בס"ד

הערות לפקודות:

[V ,D]=eig(A)

D-מטריצה מלוכסנת. V– מטריצה מלכסנת.

$$D = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix}$$

ind2sub - מקבל גודל מטריצה ואינדקס חד ממדי ומוציא אינדקס מטריצי.

find - אינדקס המטריצה בדרך הספירה של מטלב (שיטת הנחש), אם הפלט יחיד. אם רוצים את האיברים הראשונים שאינם 0 (אותם מחפשים בfind): find(M,n,'first') למט' M, n איברים ראשונים (לאחרונים 'last').

length - במטריצה מחזיר את האורך המקסימלי בין השורות לעמודות.

ones(n),zeros(n,m)- אם יש קלט אחד אזי תתקבל מטריצה ריבועית בגודל הקלט. אם שני ערכים הערך הראשון הוא מספר השורות, הערך השני מספר העמודות.

plot(x0,y0,'*')= scatter(x,y) - שרטוט נקודות ספציפיות בגרף.

Contour - מפה טופוגרפית (קווי גובה)

|| && - לסקלר, | & - לא רק. זה למטריצות עם אותם ממדים (איבר-איבר).

[X,Y]=meshgrid(x,y)

$$X = \begin{pmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_1 & \cdots & x_n \end{pmatrix}$$

$$Y = \begin{pmatrix} y_1 & \cdots & y_1 \\ \vdots & \ddots & \vdots \\ y_n & \cdots & y_n \end{pmatrix}$$

Any(M) - בודק אם בכל עמודה קיים לפחות איבר ששונה מ-0.(מוציא וקטור).

All(M) - בודק אם בכל עמודה כל האברים שונים מ0 (מוציא וקטור).

Logical - כל לא אפס יהפוך ל-1 וכל 0 יהפוך ל-0.

rand – x ~ U(0,1)

randn – x ~ N(0,1)

randi(n,sz1,sz2), כאשר:

n- מס' הכי גדול, Sz1,sz2- גדלי המטריצה

יצור משתנה מיקרי שונה: F_y פונקציית הצטברות שלו אזי $F_y^{-1}(x)$ מתפלג כמו *y*.

factorial - עצרת

integral - אינטגרציה נומרית.

fzero- מוצא נקודה שמחליפה סימן $f:\mathbb{R}\to\mathbb{R}$ (לדוגמא: 2^(x-2) לא יעבוד)

fsolve- $f:\mathbb{R}^m\to\mathbb{R}^n$

fminserch- העיקר שיהיה קרוב לניחוש ההתחלתי.

פתרון מערכת משוואות: מגדירים פונקציה כך שכל משוואה היא רכיב בוקטור. לדוגמא:

y(2)=2*x(1)-x(2)-3; y(1)=x(1)+x(2);

(פונקציה זאת מוגדרת כF)

fsolve('F',[1,1])

unique- פקודה מגניבה שמוחקת כפילויות בוקטור.

דוגמאות:

ציור בתלת ממד:

x=0:0.1:10;

y=0:0.1:10; (למשל)

[X,Y] = meshgrid(x,y);

surf (X,Y,sin(X)+cos(Y));

flip – הופך את סדר האיברים (בוקטור- ברור איך. במטריצה:

$$flip\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} => \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

$$flip\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},2\right) => \left(\begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}\right)$$

פולינומים:

כפל: *conv(p,q)*

חילוק: *deconv(p,q)*

(אם פלט יחיד תתקבל המנה, אם שניים – המנה והשארית).

נגזרת: *polyder(p)*, נגזרת מכפלה: *[n,d]=polyder(p,q)* ,*m=polyder(p,q)* (מקבלים *n/d*).

אלגוריתמים:

binary search:

נבדוק את האיבר האמצעי שבמערך הנתון. אם האיבר האמצעי הוא האיבר המבוקש, הרי מצאנו את שחיפשנו ונסיים כאן. אם לא, נבדוק מה יחס הסדר בין האיברים. אם האיבר המבוקש קטן יותר מאיבר זה, נבחר את חציו השמאלי של המערך. אם האיבר המבוקש גדול יותר, נבחר את חציו הימני של המערך. כעת נחזור על כל מה שעשינו עם תת-המערך שבחרנו. במקרה הגרוע ביותר, שבו באף אחד מהמקרים לא היה האיבר האמצעי האיבר אותו אנו מבקשים, נגיע לתת-מערך בעל איבר אחד, שהוא האיבר המבוקש (או שנגיע למסקנה שהאיבר המבוקש כלל אינו נמצא במערך).

gcd:

המחלק המשותף המקסימלי של *a,b* לפי האלגוריתם האוקלידי. נניח בה"כ *b>a* ונגדיר סדרה $r_0 = b$, $r_1 = a$, $r_{n+1} = mod(r_n, r_{n-1})$. בסוף נקבל $r_{n-1} = 0$ ברגע שזה קורה ה-*gcd* של *a,b* הוא r_n

power method:

אלגוריתם למציאת ערך עצמי הגדול ביותר בערך מוחלט

האלגוריתם: 1) נבחר ניחוש התחלתי עבור *v(1)* ונקרא לו *z(0)*.

2) נגדיר *w(1) =Az(0)* ונגדיר $\alpha_1 = \max(w(1)_i)$

3) נגדיר $z(1) = \frac{1}{\alpha_1} w(1)$

4) לחזור על התהליך.

גזירה נומרית:

נגזרת קדמית: $f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x}$

נגזרת אחורית: $f'(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x}$

נגזרת מרכזית: $f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}$

פונקציית *diff* נותנת את ההפרשים בין איברי וקטור הקלט ומוציאה וקטור הפרשים באורך *n-1*.

לדוגמא כשמחשבים וקטור נגזרת קדמית אנחנו כותבים *dfdx=diff(f(xx))./diff(xx)*.

ריבועים מינימליים:

נתונות נקודות $\{x_i, y_i\}_{i=1\dots m}$ מחפשים וקטור $c = (c_1 \dots c_n)$ כך שעבור פונקציה ספיציפית ממשפחת הפונקציות $f(x) = c_1 f_1(x) + \cdots + c_n f_n(x)$ סכום ההפרשים הריבועיים $\sum(y_i - f(x_i))^2$ הוא מינימלי.

האלגוריתם: נבנה וקטור $y = (y_1 \dots y_m)$ ואת המטריצה:

$$\begin{pmatrix} f_1(x_1) & \cdots & f_n(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_m) & \cdots & f_n(x_m) \end{pmatrix} = A$$

$$c = pinv(A) * y$$

חישוב סימבולי:

הצבת משתנה סימבולי: *syms x*

אם רוצים פונקציה סימבולית (מד"ר): *syms f(x)*

הפיכת סימבול למספר: *double(a)*

פתרון משוואה, מערכת משוואות: *solve(…..==…..,x), solve(…==….,….==…)*

הגדרת פונקציה: *f(x)=….* (כשx משתנה סימבולי).

שרטוט פונקציה בתחום: *ezplot(f,[xmin,xmax])*

גזירה ואינטגרציה:

diff(f,n) (הדיפולט עבור *n* הוא 1)

int(f) (לא מסוים), *int(f,a,b)* (מסוים)

פישוט ביטויים, פתיחת סוגריים וכו':

simplify(f) (למשל sin^2+cos^2=>1)

expand(f) (למשל כפל מקוצר)

פתרון מד"ר: *dsolve(…==….)*

פתרון מערכת משוואות: *Ax=b*

נומרית: *x=A\b*, סימבולית: *x=sym(A)\b*

מטריצות (פעולות):

det(M)=|M| , inv(M)=M^-1,

charpoly(M,x) = פ"א

eig(M) = ע"ע, *Jordan(M)* = מטריצה מז'ורדנת.

פולינומים: כתיבה (*x* הוגדר כסימבולי): *p=an*x^n+…+a0*

שורשים (**ממשיים**): *roots(p)* (אם רוצים גם מרוכבים, להשתמש ב*solve*).

factor(p) – פירוק לפולינומים ממשיים שמחלקים את *p*.

טיילור עד סדר חמישי: *taylor(f)*. סביב נקודה שאינה 0: *taylor(f,x0)*

הצבות של ערכים בביטויים, החלפת משתנה/ביטוי באחר: *subs(f,old,new)*

vpa - כשרוצים חישוב עם מספר ספרות דיוק מוגדר (ספרות דיוק לפני ואחרי הנקודה):

קביעת מספר הספרות: *digits(n)*

p=vpa(ma lehashev).

*אם מציבים ב*vpa* משהו שהמחשב כבר רואה שהוא אינסוף, התובה תהיה עדיין אינסוף. כדי לפתור בעיות כאלה כשרוצים לחשב מספרים גדולים: *x=sym(big number)* (למשל *big number=171!*).

vpa(x) ייתן ערך.

דוגמאות:

Ind2sub:

$indices = [3,4,5,6];\ s = [3,3];$

$[I,J] = ind2sub(s,M)$

$I = [3,1,2,3], J = [1,2,2,2]$

(מה שקורה זה שאם לוקחים את האיברים ה3, ה4,ה5 וה6 ממטריצה 3*3 הם יהיו במיקומים (3,1), (1,2), (2,2) ו-(3,2) בהתאמה)

לגבי *sub2ind* – הפוך: מקבל 2 מערכי מיקומים ומחזיר מיקום יחיד לכל איבר בשיטת הנחש.

find:

A = [1,0,2; ,0,1,1; 0,0,4];

find(A) => [1; 5; 7; 8; 9];

find(תנאי לוגי)could get the indices of which the condition is true.

A = [1,0,1; 0,1,0; 0,0,−1];

[r, c] = find(A > 0) =>

$r = [1; 2; 1], c = [1; 2; 3]$

Find could bring a specific number of numbers

(find((condition about A), number)
Also, it can get the numbers themselves:

for the last example:
[r, c, v] = find(A > 0) =>
(r, c *are the same*) v = [1; 1; 1]

Any:

A = [1,2; 0,0]

any(A) => [1,1],
but any$(A, 2)$ => [1; 0]

:All

A = [1,2; 0,0] , B = eye(2)

all(A) => [0,0],
but all(A, 2) => [1; 0]

כשהקלט השני ב *all* הוא 2 הוא בודק יחסית לשורות.

all(B) => [0,0], $all(B, 2)$ => [0; 0]

Conv, deconv, polyder, polyint

p = [1,1]; q = [1,2,1]

conv(p, q) => [1,3,3,1]
% $(x^2 + 2x + 1)(x + 1)$
$= (x + 1)^3$
$= x^3 + 3x^2 + 3x + 1$

deconv(p, q) => 0,
$deconv(q, p)$ = [1,1]

[a, r] = deconv(p, q) =>
$a = 0$, r = [1,1]
$[a, r] = deconv(q, p)$ =>
a = [1,1], r = [0,0,0]

polyder(p) => 1,
$polyder$([1,2,1]) => [2,2]

polyder(p, q) => [3,6,3],
$[a, r] = polyder(q, p)$ =>
a = [1,2,1], r = [1,2,1]
% ! אין צמצום

polyint(p) => [0.5000,1.0000,0],
$polyint(p, 2)$
=> [0.5000,1.0000,2.0000]

Symbolic

Solve for one equation and for two

syms x

solve(sin(x) == 1) => $pi/2$

solve(x^2 + 1 > 2) => ייתן ערך

אם המשוואה ביותר מסימבולי אחד, ברירת המחדל היא פתרון באיקס, אחרת צריך לכתוב:

solve(mishva'a, mishtane)

מערכת משוואות: syms x y

solve(sin(x)^2 == cos(y),
2 * x == y)
=>
[$pi - asin(3$^(1/2)/3) ;
$asin(3$^(1/2)/3) + pi]

Simplify

simplify(sin(x) * sin(2 * x) +
cos(x) * cos(2 * x)) => $cos(x)$

Expand

expand((x − 2) * (x − 4))
=> x^2 − 6 * x + 8

Symbolic polynomial: writing:
p = x^3 + 1

factor($x^6 - 1$)
=> [$x - 1, x + 1, x$^2 + x + 1,
x^2 − x + 1]

Subs

subs(a + b, a, 4) => b + 4
subs(a * b^2, a * b, 5) => 5 * b

VPA (general and for inf).

y = solve(x^4 − x + 1, x) ;

vpa(y) =>
the four (complex) roots

x = sym(171)

vpa(factorial(x)) =>
the value of 171! *with the requested number of digits.*

<u>אינטגרציה עם פרמטרים:</u>

```
f=@(x,c) sin(x*c);

I=@(c) integral(@(x) f(x,c),0,1)
```

<u>פונקציית הנגזרת בהינתן הפונקציה f</u>

```
h=0.01;

dfdx=@(x) (f(x+h)-f(x))/h;
```