

Tentamen TANA17 Matematiska beräkningar
Provkod: DAT1 **Godkänd:** 8p 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.

- (1p) **1:** Rita en graf över funktionen $f(x) = e^{x/2} - \cos(x)$, på intervallet $0 \leq x \leq 1$.
- (1p) **2:** Använd kolon-notation för att skapa en vektor som innehåller de udda-talen $1, 3, \dots, 17, 19$ och använd även kommandot `sum` för att beräkna talens summa.
- (1p) **3:** Beräkna lösningen till det linjära ekvationssystemet $Ax = b$ där

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 1 & -3 \\ 1 & -1 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}.$$

- (1p) **4:** Använd `filter` för att beräkna det löpande medelvärdet av två på varandra följande tal i en vektor då $x = (2, 4, 6, 8, 10, 12)$.
- (2p) **5:** Hilbert matrisen ges av

$$H_{ij} = \frac{1}{i+j}, \quad 1 \leq i, j \leq n,$$

där n är matrisens dimension. Skriv ett Matlab script som använder `for`-loopar för att generera en Hilbert matris av storlek $n = 5$.

- (3p) **6:** Vi vill beräkna skalärprodukten mellan två vektorer x och y genom att utnyttja formeln

$$S = (x, y) = \sum_{k=1}^n x_k y_k.$$

Skriv en funktion `Skalar` med två vektorer `x` och `y` som inparametrar och med skalärprodukten S som utparameter.

Utnyttja funktionen från **a)** för att beräkna skalärprodukten mellan vektorerna $x = (1, 2, 3)^T$ och $y = (2, -3, 1)^T$.

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

(2p) **7:** Använd `polyfit` för att anpassa en rät linje till följande tabell

x	0	1	2	3
y	1.7	2.2	2.9	3.4

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) **8:** Vi köper en TV på nätet för 10999kr. Då vi skal betala erbjuds vi att istället betala på avbetalning med 500kr/månad i 24 månader. Räknar vi med 2% inflation (dvs kalkylräntan på årsbasis är 2%) så blir nuvärdet av betalningarna

$$P_{nu} = \sum_{k=1}^{24} 500 \frac{1}{1.0017^k}.$$

Skriv ett Matlab script som beräknar nuvärdet av avbetalningsplanen enligt formeln ovan. Avgör om vi bör betala kontant eller välja att betala över 24 månader.

(2p) **9:** Vi vill beräkna integralen av funktionen

$$f(x) = \cos(x)e^{-2x} + 2x,$$

över intervallet $0 < x < 1$. Skriv 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 integralens värde med kommandot

```
>> I = quad( 'funk' , 0 , 1)
```

(2p) **10:** Vi vill utföra ett experiment för att verifiera ett resultat från sannolikhetsläran. Tänker vi oss ett experiment där man slår två tärningar och lägger ihop resultatet så får vi ett värde mellan 2 och 12. För att uppskatta väntevärdet simulerar vi $N = 100$ omgångar, med två tärningskast, och beräknar summan av de två kasten i varje omgång. Vi beräknar sedan medelvärdet av resultatet för alla N omgångarna.

Skriv ett Matlab script som utför ovanstående experiment. Här är det tänkt att du skall använda vektor operationer och får alltså inte använda en `for` eller `while` loop. Förlita dig istället på `sum` och `mean`. För att generera slumpalen får du använda antingen `rand` eller `randi`.

(3p) **11:** Ett lokalt maximum för en vektor $\mathbf{x}$ definieras som ett element $\mathbf{x}(i)$ för vilket villkoret,

$$\mathbf{x}(i-1) \leq \mathbf{x}(i) \geq \mathbf{x}(i+1),$$

är uppfyllt. För det första och sista elementet i vektorn skall vara ett lokalt maximum gäller ett liknande villkor.

Skriv en funktion `LokaltMax` med en vektor $\mathbf{x}$ som inparameter, och som returnerar en vektor `ind` innehållande samtliga index i sådana att $\mathbf{x}(i)$ är lokala maxima. Exempelvis skall du kunna skriva

```
>>x=[3 4 5 4 3 2 1 7 3];  
>>[ind]=LokaltMax(x);
```

och `ind` skall ges värdet `ind=[3 8]` eftersom lokala maxima finns på platserna 3 och 8 i vektorn $\mathbf{x}$.

Lösningsförslag till tentan för TANA17 April 2015.

1: Vi ritar grafen med kommandona

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

2: Vi använder

```
>> x = 1:2:19;  
>> sum(x)  
ans =  
100
```

Summan blir alltså 100.

3: Bilda matrisen och lös problemet med

```
>> A=[ 3 2 1 ; 2 1 -3 ; 1 -1 4]; b=[ 1 2 0]';  
>> x=A\b  
x =  
0.7727  
-0.5000  
-0.3182
```

4: Vi skriver

```
>> x=2:2:10  
x =  
2 4 6 8 10  
>> y=filter( [1 1],2,x)  
y =  
1 3 5 7 9
```

Där exempelvis $y(5) = 9 = (8 + 10)/2$.

5: Vi bildar Hilbert matrisen med

```
n=5;H=zeros(n,n);  
for i=1:n  
    for j=1:n  
        H(i,j)=1/(i+j);  
    end  
end  
disp(H)
```

6: Funktionen blir

```
function [S]=Skalar(x,y)
    n=length(S);S=0;
    for i=1:n
        S=S+x(i)*y(i);
    end
end
```

vilket get

```
>> Skalar( [1 2 3]' , [2 -3 1] )
ans =
    -1
```

7: Bilda vektorerna x och y samt utför polynomanpassningen med

```
>> x=[ 0 1 2 3]; y=[1.7 2.2 2.9 3.4];
>> p1=polyfit(x,y,1)
>> plot( x , polyval(p1,x) , 'b-' , x , y , 'r+');
```

Notera att vi inte behöver bilda en tärate x -vektor innan vi plottar då vi skall rita upp en rät linje.

8: Följande kommandon beräknar nuvärdet

```
>> k=1:24;
>> Pnu=500*sum( 1./ ( 1.0017.^k ) )
Pnu =
    1.1749e+04
```

Nuvärdet av en månadsbetalning på 500kr under 24 månader är alltså 11749kr. Vi bör alltså betala kontant.

9: Vi skriver en funktion

```
function [y]=funk(x)
    y=cos(x).*exp(-2*x)+2*x;
```

Då funktionen klara vektor argument kan vi skriva

```
>> I=quad('funk',0,1)
I =
    1.3935
```

8: Vi bildar en $2 \times N$ matris där varje element svarar mot ett tärningsslag. Sedan summerar vi över raderna och får en vektor där varje element är summan av två kast.

```
>> N=100;
>> A=randi([1 6],2,N);
>> S=sum(A); mean(S)
ans =
    6.8900
```

Medelvärdet blir alltså ungefär 7.

11: Funktionen LokaltMax kan exempelvis skrivas som

```
function [ind]=LokaltMax( x )
    n=length(x);k=0;      % k=antal lokala max.

    if x(2) <= x(1),k=k+1; ind(k)=1; end
    for i=2:n-1
        if ( x(i-1) <= x(i) ) & ( x(i) >= x(i+1) )
            k=k+1;ind(k)=i;
        end
    end
    if x(n-1) <= x(n),k=k+1; ind(k)=n; end
end
```

Vi testar med

```
>> x=[3 4 5 4 3 2 1 7 3]
>> LokaltMax( x )
ans =
     3     8
```