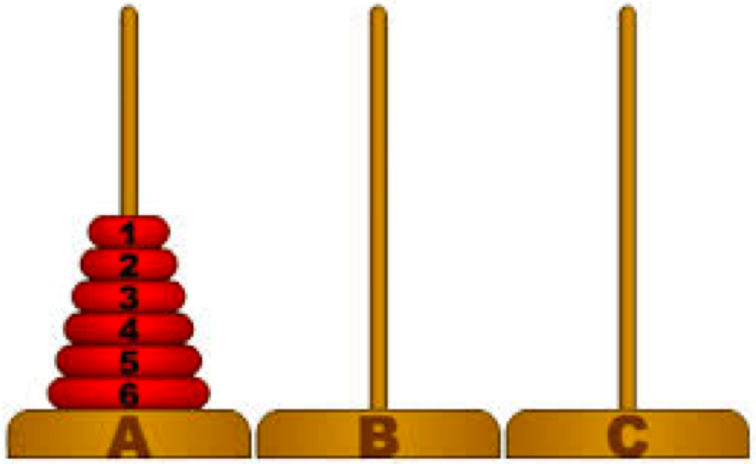
מבוא לתכנות מתמטי

תרגול 6

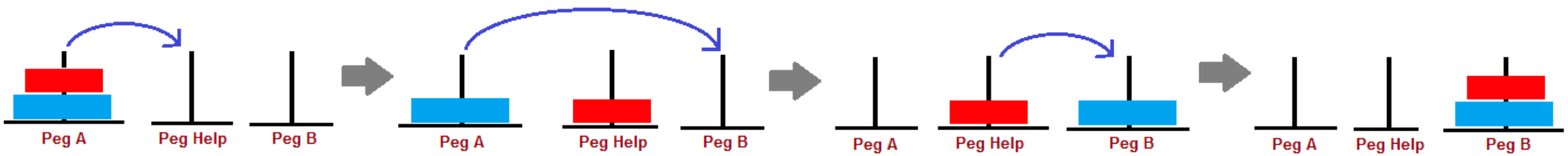
מגדלי הנוי Hanoi Tower

- **תיאור הבעיה:**
  - נתונים 3 מגדלים A B ו-C, כאשר על מגדל A מושחלות n טבעות
  - מטרה להעביר את n הטבעות ממגדל A למגדל C כך שהן יהיו מסודרות עפ"י סדר כפי שהן במגדל A (טבעת קטנה נמצאת מעל טבעת גדולה יותר)
  - אבל.....
  - בכל שלב מועברת טבעת יחידה!
  - אסור שמעל טבעת מסוימת תהיה טבעת שגדולה ממנה



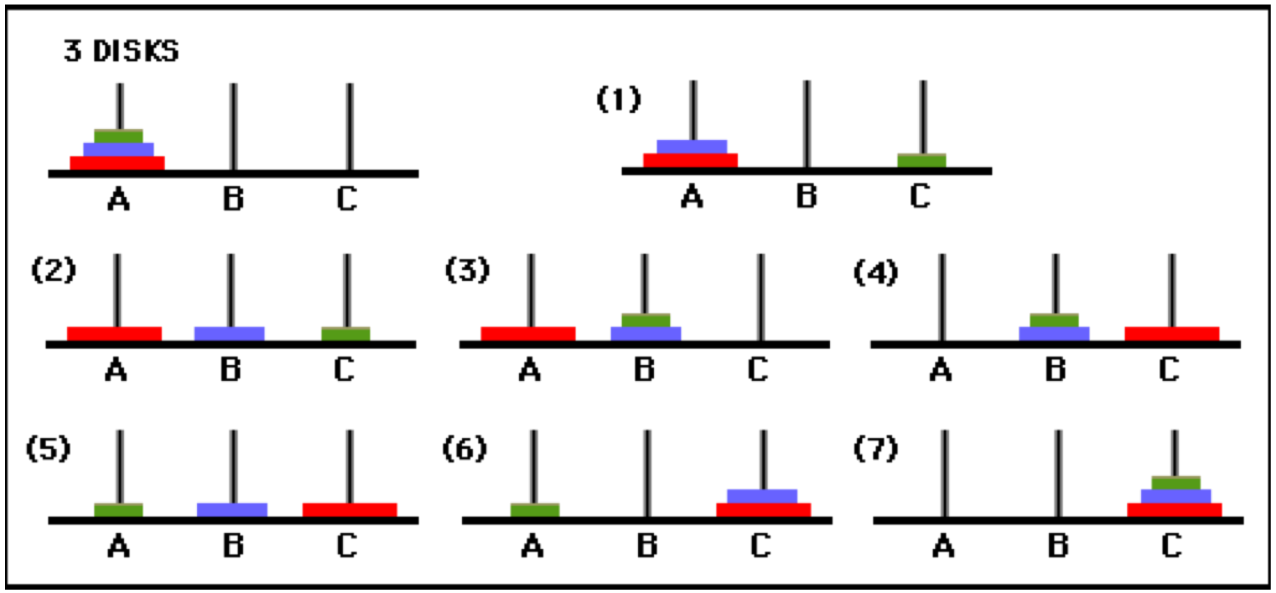
פתרון עבור n=2

- העבר את הטבעת הקטנה מ A לB
- העבר את הטבעת הגדולה לC
- העבר את הטבעת הקטנה מ B לC



פתרון עבור n=3

- נעביר את 2 הדיסקים הקטנים מ A לB
- שימו לב שאנחנו יודעים לעשות זאת!
- נעביר את הגדול לC
- נעביר את השאר מ B לC עם A כעזר



פתרון עבור n

- נעביר את הטבעות 1,2,..,n-1 ממגדל A למגדל B (C עזר)
- נעביר את הטבעת ה-n ממגדל A למגדל C
- נעביר את הטבעות 1,2,..,n-1 ממגדל B למגדל C (A עזר)

```
% n= number of disks
% init-start tower
% temp
% final-our goal
function hanoi_tower(n,init,temp,fin)
% just move from init to fin
if n ==1
    disp(['From ' init ' move to ' fin])
else
    hanoi_tower(n-1,init,fin,temp)
    hanoi_tower(1,init,temp,fin)
    hanoi_tower(n-1,temp,init,fin)
end
end
```

```
>> hanoi_tower(3,'A','B','C')
From A move to C
From A move to B
From C move to B
From A move to C
From B move to A
From B move to C
From A move to C
```

Tic toc

```
% start first timer
start_time = tic;
% loop boundaries
from = 10000;
to = 10003;
for i = from:to
    % start second timer
    find_start = tic;
    M = rand(i);
    [r,c] = find(M < 0.231);
    % stop second timer
    find_stop = toc(find_start);
    fprintf("find_%d,%d: %d seconds\n", i, i, find_stop);
end
% stop first timer
overall = toc(start_time);
fprintf("\noverall: %d seconds\n", overall);
```

```
overall: 1.520956e+00 seconds
>> tic_toc_8
find_10000,10000: 1.866040e+00 seconds
find_10001,10001: 1.855307e+00 seconds
find_10002,10002: 1.757608e+00 seconds
find_10003,10003: 1.779602e+00 seconds

overall: 7.259898e+00 seconds
```

Plot

- משמש על מנת ליצור גרפים.
- ניתן להגדיר צבעים, סגנונות וסוגים שונים של גרפים.

- Syntax:
  - Plotting a single line:

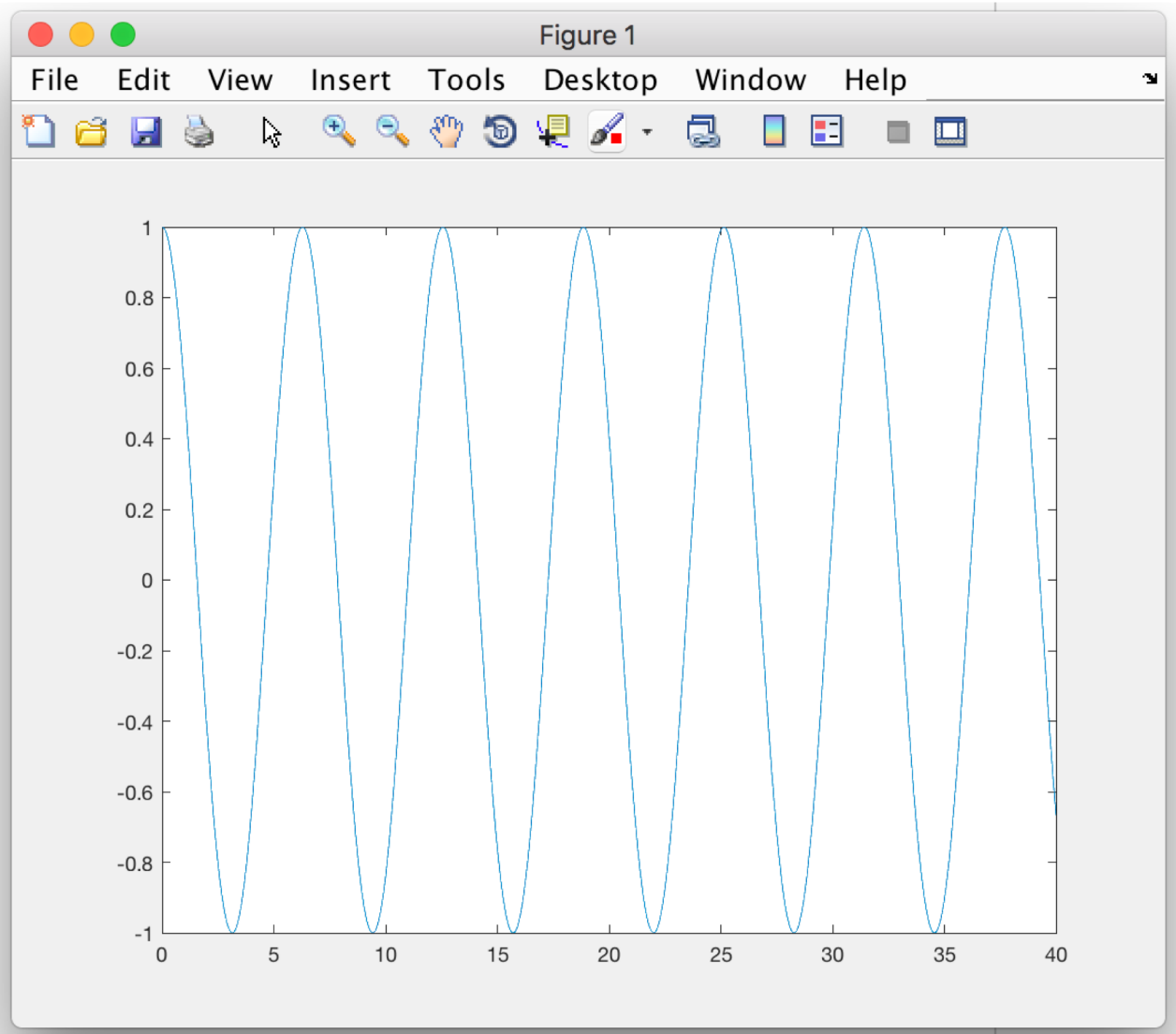
```
plot(xdata, ydata, 'color_linestyle_marker')
```

  - Plotting multiple lines:

```
plot(x1, y1, 'clm1', x2, y2, 'clm2', ...)
```

דוגמה

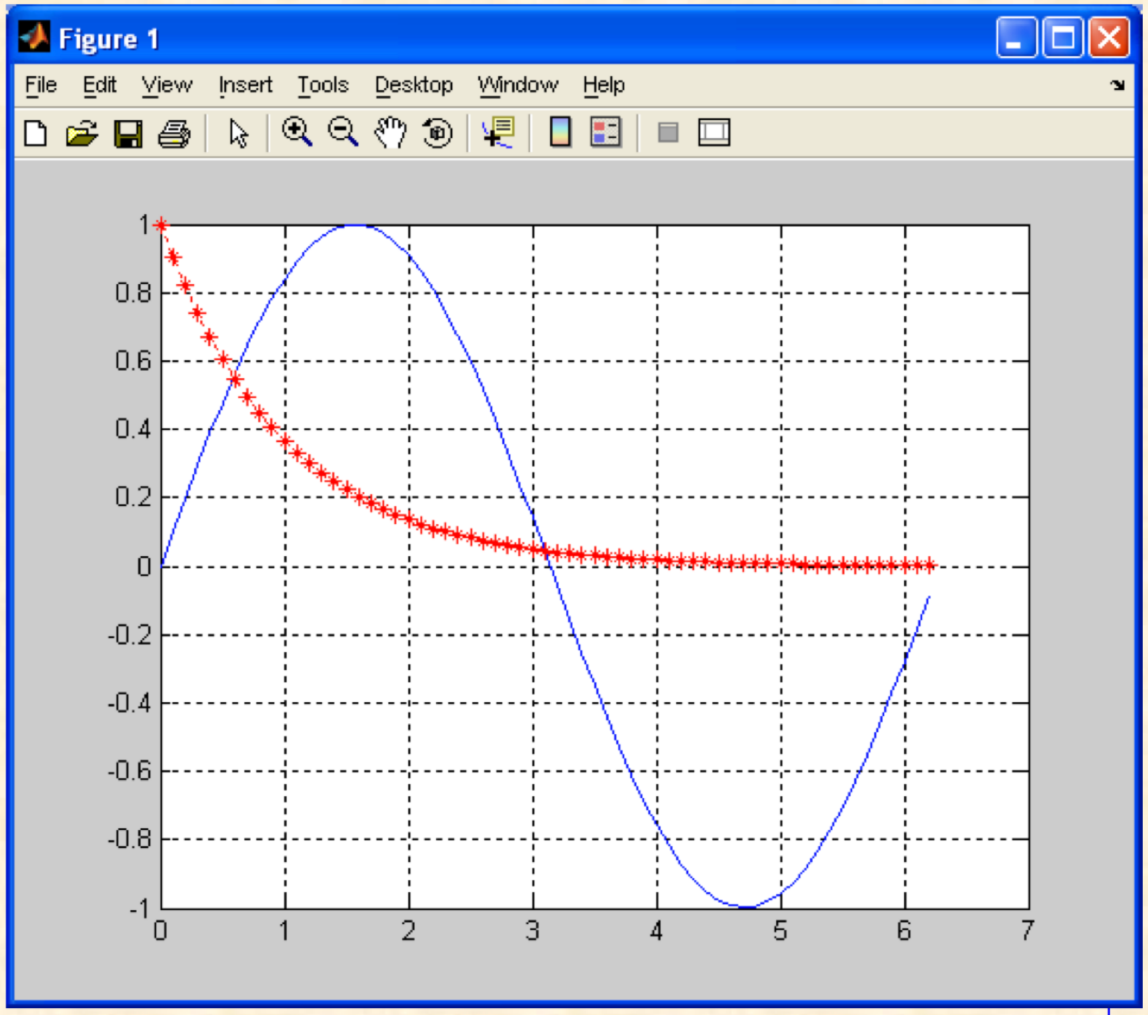
```
x=0:0.01:40;
y=cos(x)

plot(x,y)
```



Hold On

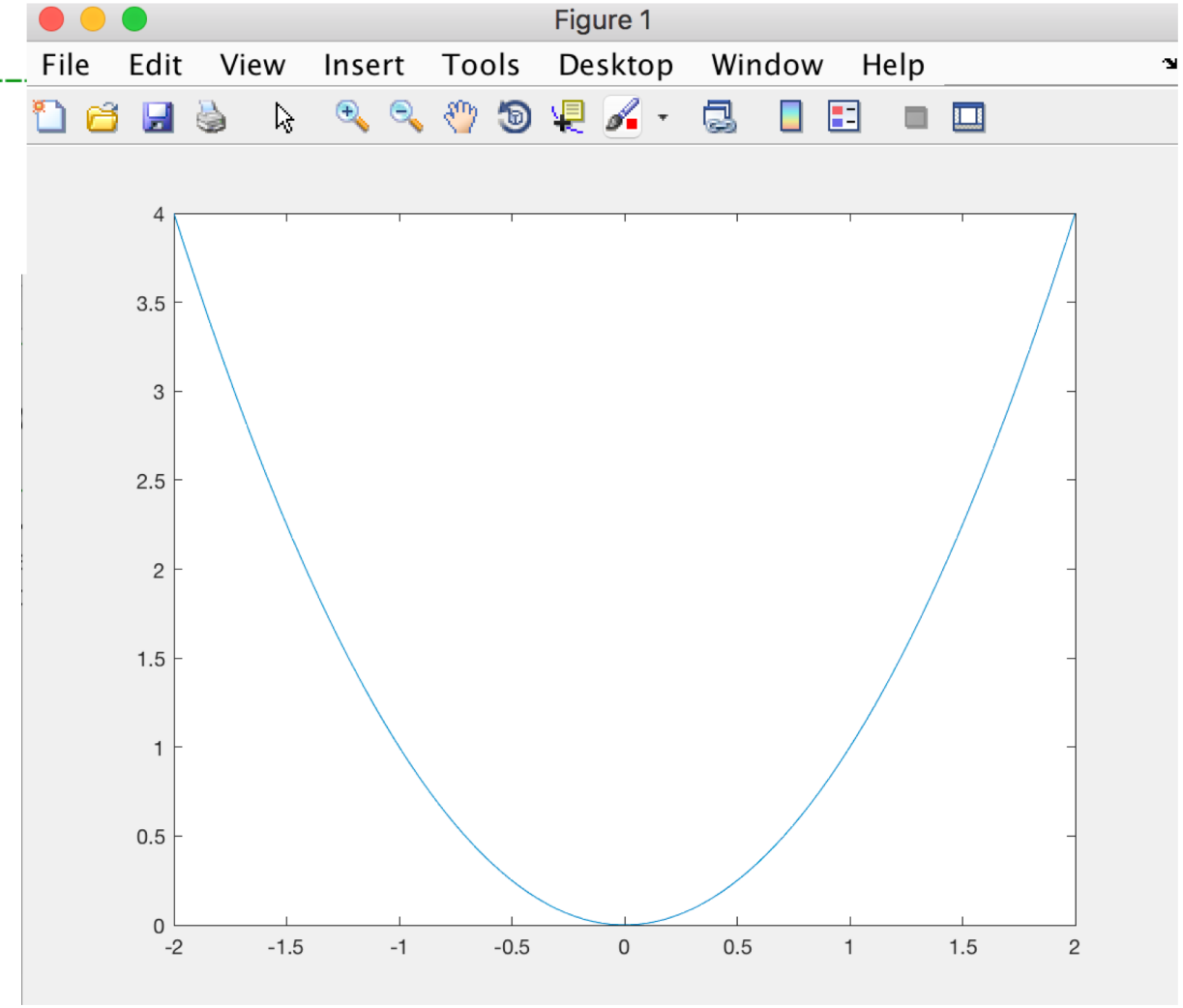
- HOLD ON holds the current plot
- HOLD OFF releases hold on current plot
- HOLD toggles the hold state

```
>> x = 0:.1:2*pi;
>> y = sin(x);
>> plot(x,y)
>> grid on
>> hold on
>> plot(x,exp(-x),'r:*')
```



Anonymous function

```
% anonymous function ----------------------------------------
% func_name =@ (input args) mathematical expression;
sum_1 =@ (x, y) x + y;

% plot y=x^2 ------------------------------------------------
power2 =@ (x) x.^2;
x_axis = linspace(-2,2,100);
plot(x_axis, power2(x_axis))
```



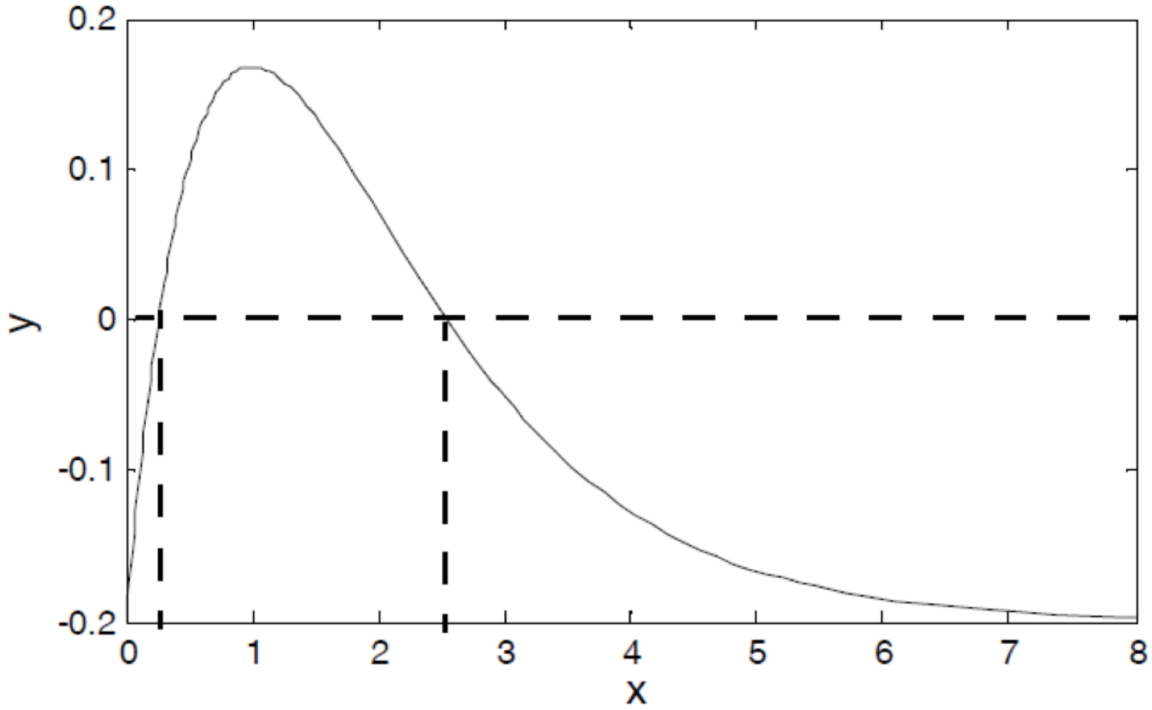
fzero

`x = fzero(@func,x0)` - חיפוש שורש ליד נק' התחלה.

תרגיל: מצאו את השורשים של הפונקציה הבאה

$$f(x) = xe^{-x} - 0.2 = 0$$

fzero




```
>> x1=fzero('x*exp(-x)-0.2',0.7)

x1 =
    0.2592
```

❶

```
>> F=@(x)x*exp(-x)-0.2
F =
@(x)x*exp(-x)-0.2
>> fzero(F,2.8)
ans =
2.5426
```

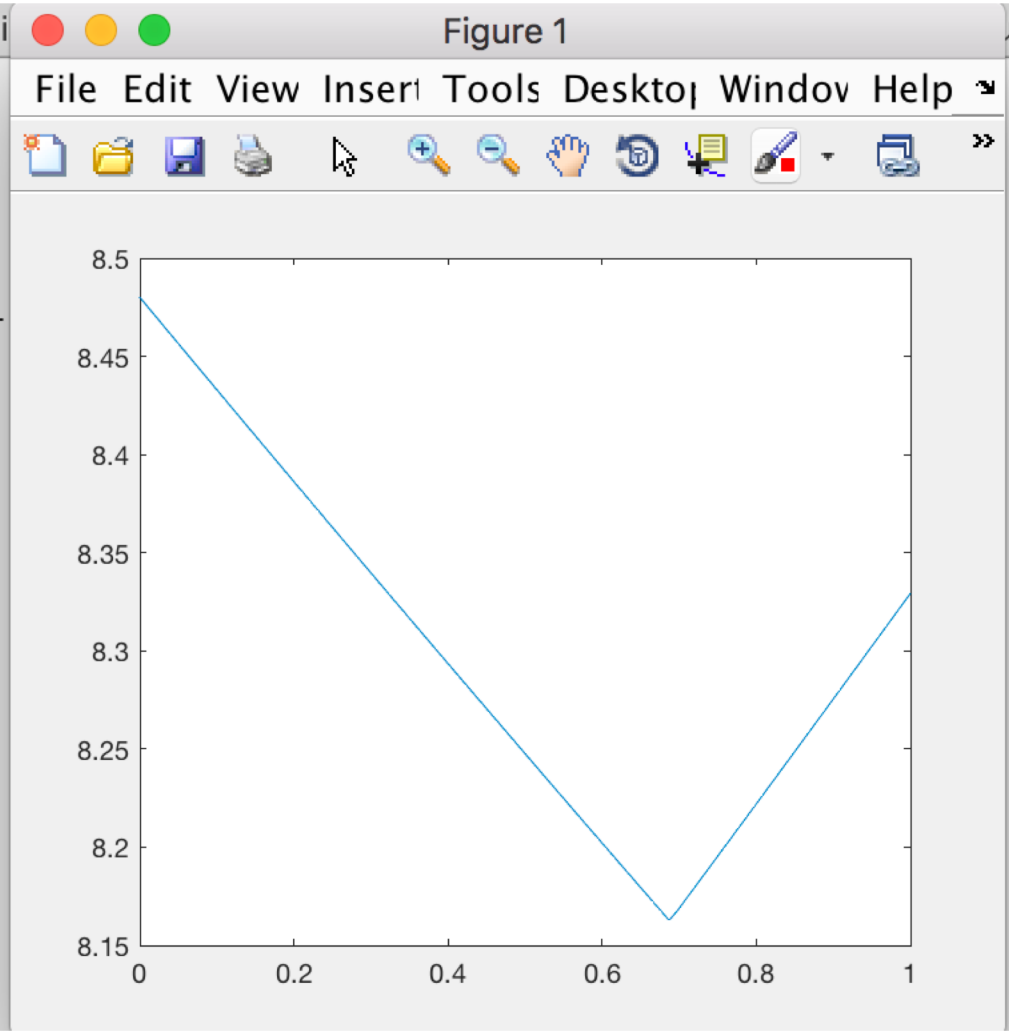
❷

תרגיל

נגדיר פונקציה $f(x)$ שהיא הערך העצמי הגדול ביותר בערך מוחלט של המטריצה $\begin{pmatrix} 1 & -7 & 2 \\ -7 & x & 4 \\ 2 & 4 & 1 \end{pmatrix}$. שרטטו את $f(x)$ בתחום $[0,1]$.

untitled3.m | untitled4* | plot_effici

```
function plot_eig_6()
N = 100;
x_axis = linspace(0,1,N);
y_axis = zeros(1,N);
for i = 1:N
    y_axis(i) = max_eig_6(x_axis(i
end
plot(x_axis, y_axis);
end

function max_val =  max_eig_6(x)
M = [1 -7 2; -7 x 4; 2 4 1];
max_val = max(abs(eig(M)));
end
```



untitled5.m

Tic Toc

```
function [] = plot_efficiency_8()
% number of itterations
N = 500;
% axis for plot
x_axis = 1:N;
y_axis = [];

for i = 1:N
    % option a - vector with n elements - O(n)
    % v = randi(100, i, i);
    % option b - n x n matrix - O(n^2)
     v = randi(100, i, i);
     x = randi(100);

    % start timer
    tic;
    search(v, x);
    % stop timer
    y_axis(end + 1) = toc;


end
plot(x_axis, y_axis)
```

```
function found = search(M, x)
found = false;
% get M size
[rows, cols] = size(M);
for i = 1:rows
    for j = 1:cols
        % search for x
        if M(i,j) == x
            found = true;
            break
        end
    end
end
end
```

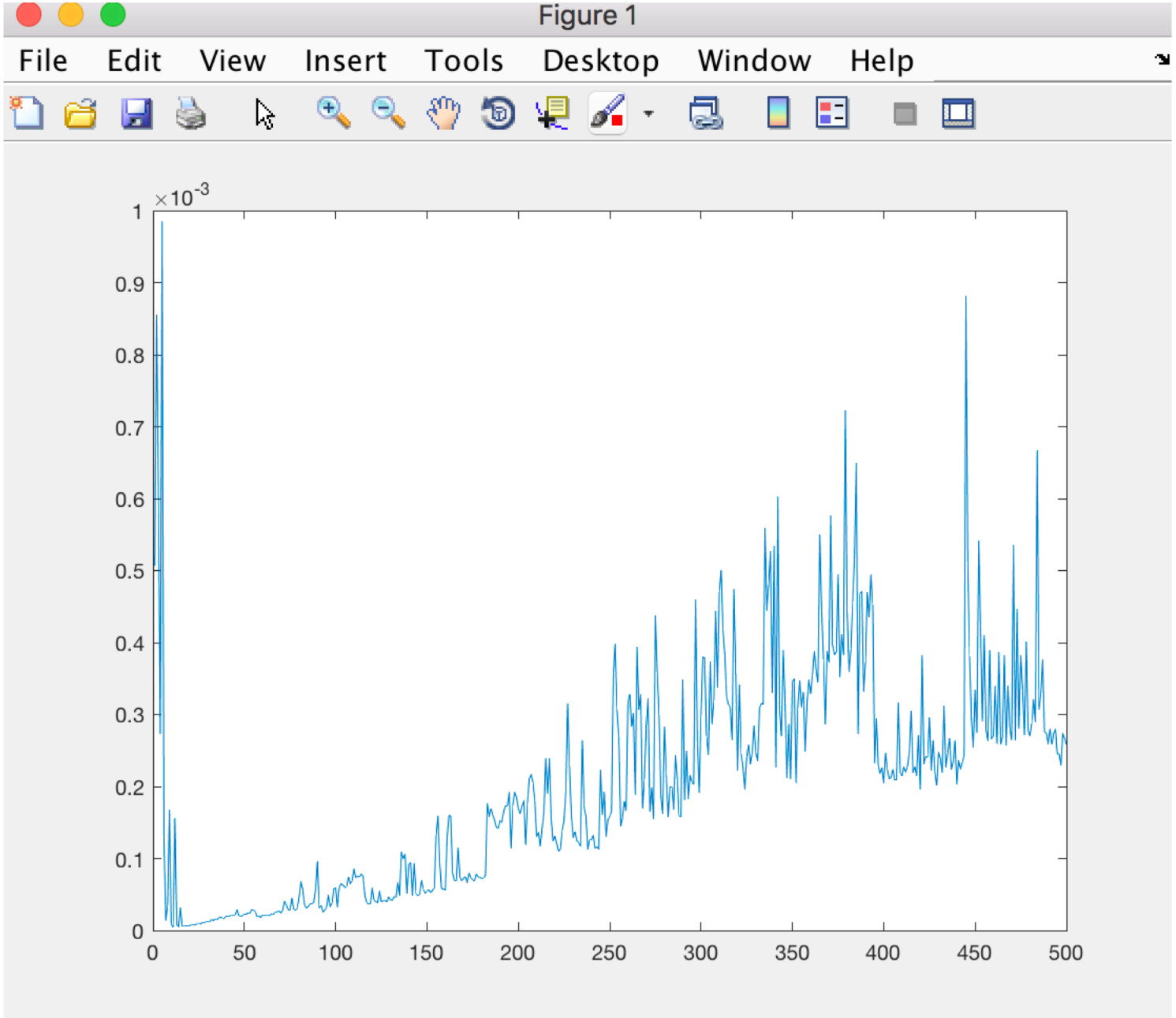


Figure 1
File Edit View Insert Tools Desktop Window Help
×10^-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 50 100 150 200 250 300 350 400 450 500

```
time_simple=zeros(20,50);
time_b1=zeros(20,50);
time_b=zeros(20,50);
Nvec=floor(logspace(3,7,20));
for k=1:50
    for i=1:length(Nvec);
        N=Nvec(i);
        target=floor(N/2);
        v=1:N;
        tic
        index=simplesearch(v,target);
        time_simple(i,k)=toc;
        tic
        index=binarysearch1(v,target);
        time_b1(i,k)=toc;
        tic
        index=binarysearch(v,target,1,N);
        time_b(i,k)=toc;
    end
end
ts=mean(time_simple,2);
tb1=mean(time_b1,2);
tb=mean(time_b,2);
figure(1)
clf
loglog(Nvec,ts,'b')
hold on
loglog(Nvec,tb1,'r')
loglog(Nvec,tb,'k')
legend('simple search','binary', 'recursive binary')
```