

## 13 სიმბოლური სტრიქონი, სიმბოლური გამონასხის დამუშავება

### შესავალი

თუმცა MATLAB –ის ძირითადი დანიშნულება რიცხვითი გამოთვლებია, ზოგჯერ საჭიროა ტექსტური მანიპულაციები. მაგალითად გრაფიკულ გამოსახულებაზე სათაურის გაკეთება, ან ღერძების სახელების დაწერა. ამისათვის MATLAB -ს გააჩნია სტრიქონი, რომელიც ასოთ გამოსახულებას შეიცავს, ანუ სიმბოლური სტრიქონი .

### 13.1 სტრიქონის კონსტრუქცია

MATLAB სიმბოლური სტრიქონი ASCII სიდიდეების ერთობლიობაა. მაგალითად:

```
>>text = 'This is a character string'
text =
This is a character string
>>size(text)
Ans =
26
>> whos
Name          Size          Bytes          Class
ans           1x26          16             double array
text          1x26          52             char array

Grand total is 28 elements using 68 bytes
```

### ASCII კოდი

სტრიქონის ყოველი სიმბოლო წარმოადგენს ერთ ელემენტს მასივში, რომელიც საჭიროებს 2 ბაიტ მესხიერებას. (რიცხვითი გამოსახულება მოითხოვს 8 ბაიტს.) ASCII კოდი 'A'- 'Z'

სიმბოლოთათვის არის ერთმანეთის მომდევნო მთელი რიცხვები 65-დან 90-მდე, ხოლო 'a' – 'z' 97-დან 122-მდე. MATLAB ფუნქცია **abs** გვაძლევს ASCII კოდს სტრიქონისათვის.

```
>> text = 'This is a character string'
text =
This is a character string
>> d = abs(text)
d =
Columns 1 through 12
84 104 105 115 32 105 115 32 97 32 99 104
Columns 13 through 24
97 114 97 99 16 101 114 32 115 116 114 105
Columns 25 through 26
110 103
```

ფუნქცია **char** ასრულებს შებრუნებულ გარდაქმნას, ASCII კოდის მიხედვით აღადგენს სტრიქონს:

```
>> char(d)
ans =
This is a character string
```

სტრიქონსა და ASCII კოდს შორის ამ დამოკიდებულების საფუძველზე მივიღებთ:

```
>> alpha = abs('a'):abs('z')
>> disp(char(alpha))
abcdefghijklmnopqrstuvwxyz
```

სტრიქონი იგივე მასივია, მაგრამ სიმბოლოთა მასივი, ამიტომ მასზე შესაძლებელია მასიური ოპერაციების წარმოება:

```
>> text = 'This is a character string';
>> u = text(11:19)
u =
character
```

ისევე როგორც მატრიცა, სტრიქონი მასივიც შეიძლება შეიცავდეს მრავალ სტრიქონს, მაგრამ ყოველი სტრიქონი სვეტების თანაბარ რაოდენობას უნდა შეიცავდეს. მაგალითად:

```
v = ['Character strings having more than'
'one row must have the same number '
'of columns - just like matrices  ']
```

მივიღებთ:

```
v =

Character strings having more than
one row must have the same number
of columns - just like matrices

>> size(v)
ans =
```

3 34

რადგან სტრიქონი იგივე მასივია, შეგვიძლია ორო სტრიქონი გავართიანოთ კვადრატული ფრჩხილების საშუალებით:

```
>> today = 'May';
>> today = [today, ' 18']
today =
May 18
```

განვიხილოთ MATLAB-ის რამდენიმე ფუნქცია სტრიქონისათვის:

<code>char(x)</code>	<code>x</code> მასივს, რომლის ელემენტებიც მთელი დადებითი რიცხვებია და წარმოადგენს ასოითი სიმბოლოს კოდებს გარდაქმნის ასოით სიმბოლოთა მასივად.
<code>int2str(x)</code>	<code>x</code> მატრიცის ელემენტებს დაამრგვალებს მთელამდე და შედეგად მიღებულ მასივს გარდაქმნის სტრიქონ მატრიცად
<code>num2str(x)</code>	<code>x</code> რიცხვით მატრიცას გარდაქმნის სტრიქონად. ეს ბრძანება სასარგებლოა გრაფიკზე აღნიშვნების გასაკეთებლად (როგორც <code>title</code> , <code>xlabel</code> , <code>ylabel</code> , <code>text</code> ბრძანებათა არგუმენტი)
<code>str2num(s)</code>	<code>s</code> სტრიქონს, რომელიც შეიცავს რიცხვით სიდიდეთა გამოსახულებას ASCII ფორმატში, გარდაქმნის შესაბამის რიცხვით მნიშვნელობად. სტრიქონი <code>s</code> შეიძლება შეიცავდეს ციფრს, ათწილადურ მძიმეს, <code>+</code> ან <code>-</code> ნიშანს, <code>'e'</code> და <code>'d'</code> სიმბოლოებს – მანტისასა და ექსპონენტას გამყოფის სახით და <code>i</code> – კომპლექსური რიცხვებისათვის.

მაგალითები:

```
>> x=abs('string')
x =

    115    116    114    105    110    103

>> char(x)
ans =
string
```

**num2str** საილუსტრაციოდ განვიხილოთ შემდეგი ბრძანებები:

როგორც ვიცით, ცვლადს - `pi` MATLAB აღიქვამს როგორც  $\pi$  მნიშვნელობას – 3.1416

```
>> ss=num2str(pi)
ss =
3.1416
```

`ss` ამ შემთხვევაში არის არა კონკრეტული რიცხვი 3.1416, არამედ სიმბოლოთა მასივი (`char array`), რომლის ელემენტებიც ASCII კოდებს წარმოადგენს. მართლაც:

```
>> size(ss)
ans =
1 6
```

```
>> abs(ss)
```

```
ans =
```

```
51 46 49 52 49 54
```

ახლა გამოვიყენოთ ბრძანება , რომელიც სტრიქონს ss გარდაქმნის რიცხვით სიდიდედ:

```
pp=str2num(ss)
```

```
pp =
```

```
3.1416
```

pp უკვე კონკრეტული რიცხვია (double array)– 3.1416.

```
size(pp)
```

```
ans =
```

```
1
```

განვიხილოთ კიდევ ერთი მაგალითი:

MATLAB შემდეგ ბრძანებებს:

```
tg = 2.2774;
```

```
xg = 144.6364;
```

```
disp(['time of flight: ' num2str(tg) ' s'])
```

```
disp(['distance traveled : ' num2str(xg) ' ft'])
```

შედეგად მოჰყვება:

```
>> time of flight: 2.2774 s
```

```
distance traveled : 144.6364 ft
```

**disp** ბრძანების პირველი არგუმენტი ჩვეულებრივი სიმბოლური სტრიქონია, მეორე – ბრძანებაა, რომელიც რიცხვით გამოსახულებას სიმბოლურ სტრიქონად გარდაქმნის, და მესამე – ისევ ჩვეულებრივი სიმბოლური სტრიქონი.

MATLAB ფუნქციები სიმბოლოთა სტრიქონისათვის:

blanks(n)	გვამღებს სტრიქონს , რომელიც n სიმბოლოს ნაცვლად n ცარიელ პოზიციას შეიცავს.
deblank(s)	თუ სტრიქონი ბოლოვდება განმეორებადი ცარიელი პოზიციით, ეს ბრძანება შეკვეცს მას პირველივე სიმბოლომდე გაუშვებს s სტრიქონს როგორც MATLAB გამოსახულებას ან ბრძანებას.
eval(s1,s2)	საშუალებას იძლევა შეცდომა ვიპოვოთ. გაუშვებს s1 შესაბამის ბრძანებას და გვამღებს შედეგს, თუ ბრძანება წარმატებით შესრულდა, ხოლო თუ ბრძანების შესრულება შეცდომას იძლევა, უშვებს s2 – შესაბამის ბრძანებას.
findstr(s1,s2)	პოულობს ერთ სტრიქონს მეორის შიგნით
ischar(s)	გვამღებს 1, თუ s სიმბოლური მასივია, სხვა შემთხვევაში –0

isletter(s)	გვაძლევს 1-ს s სტრიქონის ყოველი ელემენტისათვის, რომელიც ასოით გამოსახულებას წარმოადგენს და 0 სხვა შემთხვევაში.
isspace(s)	გვაძლევს 1-ს s სტრიქონის იმ ელემენტებისათვის, რომლებიც წარმოადგენს ცარიელ პოზიციას(space, tabs, newlines, carriage returns), 0 – სხვა შემთხვევაში

## 13.2 დრო და თარიღი

უკვე გავეცანით MATLAB ბრძანებას **date**, რომელიც გვაძლევს თარიღს, **clock** – გვაძლევს დროის მნიშვნელობას. უფრო დაწვრილებითი ინფორმაციის მისაღებად ამ საკითხთან დაკავშირებით შეგიძლიათ ისარგებლოთ MATLAB-ის **help** ბრძანებით. გავეცნოთ ზოგიერთ მათგანს.

```
>> [d,s]=weekday('9/15/2006')
```

გვაძლევს კვირის რა დღეა მითითებული თარიღი:

```
d = 6
s = Fri
```

```
>> calendar('9/15/2006')
```

რომელიც გვაძლევს:

```

                Sep 2006
  S      M      Tu      W      Th      F      S
  0      0      0      0      0      1      2
  3      4      5      6      7      8      9
 10     11     12     13     14     15     16
 17     18     19     20     21     22     23
 24     25     26     27     28     29     30
 0      0      0      0      0      0      0

```

ბრძანება tic ჩართავს ტაიმერს ხოლო toc, გამორთავს მას და მოგვცემს დროს, რომელიც გავიდა ტაიმერის ჩართვის, ანუ tic ბრძანების შემდეგ. ეს ბრძანება სასარგებლოა, როცა გვსურს დავითვალოთ რა დროს ანდომებს კომპიუტერი ჩვენს მიერ დაწერილი პროგრამის ამათუიმ ოპერაციის შესრულებას.

## 13.3 ბაზური გარდაქმნები და ბიტ ოპერაციები

MATLAB აქვს შესაძლებლობა გადაიყვანოს რიცხვები ორობითიდან ათობითში და ან რომელიმე სხვა ბაზურ სისტემაში სიმბოლური სტრიქონის ფორმით.

dec2bin(d)	გვაძლევს d –ს ორობით წარმოდგენას როგორც სტრიქონს. d უნდა იყოს არაუარყოფითი მთელი რიცხვი $<2^{52}$ .
bin2dec(b)	ახდენს b ორობითი სტრიქონის ინტერპრეტირებას და გვაძლევს შესაბამის ათობით რიცხვს

მაგალითად:

```
>> d=32;
>> dec2bin(d)

ans =

100000;

>> bin2dec('100000')

ans =

32
```

MATLAB –ს გააჩნია ფუნქციები ლოგიკური ოპერაციების განსახორციელებლად რიცხვების შესაბამის ორობით სტრიქონებს შორის. (a, b, c მთელი, არაუარყოფითი რიცხვებია).

c=bitand(a,b)	ბიტური and ორ არგუმენტს შორის
c=bitor(a,b)	ბიტური or a და b არგუმენტს შორის
c=bitcmp(a,N)	a ათობითი მთელი რიცხვის N ბიტთან ორობით წარმოდგენაში შეცვლის 0 1-ით, ხოლო 1 0-ით და მოგვცემს შესაბამის ათობით რიცხვს
c=bitshift(a,N)	მოგვცემს მთელ ათობით რიცხვს, რომელიც წარმოადგენს a რიცხვის ორობითი წარმოდგენის N ბიტით წანაცვლების შედეგს. თუ $N > 0$ , ეს ოპერაცია იგივეა, რაც გამრავლება $2^N$ , თუ $N < 0$ , - იგივეა, რაც გაყოფა $2^N$ -ზე

მაგალითად:

```
>> bitand(13,27)    ბრძანება მოგვცემს:

ans = 9
```

მართლაც რიცხვი 13 ორობითი წარმოდგენა 5 ბიტთან სისტემაში ასეთია :

```
>> dec2bin(13,5)
ans = 01101
```

ხოლო - 27-ის:

```
>> dec2bin(27,5)
ans = 11011
```

თუ შევადარებთ ერთმანეთს ორ ორობით რიცხვს: 01101 და 11011 **and** შედარების ოპერატორით, მივიღებთ 01001, რასაც შეესაბამება ათობითი რიცხვი 9. მართლაც:

```
>> bin2dec('01001')
ans = 9
```

ახლა ვნახოთ **bitcmp** ფუნქციის მოქმედების შედეგი:

```
>>bitcmp(9,5) brZanebis Sedegad miviRebT:
ans = 22
```

რადგან ათობითი რიცხვის 9 ორობითი წარმოდგენა 5 ბიტთან სისტემაში = 01001, **bitcmp** ფუნქცია შეცვლის 0 1-ით, ხოლო 1- 0-ით და მიიღება 10110, რასაც შეესაბამება ათობითი რიცხვი 22.

შევამოწმოთ:

```
>> dec2bin(9,5)
ans = 01001
```

```
>>dec2bin(22,5)
ans =10110
```

```
>>bitShift(9,2)
ans = 36
```

მართლაც 9 ათობითი რიცხვის ორობითი წარმოდგენაა 1001, **bitshift** ბრძანება დაუმატებს მას ორ ნულს მარჯვნივ და მიიღება ორობითი რიცხვი 100100, რაც შეესაბამება ათობით რიცხვს 36, ეს ოპერაცია იგივეა, რაც 9 გავამრავლოთ  $2^2$  ( $9*2^2=36$ ).

### 13.4 სიმბოლური ინფორმაციის დამუშავება

აქმდე ჩვენ MATLAB –ს განვიხილავდით როგორც საშუალებას რიცხვითი გამოთვლების საწარმოებლად. ასევე მოგაწოდეთ გარკვეული ინფორმაცია ტექსტის მანიპულირებისათვის. MATLAB –ის საშუალებით შესაძლებელია ვაწარმოოთ მოქმედებები ჩვეულებრივ ალგებრულ გამოსახულებებზე.

უფრო დაწვრილებითი ინფორმაციისათვის აკრიფეთ ბრძანებათა ფანჯარაში **help symbolic**. ასევე შესაძლებელია ამ საკითხთან დაკავშირებული სადემონსტრაციო მასალის დათვალიერება ბრძანებით **symintro**.

#### სიმბოლური გამოსახულება და ალგებრა

თუ გვსურს MATLAB-ის საშუალებით სიმბოლური ინფორმაციის დამუშავება, პირველ რიგში უნდა განვმარტოთ (გამოვაცხადოთ) საჭირო მუდმივები და ცვლადები როგორც სიმბოლური ობიექტები. მაგალითად:

```
>>syms x y
```

სიმბოლური მუდმივების გამოსაცხადებლად გამოიყენება ფუნქცია **sym**. მისი არგუმენტია სიმბოლური სტრიქონი, რომელიც შეიცავს სპეციალური ცვლადის სახელს, რიცხვით გამოსახულებას ან ფუნქციის დასახელებას.

```
>> pi = sym('pi');
>> delta = sym('1/10');
>>sqrt2 = sym('sqrt(2) ');
```

თუ სიმბოლური მუდმივი pi ასე განვსაზღვრეთ, იგი მოცემულ სამუშაო სივრცეში შეცვლის pi –ს რიცხვით მნიშვნელობას. სიმბოლური ობიექტების ტიპი განისაზღვრება როგორც **symbolic objects**.

თუ ზემოხსენებული ბრძანებების შემდეგ მივცემთ ბრძანებას **whos**, მივიღებთ:

```
>> whos
Name                Size                Bytes  Class

delta              1x1                132   sym object
pi                 1x1                128   sym object
sqrt2             1x1                138   sym object
x                 1x1                126   sym object
y                 1x1                126   sym object
```

Grand total is 20 elements using 650 bytes

### სიმბოლური გამოსახულება

სიმბოლური ცვლადი შეიძლება გამოვიყენოთ სიმბოლურ გამოსახულებაში როგორც ცვლადი ან ფუნქციის არგუმენტი. არითმეტიკული ოპერატორები  $+$   $-$   $*$   $/$  და MATLAB-ის ფუნქციებიც გამოვიყენება იგივე წესის მიხედვით, როგორც ეს ხდებოდა რიცხვითი გამოთვლების წარმოებისას. მაგალითად:

```
>> syms s t A
>> f = s^2 + 4*s + 5
f =
s^2+4*s+5
>> g = s + 2
g =
s+2

>> h = f*g
h =
(s^2+4*s+5)*(s+2)
>> z = exp(-s*t)
z =
exp(-s*t)
>> y = A*exp(-s*t)
y =
A*exp(-s*t)
```

პირველ რიგში ვაცხადებთ ცვლადებს  $t$ ,  $s$  და  $A$ , შემდეგ ვწერთ სიმბოლურ გამოსახულებებს ახალი ცვლადების  $f$ ,  $g$ ,  $h$  და  $y$  შესაქმნელად. ახლადშექმნილი ცვლადებიც სიმბოლური ცვლადებია და არა რიცხვითი გამოსახულებები.

საზოგადოდ MATLAB-ში თუ არაფერს მივეუთითებთ,  $x$  არის დამოუკიდებელი ცვლადის სახელი, მაგრამ ეს არ ნიშნავს იმას, რომ სხვა სიმბოლო არ შეგვიძლია გამოვიყენოთ დამოუკიდებელი ცვლადის აღსანიშნავად. დამოუკიდებელი ცვლადის მოსაძებნად სიმბოლურ გამოსახულებაში MATLAB -ს გააჩნია ფუნქცია **findsym**

**findsym(S)**

$S$  სიმბოლურ გამოსახულებაში ან მატრიცაში პოულობს სიმბოლურ ცვლადებს და გვაძლევს მათ, როგორც სიმბოლურ სტრიქონს, ანბანის მიხედვით დალაგებულს და მძიმეებით



გამოყოფილს. თუ  $S$  არ შეიცავს არცერთ სიმბოლურ ცვლადს, გვაძლევს ცარიელ სტრიქონს.

ჩვენს შემთხვევაში :

```
>> findsym(f)
ans =
s
>> findsym(z)
ans =
A, s, t
```

შეგვიძლია შევქმნათ სიმბოლური მატრიცა:

```
>> n = 3;
>> syms x
>> B = x.^((0:n)')*(0:n))

B =

[1, 1, 1, 1]
[1, x, x^2, x^3]
[1, x^2, x^4, x^6]
[1, x^3, x^6, x^9]
```

როგორ შევასრულოთ სიმბოლური პოლინომის ალგებრული გარდაქმნები:

<code>expand(S)</code>	სიმბოლურ გამოსახულებაში $S$ შეასრულებს ალგებრულ მოქმედებებს, შეკრებს მსგავს წევრებს და წარმოგვიდგენს პოლინომის სახით.
<code>factor(S)</code>	$S$ პოლინომს დაშლის მამრავლებად.
<code>simplify(S)</code>	გაამარტივებს სიმბოლურ გამოსახულებას $S$
<code>[n,d]=numden(S)</code>	მოგვცემს რაციონალურ წილადად დაშლილი $S$ სიმბოლური გამოსახულების მრიცხველსა და მნიშვნელს
<code>subs(S,old,new)</code>	$S$ სიმბოლურ გამოსახულებაში ცვლადს <code>old</code> შეცვლის ცვლადით <code>new</code>

მაგალითები:

- შეკრება

```
>> syms s
>> A = s^4 - 3*s^3 - s + 2;
>> B = 4*s^3 - 2*s^2 + 5*s - 16;
>> C = A + B
C =
s^4+s^3+4*s-14-2*s^2
```

- სკალარული გამრავლება

```
>> syms s
>> A = s^4 - 3*s^3 - s + 2;
>> C = 3*A
C =
3*s^4 - 9*s^3 - 3*s + 6
```

- გამრავლება

```
>> syms s
>> A = s+2;
>> B = s+3;
>> C = A*B
C =
(s+2)*(s+3)
>> C = expand(C)
C =
s^2+5*s+6
```

- მამრავლებად დაშლა

```
>> syms s
>> D = s^2 + 6*s + 9;
>> D = factor(D)
D =
(s+3)^2
>> P = s^3 - 2*s^2 - 3*s + 10;
>> P = factor(P)
P =
(s+2)*(s^2-4*s+5)
```

- გაერთმნიშვნელებიანება

განვიხილოთ გამოსახულება

$$H(s) = -\frac{1/6}{s+3} - \frac{1/2}{s+1} + \frac{2/3}{s}$$

სამი შესაკრების გაერთმნიშვნელებიანება შესაძლებელია შემდეგნაირად:

```
>> syms s
>> H = -(1/6)/(s+3) - (1/2)/(s+1) + (2/3)/s;
>> [N,D] = numden(H)
N =
s+2
D =
(s+3)*(s+1)*s
>> D = expand(D)
D =
s^3+4*s^2+3*s
```

მაშასადამე შეგვიძლია  $H(s)$  ასე წარმოვადგინოთ:

$$H(s) = \frac{s+2}{s^3+4s^2+3s}$$

- შეკვეცა

ვთქვათ გვაქვს სიმბოლური გამოსახულება:

$$H(s) = \frac{s^3+2s^2+5s+10}{s^2+5}$$

MATLAB მარტივად შეკვეცს ამ წილადს და საბოლოოდ მოგვცემს:

```
>> syms s
>> H = (s^3 + 2*s^2 + 5*s + 10)/(s^2 + 5);
>> H = simplify(H)
H =
s+2
```

თუ წილადის მნიშვნელს დავშლით მამრავლებად, ვნახავთ, რატომ მივიღეთ  $s+2$ .

```
>> factor(s^3 + 2*s^2 + 5*s + 10)
ans =
(s+2)*(s^2+5)
```

საბოლოოდ გვექნება:

$$H(s) = s + 2$$

- ცვლადის ჩანაცვლება სხვა ცვლადით

განვიხილოთ პოლინომთა თანაფარდობა:

$$H(s) = \frac{s+3}{s^2+6s+8}$$

განვსაზღვროთ მეორე გამოსახულება

$$G(s) = H(s)|_{s=s+2}$$

ჩანაცვლოთ  $s$  ცვლადი  $s+2$  –ით MATLAB საშუალებით:

```
>> syms s
>> H = (s+3)/(s^2 + 6*s + 8);
>> G = subs(H, s, s+2)
G =
(s+5)/((s+2)^2+6*s+20)
>> G = collect(G)
G =
(s+5)/(s^2+10*s+24)
```

მივიღებთ გამოსახულებას:

$$H(s) = \frac{s+5}{s^2+10s+24}$$

განვიხილოთ კიდევ რამდენიმე სასარგებლო ბრძანება:

sym2poly(P)	P სიმბოლური პოლინომისათვის გვაძლევს ვექტორს, რომლის ელემენტებია ამ პოლინომის კოეფიციენტები
poly2sym(p)	p ვექტორზე დაყრდნობით, რომლის ელემენტებიც პოლინომის კოეფიციენტებს წარმოადგენს, გვაძლევს შესაბამის სიმბოლურ პოლინომს. ცვლადს მიანიჭებს სახელს x.

მაგალითად განვიხილოთ პოლინომი:

$$A(s) = s^3 + 4s^2 - 7s - 10$$

```
>> a = [1 4 -7 -10];
>> A = poly2sym(a,s)
A =
s^3+4*s^2-7*s-10
```

პოლინომისათვის:

$$B(s) = 4s^3 - 2s^2 + 5s - 16$$

```
>> syms s
>> B = 4*s^3 - 2*s^2 + 5*s - 16;
>> b = sym2poly(B)
b =
4    -2    5    -16
```

### სიმბოლური გამოსახულების ფორმა

MATLAB-ს შედეგად მიღებული გამოსახულება ყოველთვის არ გამოჰყავს ჩვენთვის სასურველი ფორმით, მაგალითად, გარკვეული ოპერაციების შესრულების შემდეგ შედეგად შესაძლოა მივიღოთ  $1/a*b$ , და არა  $b/a$ .

### ტრიგონომეტრიული სიმბოლური გამოსახულება

განვიხილოთ შემდეგი ბრძანებები:

```
>> syms theta phi
>> A = sin(theta + phi)
A =
sin(theta+phi)
>> A = expand(A)
A =
sin(theta)*cos(phi)+cos(theta)*sin(phi)
>> B = cos(2*theta)
B =
cos(2*theta)
```

```
>> B = expand(B)
B =
2*cos(theta)^2-1
>> C = 6*((sin(theta))^2 + (cos(theta))^2)
C =
6*sin(theta)^2+6*cos(theta)^2
>> C = expand(C)
C =
6*sin(theta)^2+6*cos(theta)^2
```

როგორც ვნახეთ MATLAB –მა “კარგად გაართვა თავი” რთულ ტრიგონომეტრიულ გარდაქმნებს, თუმცა ბოლო სრულიად მარტივი გარდაქმნის შესრულება “გაუჭირდა”:

$$C = 6(\sin^2(\theta) + \cos^2(\theta)) = 6$$

### სიმბოლური გამოსახულების რიცხვითი მნიშვნელობის გამოთვლა და გრაფიკის აგება

ხშირად გვჭირდება ცვლადის მოცემული მნიშვნელობისათვის გამოვთვალოთ სიმბოლური გამოსახულების მნიშვნელობა. ამისათვის საჭიროა ფუნქციით **subs** სიმბოლურ ცვლადს მივანიჭოთ რიცხვითი მნიშვნელობა:

```
>> E = s^3 -14*s^2 +65*s -100;
>> F = subs(E,s,7.1)
```

```
F =
```

```
13.6710
```

F პირდაპირ ლებულობს რიცხვით მნიშვნელობას double array

შესაძლებელია სიმბოლური გამოსახულების მიხედვით ავაგოთ გრაფიკი:

ezplot(f)	ააგებს f(x) გრაფიკს, სადაც f სიმბოლური გამოსახულებაა და წარმოადგენს მათემატიკურ გამოსახულებას, რომელიც შეიცავს მხოლოდ ერთ ცვლადს, ვთქვათ x. მითითების გარეშე (default) x მნიშვნელობათა არეა [-2π, 2π]
ezplot(f,xmin,xmax)	ააგებს გრაფიკს x მითითებულ მნიშვნელობათა მიხედვით

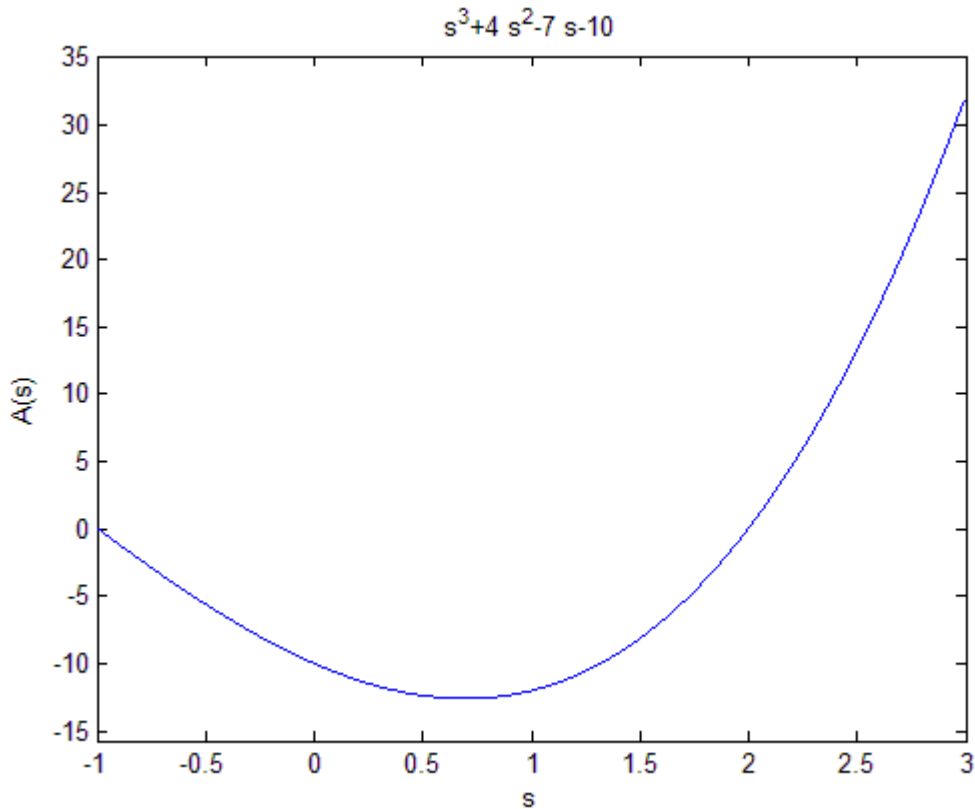
საილუსტრაციოდ განვიხილოთ ფუნქცია:

$$A(s) = s^3 + 4s^2 - 7s - 10$$

ინტერვალში [-1, 3]:

```
syms s
a = [1 4 -7 -10];
A = poly2sym(a,s)
ezplot(A,-1,3), ylabel('A(s)')
```

შედეგად მიღებული გრაფიკი ნაჩვენებია ნახ. 13.1.



ნახ. 12.1 ezplot ფუნქციის საშუალებით აგებული პოლინომის გრაფიკი

მიაქციეთ ყურადღება, რომ გამოსახულება, რომლის მიხედვითაც აიგო გრაფიკი ავტომატურად განთავსდება გრაფიკის სათაურად, ხოლო  $x$  ღერძს ავტომატურად წაეწერება ცვლადის სახელი (ამ შემთხვევაში  $s$ )

### სიმბოლური ალგებრული და ტრანსცენდენტული განტოლებები

MATLAB –ის საშუალებით შესაძლებელია ამოვხსნათ სიმბოლური ალგებრული ან ტრანსცენდენტული განტოლებები და განტოლებათა სისტემები. ტრანსცენდენტული ისეთ განტოლებას ეწოდება, რომელიც შეიცავს რომელიმე ტრანსცენდენტულ ფუნქციას:  $\cos x$ ,  $e^x$  ან  $\ln x$ . ამისათვის გამოიყენება ფუნქცია solve. არსებობს ამ ფუნქციის რამდენიმე ფორმა, ჩვენ განვიხილავთ შემდეგს:

```

solve(E1, E2, E3, . . . , EN)
solve(E1, E2, E3, . . . , EN, var1, var2, . . . , var3)

```

სადაც E1, E2, E3, . . . , სიმბოლურ გამოსახულებათა სახელებია, ხოლო var1, var2, . . . , var3 შესაბამისი ცვლადები. მიღებული ამონახსნები წარმოადგენს გამოსახულების ფესვებს. ეს არის სიმბოლური გამოსახულებანი ცვლადებისათვის შემდეგი პირობებით E1=0, E2=0, E3=0, . . . , EN=0.

ერთი განტოლებისათვის ერთი ცვლადით ამონახსნი იქნება ერთი სიმბოლური სიდიდე. მაგალითად:

```
>> syms s
>> E = s+2;
>> s = solve(E)
s =
-2
```

მივიღეთ სიმბოლური სიდიდე  $s=-2$ . იმისათვის, რომ იგი რიცხვით გამოსახულებად გარდაქმნათ საჭიროა მივმართოთ ბრძანებას **double**:

`double(S)` სიმბოლურ გამოსახულებას  $S$  გარდაქმნის რიცხვით სიდიდედ (double precision floating point number).  $S$  არ უნდა შეიცავდეს სიმბოლურ ცვლადს.

```
>> snum=double(s)

snum =

-2
```

განვიხილოთ სიმბოლური გამოსახულება – მეორე ხარისხის პოლინომი:

```
>> syms s
>> D = s^2 +6*s +9;
>> s = solve(D)
s =
[ -3]
[ -3]
```

მაგალითი ტრიგონომეტრიიდან:

```
>> syms theta x z
>> E = z*cos(theta) - x;
>> theta = solve(E,theta)
theta =
acos(x/z)
```

განტოლებებს, რომლებიც შეიცავენ პერიოდულ ფუნქციებს, შესაძლოა ჰქონდეთ ფესვების განუსაზღვრელი რაოდენობა. ასეთ შემთხვევაში **solve** ფუნქცია შემოიფარგლება ფესვებით 0-ის მახლობლობაში. მაგალითად ამოვხსნათ განტოლება:

$$2 \cos(2\theta) - \sin(\theta) = 0$$

```
>> E = cos(2*theta)-sin(theta);
>> solve(E)
ans =
[ -1/2*pi]
[ 1/6*pi]
[ 5/6*pi]
```

MATLAB –ში შესაძლებელია სიმბოლური გამოსახულების ინტეგრება და დიფერენცირება.

diff(E)	ახდენს E სიმბოლური გამოსახულების დიფერენცირებას იმ ცვლადით, რომელიც განისაზღვრება ფუნქციით findsym(E)
diff(E,v)	ახდენს E სიმბოლური გამოსახულების დიფერენცირებას v ცვლადით
diff(E,n)	ახდენს E სიმბოლური გამოსახულების დიფერენცირებას n-ჯერ (n მთელი დადებითი რიცხვია)
diff(S,v,n)	ახდენს E სიმბოლური გამოსახულების დიფერენცირებას v ცვლადით n-ჯერ (n მთელი დადებითი რიცხვია)

პოლინომის წარმოებულის გამოთვლა:

```
>> syms s n
>> p = s^3 + 4*s^2 -7*s -10;
>> d = diff(p)
d =
3*s^2+8*s-7
>> e = diff(p,2)
e =
6*s+8
>> f = diff(p,3)
f =
6
>> g = s^n;
>> h = diff(g)
h =
s^n*n/s
>> h = simplify(h)
h =
s^(n-1)*n
```

ტრანსცენდენტული ფუნქციის წარმოებულის:

```
>> syms x
>> f1 = log(x);
>> df1 = diff(f1)
df1 =
1/x
>> f2 = (cos(x))^2;
>> df2 = diff(f2)
df2 =
-2*cos(x)*sin(x)
>> f3 = sin(x^2);
>> df3 = diff(f3)
df3 =
2*cos(x^2)*x
>> df3 = simplify(df3)
df3 =
2*cos(x^2)*x
>> f4 = cos(2*x);
>> df4 = diff(f4)
df4 =
-2*sin(2*x)
```



```
>> f5 = exp(-(x^2)/2);
>> df5 = diff(f5)
df5 =
-x*exp(-1/2*x^2)
```

### ინტეგრება

int(E)	გვამღევს სიმბოლური გამოსახულების განუსაზღვრელ ინტეგრალს მისი სიმბოლური ცვლადის მიხედვით, რომელიც განისაზღვრება ფუნქციით findsym
int(E,v)	E სიმბოლური გამოსახულების განუსაზღვრელი ინტეგრალი v ცვლადის მიხედვით
int(E,a,b)	E სიმბოლური გამოსახულების განსაზღვრული ინტეგრალი ინტერვალში [a b], სადაც a და b სიმბოლური ან რიცხვითი სკალარებია
int(E,v,a,b)	E სიმბოლური გამოსახულების განსაზღვრული ინტეგრალი v ცვლადის მიხედვით ინტერვალში [a b], სადაც a და b სიმბოლური ან რიცხვითი სკალარებია

### პოლინომის ინტეგრება:

```
>> syms x n a b t
>> int(x^n)
ans =
x^(n+1)/(n+1)
>> int(x^3 + 4*x^2 + 7*x + 10)
ans =
1/4*x^4+4/3*x^3+7/2*x^2+10*x
>> int(x,1,t)
ans =
1/2*t^2-1/2
>> int(x^3,a,b)
ans =
1/4*b^4-1/4*a^4
```

### ტრანსცენდენტული ფუნქციის ინტეგრება:

```
>> syms x
>> int(1/x)
ans =
log(x)
>> int(cos(x))
ans =
sin(x)
>> int(1/(1+x^2))
ans =
atan(x)
>> int(exp(-x^2))
ans =
1/2*pi^(1/2)*erf(x)
```

$\text{erf}(x)$  წარმოადგენს ცთომილების ინტეგრალს (error function) ყოველი  $x$ -თვის, სადაც  $x$  – ნამდვილი რიცხვია, ხოლო  $\text{erf}(x)$  შემდეგნაირად განისაზღვრება:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

### წრფივი ალგებრა

მატრიცული ოპერაციების განხორციელება შესაძლებელია სიმბოლური მატრიცებისთვისაც.

```
>> A = sym([2,1; 4,3])
A =
[ 2, 1]
[ 4, 3]
>> Ainv = inv(A)
Ainv =
[ 3/2, -1/2]
[ -2, 1]
>> C = A*Ainv
C =
[ 1, 0]
[ 0, 1]
>> B = sym([1 3 0; -1 5 2; 1 2 1])
B =
[ 1, 3, 0]
[ -1, 5, 2]
[ 1, 2, 1]
>> detB = det(B)
detB =
10
```

ასევე შესაძლებელია MATLAB საშუალებით ამოვსნათ წრფივ განტოლებათა სიმბოლური სისტემა:

```
>> syms x
>> A = sym([3 2 -1; -1 3 2; 1 -1 -1])
A =
[ 3, 2, -1]
[ -1, 3, 2]
[ 1, -1, -1]
>> b = sym([10; 5; -1])
b =
[ 10]
[ 5]
[ -1]
>> x = A\b
x =
[ -2]
[ 5]
[ -6]
```

მიღებული შედეგი შეადარეთ იგივე სისტემის რიცხვით ამონახსნებს, რათა დარწმუნდეთ პასუხის სისწორეში. თუმცა, განტოლებათა სისტემის ამონახსნის რაიმე პარამეტრის მიმართ, მცირედი უპირატესობა გააჩნია სიმბოლურ მიდგომას. განვიხილოთ განტოლებათა სისტემა:

$$2x_1 - 3x_2 = 3$$

$$5x_1 + cx_2 = 19$$

ამოვხსნათ  $x_1$  და  $x_2$  როგორც  $c$  პარამეტრის ფუნქცია:

```
>> syms c
>> A = sym([2 -3; 5 c]);
>> b = sym([3; 19]);
>> x = A\b
x =
[ 3*(19+c)/(2*c+15) ]
[ 23/(2*c+15) ]
```

ბრძანების **subs** საშუალებით შეგვიძლია ვიპოვოთ განტოლებათა სისტემის ამონახსნები  $c$  სხვადასხვა მნიშვნელობათათვის:

თუ  $c=1$ , მივიღებთ:

```
x1=subs(x,1)
```

```
x1 =
```

```
3.5294
```

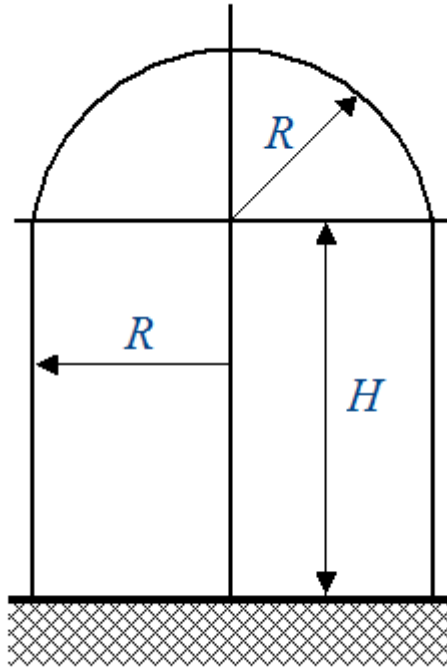
```
1.3529
```

### ცილინდრული ფორმის რეზერვუარის დიზაინი მინიმალური დანახარჯით

უნდა აიგოს ცილინდრული ფორმის რეზერვუარი ნახევარსფეროს ფორმის გადახურვით მინიმალური დანახარჯით, თუ ცნობილია, რომ ცილინდრული ზედაპირის კონსტრუქცია ღირს  $\$300/\text{მ}^2$ , ხოლო ცფერული ზედაპირის -  $\$400/\text{მ}^2$ , ხოლო რეზერვუარის ტევადობაა  $5 \cdot 10^5$  ლ. რეზერვუარი უნდა აიგოს ბეტონის ზედაპირზე, რომლის ფასიც მხედველობაში არ უნდა იქნას მიღებული.

#### 1. ამოცანის დასმა

ნახ. 12.2 –ზე მოცემულია რეზერვუარის ზომები. განვსაზღვროთ ცილინდრის რადიუსი და სიმაღლე ისე, რომ რეზერვუარის ასაგებად მინიმალური თანხა დაიხარჯოს



ნახ. 12.2 რეზერვუარის კონსტრუქცია

## 2. INPUT/OUTPUT აღწერა

ცნობილია, რომ ცილინდრული ზედაპირის აგება ღირს \$300/მ<sup>2</sup>, ხოლო ცვერული ზედაპირის - \$400/მ<sup>2</sup>. ხოლო ასაგები რეზერვუარის ტევადობაა 5\*10<sup>5</sup> ლ. უნდა განვსაზღვროთ ცილინდრის რადიუსი და სიმაღლე ისე, რომ რეზერვუარის აგებაზე მინიმალური თანხა დაიხარჯოს.

### 3. სახელდახელო ამოხსნა

ცილინდრის მოცულობა:  $V_c = \pi \cdot R^2 \cdot H$

ნახევარსფეროს მოცულობა:  $V_h = \frac{2}{3} \pi \cdot R^3$

ცილინდრის ზედაპირის ფართობი:  $A_c = 2\pi \cdot R \cdot H$

ნახევარსფეროს ზედაპირის ფართობი:  $A_h = 2\pi \cdot R^2$

რადგან რეზერვუარის მოცულობაა 500ლ = 500 მ<sup>3</sup>, შეგვიძლია დავწეროთ გამოსახულება:

$$500 = \pi \cdot R^2 \cdot H + \frac{2}{3} \pi \cdot R^3$$

აქედან

$$H = \frac{500}{\pi R^2} - \frac{2R}{3}$$

გამოვსახოთ რეზერვუარის ღირებულება როგორც ცილინდრის სიმაღლისა და რადიუსის ფუნქცია:

$$C = 300 \cdot A_c + 400 \cdot A_h = 300 \cdot (2 \cdot \pi R \cdot H) + 400(2 \cdot \pi \cdot R^2)$$

როგორც ვიცით, უწყვეტი ფუნქციის წარმოებული ნულს უტოლდება ფუნქციის მინიმუმისა და მაქსიმუმის წერტილებში. პირველ რიგში მოცულობის გამოსათვლელი განტოლებიდან

ვიპოვოთ H როგორც R-ის ფუნქცია. შემდეგ გავაწარმოთ C ფუნქცია R-ით, გავუტოლოთ მიღებული გამოსახულება 0-ს და ვიპოვოთ განტოლების ფესვები. ასე ვიპოვოთ რადიუსის მნიშვნელობას, რომელიც შეესაბამება მინიმალურ ღირებულებას, ხოლო რადიუსის საშუალებით გამოვითვლით სიმაღლის სათანადო მნიშვნელობასაც.

#### 4. MATLAB ამოხსნა

```
syms R H
V = pi*R^2*H + (2/3)*pi*R^3 - 500; % Equation for volume
H = solve(V,H); % Solve volume for height H
H = -2/3*(pi*R^3-750)/pi/R^2;
C = 300*(2*pi*R*H) + 400*(2*pi % Equation for cost
dCdR = diff(C,R); % Derivative of cost wrt R
dCdR = 400/R^2*(pi*R^3-750)+400*pi*R;
Rmins = solve(dCdR,R); % Solve dC/dR for R: Rmin
Rmins = double(Rmins);

Rmin = Rmins(1);
Hmin = double(subs(H,R,Rmin));
Cmin = double(subs(C, {R,H}, {Rmin,Hmin}));
```

#### 5. შემოწმება

პროგრამა გამოითვლის რეზერვუარის ასაგებად საჭირო მინიმალურ დანახარჯს და რადიუსის და სიმაღლის სათანადო მნიშვნელობებს.

```
Rmin =
    4.92372510921348
Hmin =
    3.28248340614232
Cmin =
    9.139421678069333e+004
```

#### ბრძანებები და ფუნქციები

char(x)	x მასივს, რომლის ელემენტებიც მთელი დადებითი რიცხვებია და წარმოადგენს სიმბოლოთა შესაბამის კოდებს, გარდაქმნის სიმბოლოთა სტრიქონად.
[n,d]=numden(S)	მოგვცემს რაციონალურ წილადად დაშლილი S სიმბოლური გამოსახულების მრიცხველსა და მნიშვნელს
bin2dec(b)	ახდენს ორობითი რიცხვის - b ინტერპრეტირებას და გვაძლევს შესაბამის ათობით რიცხვს
blancs(n)	გვაძლევს სტრიქონს , რომელიც შეიცავს n სიმბოლოს ნაცვლად n ცარიელ პოზიციას.
c=bitand(a,b)	ბიტური and ორ არგუმენტს შორის
c=bitcmp(a,N)	a ათობითი მთელი რიცხვის N ბიტიან ორობით წარმოდგენაში შეცვლის 0 1-ით, ხოლო 1 0-ით და მოგვცემს შესაბამის ათობით რიცხვს
c=bitor(a,b)	ბიტური or a და b არგუმენტს შორის
c=bitshift(a,N)	მოგვცემს მთელ ათობით რიცხვს, რომელიც წარმოადგენს a რიცხვის ორობითი წარმოდგენის N ბიტით წანაცვლების შედეგს. თუ $N > 0$ , ეს ოპერაცია იგივეა, რაც გამრავლება $2^N$ ,

	თუ $N < 0$ , - იგივეა, რაც გაყოფა $2^N$ -ზე
deblank(s)	თუ სტრიქონი ბოლოვდება განმეორებადი ცარიელი პოზიციით, ეს ბრძანება შეკვეცს მას პირველივე სიმბოლომდე
dec2bin(d)	გვაძლევს $d$ ორობით წარმოდგენას როგორც სტრიქონს. $d$ უნდა იყოს არაუარყოფითი მთელი რიცხვი.
dec2hex(d)	გვაძლევს $d$ თექვსმეტობით წარმოდგენას როგორც სტრიქონს. $d$ უნდა იყოს არაუარყოფითი მთელი რიცხვი.
diff(E)	ახდენს $E$ სიმბოლური გამოსახულების დიფერენცირებას იმ ცვლადის მიხედვით, რომელიც განისაზღვრება ფუნქციით findsym(E)
diff(E,n)	ახდენს $E$ სიმბოლური გამოსახულების დიფერენცირებას $n$ -ჯერ ( $n$ მთელი დადებითი რიცხვია)
diff(E,v)	ახდენს $E$ სიმბოლური გამოსახულების დიფერენცირებას $v$ ცვლადის მიხედვით
diff(S,v,n)	ახდენს $E$ სიმბოლური გამოსახულების დიფერენცირებას $v$ ცვლადის მიხედვით $n$ -ჯერ ( $n$ მთელი დადებითი რიცხვია)
double(S)	სიმბოლურ გამოსახულებას $S$ გარდაქმნის რიცხვით სიდიდედ (double precision floating point number). $S$ არ უნდა შეიცავდეს სიმბოლურ ცვლადს.
eval(s)	გაუშვებს $s$ სტრიქონს როგორც MATLAB გამოსახულებას ან ბრძანებას.
eval(s1,s2)	საშუალებას იძლევა შეცდომა დავიჭიროთ. გაუშვებს $s1$ შესაბამის ბრძანებას და გვაძლევს შედეგს, თუ ბრძანება წარმატებით შესრულდა, ხოლო თუ ბრძანების შესრულება შეცდომას იძლევა, უშვებს $s2$ – შესაბამის ბრძანებას.
expand(S)	სიმბოლურ გამოსახულებაში $S$ შეასრულებს ალგებრულ მოქმედებებს, შეკრებს მსგავს წევრებს და წარმოგვიდგენს პოლინომის სახით.
ezplot(f)	ააგებს $f(x)$ გრაფიკს, სადაც $f$ სიმბოლური გამოსახულებაა და წარმოადგენს მათემატიკურ გამოსახულებას, რომელიც შეიცავს მხოლოდ ერთ ცვლადს, ვთქვათ $x$ . მითითების გარეშე (default) $x$ მნიშვნელობათა არეა $[-2\pi, 2\pi]$
ezplot(f,xmin,xmax)	ააგებს გრაფიკს $x$ მითითებულ ინტერვალში
factor(S)	$S$ პოლინომს დაშლის მამრავლებად.
findstr(s1,s2)	პოულობს ერთ სტრიქონს მეორის შიგნით
findsym(S)	სიმბოლურ გამოსახულებაში ან მატრიცაში $S$ პოულობს სიმბოლურ ცვლადებს და გვაძლევს მათ, როგორც სიმბოლურ სტრიქონს, ანბანის მიხედვით დალაგებულს და მძიმეებით გამოყოფილს. თუ $S$ არ შეიცავს არცერთ სიმბოლურ ცვლადს, გვაძლევს ცარიელ სტრიქონს.
hex2dec(h)	ახდენს ორობითი რიცხვის $h$ ინტერპრეტირებას და გვაძლევს შესაბამის ათობით რიცხვს
int(E)	გვაძლევს სიმბოლური გამოსახულების განუსაზღვრელ ინტეგრალს მისი სიმბოლური ცვლადის მიხედვით, რომელიც განისაზღვრება ფუნქციით findsym
int(E,a,b)	$E$ სიმბოლური გამოსახულების განსაზღვრული ინტეგრალი ინტერვალში $[a b]$ , სადაც $a$ და $b$ სიმბოლური ან რიცხვითი სკალარებია

int(E,v)	E სიმბოლური გამოსახულების განუსაზღვრელი ინტეგრალი v ცვლადის მიხედვით
int(E,v,a,b)	E სიმბოლური გამოსახულების განსაზღვრული ინტეგრალი v ცვლადის მიხედვით ინტერვალში [a b], სადაც a და b სიმბოლური ან რიცხვითი სკალარებია
int2str(x)	x მატრიცის ელემენტებს დაამრგვალებს მთელამდე და შედეგად მიღებულ მასივს გარდაქმნის სიმბოლურ სტრიქონად
ischar(s)	გვაძლევს 1, თუ s სიმბოლური მასივია, სხვა შემთხვევაში -0
isletter(s)	გვაძლევს 1 s სტრიქონის ყოველი ელემენტისათვის, რომელიც ასოით გამოსახულებას წარმოადგენს და 0 სხვა შემთხვევაში.
isspace(s)	გვაძლევს 1 s სტრიქონის იმ ელემენტებისათვის, რომლებიც წარმოადგენს ცარიელ პოზიციას(space, tabs, newlines, carriage returns), 0 – სხვა შემთხვევაში
num2str(x)	x მატრიცას გარდაქმნის სტრიქონად. ეს ბრძანება სასარგებლოა გრაფიკზე აღნიშვნების გასაკეთებლად (title, xlabel, ylabel, text)
poly2sym(p)	p ვექტორზე დაყრდნობით, რომლის ელემენტებიც პოლინომის კოეფიციენტებს წარმოადგენს, გვაძლევს შესაბამის სიმბოლურ პოლინომს. ცვლადს მიანიჭებს სახელს x.
simplify(S)	გაამარტივებს სიმბოლურ გამოსახულებას S
str2num(s)	გარდაქმნის s სტრიქონს, რომელიც უნდა იყოს რიცხვითი სიდიდის შესაბამისი ASCII კოდი, შესაბამის რიცხვით მნიშვნელობად. სტრიქონი s შეიძლება შეიცავდეს ციფრს, ათწილადურ მძიმეს, + ან - ნიშანს, 'e' 10-ის ხარისხის მაჩვენებლს და I – კომპლექსური რიცხვებისათვის.
subs(S,old,new)	S სიმბოლურ გამოსახულებაში ცვლადს old შეცვლის ცვლადით new
sym2poly(P)	სიმბოლური პოლინომისათვის P გვაძლევს ვექტორს, რომლის ელემენტებია ამ პოლინომის კოეფიციენტები