MAI/Linköpings universitet Fredrik Berntsson

Tentamen TANA17 Matematiska beräkningar
Provkod: DAT1 **Godkänd:** 8p av totalt 20p **Tid:** 13:e januari klockan 8.00-12.00

Redovisning Lös först uppgifterna i Matlab. Då du har en färdig lösning skriv då ner de kommandon du använde på papper. Redovisa även eventuella resultat du fick då körde dina Matlab kommadon. Grafer behöver inte redovisas.

(2p) **1:** Använd kolon-notation för att skapa en vektor som innehåller talen $1, 2, 3, \ldots, 9, 10$. Använd sedan elementvisa operationer och kommandot `sum` för att beräkna summan

$$S = \sum_{k=1}^{10} k^2 - k.$$

(2p) **2:** Använd `polyfit` för att anpassa ett andragradspolynom till följande tabell

x	0	0.5	1.0	1.3	1.7
y	1.3	1.8	1.5	1.1	0.5

Plotta polynomet på intervallet $0 \leq x \leq 2$ som en blå heldragen linje. Rita även in punkterna (x_i, y_i) i samma graf med röda `x`.

(1p) **3:** Låt matrisen A vara

$$A = \begin{pmatrix} 1 & -2 & 2 \\ 3 & 2 & -1 \\ 2 & -1 & 3 \end{pmatrix}$$

Matrisens determinant kan beräknas med hjälp av funktionen `det`. Skapa först matrisen A och beräkna både $(\det(A))^{-1}$ och $\det(A^{-1})$ för att verifiera den välbekanta räkneregeln. Redovisa resultatet. Stämmer räkneregeln?

(2p) **4:** Följande kommandon är tänkta att kontrollera ifall en vektor är sorterad i stigande ordning:

```
for i=2:length(x)
  if x(i)>x(i-1)
    Stigande=1;
  else
    Stigande=0;
  end
end
```

Förklara vad som går fel och gör de ändringar som krävs för att programmet skall fungera som det är tänkt.

(2p) **5:** Låt A vara en matris av storlek $n \times m$. Skriv ett Matlabscript som använder `for`-loopar för att beräkna antalet positiva tal i matrisen. Testa ditt program på matrisen

$$A = \begin{pmatrix} 1 & -1 & 0 & 3 \\ -3 & 2 & 1 & 0 \\ 2 & -1 & 3 & -1 \end{pmatrix}.$$

(2p) **6:** Då man samlar in mätresultat är man ofta intresserade av att uppskatta hur stor variationen i mätningarna är. Ett sådant mått är *standardavvikelsen*. Den definieras som

$$\sigma = \left(\frac{1}{n-1} \sum_{i=1}^{n} (x_i - m)^2 \right)^{1/2}.$$

där m är medelvärdet

$$m = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Skriv en funktion `StandardAvvikelse` med en vektor `x` inparametrar och med σ som utparameter.

Testa funktionen för att beräkna standard avvikelsen för vektorn $x = (2\,,\,1\,,\,3\,,\,2\,,\,2\,,\,1)^T$.

OBS I Matlab finns en inbyggd funktion `std` som kan utföra precis denna beräkning. Du kan använda den för att kontrollera att du gjort rätt men får inte utnyttja den för att lösa uppgiften.

(3p) **7:** En rot till ekvationen $f(x) = xe^x - 1 = 0$ kan hittas genom att beräkna gränsvärdet för talföljden,

$$\begin{cases} x_0 = 1, \\ x_n = e^{-x_{n-1}}, \quad n = 1, 2, 3 \ldots \end{cases}$$

Skriv ett program som beräknar talföljdens gränsvärde. Du skall fortsätta att bilda nya tal x_n ända tills avbrottskriteriet $|x_n - x_{n-1}| < 10^{-4}$ är uppfyllt.

Vi vill även studera konvergenshastigheten. Ditt program skall därför skriva en tabell innehållande iterationsnummer n, x_n, samt skillnaden $x_n - x_{n-1}$ till skärmen. Utskriften skall göras enligt följande mönster:

```
   n          x(n)        x(n)-x(n-1)
 ---------------------------------------
   1       0.3678794      -6.32e-01
   2       0.6922006       3.24e-01
   3       0.5004735      -1.92e-01
```

(2p) **8:** Vi vill undersöka funktionen

$$f(x) = \cos(x)e^{-2x} - 2x, \quad 0 \leq x \leq 1.$$

a) Rita en graf över funktionen $f(x)$ på intervallet $0 \leq x \leq 1$.

b) Vi ser att ekvationen $f(x) = 0$ har en rot nära $x = 0.3$. För att kunna hitta roten skall du skriva en Matlab funktion `funk` med en vektor x som inparameter som beräknar en vektor y med motsvarande funktionsvärden.

Din funktion skall vara skriven på ett sådant sätt att det går att beräkna roten med kommandot

```
>> x = fzero( 'funk' , 0.3 )
```

(4p) **9:** Du skall skriva en funktion som för ett godtyckligt **udda** värde $N \geq 3$, skriver ut en kvadrat med ett mönster bestående av omväxlande asterixer och minustecken till en fil `kvadrat.txt`. Mönstret som skall skrivas ut skall bildas enligt mönstren i nedanstående exempel. För $N = 9$ skall filen `kvadrat.txt` innehålla

```
*********
*-------*
*-*****-*
*-*---*-*
*-*-*-*-*
*-*---*-*
*-*****-*
*-------*
*********
```

och för fallet $N = 13$ skall resultatet vara

```
*************
*-----------*
*-*********-*
*-*-------*-*
*-*-*****-*-*
*-*-*---*-*-*
*-*-*-*-*-*-*
*-*-*---*-*-*
*-*-*****-*-*
*-*-------*-*
*-*********-*
*-----------*
*************
```

Lösningsförslag till tentan för TANA17 Januari 2017.

1: Vi beräknar summans värde med kommandona

```
>> k=1:10;
>> S=sum( k.^2-k )
  S =
   330
```

2: Vi använder kommandona

```
>> x=[0 0.5 1.0 1.3 1.7];y=[ 1.3 1.8 1.5 1.1 0.5];
>> p2 = polyfit(x,y,2);
>> xx=0:0.1:2;
>> plot( xx,polyval( p2 , xx ),'b-')
>> hold on
>> plot( x,y,'rx')
>> hold off
```

Vi ser att linjen ligger nära punkterna.

3: Bilda matrisen och utförberäkningarna med

```
>> A=[ 1 -2 2 ; 3 2 -1 ; 2 -1 3];
>> 1/det(A)
  ans =
   0.0769
>> det(inv(A))
  ans =
   0.0769
```

Räkneregeln stämmer.

4: Problemet är att endast sista kontrollen räknas. Enklaste lösningen är att avbryta då vi hittar två element som inte ligger i stigande ordning med `break`. Vi får

```
for i=2:length(x)
  if x(i)>x(i-1)
    Stigande=1;
  else
    Stigande=0;
    break
  end
end
```

Det finns flera alternativa lösningar som ger enklare kod. Exempelvis

```
Stigande=1;
for i=2:length(x)
  if x(i) < x(i-1) % Vänt på jämförelsen!
    Stigande=0;
    break
  end
end
```

5: Vi bildar matrisen och beräknar antalet positiva element med

```
A=[ 1 -1 0 3 ; -3 2 1 0 ; 2 -1 3 -1];

[n,m]=size(A);
S=0;
for i=1:n
  for j=1:m
     if A(i,j)>0, S=S+1; end
   end
end
disp(S)
```

vilket ger värdet $S = 6$.

6: Funktionen blir

```
function [S]=StandardAvvikelse(x)
      n=length(x);
      m=mean(x);
      for i=1:n
         S=S+( x(i)-m )^2;
      end
      S=S/(n-1);
end
```

vilket ger

```
>> StadardAvvikelse( [2 1 3 2 2 1] )
  ans =
      0.7528
```

7: På filen `hittaRot.m` skriver vi

```
% Inför variabler x0 och x1 som representerar två på varandra
% följande approximationer
x0=-1;x1=1;n=0;    %  Med x0=-1 blir abvrottsvikkloret falskt
tol=10^-4;
fprintf(1,'  n    x(n)     x(n)-x(n-1)\n');
```

```
fprintf(1,'-----------------------------------\n');
while abs(x0-x1)>tol
    % beränka nästa approximation
    x0=x1;
    x1=exp(-x0);
    n=n+1;

    % Skriv ut en rad i tabellen
    fprintf(1,'  %i   %9.7f  %9.2e\n',n,x1,x1-x0);
end
```

8: Foljande kommandon plottar grafen

```
>> x=0:0.01:1;
>> f=cos(x).*exp(-2*x)-2*x;
>> plot( x , f ;
```

På filen `funk.m` skriver vi sedan

```
function y=funk(x)
   y=cos(x).*exp(-2*x)-2*x;
end
```

I Matlab kan vi sedan beräkna roten med

```
>> x=fzero('funk',0.3)
 x =
  0.2766
```

9: Funktionen blir:

```
function kvadrat(N);

  fid=fopen('matris.txt','w');
  for i=1:N,row(i)='*';,end;

  fprintf(fid,[row,'\n']);
  for i=2:(N-1)/2+1
    for j=i:N-i+1
      if row(j)=='*'
        row(j)='-';
      else
        row(j)='*';
      end;
    end;
    fprintf(fid,[row,'\n']);
```

```
 end;
 for i=(N-1)/2+1:-1:2
   for j=i:N-i+1
     if row(j)=='*'
       row(j)='-';
     else
       row(j)='*';
     end;
   end;
   fprintf(fid,[row,'\n']);
 end;
fclose(fid);
```