

## Tentamen TANA17 Matematiska beräkningar

**Provkod:** DAT1 **Godkänd:** 8p av totalt 20p **Tid:** 18:e augusti 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:** Skapa en vektor som innehåller de positiva talen  $1, 2, 3, \dots, 50$ . Använd sedan elementvisa operationer och kommandot `sum` för att beräkna summan

$$S = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{50^2}.$$

- (2p) **2:** Skapa ett tredjegradspolynom  $p(x)$  med rötter  $x_1 = 0.3$ ,  $x_2 = 1.5$  och  $x_3 = 2.9$  genom att använda `polyfit` för att hitta det polynom av grad 3 som interpolerar tabellen

| x | 0   | 0.3 | 1.5 | 2.9 |
|---|-----|-----|-----|-----|
| 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 vektor är sorterad i växande ordning

```
n=length(y);
for i=2:n
    if y(i-1)>y(i)
        test=0;
    else
        test=1;
    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 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 Matlabscript som skapar en sådan matris för en given dimension  $n$ . Scriptet skall dessutom lösa ekvationssystemet  $Ax = b$  för fallet  $n = 4$  och  $b = (1, 1, 1, 1)^T$ .

- (2p) 5: I ett tärningsspel skall vi slå tärningar ända tills totalsumman blir större än eller lika med 10. Skriv ett Matlabscript som simulerar en spelomgång och använder **randi** eller **rand** för att simulera tärningskast ända tills total summan blivit åtminstone 10. Programmet skall göra utskrifter enligt mönstret

```
Kast 1 gav 5
Kast 2 gav 1
Kast 3 gav 3
Kast 4 gav 4
Efter 4 kast har vi summan 13.
```

- (3p) 6: En bakteriepopulation förökar sig snabbt ifall näring tillförs. Vid varje tidssteg delar sig hälften av bakterierna. Dessutom räcker den näring som tillförs endast för  $10^5$  bakterier. Blir populationen större än så gör konkurrens om näring att ett stort antal inte får tillräckligt med energi och dör. Detta ger en modell för populationens utveckling

$$P_k = 1.5P_{k-1} - 2 \max(P_{k-1} - 10^5, 0)$$

där  $P_k$  är populationens storlek vid tidssteg  $k$ .

- a) Skriv en funktion som simulerar hur bakteriepopulationen utvecklas under  $N$  tidssteg. Funktionen skall kunna anropas enligt

```
>> N = 4; P0=100; P = Bakterier( N , P0 )
P =
    100.0000    150.0000    225.0000    337.5000
```

där P0 är bakteriepopulationens storlek då simuleringen startar.

- b) Använd funktionen för att genomföra  $N = 20$  tidssteg med startpopulationen  $P_0 = 5000$ . Plotta en graf som visar  $P_k$  som funktion av tidssteget  $k$ . Du skall även lägga till lämplig text på koordinataxlarna.

(3p) **7:** Vi vill lösa en ekvation

$$f(x) = \int_0^x \cos(t)e^{2t} dt - 2 = 0$$

- a) Skriv en funktion `IntegralFunktion` som beräknar  $f(x)$  ovan. Funktionen skall ta  $x$  som inparameter och returnera värdet  $f(x)$ . Använd `integral` för att beräkna integralens värde inne i funktionen.
- b) För att bilda oss en uppfattning av var roten finns skall vi plotta  $f(x)$  på intervallet  $0 \leq x \leq 2$ . Bilda därför en vektor `xvals` som innehåller  $n = 50$  jämt utspridda punkter på intervallet  $0 \leq x \leq 2$  och beräkna en vektor `fvals` med motsvarande funktionsvärden. Plotta sedan en graf över funktionen  $f(x)$  på aktuellt intervall.
- c) Vi ser att en rot finns runt  $x = 0.9$ . Använd nu `fzero` för att lösa ekvationen  $f(x) = 0$ .

(4p) **8:** Tecken- och stränghantering är ibland nödvändigt för att presentera beräkningsuppgifter för användare. Lös följande uppgifter:

- a) Skriv en funktion `CountLetters` med en sträng som inparameter som räknar hur många bokstäver en given sträng innehåller. Använd Matlabfunktionen `isletter`. Funktionen skall ge utskriften “Strängen innehåller <antal> bokstaver”.
- b) En vektor kan skrivas ut till skärmen med kommandot `disp`. Det blir inte särskilt snyggt om elementen i vektorn har väldigt olika storleksordning. Vi vill därför förbättra utskriften genom att skriva en funktion `displayVector` som skriver ut vektorn på följande sätt

```
>> displayVector( 'x' , [-2.845 0.323 12.3*10^-6 1002.4] )  
x=( -2.85e+00 , 3.23e-01 , 1.23e-05 , 1.00e+03 )
```

Vi vill alltså skriva varje element på exponentform med två decimaler.

## Lösningsförslag till tentan för TANA17 Augusti 2016.

### 1: Skriv

```
>> x=1:50;
>> S=sum(1./(x.^2))
S =
    1.6251
```

### 2: Skriv

```
>> x=[ 0 0.3 1.5 2.9]; y=[1 0 0 0];
>> p=polyfit(x,y,3)
p =
   -0.7663    3.6015   -4.3448    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. Alltså

```
n=length(y); test=1;
for i=2:n
    if y(i-1)>y(i)
        test=0;
    end
end
```

Det går även att avbryta med `break` ifall vi hittar ett exempel på element i avtagande ordning.

### 4: Matrisen skapas med programmet

```
N=4;
A=zeros(N,N);
for i=1:N
    for j=1:N
        A(i,j)=1/(i+j);
    end
end
```

Vi skapar sedan högerledet och löser ekvationssystemet med

```
b=ones(N,1);
x=A\b; x'
ans =
   -20.0000   180.0000  -420.0000   280.0000
```

5: Vi kan använda programmet

```
summa=0;antal=0;
while summa<10
    antal=antal+1;
    kast=randi([1 6],1);
    summa=summa+kast;
    disp(['Kast ',num2str(antal), ' gav ',num2str(kast)]);
end
disp(['Efter ',num2str(antal),' har vi summan ',num2str(summa)]);
```

6: Funktionen blir

```
function P=Bakterie( N , P0 )
    P=zeros(1,N);
    P(1)=P0;
    for k=2:N
        P(k)=1.5*P(k-1)-2*max( P(k-1)-10^5 , 0 );
    end;
end
```

Vi använder den för att plotta grafen med

```
N=20;P0=5000;
P=Bakterie(N,P0);
plot(P);
ylabel('Population: P_k');
xlabel('Tid: k')
```

7: På filen IntegralFunktion.m skriver vi

```
function [f]=IntegralFunktion(x)
    g=@(t)cos(t).*exp(2*t);
    f=integral( g , 0 , x )-2;
```

Vi plottar sedan funktionen och löser ekvationen med

```
n=50;
xvals=2*(0:n-1)/(n-1);fvals=zeros(1,n);
for i=1:n,fvals(i)=IntegralFunktion(xvals(i));end
plot(xvals,fvals)
x0=fzero(@IntegralFunktion,0.9)
```

8: Funktionerna blir

```

function [Antal]=CountLetters( str )

    n=length(str);
    Antal=0;
    for i=1:n
        if isletter( str(i) )
            Antal=Antal+1;
        end
    end
    fprintf(1,'Strängen innehåller %i bokstäver\n',Antal)
end

```

och

```

function displayVector( name , vector )

    n=length(vector);

    % skriv först ut början: <name>=(
    fprintf(1,' %s=(',name);

    % skriv nu ut ett elemnt i taget med , och mellanslag.
    for i=1:n-1
        fprintf(1,' %7.2e ',vector(i));
    end

    % skriv nu ut sista elementet utan , efter.
    fprintf(1,' %7.2e )\n',vector(n) );
end

```