ჩავწეროთ ცვლადების სია, შემდეგ საჭირო მნიშვნელობების სია. მაგ,

```
>> syms x y
>> s=x^2+y;
>> subs(s,{x,y},{3,1})
ans =  10
```

ან

```
>> subs(s,{x,y},{3,v})
ans = v + 9
```

სტრიქონები Matlab-ში

Matlab-ში სტრიქონები ინტერპრეტირებულია მასივებთან, რომელთა ელემენტებიც სიმბოლოებია.

მაგალითად

```
>> first_name = 'Jonathon';
>> last_name = 'Smith';
>> middle_initial='P';
>> name = [first_name, ' ', middle_initial, '. ', last_name]
```

name =
Jonathon P. Smith
ბოლო სტრიქონში მოცემულია სამი ცვლადის: first_name, middle_initial, last_name და ჰარების გაერთიანება ერთ name ცვლადში.
>> numel(name)
ans =
17
>> first=name(1:8)
first =
Jonathon
>> last=name(13:17)
last =
Smith
რადგან name მასივია, შეგვიძლია ჩავწეროთ შემდეგი მინიწების ოპერატორები:
>> length(name)
ans =
17
>> first=name(1:8)
first =
Jonathon
>> last=name(13:17)
last =
Smith
სტრიქონების ერთ მასივში სტრიქონებად ჩასაწერად, საჭიროა სტრიქონებს ჰქონდეს ერთნაირი ზომა, მაგ:
>> name_matrix=str2mat(first_name,middle_initial,last_name)
name_matrix =
Jonathon
P
Smith
>> size(name_matrix)
ans =
3 8
numel(name_matrix)

ans =
24
ფუნქცია strcat სტრიქონებს მიადგავს ერთმანეთს:
>> name = strcat (first_name, middle_initial, last_name)
name =
JonathonP.Smith
ფუნქცია strvcat სტრიქონებს მიადგავს ერთმანეთს ვერტიკალურად:
>> name = strvcat (first_name, middle_initial, last_name)
name =
Jonathon

P.
Smith
ლოგიკური ფუნქცია strcmp ერთმანეთს შეადარებს სტრიქონებს, თუ ტოლია შედეგი უდრის 1-ს, 0-ს წინააღმდეგ შემთხვევაში:

```
>> s1='ia';
>> s2='nia';
>> strcmp(s1,s2)
ans =
    0
```

ფუნქცია strrep: s1 სტრიქონში ეძებს s2 სტრიქონს და თუ ასეთს შეიცავს, მაშინ შეცვლის s3-ით (თვითონ s1არ იცვლება):

```
>> s1='ananiashvili';
>> s2='anania';
>> s3='marjani';
>> strrep(s1,s2,s3)
ans =
marjanishvili
>> s1
s1 =
ananiashvili
```

ფუნქცია findstr: s1 სტრიქონში ეძებს s2 სტრიქონს და თუ ასეთს შეიცავს, მაშინ დააბრუნებს s2 სტრიქონის პირველი ელემენტის რიგით ნომერს s1 სტრიქონში:

```
>> s1 =’ananiashvili’;
>> s2='shvili';
>> findstr(s1,s2)
ans =
    7
```

ფუნქცია int2str მთელ რიცხვს გადაიყვანს სიმბოლურში დამრგვალებით:

```
>> int2str(523)
ans =
523
>> int2str(523.12)
ans =
523
>> int2str(523.62)
ans =
524
```

ფუნქცია num2str მთელ რიცხვს გადაიყვანს სიმბოლურში დამრგვალების გარეშე:

```
>> num2str(523.62)
ans =
```

523.62

ფუნქცია mat2str რიცხვით მატრიცას გადაიყვანს სიმბოლურში:

```
>> A=[1 2 3; -6 7 11]
A =
    1    2    3
   -6    7   11
>> mat2str(A)
ans =
[1 2 3;-6 7 11]
```

ფუნქცია eval პირიქით სიმბოლურ მატრიცას გადაიყვანს რიცხვითში:

```
>> eval(ans)
ans =
    1    2    3
   -6    7   11
```

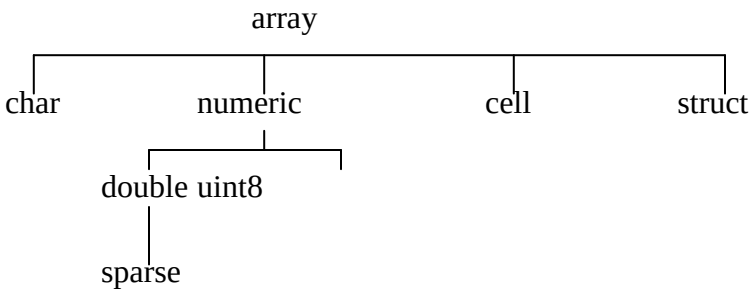
ამოცანები:

1. მოცემულ სტრიქონში დათვალეთ ხმოვნების რაოდენობა და შეადარეთ დანარჩენი სიმბოლოების რაოდენობის 1/3-ს:
2. თუ მოცემულ s სიტყვაში მე-4 პოზიციიდან წერია red შეცვალეთ ბლუე-თი, თუ წერია blue შეცვალეთ red-ით, თუ წერია yelow შეცვალეთ green-ით, ყველა დანარჩენ შემთხვევაში შეცვალეთ wight-ით.
3. მოცემულ სიტყვაში გახსნილი ფრჩხილები შეცვალეთ დახურულით და პირიქით;
4. მოცემულია s სტრიქონი, თუ ამ სტრიქონში მე-5 სიმბოლოდან დაწყებული წერია cat, შეცვალეთ dog-ით, თუ წერია man, შეცვალეთ woman-ით, თუ წერია boy, შეცვალეთ girl-ით. თუ ასეთი სიტყვა არ წერია, დაბეჭდეთ შესაბამისი შეტყობინება;
5. შეამოწმეთ მოცემული რიცხვი პალინდრომია თუ არა (პალინდრომი ჰქვია რიცხვს, თუ ის ერთნაირად იკითხება ჩვეულებრივად და შებრუნებულად. მაგ. 1221);
6. შევადაროთ მოცემულ სიტყვაში გახსნილი და დახურული ფრჩხილების რაოდენობა;
7.

ლექცია 3

ცვლადთა ტიპები

MATLAB სისტემაში განსაზღვრულია მონაცემთა 6 საბაზისო ტიპი, რომელთაგან თითოეული წარმოადგენს მრავალგანზომილებიან მასივებს. ამ მასივების ორგანზომილებიან ვერსიებს უწოდებენ მატრიცებს, მოვიყვანოთ დიაგრამა



უფრო ხშირად ჩვენ გამოვიყენებთ ორმაგი სიზუსტის (double) და სიმბოლური ან სტრიქონული ტიპის (char) მასივებს. ეს დაკავშირებულია იმასთან,რომ MATLAB-ში ყველა გამოთვლები მიმდინარეობს ორმაგი სიზუსტის მასივებთან, ხოლო დანარჩენ ტიპებს აქვთ სპეციფიკური დანიშნულება. დაუშვებელია ნუმერიც და array ტიპის განსაზღვრა, მათი დანიშნულებაა ცვლადების ერთ ჯგუფში გაერთიანება.

არსებობს M-ფაილის ორი ტიპი M-სცენარი და M-ფუნქცია.

M-სცენარი წარმოადგენს M-ფაილის ყველაზე მარტივ ტიპს. მას არ გააჩნია შემავალი და გამომავალი არგუმენტები. M-სცენარები გამოიყენებენ ცვლადებს სამუშაო არიდან და სცენარის დამთავრების შემდეგ ეს მონაცემები ინახება ისევ სამუშაო არეში და შესაძლებელია მათი გამოყენება შემდგომ გამოთვლებში.

მათლაბ სისტემის სამუშაო არე (Workspace) ეს არის მეხსიერების ის არე, რომელიც MATLAB –თან მუშაობის დროს სპეციალურად გამოიყოფა (ოპერატიულ მეხსიერებაში), რომელშიც სეანსის დროს გამოყენებული ცვლადები და მათი მნიშვნელობები ინახება.

მათლაბში ნებისმიერი ცვლადი წარმოადგენს მასივს. შესაძლოა გვქონდეს ცარიელი მასივები, ასევე მასივი ზომით 1×1.

M-ფუნქციები - წარმოადგენენ M-ფაილებს, რომლებისთვისაც დასაშვებია შემავალი და გამომავალი არგუმენტების არსებობა. M-ფუნქციებს გააჩნიათ საკუთარი

მუშა ველი რომელიც განსხვავდება MATLAB-ის მუშა ველისაგან. განვიხილოთ M-ფუნქციის-მარტივი მაგალითი:

```
function f=fact(n)
%fact(n) გამოითვლის ფაქტორიალს
%n-ის ფაქტორიალი გამოითვლება prod ფუნქციით
f=prod(1:n);
end
```

M-ფუნქციის სტრუქტურა:

M-ფუნქცია შედგება:

1) ფუნქციის განსაზვრის სტრიქონისაგან;
2) კომენტარების I-ლი სტრიქონისაგან;
3) კომენტარებისაგან;
4) ფუნქციის ტანისაგან.
5) მთავრდება end-ით

ფუნქციისგანსაზღვრის სტრიქონი MATLAB სისტემას ამცნობს, რომ ეს ფაილი არის M-ფუნქცია და განსაზღვრავს შემავალ და გამომავალ არგუმენტებს. ჩვენს მაგალითში: function f=fact(n), function გასაღები სიტყვაა M-ფუნქციის აღმნიშვნელი:

f-გამომავალი არგუმენტია;

fact- ფუნქციის სახელი;

n-შემავალი არგუმენტი.

შესაძლოა M-ფუნქციას გააჩნდეს რამდენიმე გამომავალი არგუმენტი, ასეთ შემთხვევაში ისინი ჩაიწერებიან კვადრატულ ფრჩხილებში და ერთმანეთისაგან გამოიყოფიან მძიმით. მაგ. function [x,y,z]=sphere(theta,phi,rho)

კომენტარები: თუ ბრძანებათა ველიდან მომხმარებელი მიუთითებს ბრძანებას help <ფუნქციის სახელი>, გამოჩნდება კომენტარების პირველი სტრიქონი, ამიტომ აუცილებლად კარგად უნდა იქნეს მოფიქრებული კომენტარების პირველი სტრიქონი. კომენტარები იწყება %-ის ნიშნით.

ფუნქციის ტანი: ფუნქციის ტანი შეიცავს MATLAB ენის კოდს (ანუ პროგრამას), რომელშიც სრულდება გამოთვლები და გამომავალ არგუმენტებს ენიჭებათ შესაბამისი მნიშვნელობები. ჩვენს მაგალითში ფუნქციის ტანი შედგება ერთი ოპერატორისაგან: f=prod(1:n);

M-ფუნქციის სახელი: MATLAB-ში M-ფუნქციის სახელი, ასევე ცვლადის სახელი არ უნდა აჭარბებდეს 31 სიმბოლოს, უფრო ზუსტად შესაძლოა სიმბოლოების რაოდენობა სახელში 31-ზე მეტიც იყოს, მაგრამ რედაქტორი დანარჩენ სიმბოლოებს მხედველობაში არ მიიღებს. M-ფუნქციის სახელი აუცილებლად უნდა იწყებოდეს ასოთი, შემდეგ კი დასაშვებია ასოებისა და ციფრების კომბინაცია. ამავე დროს ფრთხილად უნდა ვიყოთ როდესაც ვირჩევთ ფუნქციის სახელს, ის არ უნდა იყოს MATLAB –ის რეზერვირებული სიტყვა-ანუ არ უნდა იყოს MATLAB-ში ჩადგმული

რომელიმე მზა ფუნქციის სახელი. ამის გასარკვევად შესაძლებელია MATLAB –ის ბრძანებათა სტრიქონში ჩავწეროთ ბრძანება which <ფუნქციის სახელი>, მაგ:

>> which funq

'funq' not found

ამგვარად funq შეგვიძლია დავარქვათ ჩვენს M-ფუნქციას.

M-ფუნქციის სახელი, რომელიც შეესაბამება M-ფუნქციას, უნდა შედგებოდეს საკუთრივ M-ფუნქციის სახელისაგან და .m გაფართებისაგან. შესაძლებელია M-ფუნქციის გარე სახელი M-ფუნქციის სახელი არ ემთხვევოდეს ერთმანეთს, მაგრამ ეს სასურველი არ არის (ანუ M-ფუნქციის შენახვის დროს File Name –ში არაა რეკომენდებული სხვა სახელის მითითება).

M-ფუნქციის შესრულება

M-ფუნქციია შეიძლება გამოვიძახოთ MATLAB-ის ბრძანებათა სტრიქონიდან ან სხვა M-ფაილიდან, ამ დროს აუცილებელია მივუთითოთ ყველა აუცილებელი ატრიბუტი - შემავალი არგუმენტები მრგვალ ფრჩხილებში და გამომავალი არგუმენტები კვადრატულ ფრჩხილებში. ეს საშუალებას გვაძლევს თავიდან ავიცილოთ განმეორებითი სინტაქსური ანალიზი. ფსევდოკოდი ინახება მეხსიერებაში მანამ არ იქნება გამოყენებული ცლეარ ბრძანება ან არ იქნება დამთავრებული რედაქტორთან მუშაობის სეანსი. ბრძანება clear- ისათვის დასაშვებია შემდეგი მოდიფიკაციები: clear <ფუნქციის სახელი> სამუშაო არიდან წაშლის შესაბამის ფუნქციას;

clear functions წაშლის ყველა კომპილირებულ პროგრამას;

clear all წაშლის პროგრამებსაც და მონაცემებსაც.

შესაძლებელია კომპილირებული M-ფუნქციისა და M-სცენარის შენახვა შემდგომი სეანსებისათვის. ბრძანება ცოდე-ს საშუალებით. მაგ. ცოდე ფაცტ ბრძანება მოახდენს fact.m M-ფუნქციის სინტაქსურ ანალიზს და შეინახავს შესაბამის ფსევდოკოდს fact.p ფაილში მაგრამ რადგანაც არ საჭიროებს. -კოდის შენახვა მიზანშეწონილია ორ შემთხვევაში:

1) როდესაც საჭიროა დიდი რაოდენობით M-ფაილების სინტაქსური ანალიზი, უმეტესად გრაფიკული ინტერფეისის შესაქმნელად;

2) როდესაც მომხმარებელს სურვილი აქვს დამალოს M-ფაილში რეალიზებული ალგორითმი.

რაც შეეხება არგუმენტების გადაცემას, ეს შემდეგი წესით ხდება: თუ არგუმენტის მნიშვნელობა იცვლება ფუნქციის შესრულებისას, მაშინ MATLAB-ში ის მნიშვნელობით გადაეცემა, თუ ფუნქცია არ ცვლის არგუმენტის მნიშვნელობას და მხოლოდ გამოიყენებს მას გამოთვლებისათვის, მაშინ არგუმენტი გადაეცემა მიმართვით, რაც მეხსიერების ოპტიმალური გამოყენების საშუალებას იძლევა.

როგორც უკვე აღვნიშნეთ ყოველი M-ფუნქციისათვისდამატებით გამოიყოფა ფუნქციის მუშა არე. MATLAB-ში მუშაობის დროს შეგვიზლია მივმართოთ მხოლოდ

თვითონ სისტემის მუშა არეში განსაზღვრულ ცვლადებს. თუ ცვლადი გამოცხადებულია გლობალურად მაშინ ის მიეკუთვნება რამდენიმე მუშა არეს.

ლოკალური და გლობალური ცვლადები

MATLAB-ში ცვლადების გამოყენება M-ფაილში არაფრით განსხვავდება ცვლადების გამოყენებისაგან ბრძანებათა სტრიქონში, კერძოდ:

1) არ საჭიროებენ გამოცხადებას, ცვლადს ვერ მიანიჭებ მნიშვნელობას, თუ მარჯვენა მხარეში შემავალ ყველა ცვლადს მინიჭებული არ ექნება მნიშვნელობა.

2) მინიჭების ნებისმიერი ოპერაცია იძლევა ახალ ცვლადს, ან ცვლის უკვე არსებული ცვლადის მნიშვნელობას.

3) ცვლადის სახელი იგივე წესით აირჩევა რა წესითაც ფუნქციის სახელი.

ჩვეულებისამებრ, ფუნქციებს გააჩნიათ საკუთარი ლოკალური ცვლადები, მაგრამ თუ რომელიმე ცვლადს გამოვაცხადებთ გლობალურად რამდენიმე ფუნქციაში, მაშინ ყველა ისინი გამოიყენებენ ამ ცვლადის ერთადერთ ასლს.

გლობალურ ცვლადებთან სამუშაოდ აუცილებელია:

1) ყოველ M-ფუნქციაში, რომელშიც ეს ცვლადი გამოიყენება აუცილებელია, გამოვაცხადოთ ის გლობალურად(global ბრძანებით). იმისათვის, რომ რომელიმე ცვლადი მუშა არიდან იყოს გლობალური, აუცილებელია ის ბრძანებათა სტრიქონში გამოვაცხადოთ გლობალურად.

2) ყოველ ფუნქციაში გამოვიყენოთ გლობალ ბრძანება ცვლადის პირველ გამოყენებამდე.რეკომენდებულია გლობალ ბრძანების გამოყენება -ფაილის დასაწყისში.

გლობალური ცვლადების სახელებს არჩევენ უფრო დიდს,ვიდრე ლოკალურს, ამავე დროს მოხერხებულია რომ ისინი იყოს დიდი ასოებით ჩაწერილი, მაგ: global ALPHA BETA

არგუმენტების რაოდენობის შემოწმება

ფუნქციები nargin და nargout საშუალებას იძლევიან განვსაზღვროთ ფუნქციის შემავალი და გამომავალი არგუმენტების რაოდენობა. ეს ინფორმაცია შესაძლოა გამოვიყენოთ პირობით ოპერატორებში გამოთვლების მიმდინარეობის შესაცვლელად, მოვიყვანოთ მარტივი მაგალითი:

```
function c=testarg(a,b)
if (nargin==1)
    c=a.^2;
elseif (nargin==2)
    c=a+b;
end
```

თუ გვექნება ერთადერთი შემავალი არგუმენტი, ფუნქცია გამოითვლის შემავალი არგუმენტის კვადრატს; ორი არგუმენტის შემთხვევაში კი შესრულდება მათი შეკრების ოპერაცია.

მიმდინარე და უსახელო ფუნქციები

MATLAB -ში შესაძლებელია ჩვენთვის საჭირო მომხმარებლის ფუნქციის განსაზღვრა ბრძანებათა სტრიქონში, ისე რომ ეს ფუნქცია იარსებებს მიმდინარე სეანსის განმავლობაში. მიმდინარე ფუნქცია განისაზღვრება შემდეგნაირად:
INLINE('ფორმულა', 'არგ1', 'არგ2', …) ფორმულა არის სტრიქონი, ამიტომ ის მოთავსებულია აპოსტროფებში. არგუმენტები შეგვიძლია არ მივუთითოთ, მითითების შემთხვევაში არსებითი მნიშვნელობა აქვს მათ რიგითობას. მაგ.

```
>> g = inline('t^2')
g =
     Inline function:
     g(t) = t^2
>> g(3.5)
ans =
  12.250000000000000
```

მაგალითი 2

```
g = inline('sin(2*pi*f + r)', 'f', 'r')
g =
     Inline function:
     g(f,r) = sin(2*pi*f + r)
>> g(1,2)
ans =
   0.909297426825682
```

უსახელო (ანონიმური), ფუნქცია აღიწერება შემდეგნაირად:
ფუნქციის სახელი=@(არგ1, არგ2, …) ფორმულა
აქ ფორმულას და არგუმენტებს ვწერთ აპოსტროფების გარეშე. მაგ.

```
>> g = @(t)t^2
g =
    @(t)t^2
>> g(2)
ans =
     4
```

MATLAB-ის ოპერატორები

MATLAB-ის ოპერატორები იყოფიან სამ კატეგორიად:

1) არითმეტიკული ოპერატორები–რომლებიც უზრუნველყოფენ არითმეტიკული გამოსახულებების კონსტრუირებას და მათ გამოთვლას.

2) მიმართების ოპერატორები–უზრუნველყოფენ რიცხვითი ოპერანდების შედარებას.

3) ლოგიკური ოპერატორები–უზრუნველყოფენ ლოგიკური გამოსახულების აგებას.

არითმეტიკული გამოსახულებების გამოთვლისას მიმართების ოპერატორებს უფრო დაბალი პრიორიტეტი აქვთ, ვიდრე არითმეტიკულს და უფრო მაღალი ვიდრე ლოგიკურს (ე.ი. ჯერ შესრულდება არითმეტიკული, შემდეგ მიმართების, ბოლოს ლოგიკური).

არითმეტიკული ოპერატორები

არითმეტიკული ოპერაციები სრულდება შემდეგი პრიორიტეტის მიხედვით:

1 დონე: ტრანსპორინება ელემენტებით (.'),ელემენტების მიხედვით ახარისხება (.^), შეუღლებული მატრიცის პოვნა ('), მატრიცის ახარისხება.

2 დონე: უნარული შეკრება (+), უნარული გამოკლება (-) (ე.ი. ნიშნის აღება).

3 დონე: მასივების გამრავლება (.*), მარჯვენა გაყოფა (./), მარცხენა გაყოფა(.\), მატრიცების გამრავლება (*), წრფივ განტოლებათა სისტემების ამოხსნა (/) და (\).

4 დონე: შეკრება (+), გამოკლება (-).

5 დონე: მასივის ფორმირების ოპერატორი (:).

ოპერატორები ყოველი დონის შიგნით არიან ტოლძალოვანი, მოქმედებების რიგის შეცვლა შესაძლებელია მრგვალი ფრჩხილების გამოყენებით.

მაგ.: ვთქვათ მოცემულია ორი ვექტორი:

```
A=[3 9 5];
B=[2 1 5];
C=A./B.^2        –ოპერატორის შედეგი იქნება
C=0.7500    9.0000    0.2000
C=(A./B).2       ოპერატორით მიიღება
C=2.2500    81.0000   1.0000
```

არითმეტიკულ ოპერატორებში შესაძლოა გამოვიყენოთ ინდექსიანი გამოსახულებები.

არითმეტიკული გამოსახულებები სრულდება ერთი და იგივე ზომის მასივებისათვის, გამონაკლისია ის შემთხვევა, როდესაც ერთი ოპერანდი არის სკალარული, ასეთ დროს ხდება ე.წ. "სკალარის გაფართოება" რაც ნიშნავს შემდეგს: სკალარი ფართოვდება მეორე ოპერანდის შესაბამისად.

მონაცემთა დიაპაზონის ფორმირება: ორწერტილიანი ოპერატორი :

ბრძანებათა სტრიქონში ავკრიფოთ j= - 3:5 და ვნახოთ შედეგი:

```
>> j=-3:5
j =
   -3    -2    -1     0     1     2     3     4     5
```

როგორც ვხედავთ MATLAB-მა j ვექტორი დააფორმირა მითითებულ საზღვრებში. ამ მაგალითში მნიშვნელობები იზრდებოდა 1-ით. ზოგადად ორწერტილიან ოპერატორს აქვს სახე: k1:h:k2, რაც ნიშნავს რომ დიაპაზონის უმცირესი მნიშვნელობაა k1, უდიდესი - k2 და ცვლილება მოხდება h ბიჯით. შეგვიძლია ბიჯი არ მივუთითოთ როგორც ჩვენს მაგალითში იყო, გაჩუმების პრინციპით ბიჯი უდრის 1-ს. განვიხილოთ მაგ. :

```
>> j=-0.5:0.2:0.5
j =
  -0.5000   -0.3000   -0.1000    0.1000    0.3000    0.5000
```

როგორც ვხედავთ დიაპაზონის საზღვრებიც და ბიჯიც შეიძლება ავიღოთ რეალური რიცხვები. შევნიშნოთ რომ, დიაპაზონი იქნება ცარიელი თუ ჩავწერთ:

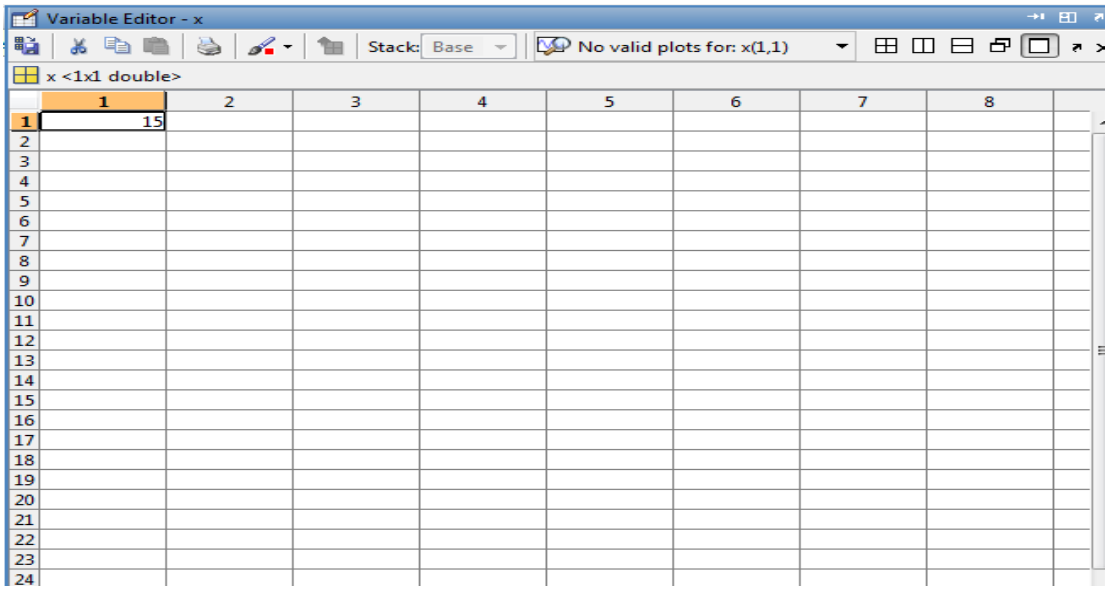
```
>> j=1:-1:5
j =
   Empty matrix: 1-by-0
```

მასივები MATLAB-ში

MATLAB-ში ნებისმიერი სიდიდე წარმოადგენს მასივს. სკალარულ სიდიდეებსაც MATLAB-ი მასივებად აღიქვამს, მაგალითად ჩავწეროთ მინიჭების ოპერატორი:

```
>> x=15
x =
    15
>> whos x
  Name      Size            Bytes  Class     Attributes
  x         1x1                 8  double
```

ანუ Size გვიჩვენებს რომ x არის 1x1ზომის მასივი. თუ workspace–ში x-ზე ორჯერ დავაწკაპუნებთ გაიხსნება variable editor-x ფანჯარა:



აქ შესაძლებელია მონაცემების რედაქტირება.

ზოგადად მასივის დასაფორმირებლად MATLAB-ში გამოიყენება [ ] ფრჩხილები. ელემენტებს შორის იწერება მძიმე ან ჰარი:

```
>> a=[1 2 -6 0.5]
a =
   1.0000    2.0000   -6.0000    0.5000
```

ან

```
>> a=[1, 2, -6, 0.5]
a =
   1.0000    2.0000   -6.0000    0.5000
```

ამგვარად მივიღეთ ვექტორ-სტრიქონი. ვექტორ-სვეტის მისაღებად უნდა ჩავწეროთ ";"

```
>> a=[1; 2; -6; 0.5]
a =
   1.0000
   2.0000
  -6.0000
   0.5000
```

ვექტორის ფორმირებისათვის კონკრეტულ სიტუაციებში შეგვიძლია გამოვიყენოთ ":" ოპერატორი, მაგ.:

```
>> b=-1:0.2:0
b =
  -1.0000   -0.8000   -0.6000   -0.4000   -0.2000         0
```

ანალოგიურად ფორმირდება ორგანზომილებიანი მასივები ანუ მატრიცები. მატრიცის ყველა სტრიქონი წარმოადგენს სტრიქონ-ვექტორს, ამიტომ მატრიცის სტრიქონის ბოლოს იწერება ";". მაგ.

```
>> A=[1 0 -5; 3 9 6]
```

```
A =
    1    0   -5
    3    9    6
```

MATLAB-ს გააჩნია მრავალგანზომილებიანი მასივებიც, მათ განვიხილავთ მოგვიანებით.

მასივის ელემენტების რაოდენობის და ზომების განსაზღვრის ფუნქციები:

size-აბრუნებს ელემენტების რაოდენობას განზომილებების შესაბამისად;
numel –აბრუნებს მასივის ელემენტების რაოდენობას;
length -აბრუნებს განზომილებებს შორის მაქსიმალურს;
ndims –ფუნქციით განვსაზღვრავთ რამდენ განზომილებიანია მასივი;

განვიხილოთ მაგალითი:

```
>> a=[1 2 3; 4 5 6]
a =
    1    2    3
    4    5    6
>> size(a)
ans =    2    3
>> numel(a)
ans =    6
>> ndims(a)
ans =  2
>> length(a)
ans =   3
```

ცარიელი მასივები

MATLAB-ში დასაშვებია ცარიელი მასივები, ეს მასივები აღინიშნება [ ], მისი ზომაა 0×0, ასევე დასაშვებია 10×0×20-ზომის მასივები, ანუ როდესაც ყველა განზომილება არ არის 0-ის ტოლი. შესაძლებელია ასევე ცარიელი მასივის განსაზღვრა ზეროს ფუნქციით. მაგ. E=zeros(0,5), ნიშნავს (0,5)-ს.

ძირითადი დანიშნულება ცარიელი მასივისა ის არის, რომ ნებისმიერი ოპერაცია, რომელიც განსაზღვრულია m×n, ზომის მასივისათვის განსაზღვრავდეს სწორ შედეგს, მაშინ როდესაც m ან n უდრის ნულს.

ინდექსები და ქვეინდექსები

ვექტორის ელემენტებზე წვდომა ხდება ინდექსის საშუალებით, ინდექსი მიეთითება მრგვალ ფრჩხილებში. ამასთან პირველი ელემენტის ინდექსი უდრის 1-ს, ყოველი მომდევნოს ნომერი იზრდება 1-ით. მაგ.

```
>> a=[-1, 2, -4, 0]
>> a(1)
ans =  -1
>> a(3)
ans =  -4
>> a(5)
```

??? Index exceeds matrix dimensions.

რადგან a ვექტორი მხოლოდ 4 ელემენტს შეიცავს, ბოლო ბრძანებაში შეცდომაც აქედან გამომდინარეა. მაგრამ თუ ჩავწერდით:

```
>> a(5)=12
a =   -1    2   -4    0   12
```

ამ შემთხვევაში მასივის ზომა გაიზარდა და შესაბამისად მე-5 ელემენტის მნიშვნელობა გახდა 12. ამ ოპერაციას ჰქვია მასივის გაფართოება. მასივის გაფართოება შეგვიძლია მოვახდინოთ ასევე Variable Editor-a ფანჯრიდან.

შესაძლებელია მასივის ელემენტების გადალაგება ინდექსის რიგის მითითებით, მაგ:

```
>> u=[1 0 2 3 0 5];
>> k=[6 1 4 2 5 3];
>> u(k)
ans =
    5    1    3    0    0    2
```

ვექტორის ელემენტების ჯამი შეგვიძლია ვიპოვოთ sum ფუნქციით:

```
>> a=[-1, 2, -4, 0]
a =
   -1    2   -4    0
>> sum(a)
ans = -3
```

მაქსიმუმისა და მინიმუმის გამოსათვლელად გამოვიყენებთ min და max ფუნქციებს:

```
>> min(a)
ans = -4
>> max(a)
ans = 2
```

შეგვიძლია ვიპოვოთ მაქსიმალური ელემენტი და მისი ნომერი ერთდროულად:

```
>> [m,k]=max([1 8 9 11 -5 7])
m =11
k = 4
```

ნამრავლს გამოვითვლით prod ფუნქციით:

```
>> a=[1 2 -5.2]
a =
   1.0000    2.0000   -5.2000
>> prod(a)
ans =  -10.4000
```

ფუნქცია sort(a) დაალაგებს a მასივის ელემენტებს ზრდადობით:

```
>> a=[6 8 -9 12 4 7];
>> sort(a)
ans =    -9     4     6     7     8    12
```

შეგვიძლია დავალაგოთ და მივიღოთ შესაბამისი ინდექსების მასივიც:

```
[aa,ind]=sort(a)
aa =    -9     4     6     7     8    12
ind =     3     5     1     6     2     4
```

შევნიშნოთ რომ –sort(-a) დაალაგებს a მასივის ელემენტებს კლებადობით.

ფუნქცია mean(a) გამოთვლის a მასივის ელემენტების საშუალო არითმეტიკულს. შესაძლებელია მასივსა და სკალარზე არითმეტიკული ბრძანებების შესრულდება მასივის ელემენტების მიხედვით:

```
>> a=[1 2 -5 0];
>> a+3
ans =   4     5    -2     3
```

ორი მასივის შეკრება და გამოკლება მოხდება ასევე ელემენტების მიხედვით. იმისათვის რომ გავყოთ, გავამრავლოთ და ავახარისხოთ მასივები შესაბამის ბრძანებას წინ უნდა დავუწეროთ წერტილი, ამასთან არითმეტიკული მოქმედებები დასაშვებია მხოლოდ ერთნაირი ზომის მასივებზე:

```
>> u=[4 -9 12 0 8];
>> v=[2 3 7 0 0];
>> u./v
ans =
   2.0000   -3.0000    1.7143       NaN       Inf
```

A მატრიცის ელემენტი i-ური სტრიქონის და j-ური სვეტის გადაკვეთაზე აღინიშნება, როგორც A(i,j). თუ მატრიცის ზომა არის 4×4, მაშინ თუ მივმართავთ ელემენტს A მატრიცის გარეთ, მაგ.: t=A(4,5), დაფიქსირდება შეცდომა. მაგრამ თუ ჩავწერთ:

```
X=A
X(4,5)=17
```

მაშინ MATLAB-ი ავტომატურად გააფართოებს მატრიცას, დარჩენილ უჯრებს მიანიჭებს მნიშვნელობას 0. განვიხილოთ საილუსტრაციო მაგალითი:

```
>> M=magic(4)
```

```
M =
   16    2    3   13
    5   11   10    8
    9    7    6   12
    4   14   15    1
>> M(4,5)=-3
M =
   16    2    3   13    0
    5   11   10    8    0
    9    7    6   12    0
    4   14   15    1   -3
```

ვთქვათ გვაქვს:

```
>> A=magic(4)
A =
   16    2    3   13
    5   11   10    8
    9    7    6   12
    4   14   15    1
```

A მატრიცის ელემენტებს შეიძლება მივმართოთ ერთი ინდექსითაც. მაგ.: A(12) ეს იგივეა რაც A(4,3):

```
>> A(2)
ans =  5
>> A(12)
ans =  15
```

ინდექსის ცვლილება შეიძლება : -ით ჩავწეროთ, მაგ.: A(1:4,3)- შედეგი იქნება A-ს მესამე სვეტი, ეს შეიძლებოდა ასეც ჩაგვეწერა A(:,3). თუ გვინდა ვიპოვოთ მესამე სვეტის ჯამი შეიძლება ჩავწეროთ sum(A(1:4,3)) ან sum(A(:,3)). sum(A(:,end))- შეაჯამებს ბოლო სვეტს. მაგრამ თუ ჩავწერთ

```
>> sum(A)
ans =
   34   34   34   34
```

ამ შემთხვევაში მივიღებთ ვექტორს, რომელშიც წერია ჯამები სვეტების მიხედვით. ანალოგიურად max, min და prod ფუნქციები გამოითვლიან შესაბამისად სვეტების მიხედვით. ასევე sort(A) დაალაგებს სვეტებს.

ფუნქცია rand(n) შემთხვევითი რიცხვებისაგან აფორმირებს მასივს ზომით $n \times n$ ნორმალიზებული ფორმით ((0,1)ინტერვალი). A=rand(m,n) შემთხვევითი რიცხვებისაგან აფორმირებს A მასივს ზომით $n \times n$. მაგ.:

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

```
    0.8147    0.1270    0.6324
    0.9058    0.9134    0.0975
```

rand-ის ანალოგიურია ფუნქცია randn შემთხვევითი რიცხვების სტანდარტული ნორმალიზებული ფორმით მიიღება ((-10,10)ინტერვალი).

მთელი რიცხვების მისაღებად შეგვიძლია გამოვიყენოთ round, fix, floor ფუნქციები. მაგ.:

```
>> B=round(randn(3,4))
B =
     0     0    -1    -2
    -1     1     0     0
    -1     3     0    -2
```

მატრიცები შეიძლება მივადგათ ერთმანეთს. მაგ.: თუ A=magic(4), შეიძლება ჩავწეროთ: B=[A A+32; A+48 A+16], B იქნება მატრიცი ზომით 8 ×8. კვადრატული ფრჩხილებით შეგვიძლია ამოვიღოთ მატრიცის სვეტი ან სტრიქონი. მაგ.: თუ

```
A=magic(4);
X=A;
X(:,2)=[ ]
```

შედეგი იქნება:

```
X= 16    3   13
     5   10   8
     9    6   12
     4   15   1
```

ელემენტარული მატრიცების ფორმირება:

ones-ერთიანებისაგან შედგენილი მატრიცი:

```
>> ones(4)
ans =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

eye -ერთეულოვანი მატრიცი:

```
>> I=eye(3)
I =
     1     0     0
     0     1     0
     0     0     1
```

magic - მაგიური კვადრატის ფორმირება:

```
>> A=magic(3)
A =
     8     1     6
     3     5     7
```

```
   4    9    2
```

diag - მატრიცის დიაგონალის აღება:

```
>> diag(A)
ans =
   8
   5
   2
```

triu – ზედა სამკუთხა მატრიცისფორმირება:

```
>> triu(A)
ans =
   8    1    6
   0    5    7
   0    0    2
```

tril – ქვედა სამკუთხა მატრიცის ფორმირება:

```
>> tril(A)
ans =
   8    0    0
   3    5    0
   4    9    2
```

ამოცანები

1. მოცემულია X ვექტორი, შეადგინეთ Y ვექტორი რომელშიც X–ის ელემენტები ჩაწერილია საპირისპირო რიგით;
2. მოცემულია X ვექტორი, X–ის კენტნომრიანი ელემენტებისაგან შეადგინეთ Y ვექტორი;
3. მოცემულია X ვექტორი, იპოვეთ X–ის კენტნომრიანი ელემენტების ჯამი;
4. მოცემულია X ვექტორი, იპოვეთ X–ის ელემენტების საშუალო არითმეტიკული;
5. მოცემულია X ვექტორი, იპოვეთ X–ის ელემენტების საშუალო გეომეტრიული;
6. მოცემულია X ვექტორი, იპოვეთ X–ის პირველი და ბოლო ელემენტების ჯამი;
7. მოცემულია X ვექტორი, დაალაგეთ X ვექტორის ელემენტები ზრდის რიგით;
8. მოცემულია X ვექტორი, დაალაგეთ X ვექტორის ელემენტები კლების რიგით;
9. მოცემულია A(3 ×3) მასივი, ჩაწერეთ A მასივის პირველი სვეტი X ვექტორში;
10. მოცემულია A(3 ×4) მასივი, დაალაგეთ A მასივის სვეტები მათი ჯამების ზრდის მიხედვით;
11. მოცემულია A(3 ×4) მასივი, დაალაგეთ A მასივის სვეტები მათი ჯამების კლების მიხედვით;
12. მოცემულია A(3 ×4) მასივი, შეადგინეთ B(4 ×3) მასივი, რომელის სვეტებად A მასივის სტრიქონებია ჩაწერილი. დაალაგეთ B მასივის სვეტები მათი ჯამების ზრდის მიხედვით;

ლექცია 4

მიმართების ოპერატორები

MATLAB სისტემაში განსაზღვრულია 6 სახის მიმართების ოპერაცია: < ნაკლები, <= ნაკლები ან ტოლი, > მეტი, >= მეტი ან ტოლი, == იგივურად ტოლი, ~= არ უდრის.

სკალრულ სიდიდეებზე მიმართების ოპერაციები სრულდება ჩვეულებრივი წესით, თუ მიმართება ჭეშმარიტია მაშინ შედეგი არის 1, წინააღმდეგ შემთხვევაში 0.

მაგ.

```
>> 2>5
ans =
    0
>> a=12; b=-2;
>> a*4<=b^6
ans =
    1
```

მასივების შემთხვევაში მიმართების ოპერაციები ასრულებენ ორი ერთნაირგანზომილებიანი მასივის ელემენტების შედარებას შესაბამისი ელემენტების მიხედვით. გამონაკლისია მხოლოდ ის შემთხვევა, როდესაც ერთი ოპერანდი არის სკალარი, მაშინ სკალარი ედარება მასივის თითოეულ ელემენტს. იმ პოზიციებში, სადაც მიმართება არის ჭეშმარიტი მიიღება მნიშვნელობა 1, სადაც მცდარია-0.

მაგ. შევადაროთ A და B

```
A=[2 7 6; 9 0 –1; 3 0.5 6];
B=[8 0.2 0; -3 2 5; 4 –1 7];
A<B
ans
    1 0 0
    0 1 1
    1 0 1
```

თუ გვინდა რომ A მასივი განვსაზღვროთ როგორც ცარიელი, უნდა ჩავწეროთ: A==[ ], თუ A-ს ზომაა 0×0 ან 1×1. თუ გამოვიყენებთ ფუნქცია isempty(A), შედეგი იქნება 1, თუ A ცარიელი მასივია და იქნება 0 სხვა შემთხვევაში.

ლოგიკური ოპერატორები

MATLAB-ის ლოგიკური ოპერატორებია:

&& (AND) და

|| (OR) ან

~ (NOT) არა

გარდა ამისა bitfun კატალოგში არის ლოგიკური ფუნქციების ჩამონათვალი, რომლებიც ასრულებენ თანრიგობრივ ლოგიკურ ოპერაციებს.

ლოგიკური ოპერაციები ახდენენ ერთი და იგივე ზომის მასივების შედარებებს ელემენტების მიხედვით. ასევეა ვექტორებისა და მატრიცებისათვის, გარდა იმ შემთხვევისა, როდესაც ერთი ოპერანდი არის სკალარი, მაშინ აქაც სკალარი ედარება მასივის ყველა ელემენტს, იმ პოზიციებში სადაც ეს დამოკიდებულება ჭეშმარიტია მიიღება 1, სადაც მცდარი-0;

ყოველ ოპერატორს შეესაბამება გარკვეული პირობები ლოგიკური გამოსახულების ჭეშმარიტების დასადგენად.

ლოგიკური გამოსახულება & (AND) ოპერატორით ჭეშმარიტია,თუ ორივე ოპერანდი ჭეშმარიტია. თუ ლოგიკური გამოსახულების ელემენტები რიცხვებია, მაშინ გამოსახულება ჭეშმარიტია თუ, ორივე ოპერანდი განსხვავდება ნულისაგან. მაგ.

```
>>u=[1 0 2 3 0 5]
>>v=[5 6 1 0 0 7]
>>u&v
ans =
   1    0    1    0    0    1
```

ან

```
and(u,v)
ans =
   1    0    1    0    0    1
```

ლოგიკური გამოსახულება OR(|) ოპერატორით არის ჭეშმარიტი, თუ ერთ-ერთი ოპერანდებიდან ან ორივე ლოგიკურად ჭეშმარიტია. გამოსახულება ლოგიკურად მცდარია, მხოლოდ მაშინ, თუ ორივე ოპერანდი მცდარია. თუ ლოგიკური გამოსახულების ელემენტები არის რიცხვები, მაშინ გამოსახულება მცდარია თუ ორივე ოპერანდი უდრის ნულს. იგივე u და v ვექტორისათვის:

```
>>u|v
ans
  [1 1 1 1 0 1]
```

ლოგიკური გამოსახულება NOT(~) ოპერატორით აგებს უარყოფას. შედეგი ლოგიკურად მცდარია, თუ ოპერანდი ჭეშმარიტია და ჭეშმარიტია თუ ოპერანდი მცდარია. თუ ლოგიკური გამოსახულების ელამენტები რიცხვებია, მაშინ ნებისმიერი

ოპერანდი რომელიც განსხვავდება ნულისაგან გადაიქცევა ნულად და პირიქით ნულოვანი ოპერანდი გადაიქცევა 1-იანად. მაგ.

```
>>~u
ans
    [0 1 0 0 1 0]
```

ლოგიკური ფუნქციები

გარდა ლოგიკური ოპერატორებისა MATLAB სისტემაში ჩართულია ლოგიკური ფუნქციები:

ფუნქცია XOR(a,b) განსაკუთრებული ან; შედეგი ჭეშმარიტია მხოლოდ მაშინ როდესაც ერთი ოპერანდი ჭეშმარიტია, მეორე კი მცდარი. რიცხვითი გამოსახულებისათვის აბრუნებს 1-ს, თუ ერთი ოპერანდი არ უდრის 0-ს, მეორე კი უდრის 1-ს.

მაგ.

```
>a=1;
b=1;
X0R(a,b)
ans
     0
```

ფუნქცია all დააბრუნებს 1-ს, თუ ვექტორის ყველა ელემენტი ჭეშმარიტია ან განსხვავდება 0-ისაგან.

ფუნქცია ani დააბრუნებს 1-ს, თუ ერთი მაინც არგუმენტი არ უდრის 0-ს, წინააღმდეგ შემთხვევაში დააბრუნებს 0-ს.

ფუნქცია isnan და isinf დააბრუნებენ 1-ს, NaN და inf-ისათვის. (ინფ- სპეციალური ცვლადია, რომელიც აღნიშნავს ∞-ს, NaN სპეციალური ცვლადია, რომელიც აღნიშნავს 0/0 ან inf/inf).

ფუნქცია isfinite ჭეშმარიტია მხოლოდ NaN და inf-ისათვის. მაგ.

```
A=[0 1 5; 2 NaN –inf];
B=[0 0 15; 2 5 inf];
C=A./B
C=
   NaN   inf      0,3333
     1      NaN    NaN
isfinite(C)          isnan(c)        isinf(C)
ans=                        ans=                        ans=
          0 0 1                                 1 0 0                          0 1 0
 1 0 0                          0 1 1                      0 0 0
```

ფუნქცია find განსაზღვრავს მასივის იმ ელემენტების ინდექსებს, რომლებიც აკმაყოფილებენ მოცემულ ლოგიკურ პირობას. ყველაზე უფრო გამოყენებადი ფორმაა k=find(<პირობა>) დააბრუნებს ვექტორს იმ ინდექსებით, რომლებიც აკმაყოფილებენ მოცემულ პირობას.

მაგ.: A=magic(4)

```
   A=
16     2   3   13
 5    11  10    8
 9     7   6   12
 4    14  15    1
k=find(A>8);
A(k)=100
A=
   100    2     3   100
     5  100   100     8
   100    7     6   100
     4  100   100   100
```

ფუნქცია [i,j]=find(x) დააბრუნებს მატრიცის არანულოვანი ელემენტების ინდექსებს.

MATLAB –ში ერთ გამოსახულებაში შეგვხვდეს არითმეტიკული და ლოგიკური ოპერაციები, ასევე ლოგიკური ფუნქციები. ამ შემთხვევაში ოპერაციათა პრიორიტეტი შემდეგია: ჯერ შესრულდება უარყოფის ოპერაცია, ლოგიკური ფუნქციები, შემდეგ არითმეტიკული ოპერაციები ზემოთ მოყვანილი პრიორიტეტების შესაბამისად, შემდეგ მიმართების ოპერაციები, ლოგიკური "და", ლოგიკური "ან". ოპერატორები ყოველი დონის შიგნით არიან ტოლძალოვანი, მოქმედებების რიგის შეცვლა შესაძლებელია მრგვალი ფრჩხილების გამოყენებით.

მაგალითები:

1. მოცემულია $A=\begin{pmatrix} 1 & 0 & 3 & 5 \\ -2 & -1 & 0 & 1 \end{pmatrix}$ და $B=\begin{pmatrix} 3 & -7 & 0 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix}$, გამოთვალეთ შემდეგი გამოსახულება ~A&B|A~B^2, ჩამოთვალეთ მოქმედებების შესრულების რიგი.
2. მოცემულია $A=\begin{pmatrix} 1 & 0 & 3 & 5 \\ -2 & -1 & 0 & 1 \end{pmatrix}$ და $B=\begin{pmatrix} 3 & -7 & 0 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix}$, გამოთვალეთ შემდეგი გამოსახულება A&~B|A&~B^2, ჩამოთვალეთ მოქმედებების შესრულების რიგი;
3. მოცემულია $A=\begin{pmatrix} 1 & 0 & 3 & 5 \\ -2 & -1 & 0 & 1 \end{pmatrix}$ და $B=\begin{pmatrix} 3 & -7 & 0 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix}$, გამოთვალეთ შემდეგი გამოსახულება ~A&B|xor(A,B)&~B, ჩამოთვალეთ მოქმედებების შესრულების რიგი;
4. მოცემულია $A=\begin{pmatrix} 1 & 0 & 3 & 5 \\ -2 & -1 & 0 & 1 \end{pmatrix}$ და $B=\begin{pmatrix} 3 & -7 & 0 & 0 \\ 0 & -1 & 0 & 3 \end{pmatrix}$, გამოთვალეთ შემდეგი გამოსახულება A&~xor(A,B)|A&~B, ჩამოთვალეთ მოქმედებების შესრულების რიგი;

5. მოცემულია X ვექტორი, X–ის ის ელემენტები, რომლებიც ნაკლებია საშუალო არითმეტიკულზე შეცვალეთ საშუალო არითმეტიკულისა და ამ ელემენტების ჯამით;

6. მოცემულია X ვექტორი, X–ის საშუალო არითმეტიკულზე ნაკლები ელემენტებით შეადგინეთ ახალი ვექტორი;

7. მოცემულია A(3 ×4) მასივი, იპოვეთ მასივის დადებითი ელემენტების ჯამი;

8. მოცემულია A(3 ×4) მასივი, იპოვეთ მასივის დადებითი და უარყოფითი ელემენტების რაოდენობა.

შეტანა-გამოტანა **MATLAB**-ში

თუ მომხმარებელს სურს ***მონაცემების შეტანა კლავიატურიდან***, მაშინ უნდა გამოიყენოს input ფუნქცია, ფუნქციის არგუმენტია 'prompt ' რიცხვითი სიდიდეების შემთხვევაში.
ავკრიფოთ ბრძანება:

```
>> x=input('prompt ')
prompt
```

ფუნქცია ელოდება რიცხვით სიდიდეს. კლავიატურაზე ავკრიფოთ x-ის მნიშვნელობა, ის შეიძლება იყოს სკალარი ან მატრიცა, შეტანა მოხდება შესაბამისი ფორმატით, ვთქვათ preferences-ში არჩეულია long loose ფორმატი
მაგ:

```
>> x=input('prompt ')
prompt 0.124
x =
  0.124000000000000
>> a=input('shemoitanet masivi a  ')
shemoitanet masivi a  [1 2 3; 4 -5 2]
a =
    1    2    3
    4   -5    2
```

სტრიქონის შეტანა ასევე შესაძლებელია კლავიატურიდან, ამ შემთხვევაში input ფუნქციას ორი არგუმენტი გააჩნია:

```
>> x=input('string prompt ', 's')
string prompt
```

აქ ჩავწერთ სტრიქონს, მაგ:

```
>> g=input('shemoitanet striqoni ', 's')
```

shemoitanet striqoni 'ananiashvili'
g =
'ananiashvili'
მონაცემების ეკრანზე გამოსატანადშეგვიძლია გამოვიყენოთ disp ფუნქცია, ის დაგვიბეჭდავს მითითებული ცვლადის მნიშვნელობას ან სტრიქონს, მაგ:

```
>> disp(pi)
   3.141592653589793
>> disp('This is a string!!!')
This is a string!!!
```

disp ფუნქციის გამოყენების უპირატესობა ისაა რომ ის არ გვიჩვენებს ეკრანზე ცვლადის სახელს.
ეს ფუნქციები შეგვიძლია გამოვიყენოთ M–ფუნქციებშიც და M –სცენარებშიც.

მონაცემთა ფორმატირებულად გამოსატანად ასევე გვაქვს ორი ფუნქცია sprintf და fprint, რომლებიც მნიშვნელობების კონვერტირებას ახდენენ სიმბოლურში, მაგ:

```
>> phi = sprintf('%0.15g',(1+sqrt(5))/2)
phi =
1.61803398874989
>> size(phi)
ans =
        1       16
```

აქედან ჩანს, რომ phi სიმბოლურია,

```
>> sprintf('%0.5g', eps)
ans =
2.2204e-016
>> sprintf('%1.25f', eps)
ans =
0.0000000000000002220446049
>> sprintf('%15.5g', 1/eps)
ans =
4.5036e+015
>> sprintf('%15.5f', 1/eps)
ans =
4503599627370496.00000
```

აქ გამოყენებული ფორმატირების სპეციფიკატორები C-დაპროგრამების ენის სპეციფიკატორების ანალოგიურია.
ქვემოთ მოყვანილია მაგალითი თუ როგორ ჩაიწერება ფორმატირებული ტექსტი data.dat ფაილში:

```
>> x=[1:10];
>> fout=fopen('data.dat','wt');
>> fprintf(fout,' k x(k)\n');
```

```
>> for k=1:length(x)
fprintf(fout,'%4d %5.2f\n',k,x(k));
end
>> fclose(fout);
```

type ბრძანებით შეგვიძლია ვნახოთ ის რაც ჩაწერია data.dat ფაილში:

```
>> type data.dat
  k  x(k)
  1  1.00
  2  2.00
  3  3.00
  4  4.00
  5  5.00
  6  6.00
  7  7.00
  8  8.00
  9  9.00
 10 10.00
```

მონაცემების ფაილში ჩასაწერად შეგვიძლია გამოვიყენოთ dlmwrite ფუნქციარომელსაც აქვს სახე: RESULT = DLMWRITE(ფაილის სახელი) ან RESULT = DLMWRITE (ფაილის სახელი, გამყოფი სიმბოლო)

გამყოფი სიმბოლო შეიძლება იყოს მაგ. ';', თუ გვინდა ჩავწეროთ ტაბულაციის სიმბოლო მაშინ ვწერთ: '\t'. ***მონაცემების ფაილიდან წასაკითხად*** შეგვიძლია გამოვიყენოთ dlmread ფუნქციარომელსაც აქვს სახე: RESULT = DLMREAD(ფაილის სახელი) ან RESULT = DLMREAD(ფაილის სახელი, გამყოფი სიმბოლო)

გამყოფი სიმბოლო შეიძლება იყოს მაგ. ';', თუ გვინდა ჩავწეროთ ტაბულაციის სიმბოლო მაშინ ვწერთ: '\t'. მაგალითად:

```
>> A=magic(3);
dlmwrite('monacemi.txt',A,';')
```

ფაილში ჩაწერილი მონაცემები შეგვიძლია ვნახოთ type ბრძანებით:

```
>> type monacemi.txt
8;1;6
3;5;7
4;9;2
```

ფაილში ჩაწერილი მონაცემების წასაკითხად ვწერთ:

```
>> B=dlmread('monacemi.txt',';')
B =
    8    1    6
    3    5    7
    4    9    2
```

მონაცემების მეხსიერებაში შესანახად ასევე შეგვიძლია გამოვიყენოთ save ბრძანება. მაგ.

```
>> s='Marta Clark';
>> save s
```

ამ მონაცემის ჩასატვირთად ვიყენებთ load ბრძანებას:

```
>> load s
>> s
s =
Marta Clark
```

ლექცია 5

MATLAB-ის ძირითადი ოპერატორები

პირობითი ოპერატორი:

if…else…elseif…end

პირობითი ოპერატორი MATLAB-ში სამი სახით შეიძლება შეგვხვდეს.

სინტაქსი კი ასეთია:

if <ლოგიკური გამოსახულება> <ინსტრუქციები > end

if <ლოგიკური გამოსახულება 1> <ინსტრუქციები 1> else <ინსტრუქცია 2> end

if <ლოგიკური გამოსახულება 1> <ინსტრუქციები 1> elseif
<ლოგიკური გამოსახულება2> <ინსტრუქციები 2> else
<ინსტრუქციები 3> end

პირველი მარტივი ფორმაა და სრულდება შემდეგნაირად: თუ <ლოგიკური გამოსახულება> ჭეშმარიტია, შესრულდება შესაბამისი <ინსტრუქციები> (ნსტრუქციებში იგულისხმებ MATLAB-ის ერთი ან რამოდენიმე ოპერატორი).

მაგ. if a<0 disp (‘a უარყოფითია’) end

მეორეც მარტივი ფორმაა და სრულდება შემდეგნაირად: თუ <ლოგიკური გამოსახულება 1> ჭეშმარიტია, შესრულდება შესაბამისი <ინსტრუქციები 1>, წინააღმდეგ შემთხვევაში შესრულდება else-ს შემდეგ მდგომი <ინსტრუქციები 2>. მაგ.:

```
if a<0 disp (‘a უარყოფითია’) else
                disp (‘a დადებითია’) end
```

მესამე სრულდება შემდეგნაირად: თუ <ლოგიკური გამოსახულება 1> ჭეშმარიტია, შესრულდება შესაბამისი <ინსტრუქციები 1>, წინააღმდეგ შემთხვევაში შემოწმდება <ლოგიკური გამოსახულება 2> თუ ჭეშმარიტია შესრულდება <ინსტრუქციები 2> წინააღმდეგ შემთხვევაში შესრულდება else-ს შემდეგ მდგომი <ინსტრუქციები 3>. მაგ.:

```
if n<0 disp(‘შემოტანილი რიცხვი უნდა იყოს დადებითი’)
    elseif rem(n,2)==0 a=n/2
```

```
else a=(n+1)/2
end
```

შესაძლებებელია ერთ if -ში რამდენიმე elseif გვქონდეს.

ამოცანები

1. შეუცვალეთ x –ს და y-ს ადგილები. დაბეჭდეთ მათ შორის მინიმალური.
2. ამოარჩიეთ x –ს და y-ს შორის მაქსიმალური. თუ რიცხვები ტოლია დაბეჭდეთ ფრაზა “მოცემული რიცხვები ტოლია.
3. მოცემულია მთელი რიცხვი k. გავზარდოთ მისი მნიშვნელობა 10-ით თუ იგი ლუწია და გავზარდოთ 5-ჯერ თუ კენტია.
4. ამოარჩიეთ a, b და c-ს შორის მაქსიმალური.
5. ამოარჩიეთ a, b, c-ს და d-ს შორის მაქსიმალური.
6. დაბეჭდეთ y ცვლადის მნიშვნელობა:

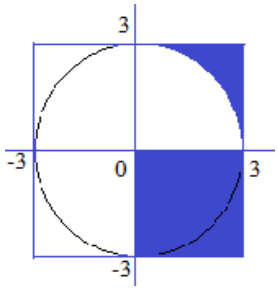
$$y = \begin{cases} x+1, tu\ x > -2 \\ 3x+1, \quad tu\ x \le -2 \end{cases}$$

7. დაბეჭდეთ y ცვლადის მნიშვნელობა:

$$y = \begin{cases} \dfrac{x^2+1}{x-1}, tu\ x \ge 0\ (x \ne 1) \\ 1, tu\ x = 1 \\ 3x^2, tu\ x < 0 \end{cases}$$

8. შეადგინეთ პროგრამა, რომელიც იპოვის მოცემული რიცხვის მოდულს.
9. n-ს მიანიჭეთ იმ მეოთხედის ნომერი რომელშიც ძევს A(x,y) წერტილი, თუ xy=0-ს დაბეჭდეთ “კოორდინატთა სათავე”.
10. მოცემული f(x)=3 x+0.5 ფუნქციისათვის შეადგინეთ m-ფუნქცია, რომელიც დაბეჭდავს ამ ფუნქციის ნიშანს მოცემული x-ისათვის.
11. ცნობილია, რომ მოცემული ოთხი a, b, c და d რიცხვებიდან სამი ტოლია, ერთი კი განსხვავებული. შეადგინეთ პროგრამა, რომელიც p ცვლადს მიანიჭებს განსხვავებული რიცხვის სიდიდეს.
12. მოცემულია სამი a, b, c რიცხვები, გამოარკვიეთ შედგება თუ არა ასეთი გვერდებით სამკუთხედი. თუ შედგება ვიპოვოთ ფართობი.

13. მოცემულია სამი a, b, c რიცხვები, გამოარკვიეთ შედგება თუ არა ასეთი გვერდებით სამკუთხედი. თუ შედგება გავარკვიოთ: არის თუ არა ტოლფერდა ან ტოლგვერდა.
14. მოცემულია a მასივი, თუ a-ს ელემენტები არანულოვანია, მაშინ ვიპოვოთ



მაქსიმალური მათ შორის, თუ a-ს გააჩნია ერთი მაინც ნულოვანი ელემენტი, მაშინ ვიპოვოთ მინიმალური ელემენტი მათ შორის.

15. მოცემულია ორი წრფის განტოლება:

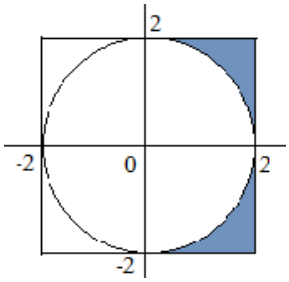
$$\begin{cases} a_1x + b_1x = c_1 \\ a_2x + b_2x = c_2 \end{cases}$$

გამოარკვიეთ ეს წრფეები პარალელურებია, ემთხვევიან ერთმანეთს თუ გადაიკვეთებიან.

16. შეამოწმეთ ეკუთვნის თუ არა მოცემული A(x,y) წერტილი გამუქებულ არეს:

ა)

ბ)



გადართვის ოპერატორი:

switch…case…otherwise…end

სინტაქსი:

swithch <გამოსახულება>

%გამოსახულება-აუცილებლად არის სკალარი ან სტრიქონი

case <მნიშვნელობა 1>

<ინსტრუქციები 1>

%შესრულდება თუ <გამოსახულება>=<მნიშვნელობა 1>

case <მნიშვნელობა 2>

…

otherwise <ინსტრუქციები >

%შესრულდება თუ <გამოსახულება> არ დაემთხვევა არცერთ <მნიშვნელობა>-ს

end

იმის მიხედვით თუ case-ის შემდეგ მდგომ რომელ მნიშვნელობას დაემთხვევა swithch ოპერატორის <გამოსახულება>

მაგ.: switch n

case –1

disp('მინუს ერთი')

case 0

disp('ნული')

case 1

disp('ერთი')

otherwise

disp('სხვა მნიშვნელობა')

end

ამ ოპერატორით (იმის მიხედვით, თუ რას უდრის n), ეკრანზე გამოვა შესაბამისი შეტყობინება.

case-ის შემდეგ შეიძლება იყოს რამდენიმე მნიშვნელობა. ამ შემთხვევაში მნიშვნელობები ჩაიწერება ფიგურულ ფრჩხილებში და ერთმანეთისაგან გამოიყოფა მძიმით:

მაგ.: swich var

case 1

disp('1')

case {2,3,4}

```
disp('2 ან 3 ან 4')
    otherwise
disp('სხვა მნიშვნელობა')
    end
```

ამოცანები

1. მოცემული f(x)=3 sin x+0.5 ფუნქციისათვის შეადგინეთ m-ფუნქცია, რომელიც დაბეჭდავს ამ ფუნქციის ნიშანს მოცემული x-ისათვის.
2. შეადგინეთ m-ფუნქცია, რომელიც გამოარკვევს მოცემული სიმბოლო არის: ციფრი, ხმოვანი, თუ რაიმე სხვა სიმბოლო.
3. შეადგინეთ m-ფუნქცია, რომელიც იმის და მიხედვით თუ რა სიმბოლოა c, დაბეჭდავს: ვერიკოს, სოფიკოს, დათოს ან უცნობს შესაბამისად რომელი სახელის პირველ ასოსაც დაემთხვევა c სიმბოლო.
4. შეადგინეთ m-ფუნქცია, რომელის პარამეტრია c, დაბეჭდავს: ვერიკოს, სოფიკოს, დათოს ან უცნობს შესაბამისად რომელი სახელიც წერია c-ში.
5. შეადგინეთ m-ფუნქცია, რომელიც იმისდა მიხედვით თუ რა სიმბოლოა c, დაბეჭდავს: ვერიკოს, სოფიკოს, დათოს ან უცნობს შესაბამისად რომელი სახელის პირველ ასოსაც დაემთხვევა c სიმბოლო.
6. მოცემულია რაიმე ნატურალური რიცხვი k, შეადგინეთ m-ფუნქცია, რომელიც გაარკვევს ეს რიცხვი ჯერადია 3-ის, 4-ის, 5-ის, თუ არ იყოფა არცერთ მათგანზე.
7. შეადგინეთ m-ფუნქცია, რომელიც month ცვლადისთვის მინიჭებული თვის შესაბამისად დაბეჭდავს წელიწადის რომელი დროა.
8. კვირის დღის ნომრის მიხედვით დაბეჭდეთ კვირის დღის დასახელება (ჩავთვალოთ, რომ კვირა იწყება ორშაბათიდან).
9. შემოდგომის თვეების ნომრების მიხედვით დაბეჭდეთ მათი სახელები.

ლექცია 6

ციკლის ოპერატორი განუსაზღვრელი ოპერაციების რაოდენობით:

while…end
სინტაქსი:

while < ლოგიკური გამოსახულება>
<ინსტრუქციები>
end

while ოპერატორი სრულდება შემდეგნაირად: თუ <ლოგიკური გამოსახულება> ჭეშმარიტია, შესრულდება <ინსტრუქციები>, შემდეგ კვლავ შემოწმდება ლოგიკური გამოსახულება და ა. შ. მანამ, სანამ გამოსახულება ჭეშმარიტია შესრულდება ინსტრუქციები და დამთავრდება როგორც კი გამოსახულება იქნება მცდარი. მაგ. თუ გამოსახულება ცარიელი მასივია, while ოპერატორი არც ერთხელ არ შესრულდება.

ციკლის ოპერატორი განსაზღვრული ოპერაციების რაოდენობით:

for…end

სინტაქსი:

for <ციკლის პარამეტრი>=<საწყისი მნიშვნელობა> : <ბიჯი> : <ბოლო მნიშვნელობა>
<ინსტრუქციები>
end

მაგ.: x(1)=-2;
for i=2:6
x(i)=2*x(i-1);
end

ციკლის ბიჯი, საწყის და საბოლოო მნიშვნელობა შეიძლება იყოს უარყოფითიც, ასევე შეიძლება იყოს ათწილადი სიდიდე. for ციკლი სრულდება შემდეგნაირად: ციკლის პარამეტრს მიენიჭება <საწყისი მნიშვნელობა> შესრულდება <ინსტრუქციები>, ციკლის პარამეტრს დაემატება <ბიჯი>, თუ პარამეტრის ახალი მნიშვნელობა ნაკლებია <ბოლო მნიშვნელობა>-ზე კვლავ შესრულდება <ინსტრუქციები>, თუ პარამეტრი აჭარბებს <ბოლო მნიშვნელობა>-ს ციკლი წყვეტს მუშაობას. შევნიშნოთ რომ თუ <საწყისი მნიშვნელობა> მეტია <ბოლო მნიშვნელობა>-ზე, მაშინ ციკლი საერთოდ არ შესრულდება.

დასაშვებია ციკლის ციკლში ჩადგმა. ამ შემთხვევაში გარე ციკლის პარამეტრს მიენიჭება შესაბამისი <საწყისი მნიშვნელობა> და ამ მნიშვნელობისათვის იმუშავებს შიგა ციკლი, გაიზრდება გარე ციკლის პარამეტრი შესაბამისი ბიჯით და თავიდან დაიწყება შიგა ციკლის შესრულება და ა. შ. გარე ციკლის ამოწურვამდე.

ციკლის პარამეტრი შეიძლება იყოს მასივი, მაგ.:

```
>> A=[-3 4 9 -11];
>> for i=A
<ინსტრუქციები>
```

```
end
```

ამ შემთხვევაში ციკლი შესრულდება 4-ჯერ i-ს შესაბამისი მნიშვნელობებისათვის: -3, 4, 9 და -11-ისათვის.

თუ მასივი ორგანზომილებიანია ასეთ შემთხვევაში ციკლის პარამეტრია ვექტორი-მასივის სვეტი, ე.ი. ციკლი შესრულდება იმდენჯერ, რამდენი სვეტიც აქვს მატრიცას.

ამოცანები

1. მოცემულია $a_1=5$, $a_n=3\ (n-1)+12$, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც ნაკლებია 10000-ზე.
2. მოცემულია $a_1=-2$, $a_n=-3\ a_{n-1}+5$, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც მეტია -10000-ზე.
3. მოცემულია $a_1=-1$, $a_n=-7\ a_{n-1}+1$, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც მეტია -1000-ზე.
4. მოცემულია $a_1=5$, $a_n=3\ a_{n-1}+7$, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც ნაკლებია 10000-ზე.
5. მოცემულია ნატურალური რიცხვი n. იპოვეთ ამ რიცხვის კენტი ციფრების ჯამი.
6. მოცემულია ნატურალური რიცხვი n. იპოვეთ ამ რიცხვის ლუწი ციფრების ნამრავლი.
7. მოცემულია ნატურალური რიცხვი n. იპოვეთ ამ რიცხვის გამყოფების ნამრავლი.
8. მოცემულია ნატურალური რიცხვი n. იპოვეთ ამ რიცხვის კენტი ციფრების კვადრატების ჯამი.
9. მოცემულია ნატურალური რიცხვი n. ამ რიცხვის ციფრებისაგან შეადგინეთ a მასივი, იპოვეთ მათ შორის მაქსიმალური.
10. მოცემულია ნატურალური რიცხვი n. იპოვეთ ამ რიცხვის 3-ის ჯერადი ციფრები.

ამოცანები

1. A(3×3) მატრიცი არის მაგიური , ჩაწერეთ ბრძანებები რომლითაც მიიღება: ა) A–ს არამთავარი დიაგონალი; ბ)ზედა სამკუთხა მატრიცა; გ) იპოვეთ ამ მატრიცის რანგი;
2. A(3×3) მატრიცი არის მაგიური, ჩაწერეთ ბრძანებები რომლითაც მიიღება: ა) A–ს მთავარი დიაგონალი; ბ)ქვედა სამკუთხა მატრიცა; გ) იპოვეთ ამ მატრიცის დეტერმინანტი;
3. A(3×3) მატრიცი არის მაგიური, ჩაწერეთ ბრძანებები რომლითაც მიიღება: ა) A–ს მთავარი დიაგონალი; ბ)ქვედა სამკუთხა მატრიცა; გ) იპოვეთ ამ მატრიცის ნულ-სივრცე;
4. A(3×3) მატრიცი არის მაგიური, ჩაწერეთ ბრძანებები რომლითაც მიიღება: ა) A–ს მთავარი დიაგონალი; ბ)ქვედა სამკუთხა მატრიცა; გ) იპოვეთ ამ მატრიცის დეტერმინანტი;
5. A(4×4) მატრიცი არის შემთხვევითი რიცხვებისაგან შედგენილი, შეადგინეთ M-ფუნქცია, რომელიც მთავარ და არამთავარ დიაგონალზე მდგომ ელემენტებს ადგილს შეუცვლის (A(1,1)↔A(4,1)).
6. A(4×4) მატრიცი არის შემთხვევითი რიცხვებისაგან შედგენილი, შეადგინეთ M-ფუნქცია, რომელიც მთავარ და არამთავარ დიაგონალზე მდგომ ელემენტებს ადგილს შეუცვლის (A(1,1)↔A(1,4)).
7. A(5×5) მატრიცი არის შემთხვევითი რიცხვებისაგან შედგენილი, შეადგინეთ M-ფუნქცია, რომელიც მთავარ და არამთავარ დიაგონალზე მდგომ ელემენტებს ადგილს შეუცვლის (A(1,1)↔A(1,5)).
8. *A(5×5) მატრიცი არის შემთხვევითი რიცხვებისაგან შედგენილი, შეადგინეთ M-ფუნქცია, რომელიც მთავარ და არამთავარ დიაგონალზე მდგომ ელემენტებს ადგილს შეუცვლის (A(1,1)↔A(5,1)). ???*
9. მოცემულია A(4×6) მატრიცა, შეუცვალეთ ადგილები პირველ და მესამე სვეტს, გამოთვალეთ A მატრიცის მეორე სტრიქონის ელემენტების საშუალო არითმეტიკული და ამავე სტრიქონში ის ელემენტები რომლებიც ნაკლებია საშუალო არითმეტიკულზე შეცვალეთ საშუალო არითმეტიკულით.
10. A(4×4) მატრიცი არის მაგიური კვადრატი, შეადგინეთ B(4×4) მატრიცა, რომელშიც მთავარ და არამთავარ დიაგონალზე მდგომ ელემენტებს ადგილი ექნებათ შეცვლილი, იპოვეთ A მატრიცის მაქსიმალური ელემენტი.
11. A(4×4) მატრიცი შემთხვევითი რიცხვებისაგან შედგენილი, იპოვეთ არამთავარ დიაგონალზე მდგომი კენტი ელემენტების ჯამი.
12. A(5×5) მატრიცი შემთხვევითი რიცხვებისაგან შედგენილი, იპოვეთ მთავარი დიაგონალზე მდგომი კენტი ელემენტების ჯამი და ლუწების ნამრავლი.
13. A(5×5) მატრიცი შემთხვევითი რიცხვებისაგან შედგენილი, შეუცვალეთ ადგილები მეორე და მეოთხე სვეტს. იპოვეთ მთავარი დიაგონალზე მდგომი ელემენტებს შორის მაქსიმალური.
14. A(5×5) მატრიცი შემთხვევითი რიცხვებისაგან შედგენილი, შეუცვალეთ ადგილები მეორე და მეოთხე სტრიქონებს. იპოვეთ არამთავარი დიაგონალზე მდგომი ელემენტების ჯამი.
15. მოცემულია A(4×6) მატრიცა, A მატრიცის ელემენტები დადებითი მთელი რიცხვებია. შეუცვალეთ ადგილები მეორე და მეოთხეე სტრიქონს, გამოთვალეთ A მატრიცის პირველი და მეორე სვეტის საშუალო არითმეტიკული და A

მატრიცაში ის ელემენტები რომლებიც ნაკლებია საშუალო არითმეტიკულზე შეცვალეთ 0-ით.

16. მოცემულია A(4×6) მატრიცა, A მატრიცის ელემენტები დადებითი მთელი რიცხვებია. შეუცვალეთ ადგილები მეორე და მესამე სვეტს, გამოთვალეთ A მატრიცის პირველი და მეორე სტრიქონის საშუალო არითმეტიკული და ამავე სტრიქონებში ის ელემენტები რომლებიც ნაკლებია საშუალო არითმეტიკულზე შეცვალეთ 0-ით.
17. მოცემულია A(4×6) მატრიცა, A მატრიცის ელემენტები დადებითი მთელი რიცხვებია. შეუცვალეთ ადგილები მეორე და მესამე სვეტს, გამოთვალეთ A მატრიცის პირველი და მეორე სტრიქონის საშუალო არითმეტიკული და ამავე სტრიქონებში ის ელემენტები რომლებიც ნაკლებია საშუალო არითმეტიკულზე შეცვალეთ 0-ით.
18. მოცემულია D(5×4) მატრიცა, D მატრიცის ელემენტები დადებითი მთელი რიცხვებია. შეუცვალეთ ადგილები მეორე და მესამე სვეტს, გამოთვალეთ D მატრიცის პირველი და მეორე სტრიქონის საშუალო არითმეტიკული და ამავე სტრიქონებში ის ელემენტები რომლებიც მეტია საშუალო არითმეტიკულზე შეცვალეთ 0-ით.
19. მოცემულია სტრიქონი, შეადგინეთ ამ სტრიქონში შემავალი: ა) თანხმოვნების მიმდევრობა; ბ) ლუწი ციფრების მიმდევრობა; გ) დაითვალეთ ჰარის, ხმოვნების და ლუწი ციფრებისაგან განსხვავებული სიმბოლოების რაოდენობა.
20. მოცემულია სტრიქონი, შეადგინეთ ამ სტრიქონში შემავალი: ა) ხმოვნების მიმდევრობა; ბ) ჰარის, ხმოვნების და ლუწი ციფრებისაგან განსხვავებული სიმბოლოების მიმდევრობა; გ) დაითვალეთ ლუწი ციფრების რაოდენობა;
21. მოცემულია სტრიქონი, შეადგინეთ ამ სტრიქონში შემავალი: ა) ხმოვნების მიმდევრობა; ბ) ჰარის, ხმოვნების და ლუწი ციფრებისაგან განსხვავებული სიმბოლოების მიმდევრობა; გ) დაითვალეთ ლუწი ციფრების რაოდენობა;
22. მოცემულია სტრიქონი, შეადგინეთ ამ სტრიქონში შემავალი: ა) ხმოვნების მიმდევრობა; ბ) ჰარის, ხმოვნების და ლუწი ციფრებისაგან განსხვავებული სიმბოლოების მიმდევრობა; გ) დაითვალეთ ლუწი ციფრების რაოდენობა;
23. მოცემულია n ელემენტიანი ნატურალურ რიცხვთა a მიმდევრობა, ამ მიმდევრობის ლუწი ელემენტებისაგან შეადგინეთ b მიმდევრობა და მიღებული მიმდევრობა დაალაგეთ კლების რიგით და იპოვეთ b-ს მაქსიმალური ელემენტი (იგულისხმება რომ a-ში ერთი წევრი მაინც ლუწია);
24. მოცემულია n ელემენტიანი ნატურალურ რიცხვთა a მიმდევრობა, ამ მიმდევრობის კენტი ელემენტებისაგან შეადგინეთ b მიმდევრობა და მიღებული მიმდევრობა დაალაგეთ კლების რიგით და იპოვეთ b-ს მინიმალური ელემენტი (იგულისხმება რომ a-ში ერთი წევრი მაინც კენტია);
25. მოცემულია n ელემენტიანი ნატურალურ რიცხვთა a მიმდევრობა, ამ მიმდევრობის ლუწი ელემენტებისაგან შეადგინეთ b მიმდევრობა და მიღებული მიმდევრობა დაალაგეთ კლების რიგით და იპოვეთ b-ს მაქსიმალური ელემენტი (იგულისხმება რომ a-ში ერთი წევრი მაინც ლუწია);
26. მოცემულია n ელემენტიანი ნატურალურ რიცხვთა a მიმდევრობა, ამ მიმდევრობის კენტი ელემენტებისაგან შეადგინეთ b მიმდევრობა და მიღებული

მიმდევრობა დაალაგეთ ზრდის რიგით და იპოვეთ b-ს მინიმალური ელემენტი (იგულისხმება რომ a-ში ერთი წევრი მაინც კენტია);

27. მოცემული x€[-4;4]-ინტერვალში h=0,5 ბიჯით გამოთვალეთ f(x), თუ:

$$f(x)=\begin{cases} x^2+7, \text{თუ } x>2 \\ 3\sin x+e^x, \text{თუ } -3<x<2 \\ \dfrac{x^2}{3\cos x-2}, \text{თუ } x\le -3 \end{cases}$$

28. მოცემული x€[-3;3]-ინტერვალში h=0,2 ბიჯით გამოთვალეთ f(x), თუ:

$$f(x)=\begin{cases} 2\sin x^2-2, \text{თუ } x>2 \\ 35x+\dfrac{x^3}{2\cos x+1.5}, \text{თუ } -2<x<2 \\ x^2-1, \text{თუ } x\le -2 \end{cases}$$

29. მოცემული x€[-4;4]-ინტერვალში h=0,5 ბიჯით გამოთვალეთ f(x), თუ:

$$f(x)=\begin{cases} x^2+7, \text{თუ } x>2 \\ 3\sin x+e^x, \text{თუ } -3<x<2 \\ \dfrac{x^2}{3\cos x-2}, \text{თუ } x\le -3 \end{cases}$$

30. მოცემული x€[-3;3]-ინტერვალში h=0,2 ბიჯით გამოთვალეთ f(x), თუ:

$$f(x)=\begin{cases} 2\sin x-1, \text{თუ } x>2 \\ \dfrac{x-2}{2\sin^2 x+3.5}, \text{თუ } -2<x<2 \\ \cos x^2-1, \text{თუ } x\le -2 \end{cases}$$

31. მოცემულია a_1=5, a_n=3 a_{n-1}+7, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც ნაკლებია 10000-ზე.
32. მოცემულია a_1=-2, a_n=-3 a_{n-1}+5, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც მეტია -10000-ზე.
33. მოცემულია a_1=5, a_n=3 a_{n-1}+7, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც ნაკლებია 10000-ზე.
34. მოცემულია a_1=-1, a_n=-7 a_{n-1}+1, იპოვეთ a_n მიმდევრობის ის მაქსიმალური წევრი და მისი ნომერი, რომელიც მეტია -1000-ზე.
35. მოცემულია რაიმე s სტრიქონი, შეადგინეთ m-ფუნქცია, რომელიც გაარკვევს ამ სტრიქონში რამდენი ხმოვანია (a,e,i,o,u,A,E,I,O,U), რამდენი ციფრია (0,1,2,3,4,5,6,7,8,9) და რამდენი სხვა სიმბოლოა.
36. მოცემულია რაიმე s სტრიქონი, შეადგინეთ m-ფუნქცია, რომელიც გაარკვევს ამ სტრიქონში რამდენი ხმოვანია (a,e,i,o,u,A,E,I,O,U), რამდენი სასვენი ნიშანი (: ; , . ! ?) და რამდენი სხვა სიმბოლოა.

37. მოცემულია რაიმე s სტრიქონი, შეადგინეთ m-ფუნქცია, რომელიც გაარკვევს ამ სტრიქონში რამდენი ციფრია (0,1,2,3,4,5,6,7,8,9), რამდენი მოქმედების ნიშანი (*, /, -,+) და რამდენი სხვა სიმბოლოა.
38. მოცემულია რაიმე s სტრიქონი, შეადგინეთ m-ფუნქცია, რომელიც ხმოვნებს: a,e,i,o,u,A,E,I,O,U შეცვლის ციფრებით: 0,1,2,3,4,5,6,7,8,9, მოცემული რიგის შესაბამისად დაითვალეთ შეცვლილი სიმბოლოების რაოდენობა.
39. მოცემულია რაიმე s სტრიქონი, შეადგინეთ m-ფუნქცია, რომელიც დაითვლის გახსნილი და დახურული ფრჩხილების რაოდენობას (,[,],) მოქმედების ნიშნების: *, /, -,+ რაოდენობას, თუ ციფრები მეტია მოქმედების ნიშნებზე დაბეჭდეთ true, წინააღმდეგ შემთხვევაში false.

ჩანაწერების მასივები

ჩანაწერთა მასივი MATLAB–ში უზრუნველყოფს სხვადასხვა ტიპის მონაცემების დასაგროვებლად. ასთი მასივების განმასხვავებელი თავისებურებაა – სახელიანი ველების არსებობა.

სტრუქტურის განსაზღვრა. სტრუქტურა – ესაა ჩანაწერების მასივი სახელიანი ველებით, რომელთა დანიშნულებაა მონაცემების შენახვა. ველი შეიძლება შეიცავდეს ნებისმიერი ტიპის მონაცემს.

მაგალითად:

განვიხილოთ patient სტრუქტურა, რომელშიც ველი name – აღნიშნავს პაციენტის სახელს, ველი billing – გადახდის ანგარიში, ველი test – სამედიცინო შემოწმების შედეგები.

patient
.name John doe
.billing 127.00
.test

79	75	73
180	178	177.5
220	210	205

ნახ. 1

სტრუქტურის აგება

სტრუქტურა შეიძლება ავაგოთ ორი ხერხით:

- მინიჭების ოპერეატორის გამოყენებით;

- ფუნქციის გამოყენებით.

მინიჭების ოპერატორის გამოყენება

იმისათვის რომ დავაფორმიროთ მარტივი სტრუქტურა ზომით 1×1, აუცილებელია მონაცემები მივანიჭოთ შესაბამის ველებს. MATLAB სისტემა ავტომატურად დააფორმირებს სტრუქტურას მისი შევსების მიხედვით.

მაგალითად დავაფორმიროთ ნახ.1-ზე მოყვანილი patient 1×1 ზომის სტრუქტურა შემდეგი ველებით:

```
patient.name='Jon Doe';
patient.billing=127.00;
patient.test=[79 75 73; 180 178 177.5; 220 210 205]
```

ახლა თუ ბრძანებათა სტრიქონში შევიტანთ სტრუქტურის სახელს:

```
>>patient
```

მივიღებთ ჩანაწერის აღწერას:

```
patient=
name='Jon Doe';
billing=127.00;
test=[3×3 double]
```

ამგვარად patient მასივია ერთი ჩანაწერით და სამი ველით. იმისათვის რომ ის გავაფართოვოთ საკმარისია სტრუქტურის სახელს დავუმატოთ ინდექსი. მაგალითად შევქმნათ მეორე ჩანაწერი patient სტრუქტურაში:

```
patient(2).name='Ann Lane';
patient(2).billing=28.50;
patient(2).test=[68 70 68; 118 118 119; 172 170 169];
```

ახლა patient სტრუქტურას აქვს ზომა 1×2. შევნიშნოთ რომ როდესაც სტრუქტურა შეიცავს ერთ ჩანაწერზე მეტს, მის მოთხოვნაზე გამოიტანება მხოლოდ ზოგადი ინფორმაცია ველებზე და არა მათი მნიშვნელობები:

```
patient
patient=
1×2 struct array with fields:
name
billing
test
```

ამ ინფორმაციის მისაღებად უნდა გამოვიყენოთ fieldnames ფუნქცია.

სტრუქტურის გაფართოებისას შეუვსებელ ველებს MATLAB მიანიჭებს ცარიელ მასივებს, მაგალითად:

```
patient.name(3)='Marta Klark';
patient
patient=
1×3 struct array with fields:
name
```

billing
test
ანუ სტრუქტურისათვის სრულდება შემდეგი პირობა:

- სტრუქტურის ყველა ჩანაწერს აქვს ერთიდაიგივე რაოდენობის ველები;
- ველების სახელები ერთიდაიგივეა ყველა ჩანაწერისათვის;

სხვადასხვა ჩანაწერისთვის ველების ზომა შეიძლება იყოს სხვადასხვა. patient სტრუქტურისათვის name ველს შეიძლება ჰქონდეს სხვადასხვა სიგრძე, ველი test შეიძლება იყოს სხვადასხვა ზომის მასივი და ა. შ.

struct ფუნქციის გამოყენება

struct ფუნქციას აქვს შემდეგი სინტაქსი:
str_array=struct('ველის სახელი','მნიშვნელობა', 'ველის სახელი', 'მნიშვნელობა', ...)
მაგალითად:
patient=stuct('name', 'Jon Doe', 'billing', 127.00, 'test',[79 75 73; 180 178 177.5; 220 210 205])
struct ფუნქციის გამოყენებით შექმნილ სტრუქტურის ყველა ველს მიენიჭება ფორმირების დროს მინიჭებული მნიშვნელობა. ანუ ჩვენს შემთხვევაში Jon Doe ყველა name ველში ეწერება, ყველა billing-ში მნიშვნელობა 127.00. შემდეგში ეს მნიშვნელობები უნდა შეიცვალოს მინიჭების ოპერატორის გამოყენებით.

სტრუქტურის მონაცემებზე და ველებზე წვდომა

სტრუქტურის რომელიმე ველზე მისამართავად უნდა ჩავწეროთ სტრუქტურის სახელი, შემდეგ წერტილი (.), შემდეგ ველის სახელი:
s=patient(2).name
s=Ann Lane
მე-3-ე პაციენტისათვის სამედიცინო ტესტის (2, 2) მონაცემისათვის მნიშვნელობის მისანიჭებლად უნდა ჩავწეროთ:
patient(3).test(2,2)=167;
სტრუქტურის რომელიმე ველის ყველა ჩანაწერის მიღება შესაძლებელია ციკლის ოპერეტორის გამოყენებით:
for i=1:length(patient)
disp(patient(i).name)
end
ჩანაწერზე წვდომისათვის საჭიროა სტრუქტურის სახელის ინდექსირება, მაგალითად
B=patient(2)

მრავალგანზომილებიანი მასივები

მრავალგანზომილებიანი მასივები –ეს ის მასივებია, რომელთა განზომილება ორზე მეტია. მრავალგანზომილებიანი მასივები ფართოდ გამოიყენებიან მრავალგანზომილებიანი მონაცემების აღწერისას. ეს მასივები შეიძლება იყოს რიცხვითი, სიმბოლური, უჯრების და სტრუქტურების მასივები.

ფუნქციები	დანიშნულება
cat	მრავალგანზომილებიანი მასივის ფორმირება
hdims	მრავალგანზომილებიანი მასივის ზომის განსაზღვრა
ndgrid	მრავალგანზომილებიანი ფუნქციისათვის ბადის გენერირება
permute, ipermute	განზომილებების გადაცვლა (გადანაცვლება)
shiftdim	მასივის განზომილების შეცვლა
squeeze	ერთ-ერთი განზომილების გაუქმება

მრავალგანზომილებიანი მასივის განსაზღვრა

მრავალგანზომილებიანი მასივები MATLAB-ის ჩვეულებრივი ორგანზომილებიანი მასივების გაფართოებას წარმოადგენენ. ორგანზომილებიან მასივებს გააჩნიათ ორი განზომილება სტრიქონების და სვეტების. ორგანზომილებიან მასივებზე წვდომა მიიღწევა ორი ინდექსის შეტანით – სტრიქონისა და სვეტისა.

განვიხილოთ სამგანზომილებიანი მასივი; ელემენტების აღსანიშნავად საჭიროა 3 ინდექსი: პირველი- სტრიქონების არსანიშნავად, მეორე-სვეტები; მესამე – გვერდები. განზომილების გაფართოებისას ემატება ახალი ინდექსები. ოთხგანზომილებიანი მასივი ხასიათდება ოთხი ინდექსით: პირველი ორი განსაზღვრავს სტრიქონს და სვეტს, ბოლო ორი მონაცემთა მესამე და მეოთხე ზომას.

მრავალგანზომილებიანი მასივების ფორმირება

მრავალგანზომილებიანი მასივების ფორმირება შესაძლებელია 3 მიდგომით:

- ინდექსების გამოყენება
- ჩადგმული ფუნქციების გამოყენება, სპეციალური სახს მასივების ფორმირებისათვის
- cat ფუნქციის გამოყენება

ინდექსების გამოყენება. მრავალგანზომილებიანი მასივების ფორმირების ერთ-ერთი ხერხი მდგომარეობს მისი როგორც ორგანზომილებიანი მასივების განთავსებაში ახალ გვერდებზე. განვიხილოთ მაგალითი:

ჯერ დავაფორმიროთ ორგანზომილებიანი A მასივი:

```
A=[5 7 8; 0 1 9; 4 3 6];
A=
5 7 8
0 1 9
4 3 6
```

ამ მასივს აქვს 2 განზომილება და ზომა 3×3. დავუმატოთ ახალი გვერდი მესამე განზომილებაში შემდეგი მინიჭების ოპერატორით:

```
A(:, :, 2)=[1 0 4; 3 5 6; 7 8 9]
```

A(:, :, 1)=
5 7 8
0 1 9
4 3 6
A(:, :, 2)=
1 0 4
3 5 6
9 8 7
დაფორმირებულია A მასივი 3 განზომილებით და ზომით 3×3×2.
სხვადასხვა განზომილებისა და ზომის მასივების ფორმირებისათვის შეგვიძლია გავნაგრძოთ სტრიქონების, სვეტებისა და გვერდების დამატება.
მრავალგანზომილებიანი მასივის განზომილების შესაცვლელად საჭიროა:

- გავზარდოთ ან შევამციროთ შესაბამისი ინდექსი;
- მივანიჭოთ ერთ ან რამდენიმე მანამდე არარსებულ ელემენტს მნიშვნელობები.

მასივის მთელ გვერდს შეგვიძლია მივანიჭოთ სკალარი და ამგვარად ერთდროულად შევავსოთ მთელი გვერდიფენა:
მაგ. A(:, :, 3)=5
A(:, :, 3)=
5 5 5
5 5 5
5 5 5
A(:, :, 2)=
1 0 4
3 5 6
9 8 7
A(:, :, 1)=
5 7 8
0 1 9
4 3 6
ეს არის 3 განზომილებიანი მასივი ზომით 3×3×3.
დავუმატოთ A მასივს მეოთხე განზომილება, შემოვიტანოთ:
A(:, :,1, 2)=[1 2 3; 4 5 6; 7 8 9];
A(:, :,2, 2)=[9 8 7; 6 5 4; 3 2 1];
A(:, :, 3,2)=[1 0 1; 1 1 0; 0 1 1];

A(:, :, 3,1)=
5 5 5
5 5 5
5 5 5

A(:, :, 3,2)=
1 0 1
1 1 0
0 1 1

A(:, :, 2,1)=
1 0 4
3 5 6
9 8 7

A(:, :, 2,2)=
9 8 7
6 5 4
3 2 1

ესაა 4-განზომილებიანი მასივი ზომით 3×3×3×2.

ჩადგმული ფუნქციების გამოყენება. ისეთი ფუნქციები, როგორიცაა randn, ones და zeros შეიძლება გამოყენებულ იქნეს მრავალგანზომილებიანი მასივების ფორმირებისათვის, ამ ფუნქციების ყოველი არგუმენტი განსაზღვრავს შესაბამისი განზომილების ზომას.

მაგალითად: დავაფორმიროთ 3 განზომილებიანი ნორმალურად განაწილებული შემთხვევითი რიცხვების 4×3×2 ზომის მასივი:

```
B=randn(4,3,2)
```

მივიღებთ

```
B(:,:,1) =
    0.5377    0.3188    3.5784
    1.8339   -1.3077    2.7694
   -2.2588   -0.4336   -1.3499
    0.8622    0.3426    3.0349
B(:,:,2) =
    0.7254   -0.1241    0.6715
   -0.0631    1.4897   -1.2075
    0.7147    1.4090    0.7172
   -0.2050    1.4172    1.6302
```

თუ გვინდა რომ დავაფორმიროთ მასივი, რომელიც კონსტანტითაა შევსებული, მოხერხებულია repmat ფუნქციის გამოყენება:

```
B=repmat(5,[3 4 2])
B(:, :,1)=
5 5 5
5 5 5
5 5 5
B(:, :,2)=
5 5 5
5 5 5
5 5 5
```

შენიშვნა: თუ მასივის ერთი მაინც განზომილება უდრის 0-ს, ეს ნიშნავს რომ მრავალგანზომილებიანი მასივი ცარიელია.

cat ფუნქციის გამოყენება. cat ფუნქციის გამოყენება მნიშვნელოვნად ამარტივებს მრავალგანზომილებიანი მასივების ფორმირებას, რამდენადაც ის საშუალებას იძლევა მითითებულ გაზომილებაში ორგანზომილებიანი მასივის განთავსებისა, შემდეგი სინტაქსის გამოყენებით:

B=cat(dim, A1, A2, …)

სადაც dim–განზომილების ნომერია, რომლის გასწვრივაც განთვასდებიან მასივები.

A1, A2, … - ორგანზომილებიანი მასივების სიაა.

მაგალითად: დავაფორმიროთ სამგანზომილებიანი მასივი, რომელიც გააერთიანებს ორ ორგანზომილებიან მასივს ზომით 2×2-ზე:

B=cat(3,[2 8; 0 5], [1 3; 7 9])

B(:,:,1)=	B(:,:,2)=
2 8	1 3
0 5	7 9

მაგალითად დავაფორმიროთ ოთხგანზომილებიანი D მასივი cat ოპერატორის შემდეგი მიმდევრობით:

A=cat(3,[9 2; 6 5], [7 1; 8 4]);

B=cat(3,[3 5; 0 1], [5 6; 2 1]);

D=cat(4,A, B, cat([1 2; 3 4], [4 3; 2 1]));

D(:,:,2,1)=	D(:,:,2,2)=	D(:,:,2,3)=
7 1	5 6	4 3
8 4	2 1	2 1

D(:,:,1,1)=	D(:,:,1,2)=	D(:,:,1,3)=
9 2	3 5	1 2
6 5	0 1	3 4

მრავალგანზომილებიანი მასივების მახასიათებლები:

იმისათვის რომ მივიღოთ მრავალგანზომილებიანი მასივების მახასიათებლები უნდა გამოვიყენოთ შემდეგი ფუნქციები:

- whos – მასივის ტიპი და გამოყენებული მეხსიერება;
- ndims – განზომილებების რაოდენობა;
- size – მასივის ზომა.

II ნაწილი

სიმბოლური გამოთვლები
წრფივი ალგებრა

მატრიცების შეკრება და გამოკლება:

MATLAB-ში შესაძლებელია ერთნაირი ზომის მატრიცების შეკრება და გამოკლება, მაგ.:

```
>>A=[0 1; 3 5];
>>B=[-1 3; 6 4];
>>A+B
        ans= -1    4
              9    9
```

სიმბოლურისათვის გვექნება:

```
>>syms a b
>>C=[a 0; 0 1];
>>D=[1 0; 0 b];
>>C+D
ans= [a+1,     0]
     [   0, 1+b]
```

მატრიცების გამრავლება

ვიპოვოთ A და B მატრიცების ნამრავლი A*B:

$$A = \begin{pmatrix} 1 & 1 & 0 & 3 \\ 2 & 4 & 5 & 3 \end{pmatrix} \qquad B = \begin{pmatrix} 4 & 2 \\ -2 & 4 \\ 1 & 3 \\ 1 & -5 \end{pmatrix}$$

```
>>A=[1 1 0 3; 2 4 5 3];
>>B=[4 2; -2 4; 1 3; 1 -5];
>>A*B
        ans= 21   -1
              8   20
```

ვექტორების სკალარული ნამრავლი:

```
>> x=[-5 4 3 0 9];
```

```
>> y=[-4 7 3 1 0];
>> dot(x,y)
ans =    57
```

იგივეა რაც:

```
>> sum(x.*y)
ans =    57
```

ვექტორების ვექტორული ნამრავლი:

```
>> syms a b c d e r
>> A=[a; b; c];
>> B=[d e r];
>> cross(A,B)
ans =
[ b*r - c*e, c*d - a*r, a*e - b*d]
```

მატრიცის ახარისხება:

```
>>A=[1 3; -2 5];
>>A^2
ans= -5   18
      -12  19
```

მატრიცის ახარისხება ელემენტების მიხედვით:

```
>>A=[1 3; -2 5];
>>C=A.^2
          C=1     9
             4      25
```

მატრიცის ტრანსპონირება:

```
>> A=[1 2 3; -4 5 6];
>> A'
ans =
   1   -4
   2    5
   3    6
```

მატრიცული მრავალწევრის გამოთვლა:

მოცემულია $P(x)=x^2-4x+2$ მრავალწევრი და $A=\begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}$ მატრიცა, იპოვეთ P(A).

```
>>A=[1 0; 2 1];
>>P=[1 –4 2];
>>PA=polyvalm(P,A)
  PA=-1   0
        -4  -1
```

მატრიცის დეტერმინანტის გამოთვლა:

```
>>A=[3 4 2; 7 5 1; 3 2];
>>det(A)
  ans= - 48
```

მატრიცის რანგის გამოთვლა:

```
>>A=[1 2 5 6; 2 3 –2 –4; 5 8 1 -2];
>>rank(A)
      ans= 2
```

მატრიცის ზედა სამკუთხა ფორმაზე მიყვანა:

```
>>A=[8 7 2 10; -8 2 7 10; 4 4 4 5; 0 4 –3 2];
>>rref(A)
        ans=  1 0 0 0
              0 1 0 0
              0 0 1 0
              0 0 0 1
```

rref-ის შესრულების საფეხურებს შეგვიძლია დავაკვირდეთ თუ rref(A) ბრძანების ნაცვლად გამოვიყენებთ rrefmovie(A) ბრძანებას.

შებრუნებული მატრიცის პოვნა:

```
>>A=[3 –4 5; 2 –3 1; 3 –5 -1];
>>AI=inv(sym(A))
        AI= [ -8,  29,  -11]
            [ -5,  18,   -7]
```

[1, -3, 1]

თუ AI=inv(sym(A)) ბრძანების ნაცვლად გამოვიყენებთ AI=inv(A) ბრძანებას, მაშინ შებრუნებული AI მატრიცა წარმოდგენილი იქნებოდა ორმაგი სიზუსტით.

მატრიცებისა და ვექტორების ნორმები.

MATLAB სისტემაში მოცემული A(mxn) მატრიცის ნორმების გამოსათვლელად გამოიყენება norm ფუნქცია. ამასთან A მატრიცის m ნორმა გამოითვლება ფუნქციით: norm(A,inf)=max(sum(abs(A'))); A მატრიცის l ნორმა გამოითვლება ფუნქციით: norm(A,1)=max(sum(abs(A))); მატრიცის k ნორმა გამოითვლება ფუნქციით: norm(a,'fro')=sqrt(sum(diag(A'*A))); ვექტორის ნორმები გამოითვლება ანალოგიური ფუნქციებით: C ვექტორის m ნორმა გამოითვლება ფუნქციით: norm(C,inf)=max(abs(C')); C ვექტორის l ნორმა გამოითვლება ფუნქციით: norm(C,1)=sum(abs(C)); ვექტორის k ნორმა გამოითვლება ფუნქციით: norm(C,2)=sum(abs(C).^2)^(1/2). მაგ.:

```
>>A=[0 1 5; 3 2 0; 6 1 2];
>>Am=norm(a,inf)
  Am=9
>>Al=norm(A,l)
  Al=9
>>Ak=norm(A,'fro')
   Ak=8.9443
```

მატრიცის ნულ-სივრცის ბაზისის, დეფექტი

MATLAB სისტემაში მოცემული null ფუნქციით შეგვიძლია განვსაზღვროთ მატრიცის ბირთვი; რიცხვითის შემთხვევაში ვიპოვით ორთონორმირებულ ბაზისს, სიმბოლური მატრიცის შემთხვევაში ვიპოვით ნულ-სივრცის ბაზისს: null(sym(A)) ან null(a,'r')-ით. colpace ფუნქციით ვიპოვით მოცემული მატრიცის სვეტების სივრცის ბაზისს. მაგ.:

```
>>A=[1 2 1; 1 0 -1; 1 1 0];
>>NA=null(sym(A))
NA=
     [ 1]
     [-1]
```

[1]

```
>>defekt=size(NA,2)
defekt=1
```

მატრიცის საკუთრივი რიცხვები და საკუთრივი ვექტორები:

MATLAB-ში მახასიათებელი პოლინომის კოეფიციენტების ფორმირებისათვის გამოიყენება ფუნქცია poly(A), თუ A მატრიცა სიმბოლურია, მაშინ პოლინომი ფორმირდება სიმბოლური სახით, poly(A,v)-საშუალებას გვაძლევს მივუთითოთ პოლინომის დამოუკიდებელი ცვლადი. solve(eq) ფუნქციით, სადაც eq ჩვენი სიმბოლური პოლინომია, შეგვიძლია ვიპოვოთ პოლინომის ფესვები. [R,D]= =eig(sym(A)), ოპერატორის შესრულების შედეგად R-ში მივიღებთ საკუთრივ ვექტორებს (ისინი R-ის სვეტები იქნებიან), ხოლო D-ს დიაგონალზე გვექნება საკუთრივი რიცხვები. მაგ.:

```
>>A=[2 –1 2; 5 –3 3; -1 0 -2];
>>[r,d]=eig(sym(A))
        r = [ -1]
            [ -1]
            [  1]
        d=
        [-1,  0, 0]
        [0,  -1, 0]
        [0,  0, -1]
```

მაგალითი 2:

```
>>A=[4 -5 2; 5 -7 3; 6 -9 4];
>>p=poly(sym(A))
         p =x^3-x^2
>> lambda=solve(p)
        lambda =[ 0]
                 [ 0]
                 [ 1]
```

ან

```
>> p=poly(A)
p =
   1.0000   -1.0000   -0.0000    0.0000
```

წრფივ განტოლებათა სისტემის ამოსახსნა

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

MATLAB სისტემაში წრფივ განტოლებათა სისტემის ამოსახსნელად გამოიყენება \, / ოპერატორები; ამასთან AX=B განტოლება ამოიხსნება: X=A\B ოპერატორით, XA=B განტოლება კი ამოიხსნება: X=B/A ოპერატორით. მაგალითად ამოვხსნათ წრფივი სისტემა:

$$\begin{cases} 4x_1 + 2x_2 + 3x_3 = -2 \\ 2x_1 + 8x_2 - x_3 = 8 \\ 9x_1 + x_2 + 8x_3 = 0 \end{cases}$$

```
>> A=[4 2 3; 2 8 -1; 9 1 8];
>> B=[-2; 8; 0];
>> X=A\B
X =
    0.0510
    0.8265
   -1.2857
```

მატრიცული განტოლებების ამოხსნა

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot X = \begin{pmatrix} 3 & 5 \\ 5 & 9 \end{pmatrix}$$

რადგან AX=B მატრიცული განტოლების ამოხსნა ალგებრულად არის $X=A^{-1}\cdot B$ მატრიცა, ამიტომ MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>> A=[1 2; 3 4];
>> B=[3 5; 5 9];
>> AI=sym(inv(A))
```

```
        AI=[-2,     1]
            [3/2, -1/2]
>> X=AI*B
        X=[-1, -1]
            [2,   3]
```

ან

```
>> X=A\B
X =
  -1.0000   -1.0000
   2.0000    3.0000
```

მათემატიკური ანალიზი

ფუნქციის ზღვრის გამოთვლა

MATLAB-ში ფუნქციის ზღვრი გამოითვლება limit ფუნქციის გამოყენებით. limit(F,x,a) განსაზღვრავს F(x) ფუნქციის ზღვარს, როდესაც $x \to a$, limit(F,a) ავტომატურად განსაზღვრავს დამოუკიდებელ ცვლადს, მაგ. t-ს და გამოითვლის F(t) ფუნქციის ზღვარს; ფუნქციები limit(F,x,a,'right') და limit(F,x,a,'left') გამოითვლიან შესაბამისად მარჯვენა და მარცხენა ზღვრებს.

მაგალითად გამოვთვალოთ შემდეგი ზღვარი:

$$\lim_{x \to 1} \frac{x^2 - 3x + 2}{x^2 - 1}$$

```
>>syms  x
>>limit((x^2-3*x+2)/(x^2-1), x, 1)
        ans= -1/2
```

ფუნქციის წარმოებულის გამოთვლა

MATLAB-ში ფუნქციის წარმოებული გამოითვლება diff ფუნქციის გამოყენებით. diff(F) თვითონ განსაზღვრავს დამოუკიდებელ ცვლადს და გამოითვლის F ფუნქციის წარმოებულს.

ფუნქციები diff(F,'v') და diff(F, sym('v')) გააწარმოებენ სიმბოლურ F გამოსახულებას v ცვლადის მიმართ.

ფუნქციები diff(F, n) და diff(F, ‘v’, n) და diff(F, n,‘v’) გააწარმოებენ სიმბოლურ F გამოსახულებას v ცვლადის მიმართ n-ჯერ.

თუ გვინდა წარმოებულის მნიშვნელობა ვიპოვოთ მოცემულ წერტილში: უნდა გამოვთვალოთ წარმოებული, დამოუკიდებელ ცვლადს მივანიჭოთ საჭირო მნიშვნელობა და ვისარგებლოთ subs(S) ფუნქციით, რომელიც S გამოსახულებაში ჩასვამს ცვლადის მნიშვნელობას და გამოითვლის მას.

მაგალითად გამოვთვალოთ შემდეგი ფუნქციის წარმოებული

$$y = (x^2 + 3)^{\sqrt{x}}$$

```
>> syms  x
>> dify=diff((x^2+3)^(sqrt(x)))
    dify=(x^2+3)^(x^(1/2))*(1/2/x^(1/2)*log(x^2+3)+2*x^(3/2)/(x^2+3))
>>pretty(dify)
          1/2
        (x    ) /         2              3/2  \
   2            |     log(x  + 3)       x        |
 (x  + 3)       | 1/2 ----------- + 2 ------       |
                |        1/2             2        |
                \     x              x  + 3    /
>> subs(dify,1)
ans =  4.7726
```

pretty ფუნქცია ჩაწერს ჩვენს გამოსახულებას მათემატიკურ ჩანაწერთან მიახლოებულად.

ფუნქციის ინტეგრალის გამოთვლა

სიმბოლური ინტეგრება

MATLAB-ში ერთი ცვლადის ფუნქციის ინტეგრალი გამოითვლება int ფუნქციის გამოყენებით. int(S) ავტომატურად განსაზღვრავს დამოუკიდებელ ცვლადს და შემდეგ ითვლის განუსაზღვრელ ინტეგრალს. int(S,v) ითვლის განუსაზღვრელ ინტეგრალს v ცვლადის მიმართ. int(S,a,b) ითვლის განსზღვრულ ინტეგრალს დამოუკიდებელი ცვლადის მიმართ a–დან b-მდე. int(S,v,a,b) ითვლის განსზღვრულ ინტეგრალს v ცვლადის მიმართ a–დან b-მდე. მაგალითად, გამოვთვალოთ ინტეგრალი:

$$\int_{-1}^{\sin t} 4x^2 t \, dx$$

```
>> syms x t
>> int(4*x^2*t,x,-1, sin(t))
ans = (4*t*(sin(t)^3 + 1))/3
```

რიცხვითი ინტეგრება

quad ფუნქციით შესაძლებელია ფუნქციათა რიცხვითი ინტეგრება მოცემული სიზუსტით. ინტეგრალის გამოსათვლელად წინასწარ ფუნქცია უნდა გავაფორმოთ M-ფუნქციად. quad–ფუნქციაში მიეთითება M-ფუნქციის სახელი, ინტეგრების საზღვრები და მიახლოებითი გამოთვლის სიზუსტე. მაგალითად გამოვთვალოთ ინტეგრალი:

$\int_0^{pi/4} x^2 \cos 2x \, dx$

ჩავწეროთ M-ფუნქცია:

```
function y = fun(x)
        y = x.^2.*cos(2*x);
end
```

ინტეგრალის გამოსათვლელად ჩავწეროთ ბრძანება:

```
>> Q = quad(@fun,0,pi/4)
Q =  0.0584
```

ან გამოვიყენოთ უსახელო ფუნქცია:

```
>> q=quad(@(x)x.^2.*cos(2*x),0,pi/4)
q =  0.0584
```

ორჯერადი ინტეგრალი გამოვთვალოთ Q = DBLQUAD(FUN,XMIN,XMAX,YMIN,YMAX): ან საზღვრების გარდა შეგვიძლია მივუთითოთ გამოთვლის სიზუსტე Q = DBLQUAD(FUN,XMIN,XMAX,YMIN,YMAX,TOL) (გაჩუმებით აიღებს 1.e-6 სიზუსტეს)

განვიხილოთ მაგალითი, გამოვთვალოთ ორჯერადი ინტეგრალი:

$$\int_{\pi}^{2\pi}\int_{0}^{\pi} x^2 \cos 2y + y^2 \cos 2x \, dxdy$$

შევადგინოთ M-ფუნქცია:

```
function r = funq(x,y)
       r = x.^2.*cos(2*y)+y.^2*cos(2*x);
   end
```

MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ ბრძანება:

```
>> Q = dblquad(@funq, pi, 2*pi, 0, pi)
```

Q = -2.0854e-008

ან სხვანაირად:

>> Q = dblquad(@(x,y)(x.^2.*cos(2*y)+y.^2*cos(2*x)), pi, 2*pi, 0, pi)

Q = -2.0854e-008

პოლინომების გამოთვლა

პოლინომის გამოთვლა შესაძლებელია ფუნქციით Y = POLYVAL(P,X). სადაც P – პოლინომის კოეფიციენტებია, X –ცვლადი. გამოითვლის Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X + P(N+1) პოლინომს. მაგალითად:

>> p=[1 3 4 -1];

>> x=5;

>> polyval(p,x)

ans = 219

თუ X ვექტორი ან მატრიცია მაშინ POLYVAL ფუნქცია ხარისხებს გამოითვლის ელემენტების მიხედვით, POLYVALM -ისაგან განსხვავებით, რომელიც X –მატრიცს აახარისხებს (მატრიცი უნდა იყოს კვადრატული). მაგალითად:

>> X=[1 3; -2 4]

X = 1 3

-2 4

>> polyval(p,X)

ans = 7 65

-5 127

>> polyvalm(p,X)

ans =

-47 102

-68 55

მიმდევრობის ჯამის გამოთვლა

symsum ფუნქციით შეგვიძლია გამოვითვალოთ როგორც სასრული ასევე უსასრულო მიმდევრობის ჯამი. მაგალითად გამოვთვალოთ ჯამი: 0+1+2+...+(k-1):

>> syms k

>> symsum(k)

ans = k^2/2 - k/2

ან

>> symsum(sym('k'))

ans =

k^2/2 - k/2

შეგვიძლია მივუთითოთ ინდექსი და საზღვრები. მაგალითად გამოვთვალოთ ჯამი:

$$s = \sum_{k=1}^{10} \frac{2k-1}{2^k}$$

>> syms k

>> symsum((2*k-1)/2^k,1,10)

ans = 3049/1024

უსასრულო ჯამის შემთხვევაში: $s = \sum_{k=1}^{\infty} \frac{2k-1}{2^k}$

>> syms k

>> symsum((2*k-1)/2^k,k,1,inf)

ans = 3

ამოცანები

1. მოცემულია A და B მატრიცები იპოვეთ მათი ჯამი და სხვაობა:

ა) $A = \begin{pmatrix} 3\frac{1}{2} & -1\frac{1}{3} & 6 \\ 2\frac{1}{3} & 1 & -2 \\ 0 & -\frac{2}{3} & -1 \end{pmatrix} \quad B = \begin{pmatrix} 2 & \frac{1}{2} & -\frac{1}{3} \\ 1 & 0 & \frac{1}{4} \\ \frac{2}{3} & -2 & -\frac{1}{2} \end{pmatrix}$

ბ) $A = \begin{pmatrix} c & 0 \\ 1 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 7 & d \\ c & 1 \end{pmatrix}$

2. ვიპოვოთ მოცემული მატრიცის ტრანსპონირებული მატრიცა:

ა) $A = \begin{pmatrix} 1 & 2 \\ -5 & 6 \end{pmatrix}$ ბ) $B = \begin{pmatrix} 1 & 2 & 3 \\ 5 & -1 & 4 \end{pmatrix}$

3. ვიპოვოთ A და B მატრიცების ნამრავლი AB და თუ შესაძლებელია BA:

ა) $A=\begin{pmatrix}1 & 1 & 0 & 3\\ 2 & 4 & 5 & 3\end{pmatrix}$ $B=\begin{pmatrix}4 & 2\\ -2 & 4\\ 1 & 3\\ 1 & -5\end{pmatrix}$

ბ) $A=\begin{pmatrix}\cos x & -\sin x\\ \sin x & \cos x\end{pmatrix}$ $B=\begin{pmatrix}\cos y & -\sin y\\ \sin y & \cos y\end{pmatrix}$ $A=\begin{pmatrix}8 & 7 & 2 & 10\\ -8 & 2 & 7 & 10\\ 4 & 4 & 4 & 5\\ 0 & 4 & -3 & 2\end{pmatrix}$

4. მოცემულია A, B და C მატრიცები იპოვეთ მათი ნამრავლები და შეამოწმეთ დისტრიბუტულობა:

$$A=\begin{pmatrix}1 & 2\end{pmatrix} \qquad B=\begin{pmatrix}3 & 4\\ 2 & 1\end{pmatrix} \qquad C=\begin{pmatrix}3 & 0 & 2\\ 5 & 1 & 0\end{pmatrix}$$

5. მოცემულია მატრიცა: $A=\begin{pmatrix}1 & 3\\ -2 & 5\end{pmatrix}$ ვიპოვოთ C მატრიცა, რომლის ელემენტები A მატრიცის შესაბამისი ელემენტების კვადრატების ტოლია.

6. მოცემულია A და B მატრიცები იპოვეთ: A^3 და B^2-4AB

$$A=\begin{pmatrix}1 & 3\\ 5 & -2\end{pmatrix} \qquad B=\begin{pmatrix}2 & 7\\ 4 & 6\end{pmatrix}$$

7. მოცემულია ორი A და B მატრიცა:

$$A=\begin{pmatrix}-1 & 1\\ 0 & 2\end{pmatrix} \qquad B=\begin{pmatrix}1 & 2\\ 0 & 2\end{pmatrix}$$

და $P(x)=2x^3+3x-4$ მრავალწევრი, იპოვეთ:

ა) P(A); ბ) P(B);

8. გამოთვალეთ შემდეგი მატრიცის რანგი:

ა) $A=\begin{pmatrix}1 & 2 & 5 & 6\\ 2 & 3 & -2 & -4\\ 5 & 8 & 1 & -2\end{pmatrix}$ ბ) $A=\begin{pmatrix}2 & -4 & 3 & 1 & 0\\ 1 & -2 & 1 & -4 & 2\\ 0 & 1 & -1 & 3 & 1\\ 4 & -7 & 4 & -4 & 5\end{pmatrix}$

9. მოცემული მატრიცა მიიყვანეთ ზედა (ქვედა) სამკუთხა ფორმაზე:

ა) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$ ბ) $\begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ 2 & -2 & 1 \end{pmatrix}$

გ)

10. ვიპოვოთ მოცემული მატრიცის შებრუნებული მატრიცა:

ა) $\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$ ბ) $\begin{pmatrix} 0 & 0 & 1 & -1 \\ 0 & 3 & 1 & 4 \\ 2 & 7 & 6 & -1 \\ 1 & 2 & 2 & -1 \end{pmatrix}$

11. ამოხსენით მატრიცული განტოლებები:

ა) $\begin{pmatrix} 1 & 2 & -3 \\ 3 & 2 & -4 \\ 2 & -1 & 0 \end{pmatrix} \cdot X = \begin{pmatrix} 1 & -3 & 0 \\ 10 & 2 & 7 \\ 10 & 7 & 8 \end{pmatrix}$ ბ) $X \cdot \begin{pmatrix} 3 & -2 \\ 5 & -4 \end{pmatrix} = \begin{pmatrix} 14 & 16 \\ 9 & 10 \end{pmatrix}$

12. ამოხსენით შემდეგი სისტემები:

ა) $\begin{cases} x + y - 2z = 6 \\ 2x + 3y - 7z = 16 \\ 5x + 2y + z = 16 \end{cases}$ ბ) $\begin{cases} ax + by + cz = a - b \\ bx + cy + az = b - c \\ cx + ay + bz = c - a \end{cases}$

გ) $\begin{cases} x_1 + x_2 + x_3 - 2x_4 = 4 \\ 2x_1 + 3x_2 - x_3 + x_4 = 1 \\ 4x_1 - 2x_2 - 6x_3 = 2 \\ 5x_1 + 4x_2 - 5x_3 + 2x_4 = 1 \end{cases}$ დ) $\begin{cases} 3x - y + z + 2u = 18 \\ 2x - 5y + t + u = -7 \\ x - t + 5u = 8 \\ 2y + z + t - u = 10 \\ x + y - 3z + t = 1 \end{cases}$

13. ვიპოვოთ მოცემული ვექტორის m, l და k ნორმები:

ა) $C = \begin{pmatrix} 1 \\ -3 \\ 0 \\ 5 \\ 9 \end{pmatrix}$ ბ) $D = \begin{pmatrix} -7 \\ 2 \\ 1 \\ 3 \\ -1 \end{pmatrix}$

14. მოცემული A მატრიცისათვის იპოვეთ კანონიკური m, l და k ნორმები:

$$A = \begin{pmatrix} 0 & 1 & 5 \\ 3 & 2 & 0 \\ 6 & 1 & 2 \end{pmatrix}$$

15. გამოთვალეთ შემდეგი მატრიცის ნულ-სივრცის ბაზისი:

ა) $A=\begin{pmatrix}11 & 2 & -8\\ 2 & 2 & 10\\ -8 & 10 & 5\end{pmatrix}$ ბ) $A=\begin{pmatrix}17 & -8 & 4\\ -8 & 17 & -4\\ 4 & -4 & 11\end{pmatrix}$

16. გამოთვალეთ X და Y ვექტორთა სკალარული ნამრავლი:

$$X=\begin{pmatrix}3\\ 2\\ 0\end{pmatrix} \quad Y=\begin{pmatrix}-7\\ 8\\ -4\end{pmatrix}$$

17. იპოვეთ მოცემული მატრიცის საკუთრივი რიცხვები და ვექტორები:

ა) $A=\begin{pmatrix}0 & 1 & 0\\ -4 & 4 & 0\\ -8 & 10 & 5\end{pmatrix}$ ბ) $A=\begin{pmatrix}7 & -12 & 6\\ 10 & -19 & 10\\ 12 & -24 & 13\end{pmatrix}$

იპოვეთ მოცემული მატრიცის საკუთრივი რიცხვები:

ა) $A=\begin{pmatrix}2 & -1 & 2\\ 5 & -3 & 3\\ -1 & 0 & -2\end{pmatrix}$

18. გამოთვალეთ შემდეგი ზღვრები:

ა) $\lim\limits_{x\to -1}\dfrac{x^3+1}{x^2+1}$ ბ) $\lim\limits_{x\to 2}\dfrac{x^2+1}{x+5}$

19. გამოთვალეთ შემდეგი ზღვრები:

ა) $\lim\limits_{x\to\infty}(\dfrac{x^3}{x^2+1}-x)$ ბ) $\lim\limits_{x\to\infty}\dfrac{3x^2-4x+11}{\sqrt{x^4-2}}$ გ) $\lim\limits_{x\to\infty}\dfrac{9x^2-1}{3x^2+x+2}$

20. გამოთვალეთ შემდეგი ზღვრები:

ა) $\lim\limits_{x\to\pi_+}\dfrac{\sqrt{1+\cos x}}{\sin x}$ ბ) $\lim\limits_{x\to 1_{\pm}} 2^{\frac{1}{x-1}}$ გ) $\lim\limits_{x\to 0^+}\dfrac{1}{x}$

21. გამოთვალეთ შემდეგი ზღვრები:

ა) $\lim\limits_{x\to 1}\dfrac{3x^4-4x^3+1}{x^3-x^2-x+1}$ ბ) $\lim\limits_{x\to 1}\dfrac{x^3-3x+2}{x^3-x^2-x+1}$ გ) $\lim\limits_{x\to 1}\dfrac{3x^4-4x^3+1}{x^3-x^2-x+1}$

22. გამოთვალეთ შემდეგი წარმოებულები:

ა) $y=\sqrt[3]{\dfrac{1+x^3}{1-x^3}}$ ბ) $f(x)=x(1+\sqrt{x})$ იპოვეთ $f'(4)$

23. გამოთვალეთ შემდეგი წარმოებულები:

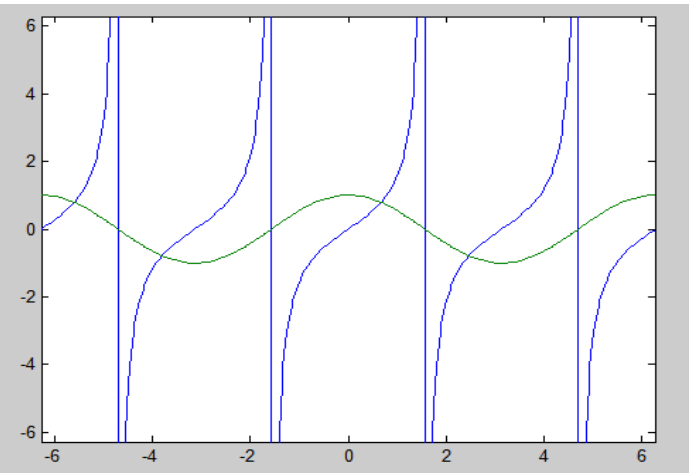
ა) $y = x^4 \ln x,\ y^{IV} = ?$ ბ) $y = x^2 \sin 2x,\ y''' = ?$

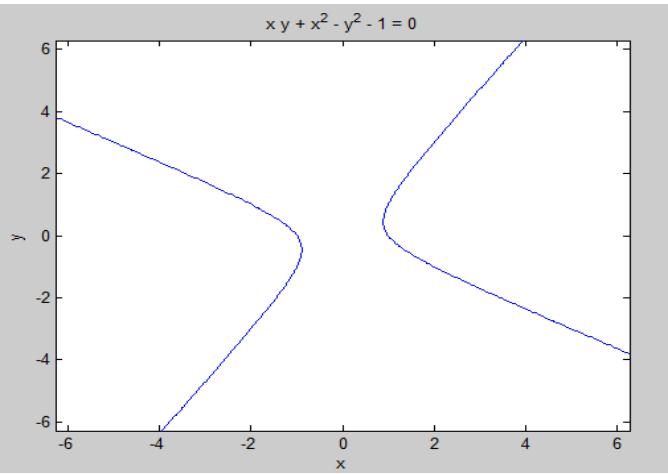
24. გამოთვალეთ შემდეგი ინტეგრალები

ფუნქციის გრაფიკები

ერთი ცვლადის ფუნქციის გრაფიკის ასაგებად MATLAB სისტემაში გამოიყენება ezplot ან fplot ფუნქცია, ან შეგვიძლია გამოვიყენოთ plot ფუნქცია. ezplot ან fplot ფუნქციები გამოიყენება მიმდინარე ან უსახელო ფუნქციებისათვის. მაგ:

>> fplot(@(x)[tan(x),cos(x)], 2*pi*[-1 1 -1 1])





>>ezplot('x.*y + x.^2 - y.^2 - 1')

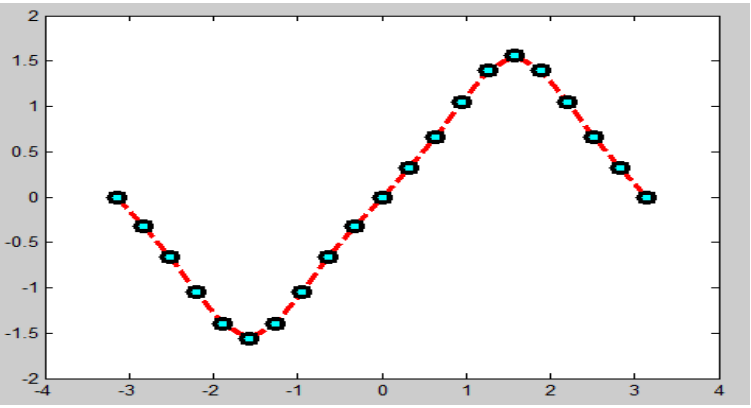
plot(x,y,s) ფუნქციაში მიეთითება წირის შესაბამისი x და y ვექტორები და წირის s მახასიათებლები, მახასიათებლები შეიძლება იყოს წირის ფერი, სტილი, მარკერი, სისქე. ცხრილში მოცემულია ეს მახასიათებლები:

წირის ფერის სპეც	ფერი	მაკერი	მაკერის ტიპი	ხაზის სტილი	
b	ლურჯი	.	წერტილი	-	უწყვეტი
g	მწვანე	0	წრეწირი	:	პუნქტირი
r	წითელი	x	x - მარკერი	-.	შტრიხ-პუნქტირი
c	ცისფერი	*	ვარსკვლავი	--	წყვეტილი
m	იისფერი	s	კვადრატი		
y	ყვითელი	d	რომბი		

k	შავი	^	სამკუთხედი		
w	თეთრი	v	სამკუთხედი		
		<	სამკუთხედი		
		>	სამკუთხედი		
		p	ხუთკუთხედი		
		h	ექვსკუთხედი		

გარდა ამისა ფერები შეგვიძლია მივუთითოთ ფერთა RGB პალიტრიდან color–ში. მნიშვნელობები აიღება [0,1] ინტერვალიდან. LineWidth –ში შეგვიძლია მივუთითოთ ხაზის სისქე, MarkerSize –ით მარკერის ზომა, MarkerFaceColor -ით მარკერის ფერი, MarkerEdgeColor -ით მარკერის კიდის ფერი. მაგ.:

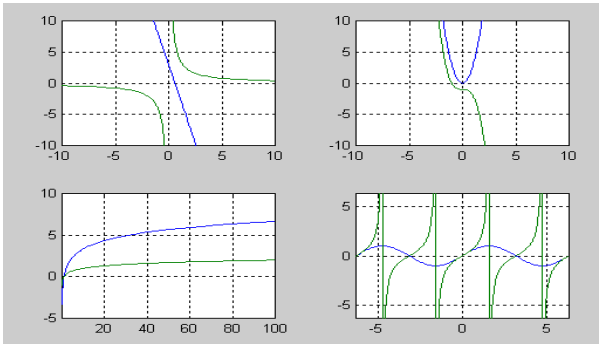
```
>>x = -pi:pi/10:pi;
   y = tan(sin(x));
    plot(x,y,'--ro','LineWidth',3,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','c',...
                'MarkerSize',8)
```



თუ გვინდა რომ რამდენიმე გრაფიკი ავაგოთ ერთ გრაფიკულ ფანჯარაში, მაშინ ეს ფანჯარა უნდა დავყოთ რამდენიმე ნაწილად subplot ფუნქციით. subplot ფუნქციით ფანჯარა დაიყოფა მატრიცულად. SUBPLOT(m,n,p)-ს m და n პარამეტრები მიუთითებენ მატრიცის სტრიქონებისა და სვეტების რაოდენობას. p-მიმდინარე გრაფიკის ნომერია, გრაფიკები დანომრვა ხდება სტრიქონულად და იწყება მარჯვენა ზედა კუთხიდან.

hold on ფუნქციით შეგვიძლია უზრუნველვყოთ ერთ გრაფიკულ ფანჯარაში რამდენიმე ფუნქციის გრაფიკის აგება.

მაგალითი: დეკარტის მართკუთხა კოორდინატთა სისტემაში ააგეთ მოცემული ფუნქციების გრაფიკები:



$$1)\ y=-5x+3;\quad 2)\ y=\frac{4}{x};\quad 3)\ y=3x^2;\quad 4)\ y=-x^3;$$

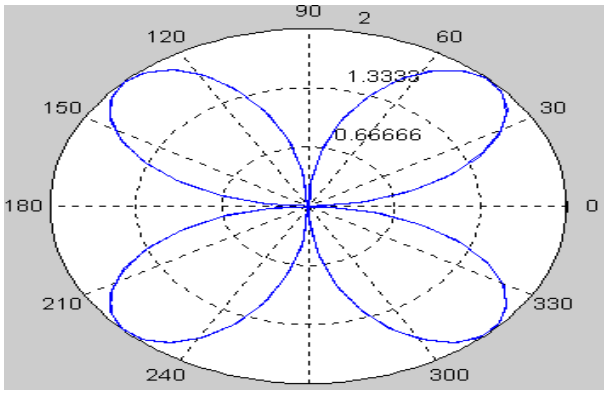
$$5)\ y=\log_2^x;\quad 6)\ y=\lg x;\quad 7)\ y=\sin x\quad 8)\ y=tg\ x$$

გრაფიკული ფანჯარა დავყოთ 4 საკოორდინატო სიბრტყედ და თითოეულში ავაგოთ ორ-ორი ფუნქციის გრაფიკი მოცემული რიგით. MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>> subplot(2,2,1);  fplot('[-5*x+3, 4/x]',[-10 10 -10 10])
>> subplot(2,2,2);  fplot('[3*x.^2, -x.^3]',[-10 10 -10 10])
>> subplot(2,2,3);  fplot('[log2(x), log10(x)]',[0.01,10])
>> subplot(2,2,1);  fplot('[sin(x), tan(x)]', 2*pi*[-1 1 -1 1])
```

პოლარულ კოორდინატებში მოცემული ფუნქციის გრაფიკი შეიძლება ავაგოთ polar ფუნქციით.

მაგალითი: ააგეთ პოლარულ კოორდინატებში მოცემული ფუნქციის გრაფიკი: r=2sinφ



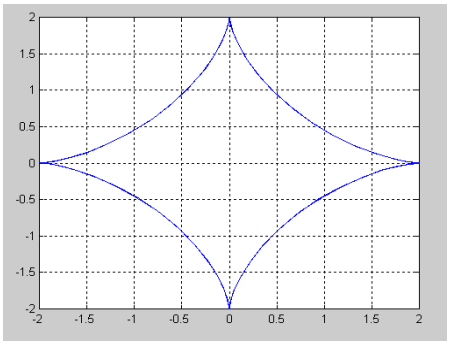
MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>>  t=-2*pi:0.01:2*pi;
>> r=2*sin(3*t);
>> polar(t,r)
```

პარამეტრული განტოლებებით მოცემული ფუნქციის გრაფიკები შეიძლება ავაგოთ plot-ით ან meshgrid-ით და contour ფუნქციებით.

მაგალითი: ააგეთ პარამეტრული განტოლებებით მოცემული ფუნქციის გრაფიკი:

$$\begin{cases} x = 2\cos^3 t \\ y = 2\sin^3 t \end{cases}$$



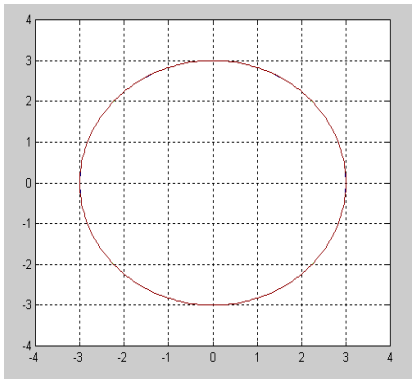
MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>> t=-2*pi:.1:2*pi;
>> x=2*cos(t).^3;
>> y=2*sin(t).^3;
>> plot(x,y); grid
```

მეორე რიგის წირების ასაგებად ვისარგებლოთ meshgrid და contour ფუნქციებით. ფუნქციის გრაფიკები სივრცეში შეიძლება ავაგოთ plot3, mesh,cotour, slice, surf ფუნქციებით.

მაგალითი: ააგეთ მოცემული ორი ცვლადის ფუნქციის გრაფიკი:

$$x^2 + y^2 = 9$$

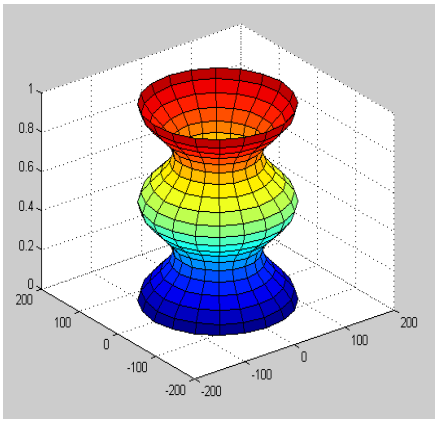


MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>> [x,y]=meshgrid([-4:0.1:4],[-4:0.1:4]);
>> z=x.^2+y.^2-9;
>> contour(x,y,z,[-0.001 0 0.001])
>> grid
```

მაგალითი: ააგეთ მოცემული ფუნქციის გრაფიკი სივრცეში:

$$\frac{x^2}{3^2} + \frac{y^2}{4^2} - \frac{z^2}{2^2} = 1$$ ((ცალკალთა ჰიპერბოლოიდი)



ამოხსნის მსვლელობა: ჯერ ეს ფორმულა გარდავქმნათ პარამეტრულად, ჩავსვათ x=3cost და y=3sint ელემენტარული გარდაქმნებით მივიღებთ:

$$z = 8(16\cos^2 t + 9\sin^2 t - \frac{1}{16})$$

MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>> t=0:pi/10:2*pi;
>> [X,Y,Z]=cylinder(-8*(16*cos(t).^2+9*sin(t).^2-1/16));
>> surf(X,Y,Z)
```

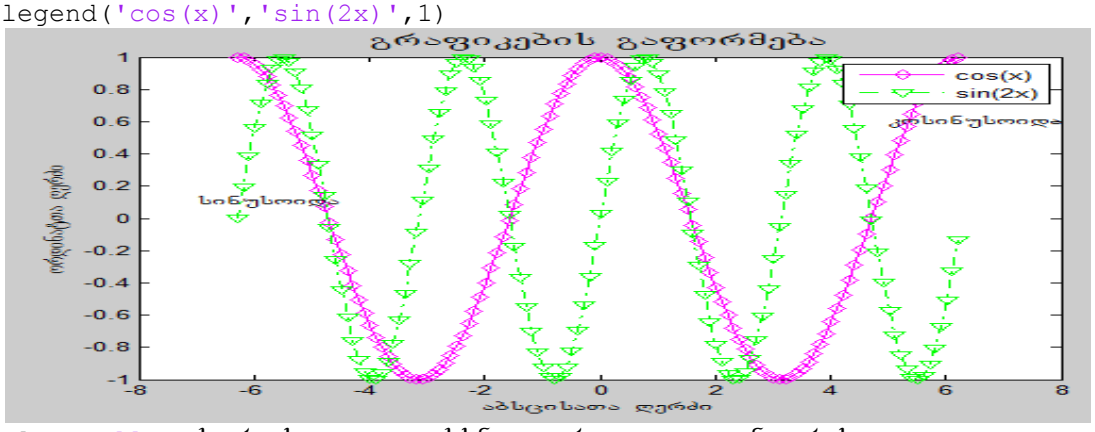
გრაფიკების გაფორმება

როგორც ზემოთ აღვნიშნეთ გრაფიკების აგებისას შეგვიძლია ავარჩიოთ წირის ფერი, მარკერის სტილი, ხაზის ტიპი.

ასევე შესაძლებელია გრაფიკის სათაურის გამოტანა title-ით, ღერძების მახლობლად წარწერების გამოტანა: xlabel, ylabel და zlabel-ით. text-ით შეგვიძლია წარწერა გამოვიტანოთ გრაფიკის არეზე. fontname–ით ავირჩევთ შრიფტს, fontsize–ით ავირჩევთ შრიფტის ზომას. თუ ერთ გრაფიკულ სიბრტყეზე რამდენიმე წირია აგებული, მაშინ მათი დახასიათება შეგვიძლია გამოვიტანოთ legend–ით. legend–ის პარამეტრებია გრაფიკზე არსებული წირების სახელები და მისი განთავსების ადგილი (განვათავსების პარამეტრის ცვლით -1, 0, 1, 2, 3, 4 წანაწერი დაიკავებს სხვადასხვა მდებარეობას).

მაგალითად:

```
close all
x=-2*pi:.1:2*pi;
y=cos(x);
f=sin(2*x);
plot(x,y,'md-',x,f,'gv-.')
title('grafikebis gaformeba', 'fontname','acadnusx','fontsize',14)
text(5,0.6,'kosinusoida','fontname','acadnusx')
text(-7,0.1,'sinusoida','fontname','acadnusx')
xlabel('abscisaTa RerZi','fontname','acadnusx')
ylabel('ordinatTa RerZi','fontname','acadnusx')
legend('cos(x)','sin(2x)',1)
```



`close all`-დახურავს ყველა გახსნილ გრაფიკულ ფანჯარას.

განტოლებათა სისტემის ამოხსნა გრაფიკულად

ამოხსენით შემდეგი განტოლებათა სისტემა:

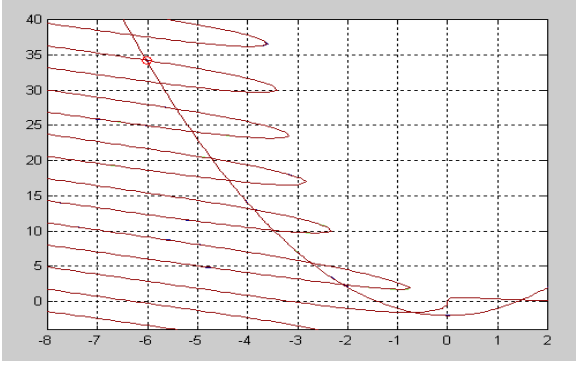
$$\begin{cases} \sin(x+y) - e^x \cdot y = 0 \\ x^2 - y = 0 \end{cases}$$

ამოვხსნათ ჯერ solve ფუნქციით, შემდეგ კი გრაფიკულად. MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>>syms x y
>>s=solve('sin(x+y)-exp(x)*y=0','x^2-y=2')
s =
   x: [1x1 sym]
   y: [1x1 sym]
s.y
ans =
34.208227234306296508646214438330
s.x
ans =
-6.017327250059306564109729711790 5
```

შენიშვნა: ამ შემთხვევაში მოიძებნა მხოლოდ ერთი ამონახსნი ამოხსნათა სიმრავლიდან. ახლა ავაგოთ ამ ფუნქციების თანაკვეთა OXY სიბრტყეზე. MATLAB-ის ბრძანებათა სტრიქონიდან შევასრულოთ შემდეგი ოპერატორები:

```
>>[X,Y]=meshgrid([-8:0.1:2], [-4:0.5:40]);
>>Z1=sin(X+Y)-exp(X).*Y;
>>Z2=x.^2-Y-2;
>>contour(X,Y,Z1,[-0.001 0 0.001]), hold on
>>contour(X,Y,Z2,[-0.001 0 0.001])
>>plot(-6.017, 34.208, 'or') grid
```



ამოცანები

1. დეკარტის მართკუთხა კოორდინატთა სისტემაში ერთ სიბრტყეზე ააგეთ

მოცემული ფუნქციების გრაფიკები:

1) $y = x^3 - 3x$ 2) $y = \dfrac{x^2 + 1}{x}$

3) $y = \dfrac{x^2}{\sqrt{x^2 - 1}}$ 4) $y = e^{-x^2}$

2. დეკარტის მართკუთხა კოორდინატთა სისტემაში ააგეთ მოცემული ფუნქციების გრაფიკები:

ა) $y = \dfrac{2 - 4x^2}{1 - 4x^2}$ ბ) $y = 2\sin x + \cos x$

3. ააგეთ პოლარულ კოორდინატებში მოცემული ფუნქციის გრაფიკი:

ა) $r = \dfrac{1}{\sqrt{2}} \cdot \dfrac{1}{\cos(\phi + \dfrac{\pi}{4})}$ ბ) $r\sin\varphi = 1$

4. ააგეთ პარამეტრული განტოლებებით მოცემული ფუნქციის გრაფიკი:

ა) $\begin{cases} x = -1 + 2\cos t \\ y = 3 + 2\sin t \end{cases}, \quad t \in [0, 2\pi]$ ბ) $\begin{cases} x = -1 + 2t \\ y = 2 - t \end{cases}, \quad t \in (-\infty, +\infty)$

5. ააგეთ მოცემული ორი ცვლადის ფუნქციის გრაფიკი:

ა) $\dfrac{x^2}{16} + \dfrac{y^2}{64} = 1$ ბ) $x^2 y - 7xy + 10y = 0$

6. ააგეთ მოცემული ფუნქციის გრაფიკი სივრცეში:

ა) $\dfrac{x^2}{9} + \dfrac{y^2}{4} + \dfrac{z^2}{25} = 1$ ბ) $x^2 + y^2 - z^2 = -1$ გ) $Z = \dfrac{\sin(xy)}{xy}$

7. ამოხსენით შემდეგი განტოლებათა სისტემა:

ა) $\begin{cases} y = \dfrac{(x^2 - 5)^3}{125} \\ y = (x^2 + 1)e^{\frac{-x^2}{2}} \end{cases}$ ბ) $\begin{cases} y = \dfrac{\cos(x - y)}{e^x} \\ y = tg\, x^2 + 2 \end{cases}$