

Tentamen TANA17 Matematiska beräkningar
Provkod: DAT1 **Godkänd:** 8p av totalt 20p **Tid:** 21:a April klockan 8.00-12.00

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

- (2p) **1:** Använd kolon-notation för att skapa en vektor som innehåller talen $1, 2, 3, \dots, 9, 100$. Använd sedan elementvisa operationer och kommandot `sum` för att beräkna summan

$$S = \sum_{k=1}^{100} 3k - 5\sqrt{k}.$$

- (2p) **2:** Vi vill hitta ett polynom med nollställena $x_1 = 1$, $x_2 = 1.7$, $x_3 = 2.9$ och $x_4 = 3.6$. Ett sätt är att hitta ett polynom som interpolerar följande tabell

x	0.0	1.0	1.7	2.9	3.6
y	1.0	0.0	0.0	0.0	0.0

Använd `polyfit` för att hitta polynomet. Tänk på att det är viktigt att välja precis rätt gradtal på polynomet. Plotta dessutom polynomet på intervallet på intervallet $0 \leq x \leq 4$ som en blå heldragen linje. Fick polynomet rätt nollställena?

- (2p) **3:** Utnyttja `for` loopar för att skapa en $m \times n$ matris med elementen $a_{ij} = 1/(1 + |i - j|^2)$. Redovisa matrisen för fallet $n = m = 3$.
- (2p) **4:** Ett lokalt maximum definieras av att $x_{i-1} < x_i > x_{i+1}$. Följande kommandon är tänkta att kontrollera om en vektor har ett lokalt maximum:

```
for i=2:length(x)-1
    if x(i)>x(i-1) & x(i)>x(i+1)
        Lokalt=1;
    else
        Lokalt=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:** 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 **x** som innehåller $N = 50$ jämnt utspridda tal på intervallet $[0, 1]$. Beräkna även en vektor **f** med motsvarande funktionsvärden och utnyttja dessa vektorer för att plotta funktionen på aktuellt intervall.
- b) Använd Matlabkommandot **min** för att hitta det minsta värdet i vektorn **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 +.

(2p) **6** En rot till ekvationen $f(x) = x \sin(x) - 1 = 0$ kan hittas genom att beräkna gränsvärdet för talföljden,

$$\begin{cases} x_0 = 1, \\ x_n = \frac{x_{n-1}^2 \cos(x_{n-1}) + 1}{\sin(x_{n-1}) + x_{n-1} \cos(x_{n-1})}, \quad n = 1, 2, 3, \dots \end{cases}$$

Skriv ett program som beräknar talföljdens gränsvärde. Du skall fortsätta att bilda nya tal x_n ända tills avbrottskriteriet $|x_n - x_{n-1}| < 10^{-8}$ 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	1.1147287	1.15e-01
2	1.1141571	-5.72e-04
3	1.1141571	1.40e-08

(2p) **7:** Vi vill hitta ett tal a sådant att

$$f(a) = \int_0^a \frac{1 + \cos(x)}{1 + \sqrt{x}} dx = 4$$

Gör följande:

- a) Skriv en Matlab funktion **IntegralFunktion**, med inparameter a , som beräknar utparametern $g(a) = f(a) - 4$. Du skall använda Matlabfunktionen **integral** för att beräkna integralens värde $f(a)$ för ett givet a .
- b) Använd **fzero** för att hitta den positiva roten till ekvationen $g(a) = f(a) - 4 = 0$. Roten bör ligga omkring $a \approx 10$.

- (3p) 8: I ett tärningsspel skall vi slå tärningar ända tills totalsumman blir större än eller lika med 20. Speciellt för detta spel är att om vi slår en sexa får vi slå en extra tärning under den omgången. Slår vi då ytterligare en sexa får vi slå om igen, och så vidare. Skriv ett Matlabscript som simulerar en spelomgång och använder `randi` eller `rand` för att simulera tärningskast ända tills total summan blivit åtminstone 20. Programmet skall göra utskrifter enligt mönstret

```
Kast 1 gav 1
Kast 2 gav 3
Kast 3 gav 6(+6+4)
Efter 3 kast har vi summan 20.
```

Där alltså tredje kastet gav två sexor följt av en 4:a.

Tips Det är klurigt att få till parenteserna vid upprepade sexor. Lös först uppgiften utan dessa.

- (3p) 9: Ofta används slumpmodeller för att simulera börskurser för företag. Ett exempel på en enkel modell är följande: Vid tiden t_k är börskursen $y_k = t(t_k)$. Kursförändringar beskrivs med hjälp av slumptal. Börskursen vid tiden t_{k+1} beräknas enligt

$$y_{k+1} = (1 + \alpha)y_k + \delta_k,$$

där α representerar den förväntade värdeökningen och δ_k är ett likformigt fördelat slumptal i intervallet $[-1, 1]$ som representerar slumpmässiga variationer. Gör följande:

- Skriv en Matlab funktion `EttSteg`, med inparameter α och y_k som beräknar y_{k+1} enligt beskrivningen ovan. Använd funktionen `rand` för att generera de slumptal du behöver.
- Låt tidsvariabeln t_k betyda timmar. Skriv en funktion `SimuleraKurs` som simulerar börskursen under ett antal timmar. Inparametrar skall vara y_0 , α , och antalet timmar T som simuleringen skall köras. Utparameter skall vara vektorer t och y med tider respektive börskurser.
Redovisa ett exempel där start kursen är $y_0 = 100$, och där $\alpha = 2.7 \cdot 10^{-3}$ så att kursen i genomsnitt ökar. Låt simuleringen pågå under 10 dagar. Plotta börskursen som funktion av tiden. Lägg även till lämplig text på koordinat axlarna i grafen.
- Det förväntade värdet på aktien efter 10 dagar bör vara $(1 + \alpha)^{10 \cdot 24} y_0$. Använd din funktion `SimuleraKurs` för att beräkna $m = 1000$ möjliga aktiekurser efter 10 dagar. Beräkna även det genomsnittliga värdet på aktien vid slutet av simuleringen och jämför med det teoretiska värdet.

Lösningsförslag till tentan för TANA17 Januari 2017.

1: Vi beräknar summans värde med kommandona

```
>> k=1:100;
>> S=sum( 3*k-5*sqrt(k) )
S =
1.1793e+04
```

2: Vi använder kommandona

```
>> x=[0 1.0 1.7 2.9 3.6];y=[1 0 0 0 0 ];
>> p4 = polyfit(x,y,4);
>> xx=0:0.1:4;
>> plot( xx,polyval( p4 , xx ),'b-')
```

Vi ser att linjen ligger nära punkterna.

3: Bilda matrisen med

```
m=3;n=3;A=zeros(m,n);
for i=1:m
    for j=1:n
        A(i,j)=1/(1+abs(i-j)^2);
    end
end
A =

1.0000    0.5000    0.2000
0.5000    1.0000    0.5000
0.2000    0.5000    1.0000
```

4: Problemet är att endast sista kontrollen räknas. Enklaste lösningen är att avbryta då vi hittar ett lokalt maximum med **break**. Vi får

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

Det finns flera alternativa lösningar som ger enklare kod. Exempelvis

```

Lokalt=0;
for i=2:length(x)-1
    if x(i)>x(i-1) & x(i)>x(i+1);
        Lokalt=1;
        break
    end
end
end

```

5: Vi löser uppgiften med programmet

```

N=50;x=linspace(0,1,N);
f=exp(-x/5).*cos(4*x);
plot(x,f,'b-');
[m,k]=min(f);
x(k),m
hold on
plot(x(k),m,'r+')
hold off

```

Vi ser att minimum blir $m = -0.8557$ vilket antas för $x_{39} = 0.7755$.

6: 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 avrottsvikkloret falskt
tol=10^-8;
fprintf(1,'  n      x(n)      x(n)-x(n-1)\n');
fprintf(1,'-----\n');
while abs(x0-x1)>tol
    % beräkna nästa approximation
    x0=x1;
    x1=(x0^2*cos(x0)+1)/(sin(x0)+x0*cos(x0));
    n=n+1;

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

```

7: Funktionen blir

```

function [g]=IntegralFunktion( a )
    f=integral(@(x) (1+cos(x))/(1+sqrt(x)) , 0,a);
    g=f-4;
end

```

Vi använder funktionen för att hitta roten med

```
a=fzero( @IntegralFunktion , 10 )
```

8: Vi simulerar spelomgången med

```
summa=0;k=0;
while summa<20
    k=k+1;
    slutparentes=0;startparentes=0;
    kast=randi([1 6],1);
    summa=summa+kast;
    str=['Kast ',num2str(k),' gav ',num2str(kast)];
    while kast==6
        slutparentes=1;
        if startparentes==0,
            str=[str,'('];startparentes=1;
        end;
        kast=randi([1 6],1);
        summa=summa+kast;
        str=[str,+',',num2str(kast)];
    end;
    if slutparentes==1
        str=[str,')'];
    end;
    disp(str)
end;
disp( ['Efter ',num2str(k),' kast har vi summan ',num2str(summa)]);
```

9: Funktionerna blir

```
function [y1]=EttSteg(y0,alpha)
    delta=-1+2*rand(1);
    y1=(1+alpha)*y0+delta;
end
```

```
function [t,y]=SimuleraKurs( y0 , alpha, T );

t=(0:T)'; % Hela timmar!
y=zeros(size(t));y(1)=y0;

for k=1:T
    y(k+1)=EttSteg( y(k) , alpha );
end
end
```

Vi kör simuleringen en gång och plottar med

```

>> y0=100;T=24*10;alpha=2.7e-3;
>> [t,y]=SimuleraKurs( 100 , alpha , T );
>> plot(t,y)
>> xlabel('Tid i timmar');
>> ylabel('Aktiekurs y(t)');

```

Vi beräknar medelvärde efter 1000 simuleringar med

```

m=1000;kurs=zeros(m,1);
for i=1:m
    [t,y]=SimuleraKurs( 100 , alpha , T );
    kurs(i)=y(end);
end
mean(kurs)
y0*(1+alpha)^(24*10)

```