

ლევან შოშიაშვილი

ფიზიკის ამოცანების მოდელირების  
საფუძვლები

(პრაქტიკული)

თბილისი

2008

# სარჩევი

სარჩევი	1
1 Matlab ბრძანებები	2
პროცენტის ნიშანი "%" . . . . .	2
<i>help</i> . . . . .	2
<i>lookfor</i> . . . . .	2
სამი წერტილი "..."	3
წერტილმძიმე ";" . . . . .	3
კლავიატურის ღილაკები . . . . .	3
2 Matlab ობიექტები: სკალარი, ვექტორი, მატრიცა	5
3 მოქმედებები ვექტორებზე და მატრიცებზე	8
4 ფუნქციები	10
5 2D გრაფიკა	11
5.1 ბრძანება <i>plot</i> . . . . .	11
5.2 ამოცანები . . . . .	11
5.3 <i>plot</i> თვისებები: ფერი და სტილი . . . . .	12
5.4 2D გრაფიკის დამატებითი შესაძლებლობები . . . . .	13
5.5 2D ანიმაცია. სინუსოიდალური ტალღის მოდელირება . . . . .	13

## თავი 1

# Matlab ბრძანებები

### პროცენტის ნიშანი ”%”

%ნიშანი ნიშნავს კომენტარებს. მის შემდეგ რაც წერია მოცემულ ხაზზე ამ ნიშნის შემდეგ მაგლავი იგნორირებას უკეთებს. ეს მოსახერხებელია პროგრამაში კომენტარების გასაკეთებლად<sup>1</sup>.

### help

*help* – ბრძანება გამოიგანს იმ თემებს რის შესახებაც შეიძლება ინფორმაციის ნახვა.

*help სიტყვა* – ბრძანება გამოიგანს ინფორმაციას *სიტყვა* შესახებ.

#### მაგალითი:

*help graphics* გამოიგანს ინფორმაციას *graphics* შესახებ

### lookfor

*lookfor სიტყვა* – ბრძანება გამოიგანს ინფორმაციას ბრძანებების შესახებ, რომელიც მსგავსია სიტყვისა *სიტყვა*.

#### მაგალითი:

მაგალითად არ იცით როგორ მუშაობს და როგორია მუსკის სინტაქსი ბრძანებისა რომელიც იღებს ფესვს მაგრიციდან. აკეთებთ:

*lookfor sqr*

გარკვეული დროის შემდეგ<sup>2</sup> ეკრანზე გამოჩნდება:

```
>> lookfor sqr
REALSQRT Real square root.
SQRT Square root.
SQRTM Matrix square root.
```

<sup>1</sup>Matlab ბრძანებების ფანჯარაში შეუძლებელია ქართული უნიკოდ გეჟსტის შეგანა, მაგრამ თუ მაგლავის M ფაილს გააკეთებთ რაიმე ედიტორში, რომელსაც ესმის უნიკოდი Matlab (მეშვიდე ვერსია) ასეთ ფაილებთან უპრობლემოდ იმუშავებს

<sup>2</sup>ამ ბრძანებას სჭირდება დრო რომ მოძებნოს შესაბამისი ბრძანებები და გამოიგანოს ეკრანზე. ამიგომ ერიდეთ მის ხშირად გამოყენებას.

LSQR LSQR Implementation of Conjugate Gradients on the Normal Equations.  
 vsqrtm.m: % function out = vsqrtm(mat)  
 SUMSQ Sum squared elements of a matrix.  
 SQRMTX Script file of LINF minimal realization.  
 SQRT Symbolic matrix element-wise square root.

საძიებელი ბრძანება ყოფილა *SQRTM*.

### სამი წერტილი "..."

Matlab არის სკრიპტული ენა. Matlab ძრავა ინტერპრეტირებას უკეთებს ყოველ ხაზს. თუ ბრძანება არ ეგევა ერთ ხაზზე(მაგალითად ადვილად წასაკითხი რომ იყოს გამოსახულება) ის უნდა გაიყოს და დაიწეროს სხვადასხვა ხაზებზე, მაგრამ ისე რომ Matlab მა ეს აღიქვას როგორც ერთ ხაზზე დაწერილი ბრძანება. ამას აკეთებს "...".

**მაგალითი:**

```
>> 5+6*12+7+...
12*15+sqrt(pi)...
+sin(12) %დაგაჭიროთ Enter-ს და მიგიღებთ
ans =

265.2359
```

### წერტილები ";"

როგორც გემოთ ავლნიშნეთ Matlab არის სკრიპტული ენა და ინტერპრეტირებას უკეთებს ყოველ ხაზს. შესაბამისად შედეგი გამოაქვს ეკრანზე ყოველი ბრძანების ხაზის შემდეგ. თუ არ გვსურს ამ შედეგის ნახვა, რომ არ გადავგვირთოთ სამუშაო ფანჯარა ბრძანების შემდეგ უნდა დავწეროთ წერილები ";".

**მაგალითი:**

```
>> x=2*2

x =

4

>> x=2*2;
>>
```

### კლავიატურის ღილაკები

"↑" ("ისარი გემოთ") და "↓" ("ისარი ქვემოთ") – გამოიყვანს წინა და შემდგომ ბრძანებებს. იგივეს აკეთებს ღილაკების კომბინაცია "Ctrl – p" და "Ctrl – n". "Ctrl – a" კურსორი გადაჰყავს ხაზის დასაწყისში, ხოლო "Ctrl – e"-ს ხაზის ბოლოს. "Ctrl – k" მლის ხაზის ნაწილს კურსორის მარჯვნივ.

"Tab" ასრულებს არასრულად აკრეფილ ბრძანებას.

## თავი 2

# Matlab ობიექტები: სკალარი, ვექტორი, მაგრიცა

ბაზისური ობიექტებია სკალარები, ვექტორები და მაგრიცები.

$N = 5$  სკალარი

კვადრატული განტოლების ამოხსნა:

განვიხილოთ კვადრატული განტოლების მაგალითი

$$a \cdot x^2 + b \cdot x + c = 0$$

აკრიფეთ შემდეგი ტექსტი Matlab ბრძანებების ფანჯარაში:

```
> a=3
> b=5
> c=-7
> disc=(b^2-4*a*c)^0.5
> root1=(-b+disc)/(2*a)
> root2=(-b-disc)/(2*a)
```

ასევე შეგიძლია შევამოწმოთ პასუხის სიზუსტე.

```
> a*root1^2+b*root1+c
```

ანალოგიურად მერე ფესვისათვის  $root2$ .

$v = [1 \ 0 \ 0]$  აღნიშნავს ვექტორ სტრიქონს  $(1 \ 0 \ 0)$

$v = [1; 2; 3]$  აღნიშნავს ვექტორ სვეტს  $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

$v = v'$  – ვექტორის ტრანსპორირება. სვეტი გადაჰყავს სტრიქონში და პირქით.

$v = [1 : 0.5 : 3]$  – ვექტორი საზღვრებით. ვექტორი სტრიქონი მნიშვნელობებით 1დან 3მდე ბიჯით 0.5.

მაგალითი:

```
>> v = [1 : 0.5 : 3]
```

```
v =
```

## თავი 2. MATLAB ობიექტები: სკალარი, ვექტორი, მატრიცა

1.0000    1.5000    2.0000    2.5000    3.0000

>>

$v = \pi * [-4 : 4] / 4$  – ესაა ვექტორი სტრიქონი. ჯერ იქმნება ვექტორი მნიშვნელობებით -4 დან 4მდე ბიჯით 1. შემდეგ მრავლდება პიზე და იყოფა 4ზე.

**მაგალითი:**

>>  $v = \pi * [-4 : 4] / 4$

$v =$

Columns 1 through 8

-3.1416    -2.3562    -1.5708    -0.7854       0    0.7854    1.5708    2.3562

Column 9

3.1416

>>

$v = []$  – ცარიელი ვექტორი.

$m = [1 \ 2 \ 3; 4 \ 5 \ 6]$  – მაგრიცა. პირველი პარამეტრია სვეტის მნიშვნელობები, ხოლო მეორე სტრიქონი

**მაგალითი:**

>>  $m = [1 \ 2 \ 3; 4 \ 5 \ 6]$

$m =$

1    2    3  
4    5    6

>>

$m = \text{zeros}(2, 3)$  იქმნება მაგრიცა  $2 \times 3$  ნულოვანი მნიშვნელობებით.

$v = \text{ones}(1, 3)$  იქმნება მაგრიცა  $1 \times 3$  ერთიანებით.

$m = \text{eye}(3)$   $3 \times 3$  ერთეულოვანი მაგრიცა.

$v = \text{rand}(3, 2)$  იქმნება მაგრიცა  $2 \times 3$  შემთხვევითი მნიშვნელობებით 0.1 დიაპაზონში.

$m(1, 3)$  – მივმართავთ  $m$  მაგრიცის (1, 3) ელემენტს (სტრიქონი, სვეტი).

**მაგალითი:**

>>  $m = [1 \ 2 \ 3; 4 \ 5 \ 6]$

$m =$

1    2    3  
4    5    6

```
>> m(1,3)
```

```
ans =
```

```
3
```

```
>>
```

$m(2, :)$  – მიემართავთ  $m$  მატრიცის 2 სტრიქონს.

**მაგალითი:**

```
>> m(2, :)
```

```
ans =
```

```
4 5 6
```

```
>>
```

$m(:, 1)$  – მიემართავთ  $m$  მატრიცის 1 სვეტს.

**მაგალითი:**

```
>> m(:, 1)
```

```
ans =
```

```
1
```

```
4
```

```
>>
```

$size(m)$  – იძლევა  $m$  მატრიცის ზომას.

$size(m, 1)$  – იძლევა  $m$  მატრიცის სტრიქონების რაოდენობას.

$size(m, 2)$  – იძლევა  $m$  მატრიცის სვეტების რაოდენობას.

**მაგალითები:**

$m1 = zeros(size(m))$  – ქმნის ახალ  $m$  მატრიცის ზომის მატრიცას რომლის ელემენტებია 0.

$v = [123]$ ; ქმნის ვექტორს  $v$ .

$v(3)$  ნიშნავს მიმართვას ვექტორ  $v$  მესამე ელემენტზე<sup>1</sup>

---

<sup>1</sup>ინდექსირება იწყება 1-დან

## თავი 3

# მოქმედებები ვექტორებზე და მატრიცებზე

Matlab განმარტებულია ყველა ის ცნობილი ოპერაცია რაც სრულდება ჩვეულებრივ რიცხვებზე. მაგალითად

```
a = [1 2 3 4]; % ვექტორი
2 * a % სკალარზე გამრავლება მრავლდება ყველა ელემენტი
a / 4 % გაყოფა. იყოფა ყველა ელემენტი
b = [5 6 7 8]; % ვექტორი
a + b % შეკრება
a - b % გამოკლება . ორივე ვექტორის ზომა უნდა იყოს ტოლი.
a .^ 2 % ახარისხება. ".^" ნიშნავს რომ თითოეული ელემენტი ადის ხარისხში
a .* b % გამრავლება. ".*" ნიშნავს რომ მრავლდება თითოეული ელემენტი
a ./ b % გაყოფა "./" ნიშნავს იყოფა თითოეული ელემენტი.
%სცადეთ ეს მოქმედებები ბრძანებათა ფანჯარაში და იხილეთ შედეგი.
```

მატრიცები და ვექტორები შეიძლება მიეწოდოს ფუნქციებს. ფუნქცია შესრულება ყოველ ელემენტზე. მაგალითად

```
log( [1 2 3 4] )
round( [1.5 2; 2.2 3.1] )
a = [1 4 6 3] % ვექტორი
sum(a) % ვექტორის ელემენტების ჯამი
max(a) % მაქსიმალური მნიშვნელობა

a = [1 2 3; 4 5 6] % მატრიცა
a(:) % მატრიცის ვექტორული ვერსია
max(a) % მაქსიმალური მნიშვნელობა ყოველ სვეტში
max(max(a)) % მატრიცის მაქსიმუმი
[1 2 3] * [4 5 6]' % ხაზი ვექტორი 1x3 მრავლდება სვეტ ვექტორზე 3x1
% ანუ სკალარული ნამრავლი

[1 2 3]' * [4 5 6] % სვეტ ვექტორ 3x1 მრავლდება 1x3 ხაზ ვექტორზე
% შედეგი 3x3 მატრიცა
```

### თავი 3. მოქმედებები ვექტორებზე და მატრიცებზე

---

% ე.წ. გარე ნამრავლი

a = rand(3,2) % 3x2 მატრიცა

b = rand(2,4) % 2x4 მატრიცა

c = a \* b % 3x4 მატრიცა

a = [1 2; 3 4; 5 6] % 3 x 2 მატრიცა

b = [5 6 7]; % 1 x 3 ვექტორი

b \* a % მატრიცული ნამრავლი

a' \* b' % მატრიცული ნამრავლი

შემთხვევითი ბრძანებები სცადეთ Matlab ბრძანებათა ფანჯარაში.

## თავი 4

# ფუნქციები

MATLAB აქვს "ჩაშენებული" ფუნქციები. მაგალითად:

*sin, cos, tan, cosec, sec, cot, asin, acos, atan, acosec, asec, acot, sinh, cosh, tanh, cosech, sech, coth, asinh, acosh, atanh, acosech, asech, acothexp, log, log10, log2, sqrt*

შენიშვნა:

გრიგონომეტრიული ფუნქციები იყენებენ რადიანს

შებრუნებულ ფუნქციებს აქვთ პრეფიქსი *a*. მაგალითად *sin* და *asin*

*log* აღნიშნავს ნატურალურ ლოგარითმს და არა ლოგარითმს 10-ის ფუძით.

სცადეთ შემდეგი მაგალითები

```
» sin(pi/2) (The variable pi is predefined and contains 3.1415..)
» 2*asin(1)
» log(0)
» sin(0.2)^2+cos(0.2)^2
```

## თავი 5

# 2D გრაფიკა

### 5.1 ბრძანება plot

გრაფიკების ასაგებად Matlab იყენებს ბრძანებას *plot*. განვიხილოთ მაგალითი:

```
x=0:10  
y=sin(x)  
plot(x,y)
```

იგება გრაფიკი  $\sin(x)$ .  $x$  იცვლება 1-დან 10-მდე 1 ბიჯით. შეგვიძლია გაგზარდოთ რემოლუცია. ბიჯს ვამცირებთ 10-ჯერ:

```
x=0:0.1:10  
y=sin(x)  
plot(x,y)
```

შეგვიძლია ავაგოთ რამდენიმე გრაფიკი 1 სურათზე:

```
x=0:0.1:10;  
y=sin(x);  
z=cos(x);  
plot(x,y,x,z)
```

ახალი გრაფიკი დაემატა ახალი ფერით. ანალოგიურად შეგვიძლია დავამატოთ 2, 3, 4 და ა.შ. გრაფიკი.  $\text{plot}(x, y, x1, y1, x2, y2, \dots)$  ბრძანების საშუალებით.

### 5.2 ამოცანები

**დავალემა 1** აკრიფეთ Matlab ბრძანებათა ფანჯარაში შემდეგი ბრძანებები:

```
t=0:0.01:5;  
x=sin(2*pi*t);  
plot(t,x)
```

ჩათვალიეთ  $t$  (წმ.) ცვლადი დროდ.

1. რამდენი პერიოდი მოთავსდა 5 წამში?
2. რა არის სინუსოიდალური ტალღის სიხშირე?
3. გააკეთეთ იგივე  $x = \sin(2 * \pi * 0.5 * t)$ ; და  $x = \sin(2 * \pi * 2 * t)$ -თვის. მოგადად გვაქვს  $x = \sin(2 * \pi * f * t)$ ; რა კავშირია  $f$ -ს, სიხშირეს და პერიოდს შორის?

**დავალება 2** აკრიფეთ Matlab ბრძანებათა ფანჯარაში შემდეგი ბრძანებები:

```
x=-3:0.01:3;
y=cosh(x);
z=exp(x)/2;
plot(x,y,x,z)
```

1. აღვილი შესამხნევია, რომ  $\cosh(x)$  და  $\exp(x)/2$  დიდი  $x$  შემთხვევაში ერთნაირად იქცევა და გოლია. რატომ?
2. გემოთ მოყვანილ კოდს დაუმატეთ შემდეგი და ააგეთ გრაფიკი:

```
zm=exp(-x)/2;
plot(x,y,x,z,x,zm)
```

3. რა კავშირია ამ გრაფიკსა  $\cosh(x)$  შორის<sup>1</sup>?

**დავალება 3** აკრიფეთ Matlab ბრძანებათა ფანჯარაში შემდეგი ბრძანებები:

```
x=-2:0.01:2;
plot(x,cosh(x),x,sinh(x),x,tanh(x))
```

1. როგორია დაკავშირებული  $\tanh(x)$  დაკავშირებული  $\sinh(x)$  და  $\cosh(x)$ -თან?
2. რა კავშირია  $\tanh(x)$   $\sinh(x)$  შორის პატარა  $x$ -თვის?
3. რა მოსდის  $\tanh(x)$  როცა  $x$  იზრდება?

**დავალება 4** აკრიფეთ Matlab ბრძანებათა ფანჯარაში შემდეგი ბრძანებები:

```
x=-10:0.01:10;
plot(x,sech(x),x,tanh(x))
```

1. რამდენ წერტილში იკეთება ეს წირები? აღვილი მისახვედრია რომ  $\operatorname{sech}(x)$  არის  $\sec(x)$ -ის ჰიპერბოლური ეკვივალენტი და განმარტებულია როგორც  $\frac{1}{\cosh(x)}$ . გამოიყენეთ "გაზიდვის" ("zoom") დილაკი ("+") და გააღიდეთ ამ გრაფიკების გალაკვეთის არე. გალაკვეთა წარმოადგენს

$$\tanh(x) = \operatorname{sech}(x)$$

განგოლების ამონახსენს. ამოხსენით ეს განგოლება ანალიზურად და შეადარეთ გრაფიკული წესით მიღებულ პასუხს.

### 5.3 plot თვისებები: ფერი და სტილი

გრაფიკის თვისებები შეიძლება შეიცვალოს ორი გზით.

ა) გრძანებებში გრაფიკის თვისებების დამატებით.

აკრიფეთ Matlab ბრძანებათა ფანჯარაში შემდეგი ბრძანებები:

```
x=[-1.5:0.1:1.5];
y=cot(x);
plot(x,y,'r')
plot(x,y,'g*')
plot(x,y,'k:')
plot(x,y,'cs--')
```

<sup>1</sup>zm შემოტანის ნაცვლად შეგიძლიათ აკრიფოთ  $\operatorname{plot}(x, \cosh(x), x, \exp(x)/2, x, \exp(-x)/2)$

"" მოთავსებული პარამეტრებიდან პირველი აღნიშნავს ფერს ხოლო მეორე ხაზის სტილს.

გავექვს ფერის შემდეგი პარამეტრები:

r (red), g (green), m (magenta), c (cyan), w (white), y (yellow), b (blue)

ხაზის სტილის პარამეტრები:

- (ჩვეულებრივი ხაზი), : (წერტილოვანი), - (წყვეტილი), -. (წყვეტილ-წერტილი)

წერტილის პარამეტრები:

+ ,o ,x ,\* ,. ,s (კვადრატი), d (diamond), (სამკუთხედი ზემოთ),v ,> ,< ,p (ხუთ

წერტილოვანი ვარსკვლავი), h (ექვს წერტილოვანი ვარსკვლავი)

ხაზის სისქე შგვიძლია ვცვალოთ შემდეგი ბრძანებით:

```
plot(x,y,'LineWidth',3)
```

ბ) ასევე შეგვიძლია ვცვალოთ გრაფიკის თვისებები გრაფიკის ფანჯრიდან, რომელიც გამოდის ეკრანზე  $plot(x,y)$  ბრძანების შემდეგ. დააწკაპუნეთ კნოპკაზე "Edit". მონიშნეთ "Current Object Properties..." აირჩიეთ "style". შეგიძლიათ შეცვალოთ გრაფიკის სტილი.

## 5.4 2D გრაფიკის დამატებითი შესაძლებლობები

Matlab აქვს 2D გრაფიკის დამატებითი ფუნქციები. გრაფიკების ლოგარითულ სკალაზე ასაგებად. *semilogx*, *semilogy*, *loglog* ბრძანებები ნიშნავს  $x$ ,  $y$  და ორივე -  $x$  და  $y$  ცვლადებს ლოგარითულ მასშტაბში. მაგალითად, ქვემოთ მოყვანილი ფუნქცია არის ჰარმონიული ოსცილატორის სიხშირის გამოძახილის ფუნქცია  $\zeta$  მილევის კოეფიციენტით:

$$H(f) = \frac{1}{(f^2 - f_0^2)^2 + 4\zeta^2 f^2 f_0^2}$$

```
f=[0:0.1:100];
f0=10;
zeta=0.01;
H=1./((f.^2-f0.^2).^2+4*f.^2*zeta^2*f0^2);
plot(f,H)
semilogy(f,H)
loglog(f,H)
```

### დავალება 5

ააგეთ *semilogx*, *semilogy* და *loglog* გრაფიკები ფუნქციებისთვის:

ა)  $y = 4x^3$ , ბ)  $y = 6^{3x^8}$ , გ)  $y = \ln(4x^3)$

რომელი ბრძანება იძლევა სწორ ხაზს? რატომ?

## 5.5 2D ანიმაცია. სინუსოიდალური გაღლის მოდელირება

Matlab იძლევა საშუალებას ბრძანებები გაერთიანებულ იყოს ერთ ფაილად. ამისათვის:

i) დააწკაპუნეთ "File" მენიუზე

ii) "New"

iii) აირჩიეთ M-ფაილი.

აკრიფეთ შემდეგი ბრძანებები:

```
t=[0:0.01:10];
w=3;
plot(t,sin(w*t))
```

შეინახეთ ეს ფაილი რაიმე სახელით. დააწკაპუნეთ კლავიშზე F5. დაამატეთ ფაილში ბრძანება "close all" მეორე ხაზზე. ფაილს ექნება სახე:

```
t=[0:0.01:10];%t იცვლება 0 დან 10მდე ბიჯით 0.01
close all% დახურე ყველა არსებული ფანჯარა
for w=0:10 %w იცვლება 0 დან 10მდე ბიჯით 1;
    plot(t,sin(w*t)) %იგება გრაფიკი
end
```

"close all" ხურავს ყველა არსებულ გრაფიკის ფანჯარას და უზრუნველყოფს გრაფიკების დახატვას ახალ ფანჯარაში. ზემოთ მოყვანილ კოდით შეგვიძლია ვივარაუდოთ რომ დაიხატება 11 გრაფიკი ფუნქციისა  $\sin()$ , სადაც  $w$  იქნება 0 დან 10-მდე, მაგრამ Matlab ხატავს გრაფიკს მხოლოდ მას შემდეგ რაც ციკლი დამთავრდა ანუ დახატავს ბოლო გრაფიკს. შეგვიძლია ვაიძულოთ Matlab დახატოს ყველა გრაფიკი ამისათვის არის ბრძანება "drawnow". მაგალითად, სცადეთ შემდეგი შეცვლილი სკრიპტი

```
t=[0:0.01:10];
close all
for w=0:10
    plot(t,sin(w*t))
    drawnow
end
```

მაგრამ ესლა გრაფიკები იმდენად სწრაფად იცვლება, რომ ვერ ვასწრებთ დანახვას. ამის გამოსასწორებლად შეგვიძლია გამოვიყენოთ ბრძანება "pause(x)", რაც აიძულებს სისტემას შეყოვნდეს x წამის განმავლობაში.

```
t=[0:0.01:10];
close all
for w=0:10
    plot(t,sin(w*t))
    pause(0.1)
end
```

"pause(x)" ასუფთავებს ასევე გრაფიკულ ბუფერს, რის შედეგადაც არ იმუშავებს ბრძანება "drawnow" ამიტომ ეს ბრძანება აღარაა კოდის ბოლო ვარიანტში. შეგვიძლია ცვალოთ "pause(x)" ფუნქციის არგუმენტი, რომ მიიღოთ "რეალისტური" ანიმაცია. შეცვალეთ w ბიჯით 0.1 და დააბრუნეთ "drawnow" ბრძანება. როგორ შეცვალა გრაფიკი?

შექმენით ახალი ფაილი შემდეგი კოდით:

```
x=[0:0.01:2];
c=343;
f=1000;
w=2*pi*f;
lambda=c/f;
k=2*pi/lambda;
close all
```

```
for t=0:1e-5:0.002
    plot(x,sin(w*t-k*x))
    drawnow
end
```

გაუშვით ეს ფაილი შესრულებამდე (F5). თქვენ იხილავთ მარჯვნივ მოძრავი გალლის ანიმაციას. ეს მაგალითი წარმოადგენს ჰაერში გავრცელებადი 1 კჰც სიხშირის წნევის გალლის გავრცელების ვიზუალიზაციას. x-ღერძი წარმოადგენს მანძილს და ყოველი გრაფიკი წარმოადგენს გალლის სურათს ღროის t მომენტში.

**დავალება 6 რა მოხდება პარამეტრების ქვემოთ მოყვანილი ცვლილებისას?**

ა) შეცვალეთ სიხშირე 300ჰც.

ბ) შეცვალეთ  $\sin(w*t-k*x)$  გამოსახულებით  $\sin(w*t+k*x)$ .

გ) შეცვალეთ ხმის სიჩქარე სიდიდით 1500 მ/წმ (ხმის სიჩქარე წყალში)

ანიმაციისას შენიშნავთ რომ სურათი ციმციმებს. ასე იმიტომ ხდება რომ ყოველ ბიჯზე ხდება სურათის წაშლა და თავიდან დახატვა და ჩვენ არ ვქმნით სურათების ჭეშმარიტ მიმდევრობას, როგორც მაგალითად გვაქვს ფილმში. ანიმაციის უკეთესი ფორმ Matlab -ში არის სპეციალური ფუნქცია რომელიც ქმნის ფილმს. ქვემოთ მოყვანილი კოდი წარმოადგენს გავრცელებადი გალლის სწორ მოდელირებას/ანიმაციას. ცვლადი F ინახავს სურათების მიმდევრობას ფილმისათვის :

```
x=[0:0.02:2] ;
c=343;
f=1000;
w=2*pi*f;
lambda=c/f;
k=2*pi/lambda;
close all
n=1;
clear F
for t=0:2e-5:0.002
    plot(x,sin(w*t-k*x))
    F(n)=getframe;
    n=n+1;
end
```

გაუშვით სკრიპტი შესრულებამდე. შედეგი არ იქნება დიდად განსხვავებული წინა შედეგებისგან, მაგრამ შესრულების შემდეგ შეგიძლიათ ბრძანებების ფანჯარაში აკრიფოთ "whos" და ნახავთ რომ გაჩნდა ახალი ცვლადი F როელიც შეიცავს ინფორმაციას ანიმაციის შესახებ. აკრიფეთ ბრძანებათა ფანჯარაში

```
movie(F)
```

ეს ბრძანება აჩვენებს ანიმაციას ორჯერ. ერთხელ როცა იგვირთება, ხოლო მეორედ როცა მიმდინარეობს ჩაგვირთულის ჩვენება. შესაძლებელია ანიმაციის სიჩქარის შეცვლა.

```
movie(F,2,25)
```

გემოთ მოყვანილი ბრძანება აჩვენებს ანიმაციას 2ჯერ სიჩქარით 25 კადრი წამში. თუ ეს პარამეტრი არაა მითითებული მაშინ აღებულია 12 კადრი/წმ. მიღებული ანიმაცია შეიძლება ჩაწეროთ როგორც .avi ფაილი ბრძანებით:

```
movie2avi(F,'failissaxeli')
```