בס"ד
שימושי מחשב במתמטיקה 88-151, פרופ' ג'. שיף
מועד א', סמסטר ב', תשע"א - פתרונות

זמן המבחן: שעתיים.
מותר להשתמש בכל חומר עזר ובמחשב כיס.
בחלק א' (50% של הציון) יש לענות על כל השאלות. ניקוד כל השאלות בחלק שווה.
בחלק ב' (50% של הציון) יש לענות על 2 מתוך 3 השאלות. ניקוד כל השאלות בחלק שווה.
יש לנמק היטב כל תשובה.

שים לב - יש הרבה פתרונות אפשריים לכל שאלה!!

חלק א'

1. כתוב פונקציה ב- Matlab אשר מקבלת כקלט ווקטור $v = (v_1, v_2, \ldots, v_n)$ ומחזירה כפלט את הווקטור w בעל אורך $n-4$ עם רכיבים

$$w_i = \frac{1}{5}\left(v_i + v_{i+1} + v_{i+2} + v_{i+3} + v_{i+4}\right)\ , \qquad i = 1, 2, \ldots n-4$$

```
function w=q1(v)
n=length(v);
w=zeros(1,n-4);
for i=1:(n-4)
    w(i) = 1/5*(v(i)+v(i+1)+v(i+2)+v(i+3)+v(i+4));
end
```

---

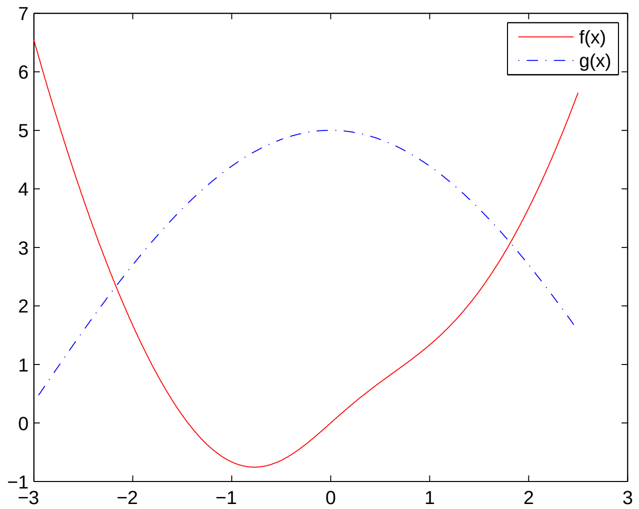
2. כתוב פרוצדורה ב- Mupad אשר מקבלת כקלט רשימה $v = [v[1], v[2], \ldots, v[n]]$ ומחזירה כפלט את הרשימה w בעלת אורך $n-4$ עם רכיבים

$$w[i] = \frac{1}{5}\left(v[i] + v[i+1] + v[i+2] + v[i+3] + v[i+4]\right)\ , \qquad i = 1, 2, \ldots n-4$$

```
q2:=proc(v)
local i,w,n;
begin
    n:=nops(v);
    w:=[0$(n-4)];
    for i from 1 to n-4 do
        w[i]:=1/5*(v[i]+v[i+1]+v[i+2]+v[i+3]+v[i+4]);
    end_for;
return(w);
end_proc
```

3. למטה מופיעים גרפים של שתי הפונקציות

$$f(x) = \frac{x^4+3x}{x^2+2}\ , \qquad g(x) = 5\cos\frac{x}{2}$$



כתוב פקודות ב- Matlab למצוא

(א) את קואורדינטות ה- x של שתי נקודות החיתוך של שני הגרפים.

(ב) את השטח החסום על ידי שני הגרפים.

(ג) את המרחק האנכי הכי גדול בין שני הגרפים בתחום בין שתי נקודות החיתוך.

לכל הסעיפים יש רק צורך בהפרש $f-g$. לכן נכין פונקציה כדלהלן:

```
function y = q3(x)
y = 5*cos(x/2) - (x.^4+3*x)./(x.^2+2)
end
```

ואחר כך נכתוב

(א) a=fzero(@q3,-2)
b=fzero(@q3,2)

(ב) A=quad(@q3,a,b)

(ג) [c d]=fminsearch(@(x) -q3(x), -0.5)
$-d$ הוא המרחק שמחפשים.

---

4. כתוב את הפקודות הרלוונטיות לפתור את השאלה הקודמת ב- Mupad. מספיק למצוא תשובות נומריות, לא ניתן למצוא את נקודות החיתוך באופן אנליטי.

יש צורך בשימוש חכם בפקודה op. על זה לא הורדתי נקודות!

```
f:=(x^4+3*x)/(x^2+2);    g:=5*cos(x/2);
```

$$\frac{x^4+3\,x}{x^2+2}$$

$$5\,\cos\left(\frac{x}{2}\right)$$

```
x1:=numeric::fsolve(f=g,x=-2);    x2:=numeric::fsolve(f=g,x=2)
```

$[x = -2.169967441]$

$[x = 1.815069499]$

```
int( g-f, x=op(op(x1,1),2)..op(op(x2,1),2))
```

14.2688736

```
y:=diff(g-f,x)
```

$$\frac{2\,x\,(x^4+3\,x)}{(x^2+2)^2}-\frac{4\,x^3+3}{x^2+2}-\frac{5\,\sin(\frac{x}{2})}{2}$$

```
x3:=numeric::fsolve(y=0,x=-0.5)
```

$[x = -0.5528921804]$

```
float(subs(g-f,x=op(op(x3,1),2)))
```

5.489013108

---

5. כתוב פונקציה ב- Matlab אשר יחשב את

$$f(p)=\int\int_D e^{-x^2-y^2}\ dxdy$$

כאשר התחום D הוא המשולש $\{(x,y)|x>0,y>0,x+y<p\}$. (כאן p הוא תמיד חיובי.)

התחום הוא משולש, שנמצא בתוך המלבן $\{(x,y)|0\le x\le p, 0\le y\le p\}$.

```
function [ z] = q5(p)
   z = dblquad( @(x,y) exp(-x.^2-y^2).*(x+y<p), 0,p,0,p);
end
```

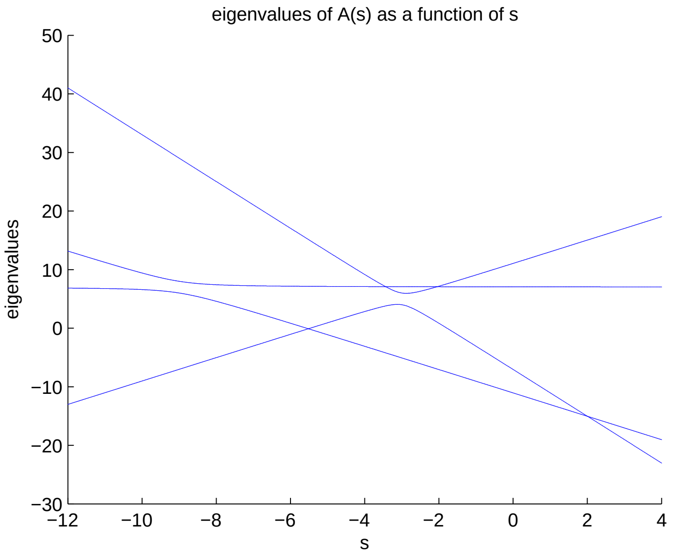
חלק ב'

1. (א) איך היית משתמש ב- Matlab לצייר (על אותם צירים) גרפים של ארבעת הערכים העצמיים של המטריצה

$$A(s)=\begin{pmatrix} -s & 2s+9 & 3 & s \\ 2s+9 & -s & s & 1 \\ 3 & s & -s & 2s+9 \\ s & 1 & 2s+9 & -s \end{pmatrix}$$

כפונקציות של s ?

(ב) למטה מופיעים גרפים של הערכים העצמיים של המטריצה $A(s)$ כפונקציות של s:



איך היית משתמש ב- Matlab למצוא את השטח החסום בין הגרף של הע"ע הכי הגדול ובין הגרף של הע"ע השני בגודלו, ואת השטח החסום בין הגרף של הע"ע השלישי בגודלו והגרף של הע"ע הכי קטן ?

(ג) איך היית משתמש ב- Matlab למצוא את הערך של s שעבורו ההפרש בין הע"ע השני בגודלו והע"ע השלישי בגודלו הוא הכי קטן ?

(א) יש כמובן הרבה דרכים. אני עשיתי

```
figure(1)
hold on
for  s=-12:0.01:4
    A = [ -s 2*s+9  3 s
          2*s+9 -s  s 1
          3    s -s 2*s+9
          s   1 2*s+9 -s];
     v=eig(A);
     plot(s,v(1),'.','Markersize',1)
     plot(s,v(2),'.','Markersize',1)
     plot(s,v(3),'.','Markersize',1)
     plot(s,v(4),'.','Markersize',1)
end
```

האופציה Markersize שולט על גודל הנקודות. לא צריך, אבל רציתי תמונה יפה יותר למבחן!

(ב) נכתוב פונקציה אשר מחשבת את ההפרש בין הע"ע הכי גדול והע"ע השני:

```
function z=qb1b(s)
A = [ -s      2*s+9      3         s
      2*s+9   -s         s         1
      3        s        -s      2*s+9
      s        1      2*s+9     -s   ];
v=eig(A);
z=v(4)-v(3);
end
```

הע"עים יוצאים בסדר עולה, ולכן מה שרוצים זה v(4)-v(3). הבעיה היא ש- fzero לא עובד על זה כי הפונקציה היא תמיד חיובית (רק מגיע ל- s לאיזה שהוא ערך מיוחד של s שלא ניתן לקבל במחשב, בגלל שגיאת עיגול). למצוא את נקודות החיתוך יש לעשות a=fminsearch(@qb1b,-3) ו- b=fminsearch(@qb1b,-2). למצוא את השטח היינו רוצים לעשות A=quad(@qb1b,a,b) אבל זה לא עובד בגלל ש- qb1b רק עובד על סקלר, ולא וקטור!! יש לשדרג ל-

```
function z=qb1b(S)
z=zeros(size(S));
for i=1:size(S,1)
   for j=1:size(S,2)
       s = S(i,j);
       A = [ -s      2*s+9      3         s
             2*s+9   -s         s         1
              3        s        -s      2*s+9
              s        1      2*s+9     -s   ];
       v=eig(A);
       z(i,j)=v(4)-v(3);
   end
end
end
```

לשטח השני ממש אותו דבר רק v(2)-v(1) במקום v(4)-v(3) .

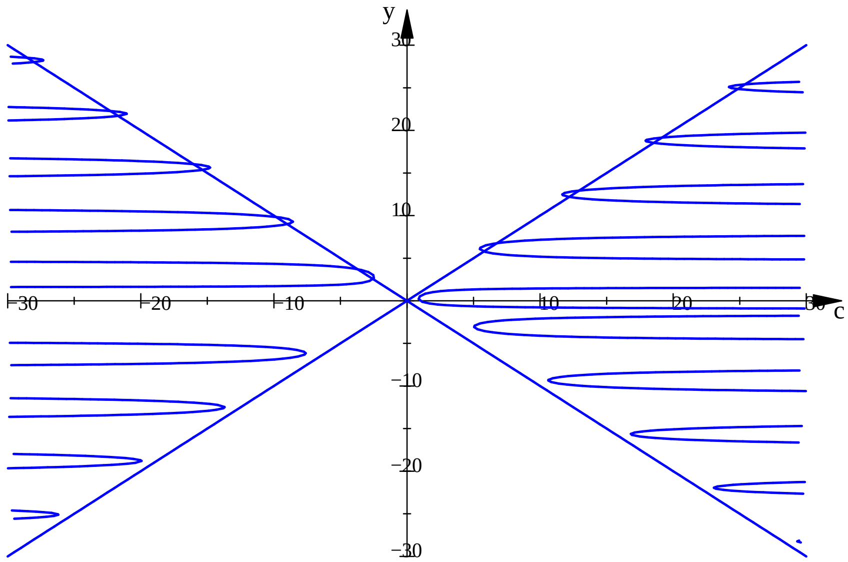
(ג) אותה פונקציה רק עם v(3)-v(2) במקום v(4)-v(3) . רואים בגרף שיש שתי אופציות ל- s המתאים, אחד קרוב ל- -9 והשני קורב ל- -3. דווקא יוצא שבדיוק שני ערכים אלה של s נותנים הפרש מינימלי בערך 2. הפקודות הרלוונטיות: [a b]=fminsearch(@qb1b,-3), [a b]=fminsearch(@qb1b,-9). (אחרי השינוי שצריך ב- qb1b.m .)

---

2. (א) ב- Mupad כתבתי את הפקודות

```
qq1:=plot::Implicit2d( c*(1+y)*cos(y)=1+y^2, c=-30..30, y=-30..30);
qq2:=plot::Function2d(c, c=-30..30);
qq3:=plot::Function2d(-c, c=-30..30);
plot(qq1,qq2,qq3)
```

וקבלתי את הגרף הזה:



הסבר את הפקודות האלה.

(ב) כתוב פונקציה ב- Matlab אשר מקבל כקלט ערך של המשתנה c ומחזיר כפלט את כל הפתרונות y של המשוואה

$$c(1+y)\cos y = 1+y^2$$

יש להעזר בגרף בסעיף הקודם להחליט, לכל c נתון, בערך כמה ערכים של y מחפשים, ובערך איפה הם נמצאים, ועל סמך זה לבנות את הפונקציה.

(א) יש פה בנייה של איור עם 3 רכיבים: הגרף $y=c$, הגרף $y=-c$ והעקומה המוגדרת על ידי המשוואה $c(1+y)\cos y = 1+y^2$. היות ובשני הראשונים y מוגדרת כפונקציה של c ניתן להשתמש בפקודה plot::Function2d. אבל בשלישי יש צורך לפקודה plot::Implicit2d ולתת תחומים גם ב- c וגם ב- y.

(ב) קשה למצוא משהו שעובד בכל המקרים. כאשר c גדל ונולד זוג חדש של פתרונות, הם מאוד קרובים וקשה להבדיל אחד מהשני. יש לשים לב שמספר הפתרונות גדל פחות או יותר כמו $|c|$ - כלומר מספר הפתרונות גדל באופן ליניארי עם הערך המוחלט של הפרמטר c. הפתרונות הם פחות או יותר בתחום $-|c| \le y \le |c|$. לכן פה אני דווקא ממליץ את הפתרון ה"גס" הבא - פשוט לקחת הרבה ערכים של y בתחום $-|c|-1 \le y \le |c|+1$, להתחיל fzero מנקודות אלה, ולהוציא את כל הפתרונות השונים שמקבלים.

```
function w=qb2(c)
% solve starting at many different points, put the answers in the vector z
z=[];
for x=(-abs(c)-1):0.1:(abs(c)+1)
    z=[z,fzero(@(y) c*(1+y)*cos(y) - 1-y^2, x)];
end
% extract the distinct solutions, put them in the vector w
w=[];
for i=1:length(z)
    t=1;
    for j=1:length(w)
        if abs(z(i)-w(j))<1e-10
           t=0;
```

```
        end
    end
    if t==1
        w=[w,z(i)];
    end
end
```

דוגמה של הפעלה שעשיתי: כתיבת $\mathrm{qb2}(20)$ נתן את הפתרונות עבור $c = 20$:

-10.3774	-16.2277	-15.0835	-8.3562	-4.4080
-1.8353	-0.8652	1.5056	4.9273	7.5100
11.5613	13.4560	18.3382	19.2578	

---

3. (א) הפרוצדורה $\mathrm{rat}(N,x)$ למטה מוצאת את השבר עם מכנה N הכי קרוב למספר נתון x. הסבר איך היא עובדת. האם היא תמיד נותנת את התשובה היחידה?

```
rat:=proc(N,x)
   local f;
   begin
    f:=floor(N*x);
    if N*x-f<1/2 then
       return(f/N);
    else
       return((f+1)/N);
    end_if;
end_proc
```

(ב) הפרוצדורה $\mathrm{bestrat}(N,x)$ למטה מוצאת את השבר עם מכנה קטן או שווה ל- N הכי קרוב למספר נתון x. הסבר איך היא עובדת. למה ניתן להגביל את הלולאה לרוץ מ- $\mathrm{floor}(N/2)$ עד N (במקום מ- 1 עד N)?

```
bestrat:=proc(N,x)
    local i,d,dnew,app,appnew;
    begin;
    d:=1;
    for i from 1 to N do
        appnew:=rat(i,x);
        dnew:=abs(appnew-x);
        if dnew<d then
           d:=dnew;
           app:=appnew;
        end_if;
     end_for;
     return(app);
end_proc
```

(ג) בהנתן x, אומרים ששלם $N > 1$ הוא מכנה טוב ל- x אם $\mathrm{bestrat}(N,x)\neq\mathrm{bestrat}(N\text{-}1,x)$. ($N = 1$ הוא תמיד מכנה טוב.) כתוב פקודות למצוא כל מכנה טוב של מספר נתון x עד 1000. יש לנסות לעשות כן ללא חישובים מיותרים. הערה: הפקודה denom מוצא מכנה של שבר.

(א) לכל מספר x קיים שלם f כך ש-

$$\frac{f}{N} \leq x \leq \frac{f+1}{N}$$

השבר עם מכנה N הכי קרוב ל- x הוא או $\frac{f}{N}$ או $\frac{f+1}{N}$. הפרוצדורה הנתונה מוצאת את $f = \mathrm{floor}(xN)$ ובודק מה יותר קרוב ל- x, $\frac{f}{N}$ או $\frac{f+1}{N}$. במידה ו- x נמצא ממש באמצע הפרוצדורה מחזירה $\frac{f+1}{N}$ - במקרה זה יש תשובה אפשרית נוספת.

(ב) הפרוצדורה הזאת רצה על כל מכנה אפשרי בין 1 ל- N, מוצאת את הקירוב הכי טוב עם מכנה זה, ובודק האם זה יותר קרוב ל- x מכל קירוב קודם. אם כן, הפרוצדורה שומרת את הקירוב החדש. ניתן לצמצם את הלולאה לרוץ מ- $i = \mathrm{floor}(N/2)$ עד N כי ניתן לכתוב כל שבר עם מכנה מ- 1 עד $N/2$ כשבר עם מכנה מ- $N/2$ עד N - פשוט על ידי כפל למעלה ולמטה על ידי חזקה מתאימה של 2.

(ג) היינו יכולים לבדוק, לכל N מ- 2 עד 1000, האם $\mathrm{bestrat}(N,x) = \mathrm{bestrat}(N-1,x)$ או לא. אבל זה ידרוש הרבה חישובים מיותרים, כי כל פעם שאנחנו קוראים ל- $\mathrm{bestrat}(N,x)$ זה מחשב את $\mathrm{rat}(i,x)$ לכל i מ- $\frac{N}{2}$ עד N. לכן עדיף להתבסס באופן ישיר על הפרוצדורה rat. לדוגמה:

```
qb3:=proc(x)
local dens,i,d,dnew;
begin;
  dens:=1;
  d:=abs(x-rat(1,x));
  for i from 2 to 1000 do
     dnew:=abs(x-rat(i,x));
     if dnew<d then
        dens:=dens,i;
        d:=dnew;
     end_if;
  end_for;
return([dens]);
end_proc
```
