

Tentamen TANA17 Matematiska beräkningar
Provkod: DAT1 **Godkänd:** 9p av totalt 20p **Hjälpmedel:** MATLAB

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 kommandon. Grafer behöver inte redovisas.

(2p) **1:** Använd `polyfit` för att anpassa en rät linje till följande tabell

x	1.0	1.5	2.0	2.5
y	1.76	2.04	2.21	2.44

Plotta även den räta linjen på intervallet $0 \leq x \leq 3$ som en blå heldragen linje. Rita även in punkterna (x_i, y_i) i samma graf med röda x.

(2p) **2:** Följande MATLAB script är tänkt att kontrollera om alla element i en vektor är positiva

```
n = length(x);
for i = 1:n
    if x(i)<0
        allaPositiva=0;
    else
        allaPositiva=1;
    end;
end
```

Förklara vad som går fel och gör de förändringar som behövs för att få förväntat resultat.

(3p) **3:** Vi vill bilda en kvadratisk matris A där elementen ges av $a_{ij} = 1/(i+j)$, $1 \leq i, j \leq n$. Exempelvis, för $n = 3$, är alltså

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \\ 1/4 & 1/5 & 1/6 \end{pmatrix}$$

Skriv ett Matlab script som skapar en sådan matris för en given dimension n . Scriptet skall dessutom lösa ekvationssystemet $Ax = b$ för fallet $n = 4$ och $b = (1, 1, 1, 1)^T$.

(3p) **4:** Vi vill studera funktionen

$$f(x) = e^{-x/5} \cos(4x),$$

på intervallet $0 \leq x \leq 1$, och undersöka var dess minimum inträffar.

- a) Skapa en vektor $\mathbf{x}$ som innehåller $N = 50$ jämnt utspridda tal på intervallet $[0, 1]$. Beräkna även en vektor $\mathbf{f}$ med motsvarande funktionsvärden och utnyttja dessa vektorer för att plotta funktionen på aktuellt intervall.
- b) Använd Matlab kommandot `min` för att hitta det minsta värdet i vektorn $\mathbf{f}$. Detta approximerar funktionens minsta värde på intervallet. Hitta även det x -värde där funktionen antar sitt minsta värde.
- c) Markera punkten $(x_{\min}, f_{\min})$ du hittat i **b)** uppgiften i den graf du ritat upp i **a)** uppgiften med ett rött +.

(3p) **5:** För att beräkna kvadratroten av ett positivt tal a kan vi använda iterationen,

$$x_{k+1} = \frac{1}{2}(x_k + a/x_k), \quad x_0 = 1;$$

- a) Skriv ett Matlab program som beräknar $\sqrt{a}$ för ett givet positivt tal a genom att utnyttja iterationsformeln ovan. Du skall avbryta iterationerna då två på varandra följande approximationer uppfyller $|x_{k+1} - x_k| < 10^{-8}$.
- b) Gör om ditt program från **a)** till en funktion med namnet `KvadratRot`. Talet a skall vara inparameter och den beräknade kvadratroten skall vara utparameter. Du skall kunna använda funktionen på följande sätt

```
>> y = KvadratRot( 5 )  
y =  
2.2361
```

Tips Svarar du på **b)** så behöver du inte svara på **a)** eftersom Matlab raderna som utför iterationen även kommer att ingå i funktionen.

- (3p) **6:** Vi är intresserade av att undersöka storleken på en insektspopulation. Vi gör en serie mätningar och får en vektor,

$$F = (F_1, F_2, \dots, F_n)^T,$$

där F_i är det uppmätta antalet insekter vid tiden t_i . För att analysera mätningarna antar vi att dessa kan beskrivas med modellen

$$F(t) \approx c_1 + c_2 \cos(t) + c_3 \sin(t),$$

där c_0 , c_1 , och c_2 är okända parametrar vi önskar bestämma.

- a) Modellen ovan leder till ett överbestämt ekvationssystem $Ax = b$ som kan lösas med minsta kvadratmetoden. Antag att följande mätningar är givna

t_k	0	1	2	3	4
$F(t_k)$	1.14	2.11	2.79	2.40	1.43

Skriv de Matlab kommandon som krävs för att bilda både matrisen A och högerledet b . Lös dessutom minsta kvadrat problemet och hitta koefficienterna c_1 , c_2 och c_3 .

- b) Plotta den funktionskurva $F(t)$ som du hittade i Uppgift a). Rita även upp de givna mätvärdena (t_k, F_k) i samma Figur (som svarta $\circ$).

- (4p) **7:** Vi vill skriva ett simuleringssystem som kan användas för att undersöka tärnings-spelet spelet Yatzy. Systemet skall kunna simulera ett stort antal spelomgångar och bestämma sannolikheten för att olika kombinationer dyker upp. Som en del i projektet att utveckla systemet skall följande uppgifter lösas.

- a) Skriv en Matlab funktion som simulerar en spelomgång. Funktionen skall returnera en vektor innehållande resultatet från 5st tärningskast. Funktionen skall kunna användas på följande sätt

```
>> x = YatzyOmgang
x =
     4     6     1     3     1
```

där alltså vektorn x tilldelas 5st slumpmässiga heltal mellan 1 till 6. Använd antingen `rand` eller `randi` för att skapa slumpalen.

- b) Att få Yatzy innebär att alla tärningskast ger samma resultat. Skriv en funktion `KontrolleraYatzy`, med en vektor som representerar en spelomgång som inparameter, som kontrollerar om vi har fått Yatzy. Funktionen skall ha en utparameter som getts värdet *sant* om vi har Yatzy och *falskt* annars
- c) Kontrollera hur vanligt det är att få Yatzy genom att skriva ett script som simulerar $N = 10000$ spelomgångar. Räkna efter hur många spelomgångar som ger Yatzy. Du skall använda dina två funktioner från uppgifterna a) och b).

Lösningsförslag till tentan för TANA17 Augusti 2015.

1: Vi ritar grafen med kommandona

```
x = [1.0 1.5 2.0 2.5]; y=[1.76 2.04 2.21 2.44];  
p1=polyfit(x,y,1);  
xx=0:0.1:3.0;  
plot( xx , polyval( p1,xx),'b-',x,y,'rx')
```

2: Problemet är att endast den sista jämförelsen då $i = n$ påverkar resultatet. Sätt allaPositiva=1 först och ändra till noll och avbryt loopen ifall ett negativt hittas.

```
n = length(x);  
allaPositiva=1;  
for i = 1:n  
    if x(i)<0  
        allaPositiva=0;  
        break  
    end;  
end
```

3: Vi skapar matrisen och löser $Ax = b$ med kommandona:

```
n=4;  
A=zeros(n,1);  
for i=1:n  
    for j=1:n  
        A(i,j)=1/(i+j);  
    end  
end  
b=ones(n,1);  
x=A\b; x'  
ans =  
    -20.0000    180.0000   -420.0000    280.0000
```

4: Vi plottar funktionen med

```
N=50;
x=(0:N-1)/(N-1); % Precis 50 x-värden
f=exp(-x/5).*cos(4*x);
plot(x,f)
```

vi hittar sedan minimum och markerar det i grafen med

```
[m,k]=min(f);disp( m ),disp( x(k) )
hold on
plot( x(k),f(k) , 'r+' )
hold off
```

5: Vi inför variabler x_0 och x_1 för två på varandra följande x -värden.

```
function [x1]=KvadratRot( a )

x0=1;x1=(x0+a/x0)/2;
while abs(x0-x1)>10^-8
    x0=x1;
    x1=(x0+a/x0)/2;
end
end
```

Vi behöver ingen räknare som håller reda på antalet termer.

6: Vi löser minsta kvadrat problemet med

```
t = [0 1 2 3 4]'; % Måste vara kolumnvektor
F=[1.14 2.11 2.79 2.40 1.43]'; b=F;
A=[ ones(size(t)) cos(t) sin(t) ]; % blir 5x3 matris
c=A\b; c'
ans =
    1.7128    -0.5857     0.8823
```

Vi plottar sedan modellen och punkterna med

```
tt = 0:0.1:4;
FF = c(1)+c(2)*cos(tt)+c(3)*sin(tt);
plot( tt , FF , 'b-',t,F,'ko')
```

Ger helt ok överensstämmelse.

7: Använder vi funktionen `randi` blir Yatzy funktionen

```
function [ Kast ]=YatzyOmgang()  
    Kast = randi( [1 6],5,1);  
end
```

För att kontrollera om alla tal är lika jämför vi de övriga med det först kastet.

```
function [Test]=KontrolleraYatzy( Kast )  
    Test=1; % Utgå ifrån sant  
    for i=2:5  
        % Testa ifall Kast(i) är samma som Kast(1) för i=2,3,4,5  
        if Kast(i) ~= Kast(1)  
            % Olika så Test blir falskt och vi avbryter  
            Test=0;  
            break  
        end  
    end  
end
```

Huvudprogrammet blir nu

```
N=10000; % Antalet omgångar totalt  
AntalYatzy=0;  
for i=1:N  
    Kast = YatzyOmgang();  
    if KontrolleraYatzy( Kast )  
        AntalYatzy=AntalYatzy+1;  
    end  
end  
disp( AntalYatzy )
```

Omkring 10 Yatzy är rimligt på 10000 omgångar.