

עיבוד תמונה

הרצאה 2

מבוא למאטלב

שפת תכנות מתמטית.

Case sensitive

אפשר להריץ שורה אחרי שורה

המשתנה הבסיסי במאטלב הוא מטריצה

או שמשתמשים בפונקציות או בסקריפטים

הסיומת של שם קובץ תהיה filename.m

סתר דוגמא

יצירת מטריצה רנדומלית בגודל של 8x8

```
tmMatlab.m x trainNetwork.m x m  
spmd  
    R = rand(8);  
end
```

מן הסתם אפשר לשמור אותה $a = \text{rand}(8)$ וזה יישמר לנו בסביבת עבודה

משתנים פרימיטיביים

אין צורך להגדיר משתנים, סוג המשתנה נקבע בעת ההשמה (כמו בפייתון)

כדי למנוע הצגה של הפלט נשתמש בנקודה פסיק ;

פעולות חישוב בסיסיות

```
>> sqrt(a)  
  
ans =  
  
    2.2361  
  
>> mod(a,b)  
  
ans =  
  
    2
```

```
>> 3^2  
  
ans =  
  
    9  
  
>> log(a)  
  
ans =  
  
    1.6094  
  
>> log(b)  
  
ans =  
  
    1.0986
```

```
>> a=5;  
>> b=3;  
>> a*b  
  
ans =  
  
    15
```

סקלר הוא מטריצה בגודל 1 על אחד

וקטור הוא מטריצה ממימד 1

ייצור מטריצות וקטורים..

סקלר = 5 = a

וקטור שורה: a = [1 3 4 7 9]

וקטור שורה עם קפיצות של 2
מ 1 עד 9

```
>> a = [1:2:9]
a =
     1     3     5     7     9
```

וקטור עמודה

```
>> a = [1;3;4;7;9]
a =
     1
     3
     4
     7
     9
```

ייצור מטריצה

```
>> A = [11 12 13; 21 22 23; 31 32 33]
A =
    11    12    13
    21    22    23
    31    32    33
```

לייצר תת מטריצה מ A שתכיל

את שורות 2 ו 3 ועמודות 2 ו 3

```
>> B = A(2:3, 2:3)
B =
    22    23
    32    33
```

תקח את כל השורות החל משורה 2
ואת העמודות 1 2 3 בהן

```
>> B=A(2:end ,1:3)
```

```
B =  
  
    21    22    23  
    31    32    33
```

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

```
B =  
  
    11    13  
    21    23  
    31    33
```

מטריצות zeros ones – מטריצה שמכילה רק אפסים ומטריצה שמכילה רק אחדות

Eye מטריצת יחידה

Rand מטריצה אקראית (אפשר גם להכפיל אותה בכל מספר כדי להגדיל ערכים)

```
>> eye(3,3)
```

```
ans =  
  
    1    0    0  
    0    1    0  
    0    0    1
```

```
>> rand(2,2)
```

```
ans =  
  
    0.8147    0.1270  
    0.9058    0.9134
```

```
>> zeros(5,6)
```

```
ans =  
  
    0    0    0    0    0    0  
    0    0    0    0    0    0  
    0    0    0    0    0    0  
    0    0    0    0    0    0  
    0    0    0    0    0    0
```

```
>> ones(2,3)
```

```
ans =  
  
    1    1    1  
    1    1    1
```

אפשר לעשות את כל הפעולות שיש על מטריצות

$C = A+B;$	– חיבור מטריצות: אותו גודל
$C = A+5;$	או עם סקלר
$C = A*B;$	– כפל מטריצות
$A.*B \quad A.^2 \quad A./B$	– פעולות איבר-איבר (להוסיף . לפני האופרטור
$\text{sqrt}(A), \text{abs}(A), \text{eig}(A)$	– להפעיל פונקציה על מטריצה
$v = A(:);$	– להפוך מטריצה לוקטור עמודה
$\text{sumCol}=\text{sum}(A,1); \text{sumRow}=\text{sum}(A,2);$	– סכום איברים לאורך עמודות/שורות
$\text{sum}(A(:))$ or $\text{sum}(\text{sum}(A,1),2)$	– סכום איברים של כל המטריצה
$[\text{numRows} , \text{numCols}] = \text{size}(A);$	– בירור גודל של מטריצה

$\text{Eig}(A)$ הערכיים העצמיים של A

אפשר לסכום את כל המטריצה, לפי שורות\ לפי עמודות

גודל מטריצה:

```
>> [R_A C_A]=size(A)
```

```
R_A =
```

```
3
```

```
C_A =
```

```
3
```

האינדקס הראשון הוא שורה והשני עמודה

נשתמש בc and r במקום x וy

כשבונים וקטור ממטריצה קודם נלקחים איברי העמודה הראשונה וכך הלאה

אבל אפשר גם לסדר לפי שורות

סוגי משתנים

U מציין מספרים חיוביים בלבד והמספר שאחריו מסמן כמה בתים מוקצים

נגיד uint8

אפשר להגדיר ואפשר לא להגדיר

אפשר להמיר באופן מפורש: `A = uint8(A)`
 אם מקצים, אפשר לקבוע ('int16', [200 300]) `A = zeros`
 לוודא שלא חורגים מהתחום של המשתנה
 איך יודעים מה סוגי המשתנים?

`>> whos`

Name	Size	Bytes	Class
a	3x3	9	uint8
ans	6x2	96	double
b	3x3	9	uint8
c	17x1	136	double
r	17x1	136	double

– הפקודה `whos`
 – אפשר גם בחלון `workspace`

ברירת המחדל היא `double`, אבל יש פונקציות שמחזירות סוגים אחרים...
 לעיתים נבחר משתנים כמו `uint8` כדי לחסוך בזיכרון...

השימוש במילה
`whos`

מביא לנו את כל סוגי
 המשתנים שעבדנו
 איתם

`>> whos`

Name	Size	Bytes	Class	Attributes
A	3x3	72	double	
B	3x3	72	double	
C_A	1x1	8	double	
R_A	1x1	8	double	
a	5x1	40	double	
ans	1x2	16	double	
b	1x1	8	double	
index	1x1	8	double	

תנאים לוגיים וחיפושים

אופרטורים לוגיים:

– Not (~),

– שוויון == ,

– שונה מ ~ = ,

– גדול או קטן מ < >

תן את המספרים שהמודולו 2 שלהם
יהיה שווה ל 0

```
>> a = [1 2 3 ; 4 5 6 ; 7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> inds = find(mod(a , 2) == 0)
inds =
     2
     4
     6
     8
```

תן את המיקומים של האיברים שמקיימים את
המודולו הזה
צד שמאל זה עמודה צד ימין זה שורה

```
>> [r,c] = find(mod(a , 2) == 0); [r c]
ans =
     2     1
     1     2
     3     2
     2     3
```

```
>> inds = find(mod(a,2) , 2 , 'first')
inds =
     1
     3
```

גרפיקה ותצוגה

figure פתח חלון שאין בו כלום אבל יש חלון כרגע

Plot(x,y) לצייר את X ו Y

Mesh פונקציה דו ממדית

Contour מטריצה דו ממדית בצורה טופוגרפית

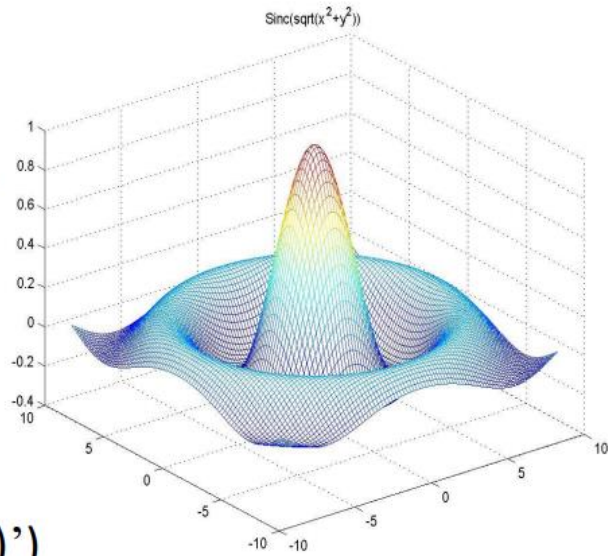
Subplot לחלק את החלון לכמה גרפים באותה צורה

axis([xmin xmax ymin ymax]) = לקבוע גבולות

title('This is the title') adds a title to
current axes

Sinc (דוגמה) 2D-3D

```
>> a=-9:0.2:9;  
>> [x,y] = meshgrid(a);  
>> R = sqrt(x.^2+y.^2)+eps;  
>> Z = sin(R)./R;  
>> mesh(a,a,Z)  
    ✖  
>> mesh(x,y,Z)  
>> title('Sinc(sqrt(x^2+y^2))')
```



הערה כותבים אחרי %

להמשיך בשורה נוספת כשהפקודה ארוכה שמים "..."

קצת על Debugging

כיוון שמטלב הוא אינטרפטר, אפשר לעצור אחרי כל פקודה, לבדוק את מצב המשתנים, לשנות את ערכם, לבדוק השפעה של שורות קוד ספציפיות, וכו' – מאד נוח.

לשים breakpoint – לוחצים על ה "-" ליד מספר השורה

מקשים שימושים:

– F5 – המשך לרוץ

– F10 – להריץ את הפקודה הבאה

– F11 – להריץ את הפקודה הבאה, ואם זאת פונקציה, להיכנס לתוכה

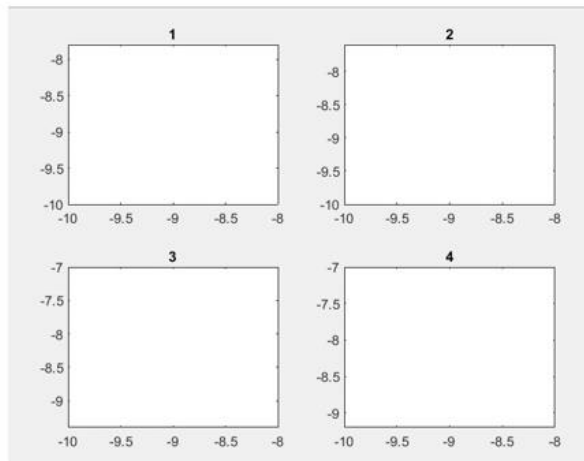
– F9 – להריץ את הקוד המסומן (כמו להעתיק לשורת הפקודה)

כלי נוח: %% Cell Coding
 – כל הקוד בין %% ל%% הבא נחשב אזור אחד
 – מאפשר להריץ בקלות אוסף פקודות מתוך סקריפט או ב debugging
 – מומלץ לקרוא עוד ב HELP.
 לעיתים נרצה לשמור את הסביבה (כלומר המשתנים):
 – save('filename.mat');
 filename.mat
 – load('filename.mat');
 טוען את כל המשתנים מהקובץ (אם קיים
 כבר משתנה כזה, אז דורס אותו!)
 – save('filename.mat', 'a', 'Mat', 'Im');
 המשתנים a, Mat, Im (לשים לב – מרכאות סביב שמות
 המשתנים).
 – load('filename.mat', 'Mat', 'Im');
 טוען את המשתנים
 האלה (אם לא קיימים, מטלב נותן אזהרה)

פקודות

```
switch (rem(n,3)==0) & (rem(n,2)==0)
    case 0
        disp('n is not dividable by 6')
    case 1
        disp('n is dividable by 6')
    otherwise
        error('This is impossible.')
end
```

```
if i==1,
    statement;
end
if res(n,2) ~= 0,
    statement;
else,
    statement;
end
```

```
for n=1:1:4
    subplot(2,2,n)
    plot(a(:,1),a(:,n+1))
    title(num2str(n))
end
```

```
a = 4; fa = sin(a);
b = 2; fb = sin(b);
while a - b > 5 * eps,
    x = (a+b)/2; fx = sin(x);
    if sign(fx) == sign(fa),
        a = x; fa=fx;
        break;
    else
        b = x; fb = fx;
    end
end
```

Tic toc אומר כמה זמן לקח להריץ את הדבר (מחשב את הזמן)

```
>> a = rand(3000);
>> b = a; tic; for c1 = 1 : numel(a), b(c1) = a(c1).^2 + 3; end; toc;
Elapsed time is 18.259007 seconds.
>> tic; c = a.^2 + 3; toc
Elapsed time is 0.175207 seconds.
```

הצגת מטריצה/תמונה

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

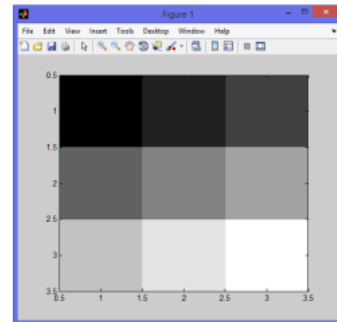
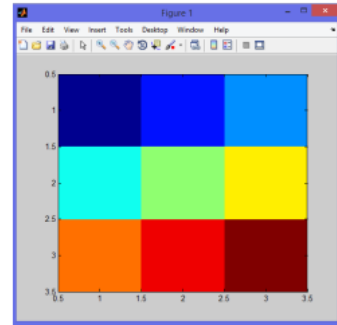
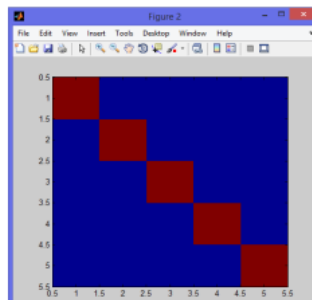
```
>> imagesc(A) →
```

```
>> colormap gray
```

```
>> B = eye(5);
```

```
>> figure, imagesc(B)
```

```
>> close all
```



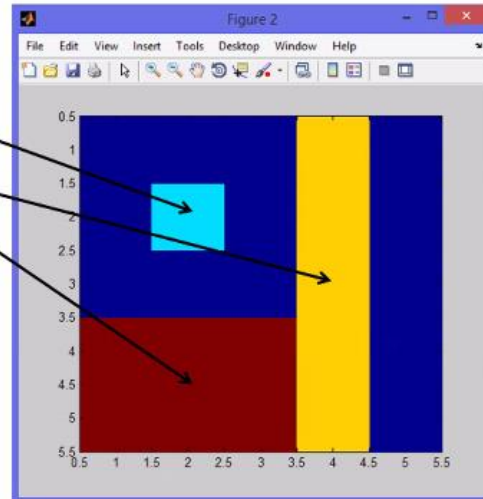
גישה לערכים במטריצה

Matrix(row, column)

Matrix(:,2) = כל השורות בעמודה מסוימת נגיד 2

השמת ערכים במטריצה קיימת

```
>> A=zeros(5);
>> A(2,2)=1;
>> A(:,4)=2;
>> A(4:5,1:3)=3;
>> imagesc(A)
```



מטריצה
גבוהה

מטריצות מסדר גבוה

```
A3 = zeros(3,3,4); % create 3X3X4 matrix of zeros
```

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

```
A3(:, :, 1) = A;
```

```
A3(:, :, 2) = A;
```

```
A3(:, :, 3) = A;
```

```
A3(:, :, 4) = A;
```

```
A3(1,2,4) → 2
```

```
A3(3,2,1) → 8
```

```
A3(2,2,:) → ??
```

A3=



תמונות במאטלב

נייצג תמונות על ידי מטריצה דו ממדית

תמונות צבע יהיו כמטריצה תלת ממדית (מטריצה עבור כחול, עבור אדום ועבור ירוק)
והקומבינציה תהיה תמונת הצבע

imread קוראת תמונות בפורמטים רבים

imwrite כותבת בפורמטים רבים

לדוגמא נרצה לשמור בפורמט TIF ללא כיווץ נכתוב:

```
imwrite(I, 'path\filename.tif', 'tif', 'compression', 'none');
```

הפונקציה imfinfo נותנת מידע על התמונה מבלי לקרוא אותו ממש

אפשר גם לשמור חלון על ידי

```
print -djpeg filename or print -dtiff filename
```

figure פותחת כל פעם חלון תצוגה חדש

figure(k) עוברת לחלון מספר K אם קיים

imshow(im) פונקציה שמציגה את התמונה בחלון הנוכחי

surf הפונקציה מאפשרת תצוגה כמו של מפות גבהים

הכלי הגרפי מאפשר לעשות Zoom, לבדוק ערך של פיקסל, ועוד.

כלים שימושיים נוספים:

– impixelinfoval

– imcrop

– zoom on או zoom off

– Imresize מאפשר לשנות גודל של תמונה