



# Matlab

Last Update: March 2009  
Nimrod Peleg

# MATLAB

- Matlab היא שפת תכנות מתקדמת שמאפשרת עיבוד מידע, הדמיה, והצגה חזותית.
- ההבדלים העיקריים בין Matlab לשפות אחרות:
- המשתנים כאן מבוססים על וקטורים ומטריצות
- מספר גדול של פונקציות מובנות, החוסכות זמן למשתמש
- מאגר נרחב של פונקציות המיועדות לתחומים הנדסיים ספציפיים, כמו: עיבוד תמונה, עיבוד אותות, בקרה, סטטיסטיקה, פיננסים ועוד ועוד
- מימוש מהיר מאד לפעולות וקטוריות / מטריציות
- גרסה נוכחית: 2009a

# Help

- נותן מידע על כל הפונקציות ב Matlab
- מכיל את תתי השדות: Contents, Index, Search, Demos, Favorites
- Contents: מכיל את המידע לפי נושאים ובצורה מסודרת.
- Index: נותן תשובות לשאלות כאשר שם הפונקציה אינו ידוע (...image)
- Search: נותן תשובה כאשר שם הפונקציה ידוע. (ניתן גם לכתוב <שם הקובץ < help)
- Demos: מראה קוד ותוצאה של נושאים (FFT).

ראו גם: lookfor וכן הוראת: doc (החל מגרסה 6)

# שמות שמורים

■ **Inf**: ערך שואף לאינסוף –  $x/0$

■ **NaN**: Not a Number –  $0/0$ ,  $inf/inf$

■ **Pi**:  $\pi = 3.1416 \dots$

■ **Eps**: אפסילון – הערך הקטן ביותר במערכת.

■ ; - לא להציג תוצאת החישוב !

■ **ans**: תוצאת החישוב האחרון

■ **Format**: קובע מבנה תצוגה

■ Note the “long”, “Short” and more presentation precision  
“format” command

■ ועוד ...

# Matrices / Vectors Operations

- כל האיברים שאנחנו מציבים מוכנסים כמטריצות : גם "סקלרים" נחשבים למטריצות (בגודל  $1 \times 1$ ).
- ערכי הערכים במטריצות ייכתבו בתוך  $[ ]$ , כאשר הפרדה של שורה מסומנת ב ; או בכתיבת הערכים בשורה חדשה.
- כדי לפעול /להתייחס לערך של מטריצה  $X$ , כותבים  $X(i,j)$  – כאשר  $i$  הוא מס' השורה ו-  $j$  מס' העמודה.
- ניתן גם לבנות מטריצות מיוחדות:  
■  $eye(1)$ ,  $ones(3,3)$ ,  $zeros(2,6)$ .
- כדי למצוא גודל של מטריצה  $b$ , מציבים  $[x,y]=size(b)$

# Matrices / Vectors

■ פעולות על מטריצות:

■  $a=b+c, a=b-c, a=b*c', a=b^2, a=b/3, a=inv(b)$

■ ניתן גם לבצע פעולות על איברים מתאימים

במטריצות:  $a=b.*c$  - הכוונה ש-

$$a(1,1)=b(1,1)*c(1,1)$$

■ (כנ"ל  $a=b./c$ ).

לדוגמא:  $b.^2$  לעומת  $b^2$

# Matrices / Vectors

■ פקודות נוספות:

■  $\text{ceil}(b)$  – עובר על אברי המטריצה ומעגל למעלה  
מספרים לא שלמים לערך השלם הקרוב ביותר אליהם.

■  $\text{floor}(b)$  – כנ"ל – מעגל כלפי מטה.

■  $\text{round}(b)$  – מעגל ערכים במטריצה לערך הקרוב.

■  $\text{abs}(b)$  – נותן את הערכים המוחלטים של המטריצה (גם  
אם היא קומפלקסית)

■  $\text{sort}(b,d)$  – ממיין את המטריצה עפ"י מימד  $d$ .

# Matrices / Vectors

- כדי למצוא מיקום של ערך מסוים במטריצה, נשתמש בפונקצית `find`:  
 $[i,j,v] = \text{find}(b)$  :  $i,j$  מייצגים את מיקום הערכים השונים מ-0, ו-  $v$  את ערכם.
- כדי למצוא ערך ספציפי:  $[i,j]=\text{find}(b==3)$  – נקבל את מיקום הערכים השווים ל-3.
- כדי לעבוד על מימד אחד מתוך המטריצה נשתמש ב ":"  
 $a(2,:)= [1,2,3]$  .



# Matrices / Vectors

## ■ וקטורים:

□  $v=0:0.1:1$  – מגדיר וקטור בעל ערכים המתחילים ב 0 , תוך קפיצות של 0.1 ועד ל- 1.

□  $V=1:6$  – במקרה כזה הקפיצות הם בגודל 1.

□ ניתן גם להגדיר אותם כמטריצה בעלת שורה / עמודה אחת

$(v=[1 \ 2 \ 3], k=[5;3;1])$ .

□ דרך נוספת היא ע"י  $x=\text{ linspace}(0,2,12)$  : שמתאר וקטור עם

12 ערכים במרחק שווה זה מזה החל מ- 0 וכלה ב- 2

# Matrices/ Vectors

■ הפקודות הבאות עוברות על העמודות של המטריצה ומחזירות וקטור תוצאה:

■ *mean(b)*, *min(b)*, *max(b)*, *sum(b)*

לכן, כדי לחשב סכום כללי של מטריצה, צריך לכתוב  
*sum(sum(b))*

■ אורך של וקטור יחושב ע"י *length(k)*

■ מכפלה של ערכי וקטור/מטריצה יחושבו ע"י *prod(k)*

■ *Reshape* משמש למעבר מוקטור למבנה מטריצי

■ *Fliplr, flipud* : היפוך סדר עמודות/שורות

■ *Repmat* : שכפול תבנית של מטריצה

# Relations

■  $C = (b == 2)$  : מקבל את המבנה של  $b$  ושם "1"  
בכל מקום בו  $b=2$  הוא true.

■  $C = (b > 2 \ \& \ b \leq 4)$

■  $C = (b \sim = 2)$

## Sub-Matrices

•  $A = [1:12]$

•  $c = \max(b(:))$  ,  $C = b(:)$

•  $b(2:3, 1:2:4)$  ,  $b(2,:)$  ,  $B(:,2)$

•  $\text{Sum}(c(:))$

# Control Flow

- **If:**

```
if exp (בלי סוגריים)
    statement
elseif exp
    statement
else
end
```

- **For:**

```
for i=1:2:11
    statement
end
```

- **While:**

```
while exp
    statement
end
```

# Speech Related Functions

פונקציות המטפלות בקולות:

- Wavread: קוראת קבצים בפורמט MS-Wav:  
□  $(Y, FS, NBITS) = WAVREAD(FILE)$  : מקבלת את האות הקולי, וכן את תדר הדגימה והרזולוציה.
- Sound: משמיעה וקטור כקובץ קול  
□ Soundsc: כנ"ל, עם Linear Scaling
- LPC: מבצעת חיזוי לינארי למקדמי מסנן לפי סדר N
- Lloyds: כימוי אופטימלי (לפי מינימום שגיאה ריבועית ממוצעת)
- AWGN: הוספת רעש גאוס ללבן לאות (פרמטר: SNR)

# Image Related Functions

- ישנן מס' פונקציות המטפלות בתמונות:
- imread : קורא תמונה למטריצה, כאשר ערכי המטריצה הם מ-0 ועד 255 (עפ"י דרגות האפור).  
*mat=imread('picture\_name.xxx')* □
- imshow : מציג את המטריצה כתמונה. אם בצענו פעולה
- על המטריצה (כמו התמרת dct יש צורך להוסיף את הפקודה:
- *mat=uint8(mat)* לפני imshow(mat)
- imhist : בונה היסטוגרמה של התמונה עפ"י דרגות האפור שלה.

# Image Functions

■ *Mat1=blkproc(mat,[m,n],func)*

מפעילה את פונקציה func על בלוקים בגודל  $m \times n$  על התמונה שנקראה ב mat.

■ *Mat1=dct(mat)*

■ ביצוע התמרת dct על התמונה שנמצאת ב-mat.  
המטריצה המתקבלת ב Mat1 היא מקדמי ה-DCT

■ IMAGEESC : Scale data and display as image

# Graph Functions

- *plot (x,y option: color, shape)*
- *subplot (rows, col, graph num)*
- *fill (x,y, option:color)*
- *figure*
- *hold on/off*
- *xlabel / ylabel ('text')*
- *grid on*
- *Surf(2d matrix)*



# Mask's

*$[x,y]=\text{Meshgrid}(xmin,xmax,ymin,ymax)$*  ■

יוצר שתי מטריצות בתחום המוגדר כאשר שילוב של שתיהן נותן לנו מערכת קורדינטות מלאה של התחום

*$m=\text{conv2}(im,mask)$*  ■

מבצע קונבולוציה של התמונה עם המסכה  
ניתן להשתמש בפרמטר 'same' לשמור על הגודל

# Files/ Functions

- בד"כ נהוג לשמור כל פונקציה בקובץ נפרד, כך ששם הקובץ הוא שם הפונקציה.
- הקבצים מתחלקים לשלושה סוגים:
  - קבצי M (קבצי תכנות), מורצים ע"י command window
  - קבצי mat : שומרים ערכי משתנים
  - קבצי fig : GUI.

# Files/ Functions

- פונקציות יכולות לקבל ארגומנטים ולהחזיר מס' משתנים (בגדלים שונים).
- ניתן לשנות בפונקציה את ערכי הארגומנטים (אבל בסיום הפונקציה כל הערכים, חוץ מהמוחזרים מתבטלים/ חוזרים לערכם הקודם).
- פונקציה נקראת בצורה הבאה:
- *function [x,y,z]= example (mat)*

ארגומנט      שם הפונקציה      ערכים מוחזרים