Matlab בסיסי

ניר אדר

Nir Adar
Email: underwar@hotmail.com
Home Page: http://underwar.livedns.co.il

אנא שלחו תיקונים והערות אל המחבר.

1. תוכן עניינים

2. מבוא

Matlab היא שפת תכנות הכוללת סביבה אינטראקטיבית לחישוב מדעי והנדסי, סימולציה, ויזואליזציה ותכנון אלגוריתמים. הבדלים בינה לבין שפות אחרות:

- המשתנים הבסיסיים ב-Matlab הינם ווקטורים ומטריצות, ולא סקלרים.
- כוללת מגוון רחב של פונקציות וספריות המיועדת לתחומים הנדסיים ספציפיים כמו עיבוד אותות, עיבוד תמונות, בקרה ועוד.

2.1. קבלת עזרה

כדי לקבל עזרה על פונקציה:	help func_name
כדי לחפש פונקציה ששמה לא ידוע	lookfor keyword
כדי לקבל כלי עזרה אינטראקטיבי	helpwin/helpdesk
קבלת איזה משתנים קיימים בשדה העבודה	whos

2.2. סוגי קבצים ב-Matlab

1. סיומת m

קבצים אלו מבצעים פקודות Matlab. עריכתם והרצתם נעשית דרך חלון ה-Editor/Debugger.

א) script – אינם מקבלים קלט או מחזירים פלט – פועלים על מידע הנמצא ב-workspace. הרצת הקובץ נעשית על ידי F5.

ב) function – יכולים לקבל קלט ומחזירים פלט. המשתנים לוקליים לפונקציה.

2. סיומת mat

קבצים אלו מאכסנים משתנים וערכיהם.

3. סיומת fig

מאכסנים גרפים.

2.3. פונקצית clc

פונקציה זו מנקה את חלון ה-command window ומביאה את הסמן אל ראש החלון.

2.4. פונקצית clear

פונקציה זו מוחקת את כל המשתנים המוגדרים ב-workspace.

2.5. שמות משתנים

מוקצים עד 31 תווים לכל שם משתנה, כאשר Matlab מתעלם מכל תו נוסף.
שם משתנה יכול להיות מורכב ממספרים, אותיות וקו תחתון, כאשר Matlab מבדילה בין אותיות גדולות לקטנות.

2.6. מספרים ב-Matlab

חזקות - $2e5 = 2 \cdot 10^5$.

מספרים קומפלקסים - $2+3i, 3+7j$, כאשר $i, j = \sqrt{-1}$.

אינסוף - $\inf = \infty$.

לא מספר – NaN.

2.7. סיום העבודה ב-Matlab

נעשה על ידי הפקודה quit או צירוף המקשים Ctrl+Q.

3. הגדרת מטריצות

3.1. הגדרה ידנית

הגדרה ידנית היא הגדרת משתנה מסוג מטריצה והצבת ערכים בתוכו. צורות הכתיבה הבאות שקולות:

1. פסיקים A=[2, 1, 5; 6, 3, 2]
2. רווח בין כל מספר A = [2 1 5 ; 6 3 2]

הערה: אם לא נגדיר שם משתנה השפה תגדיר משתנה בשם ans ותשמור בו את תוצאת החישוב.
אם נגדיר משתנה נוסף ללא שם השפה תמחק את הערך הקודם שהיה ב-ans ותכניס ערך חדש למשתנה ans. אם כן נגדיר משתנה, ans יישאר ללא שינוי.

- **הסימון ;** בסוף שורה תגרום ל-Matlab לא להדפיס את המטריצה שנוצרה לאישור.
- **מציאת גודל מטריצה קיימת** על ידי הפקודה size(A). הפונקציה מחזירה שני מספרים: m, n, כאשר m הוא מספר השורות ו-n מספר העמודות.
- **מציאת אורך ווקטור** על ידי הפקודה length(b).

3.2. פונקציות ליצירת מטריצות

- **יצירת מטריצה** $m \times n$ **שכולה אפסים** על ידי: Z=zeros(m, n)
- **יצירת מטריצה** $m \times n$ **שכולה אחדות** על ידי: Z=ones(m, n)
- **יצירת מטריצה אלכסונית**:
  1. יצירת ווקטור A, שהוא האלכסון של המטריצה. לדוגמא: A=[1; 2; 3].
  2. נשתמש בפקודה diag(A) – שייצר את המטריצה האלכסונית המתאימה.
- **קבלת האלכסון של מטריצה** A יעשה על ידי diag(A).
- **יצירת מטריצת יחידה** נעשית כך: I = eye(n) כאשר n הוא גודל המטריצה המבוקש.
- **הרכבת מטריצה מתת מטריצות** כך: B=[a b; c d] כאשר a, b, c, d מטריצות.

3.3. יצירת ווקטור בעל ערכים עוקבים

יצירת ווקטור בעל ערכים עוקבים נעשית בשתי דרכים.

דרך אחת: על ידי X = s:d:f.

כאשר s הוא הערך ההתחלתי של הווקטור, d הוא פקטור גידול או הקטנת ערכי הווקטור, ו-f הוא הערך הסופי. הווקטור הנוצר הוא: $X = [s, s+d, s+2d, ..., s+(n-1)d]$ כאשר $n = \frac{t-s}{d} + 1$.

הערות:

- רישום ללא ציון פקטור d זהה לקביעת d=1.
- אם $t \neq s + (n-1) \cdot d$ מספר הערכים יעוגל כלפי מטה.

דרך שנייה: פונקצית linspace.

יצירת ווקטור בעל איברים שמתחילים בערך מסוים וגדלים (או קטנים) בפקטור קבוע עד לערך אחר, אך כעת המשתמש קובע את מספר הערכים הרצויים בין שני הקצוות והפקטור מחושב באופן אוטומטי.

X = linspace(s, f, n)

פקודה זו תיצור ווקטור בעל n איברים שהאיבר הראשון בו הוא s, והאיבר האחרון f.

לדוגמא:

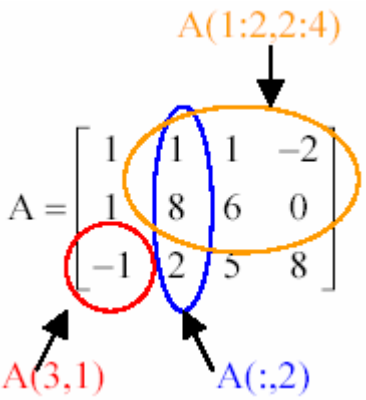
```
>> linspace(2, 10, 8)
ans = 2.0000    3.1429    4.2857    5.4286    6.5714    7.7143
8.8571   10.0000
>> linspace(2, 10, 9)
ans =  2     3     4     5     6     7     8     9    10
```

4. מניפולציות על מטריצות

4.1. אינדקסים למטריצה

ניתן לגשת לחלק מהאיברים בתוך מטריצה באמצעות סוגריים עגולות ואינדקסים. האיבר הראשון מציין מספר שורה והאיבר השני מציין מספר עמודה. האינדקס של האיבר השמאלי העליון הינו (1, 1).

המשמעות של האופרטור ":" היא כל האיברים. למשל, A(:, 2) היא כל השורות והעמודה השנייה.
A(1, :) היא השורה הראשונה וכל העמודות. הביטוי A(1, :) שקול לביטוי A(1, 1:4), ובמידה והמטריצה היא באורך 4 אזי הוא גם שקול לביטוי A(1, 1:end).
כדי לגשת אל איברי ווקטורים אין צורך לציי גם את השורה וגם את העמודה. מספיק אינדקס יחיד על מנת לגשת אליהם.
כתיבת end בסוף ווקטור האינדקס מחזירה אוטומטית את משפר השורה או העמודה המקסימאליים, בהתאם למיקומם בסוגריים העגולות. המחשה:



(ההמחשה לקוחה מחוברת שכתב **דורי פלג** בנושא Matlab הניתנת לרכישה בטכניון).

4.2. אופרטורים מטריציים

4.2.1. Transpose

עבור מטריצה ממשית A, נקבל את ה-Transpose שלה על ידי 'A.

עבור מטריצה מרוכבת A, 'A הינו הצמוד המרוכב שלה, ואילו 'A. הינו ה-Transpose שלה.

דוגמא:

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

A =
     1     2     3
     4     5     6
     7     8     9

>> A'

ans =
     1     4     7
     2     5     8
     3     6     9
```

4.2.2. סכימת סקלר ומטריצה

סכימת סקלר ומטריצה מוסיפה את הסקלר אל כל אחד מאיברי המטריצה.

4.2.3. חישוב דטרמיננטה

תהי A מטריצה ריבועית, אזי הפקודה det(A) תחזיר את הדטרמיננטה שלה.

4.2.4. היפוך מטריצה

תהי A מטריצה ריבועית $n \times n$. ישנן 3 דרכים למצוא את המטריצה ההופכית שלה:

1. inv(A)
2. A^(-1)
3. eye(n)/A

4.2.5. פונקצית sum

הפונקציה מקבלת ווקטור ומחזירה את סכום איבריו, או לחילופין היא מקבלת מטריצה ומחזירה את ווקטור סכום העמודות.

X ווקטור:

$$sum(X) = \sum_{i=1}^{n} X_i$$

X מטריצה:

$$sum(X) = \left[\sum_{i=1}^{n} X_{i,1}, ..., \sum_{i=1}^{n} X_{i,n} \right]$$

4.2.6. פונקצית cumsum

הפונקציה מקבלת ווקטור ומחזירה את ווקטור באותו אורך שבכל תא סכום איבריו המצטבר.

לחילופין היא מקבלת מטריצה ומחזירה את מטריצת סכום העמודות המצטבר.

4.2.7. פונקצית prod

הפונקציה מקבלת ווקטור ומחזירה את מכפלת איבריו, או לחילופין היא מקבלת מטריצה ומחזירה את ווקטור מכפלת העמודות.

X ווקטור:

$$prod(X) = \prod_{i=1}^{n} X_i$$

X מטריצה:

$$sum(X) = \left[\prod_{i=1}^{n} X_{i,1}, \ldots, \prod_{i=1}^{n} X_{i,n} \right]$$

4.2.8. אופרטור הנקודה

פעולות המבוצעות באמצעות אופרטור הנקודה משנות את המשמעות של המטריצות למערכים.

פעולת נקודה בין שתי מטריצות היא פעולה בין שני מערכים – פעולה איבר מול איבר.

אופרטור הנקודה ממוקם לפני אופרטור הפעולה החשבונית.

באמצעות אופרטור הנקודה ניתן לחשב ביטויים מתמטיים מסובכים ללא לולאות.

לדוגמא, חישוב הביטוי $s = \sum_{x=1}^{5} x^x$ יכול להיעשות בצורה הבאה:

```
x = 1:5;
s = sum(x.^x);
```

הערה חשובה: בחלוקת ווקטור בסקלר יש להשתמש תמיד באופרטור הנקודה, או לחילופין להכפיל את הווקטור ב- $\frac{1}{\text{scalar}}$ במקום לחלק.

4.2.9. פונקצית find

אופרטורים לוגיים ב-Matlab: ==, <, >, ~= (not equal), ~ (not)

הפונקציה find מקבלת תנאי לוגי ומחזירה את כל האיברים השונים מאפס במטריצה שמקיימים את אותו תנאי לוגי.
למשל הפקודה I=find(A>100) תחזיר את האינדקסים של כל האברים במטריצה A הגדולים מ-100.
הפונקציה מחזירה אינדקס יחיד במטריצה (שרץ לאורך העמודות) או זוג אינדקסים (x,y) (תלוי במספר פרמטרי החזרה), לדוגמא [I, J] = find(A>100) ישים את האינדקסים של העמודות ב-i והשורות ב-j.

שימוש נוסף של find הוא החזרת שורות שונות מ-0. הביטוי הבא: [I,J] = find(X)
מחזיר את כל העמודות והשורות של האיברים השונים מ-0 במטריצה (שימושי עבור מטריצות דלילות).
הביטוי [I,J,V] = find(X) מחזיר את כל העמודות והשורות של האיברים השונים מ-0 במטריצה, אולם בנוסף מחזירה ווקטור שלישי המכיל את האיברים הנ"ל.

נשים לב להבדל שבין [I,J,V] = find(X) לביטוי [I,J,V] = find(X~=0). התאים שיוחזרו הם אותם תאים, אבל הערך של V במקרה השני יהיה ווקטור אחדות.

4.3. פתירת מערכת משוואות ליניארית

נתונה מערכת של m משוואות ו-n נעלמים. נכתוב אותה בצורה: $A \cdot x = b$.
במידה ו- $n = m$, אז A היא מטריצה ריבועית, וניתן לפתור את מערכת המשוואות בצורה הבאה:

```
x=inv(A)*b;
```

עבור המקרה הלא ריבועי, נחשב את הפיתרון כך:

```
x=inv(A'*A)*A'*b;
```

4.4. מחרוזות

מחרוזת היא ווקטור שכל איבר בו מכיל תו יחיד. מחרוזת מוגדרת כטקסט בעל גרשיים משני צדדיו.
לדוגמא:

```
myString = 'Hello, World'
```

ניתן ליצור מטריצה של מחרוזות, וניתן גם להפעיל את הפונקציות שפועלות על מטריצות גם על מחרוזות.

כתיבת מערכי מחרוזות: במידה ונרצה ליצור מטריצה, או מערך מחרוזות, ממספר מחרוזות שאינן שוות באורכן נשתמש בפונקציה str2mat, למשל str2mat('one', 'two', 'three'). הפונקציה תרפד את המחרוזות הקצרות יותר ברווחים על מנת ליצור מטריצה בעלת אורך שורות שווה.

הפיכת מספרים למחרוזות: הפונקציה num2str ממירה ערכים מספריים למחרוזות.

5. גרפיקה

5.1. כללי

כל פונקציה ליצירת גרפים יוצרת באופן אוטומטי חלון גרפיקה בו משורטט הגרף.

הרצת פונקציה ליצירת גרפים כאשר כבר קיים גרף גורמת למחיקת הגרף הקודם.

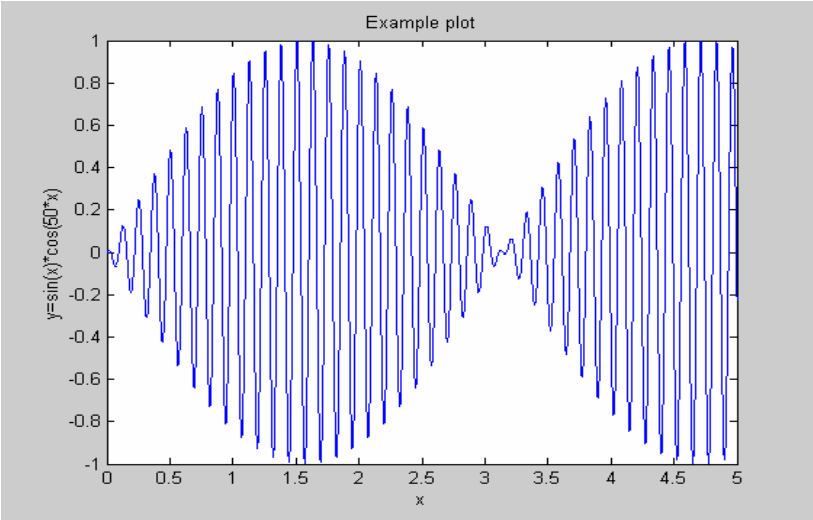
במקרים רבים נרצה ליצור מספר חלונות שיכילו גרפים. נוכל לפתוח חלון גרפי חדש על ידי הפונקציה figure(n) כאשר n הוא חלון גרפיקה שמספרו n. אם נכתוב את הפקודה figure ללא פרמטרים ייפתח חלון גרפיקה עם המספר הפנוי הבא.

5.2. דוגמא לגרף פשוט

נכתוב את הקוד הבא:

```
x=0:0.01:5;
y=sin(x).*cos(50*x);
plot(x,y)
title('Example plot')
xlabel('x')
ylabel('y=sin(x).*cos(50*x)')
```

התוצאה:



6. מבני בקרה

6.1. פקודת if

פקודה זו מתנהגת באופן דומה לפקודת if בשפות אחרות. התחביר שלה הוא כלהלן:

```
if  (expression),
        statement;
elseif (expression),
        statement;
elseif expression,
        statement;
else,
        statement;
end
```

6.2. לולאת while

לולאת while מאפשרת לחזור על קטע קוד מספר פעמים, כל עוד מתקיים התנאי הניתן לה. תחביר:

```
while expression
        statements
end
```

6.3. לולאת for

לולאת for מבצעת קטע קוד מספר קבוע של פעמים. התחביר כללי של for הינו:

```
for i = s:d:f
        statements
end
```

מומלץ לא להשתמש בלולאות for ולהשתמש בפונקציות סטנדרטיות מקבילות על מטריצות, כאשר המהירות חשובה.

7. פונקציות

ניתן להגדיר קטעי קוד Matlab בתור פונקציות על מנת לעשות בהם שימוש חוזר בעתיד. תחביר:

```
function <return values> = <function_name>(parameters)
% <function descprition>
<function body>
```

כאשר אנחנו יוצרים פונקציה עלינו לשמור אותה בקובץ m ששמו כשמה שם הפונקציה.

דוגמא:

```
function [c,d,e]= pyt(a,b)
% returns the hyotensus (yeter) in a right angle
% triangle according to Pythagoras theorem
% c        is the hyotensus
% d and e are the two sharp angles
c=sqrt(a.^2+b.^2);
d=atan(b/a);
e=pi/2-d;
```