

Tentamen TANA17 Matematiska beräkningar
Provkod: DAT1 **Godkänd:** 8p av totalt 20p **Tid:** 1:a april 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 kommandon. Grafer behöver inte redovisas.

- (2p) **1:** Använd kolon-notation för att skapa en vektor som innehåller de udda talen $1, 3, 5, \dots, 99$. Använd sedan elementvisa operationer och kommandot `sum` för att beräkna summan

$$S = 1 + \sqrt{3} + \sqrt{5} + \dots + \sqrt{99}.$$

- (2p) **2:** Skapa ett tredjegradspolynom $p(x)$ med rötter $x_1 = 1$, $x_2 = 1.7$ och $x_3 = 2.5$ genom att använda `polyfit` för att hitta det polynom av grad 3 som interpolerar tabellen

x	0	1.0	1.7	2.3
y	1.0	0.0	0.0	0.0

Plotta även polynomet 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 så att det tydligt syns att polynomet får rätt rötter.

- (2p) **3:** Följande kommandon är tänkta att kontrollera ifall en matris A endast har positiva element:

```
[n,m]=size(A);  
for i=1:n  
    for j=1:m  
        if A(i,j) <= 0  
            test = 0;  
        else  
            test = 1  
        end  
    end  
end
```

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

- (2p) **4:** Vi vill bilda en matris enligt följande regel:

$$a_{ij} = (5 + |i - j|)^2, \quad 1 \leq i, j \leq N,$$

där N är matrisens dimension. Skriv ett Matlabprogram som skapar denna matris för dimension $N = 10$.

(2p) **5:** Vi vill beräkna en integral

$$I = \int_0^1 f(x) dx, \quad \text{där } f(x) = \frac{1 + \sin(x)}{1 + \cos(x)}.$$

Skapa först ett funktionshandtag eller en funktionsfil som implementerar funktionen $f(x)$. Funktionen måste klara att ha en vektor $\mathbf{x}$ som inparameter. Använd sedan funktionen `integral` för att beräkna ett värde på integralen I ovan.

(3p) **6:** Vi vill studera hur summan

$$S_N = \sum_{k=1}^N N(N - k + 1) - k^2,$$

beror på N . Vi skall därför göra följande:

- a) Skriv en funktion som beräknar summan för ett givet värde på N . Funktionen skall kunna anropas enligt

```
>> N = 10; S = Summan( N )  
S =  
165
```

- b) Skapa en vektor som innehåller heltalen $N = 10, 20, 30, \dots, 100$. Beräkna sedan en ny vektor som innehåller motsvarande summor S_N med hjälp av din funktion från a). Plotta en graf som visar S_N som funktion av N . Använd `semilogy` istället för `plot` för att tydligare se hur S_N beror på N . Du skall även lägga till lämplig text på koordinataxlarna.

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

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

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

(4p) 8: Ett instrument ger ibland underliga mätresultat. En viss storhet som vi vill mäta är nästan konstant men ibland ger mätinstrumentet en alldeles för stor avvikelse ifrån det konstanta värdet.

a) Vi antar att mätserien har lagrats i en vektor F och vi vill hitta konstiga mätvärden genom att göra följande

- Beräkna först medelvärdet $m = \frac{1}{n} \sum_{k=1}^n F_k$.
- Hitta de index k sådana att $|F_k - m| > \text{tol}$, där tol är en given tolerans.

Skriv en funktion

```
>> [ind]=HittaAvvikelser( F , tol )
```

som utför ovanstående steg. Testa din funktion på fallet

$$F = (2.1, 1.9, 0.1, 2.1, 2.2, 3.7, 1.9, 1.8, 2.1),$$

och $\text{tol} = 1.0$. Du skall få $\text{ind} = (3, 6)$ eftersom dessa platser i vektorn avviker mycket från medelvärdet.

b) För att kunna testa funktionen `HittaAvvikelser` vill vi skriva en funktion som skapar simulerade mätserier av given längd. Funktionen skall kunna anropas enligt

```
>> F = SkapaData( m , tol , ind , N );
```

där m är önskat medelvärde, tol är den tolerans som skall användas, ind är en vektor som innehåller de index där vektorn F skall avvika från medelvärdet, och N är önskad storlek på vektorn F . Använd följande metod.

- Skapa en konstant vektor $F_1 = (1, 1, \dots, 1)^T$ som innehåller N st element.
- Skapa en vektor F_2 som innehåller nollor förutom de platser vars index ingår i vektorn ind . På dessa platser skall det istället finnas $+1$ eller -1 med samma sannolikhet.
- Använd funktionen `randn` för att skapa en vektor F_3 av längd N som innehåller slumpmässiga tal.

Bilda sedan vektorn F som

$$F = m \cdot F_1 + 1.5 \cdot \text{tol} \cdot F_2 + 0.1 \cdot \text{tol} \cdot F_3.$$

Då du är klar skall du testa att skapa en simulerad mätserie och även testa att `HittaAvvikelser` fungerar genom att skriva

```
>> F = SkapaData( 2.3 , 1.0 , [3 7 8 11] , 20 );
>> plot( F )
>> ind2 = HittaAvvikelser( F , 1.0 )
```

Du skall då hitta de platser som du valt ut som avvikande. Plotten skall även visa att mätserien ser rimlig ut.

Lösningsförslag till tentan för TANA17 Januari 2016.

1: Skriv

```
>> x=1:2:99;
>> S=sum(sqrt(x))
S =
    333.4153
```

2: Skriv

```
>> x=[ 0 1.0 1.7 2.3]; y=[1 0 0 0];
>> p=polyfit(x,y,3)
p =
    -0.2558    1.2788   -2.0230    1.0000
>> xx=0:0.01:3;
>> plot(xx,polyval(p,xx),'b-',x,y,'rx')
```

Man ser tydligt att polynomet får rätt nollställen.

- 3: Problemet är att endast sista kontrollen räknas. Hittar vi ett fall där $a_{i,j}$ är negativt eller noll så sätter vi `test=0` och vi kan aldrig därefter sätta tillbaka `test` till 1. Alltså

```
[n,m]=size(A);
test=1;
for i=1:n
    for j=1:m
        if A(i,j) <= 0
            test = 0;
        end
    end
end
```

Det går att avbryta med `break` men tänk isåfall på att `break` bara avbryter en `for`-loop så det måste bli en extra `if`-sats av typ `if test==0,break,end` innan sista `end`.

- 4: Matrisen skapas med programmet

```
N=10;
A=zeros(N,N);
for i=1:N
    for j=1:N
        A(i,j)=(5+abs(i-j))^2;
    end
end
```

5: Integralen beräknas med hjälp av

```
>> f=@(x) (1+sin(x))./(1+cos(x));
>> I=integral( f,0,1 )
I =
    0.8075
```

6: Funktionen blir

```
function [S]=Summan(N)
    S=0;
    for k=1:N
        S=S+N*(N-k+1)-k^2;
    end
end
```

Vi använder den för att plotta grafen med

```
N=10:10:100;
S=zeros(size(N));
for k=1:length(N)
    S(k)=Summan(N(k));
end
semilogy(N,S)
xlabel('N');ylabel('Summan: S_N');
```

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 avrottsvikkloret falskt
tol=10^-4;
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=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: Funktionen HittaAvvikelser blir

```

function [ind]=HittaAvvikelser( F , tol )
    m=mean(F);
    n=0;
    for k=1:length(F)
        if abs(F(k)-m)>tol
            n=n+1;
            ind(n)=k;
        end
    end
end

```

Vi använder sedan funktionen för att skapa grafen med

```

>> F=[2.1 1.9 0.1 2.1 2.2 3.7 1.9 1.8 2.1];
>> ind=HittaAvvikelser( F , 1 )
>> ind =
     3     6
>> plot(F); hold on, plot( ind , F(ind) , 'ro'), hold off

```

För att kunna skapa testdata följer vi metoden som beskrivs och skriver en funktion

```

function [ F ] = SkapaData( m , tol, ind , N )
    F1=ones(N,1);
    F2=zeros(N,1);
    for i=1:length(ind)
        draw=randi([0 1]);
        if draw==0
            F2(ind(i))=-1;
        else
            F2(ind(i))=+1;
        end
    end
    F3=randn(N,1);
    F=m*F1+1.5*tol*F2+0.1*tol*F3;
end

```

Plottar man mätserien får man det utseende man förväntar sig.