Introduktion i MATLAB

Övningsuppgifter

Hampus Malmberg
Jesper Karlsson
Sven Jacobsson

31 augusti 2011

Förord

Detta dokument innehåller uppgifter samt lösningsförslag till kompendiet *Introduktion i MATLAB*. Uppgifterna är lagda på en grundläggande nivå och kräver ingen kunskap i MATLAB förutom den som presenteras i kompendiet.

Kontakt

Vi som har håller i introduktionen och skrivit undervisningmaterialet är:

Hampus Malmberg *<hammal@student.chalmers.se>*
Jesper Karlsson *<jeskarl@student.chalmers.se>*
Sven Jacobsson *<jsven@student.chalmers.se>*

Om ni har en fråga om kursens innehåll eller upptäckt något fel är ni alltid välkomna att höra av er till någon av oss.

Innehåll

1 Uppgifter

1.1 Grundläggande hantering av värden

1. Deklarera variablerna

   (a) $a = 4$

   (b) $b = 1$

   (c) $c = a - b$

2. Skapa en lista

   (a) från 0 till 10 med steglängden ett, alltså $\begin{bmatrix}0 & 1 & 2 & \dots & 9 & 10\end{bmatrix}$.

   (b) från 0 till 10 med steglängden 2.

   (c) från 5 214 till 8 286 med 7 jämnt utspridda värden.

   (d) från 174 417 till 1 529 842 med 26 värden.

3. Deklarera en variabel d som 361 jämnt utspridda värden mellan 0 och 2π. Vad blir sinus för det 181a värdet i d? Var det vad du hade förväntat dig? Varför eller varför inte?

4. Konstruera en 3 x 3 lista (T) med hjälp av fyra stycken mindre listor (a, b, c och d) med olika antal värden.

   Ledning: Om $a = 2$ *och* $b = 1$ *blir* $C = \begin{bmatrix}a; b\end{bmatrix} = \begin{bmatrix}2\\1\end{bmatrix}$.

5. Deklarera variabeln $x = \begin{bmatrix}1.2 & 3.2 & 1.9\end{bmatrix}$.

   (a) Öka alla värden i x med 2 och spara i variabeln y.

   (b) Minska det andra värdet i y med 3 och spara endast detta värdet i en ny variabel z.

6. Deklarera variabeln $A = \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$ och kvadrera varje värde i A med hjälp av endast ett kommando istället för varje värde var för sig. Kontrollera att resultatet stämmer med det du förväntat dig.

7. Vad händer om vi multiplicerar de två listorna $a = \begin{bmatrix}3 & 1\end{bmatrix}$ och $b = \begin{bmatrix}2 & 2\end{bmatrix}$? Börja med a .* b. Vad händer om vi försöker multiplicera a och b utan punkten?

1.2 Hjälpkommandon

1. Läs om funktionen *roots* i MATLABs dokumentation och hitta sedan för vilka x funktionen $f(x) = x^3 - 2x^2 - x + 2 = 0$.

2. Kommandot `doc size` och `doc length` öppnar dokumentationen för *size* respektive *length*-funktionerna i MATLAB. Försök att förstå hur dessa fungerar och använd funktionerna för att beräkna:

   (a) längden på listan $\begin{bmatrix}1 & 3 & 5 & 7 & 9\end{bmatrix}$.

   (b) längden på listan `primes(37)`.

   (c) samtliga dimensioner av `perms([1 2 3])`.

1.3 Tal och Matematiska funktioner

1. Vi vill räkna ut sinus för 32°. Vilken funktion gör detta i MATLAB och vad blir svaret?

2. Beräkna

   (a) $\ln(3)$

   (b) $\ln(e^2)$

   (c) $\sqrt{\log_{10}(100) + \sqrt{21 + 17 + 11}}$

   (d) $\sin\left(\frac{7\pi}{12}\right)$

   (e) $\sqrt[3]{-8}$

   (f) Räkna ut $|\sin(x)|$
   för en lista med ett lämpligt antal värden då
   $\forall\ x : -1 \leq x \leq 1$

   (g) $\sqrt{-1}$

3. Deklarera variabeln x från -5 till 5 och beräkna absolutbeloppet av alla värden i x med ett kommando i en lista med lämpligt antal punkter.

4. Skapa en lista med tio slumpmässigt valda värden mellan 5 och 15 och avrunda dessa värden till närmaste heltal.

5. Vi vill simulera ett tärningskast. Låt MATLAB välja ett slumpmässigt värde mellan ett och sex. Vi fuskar lite i tärningspelet och väljer att avrunda alla värden upp till närmaste heltal. Hur gör vi?

1.4 Scriptfiler

1. Skapa en script-fil *matte.m* och skriv in alla kommandon för att lösa uppgift 1.3.3-1.3.5. Kör filen och kontrollera att allt fungerar.

2. Kommentera koden i föregående uppgift.

3. Publicera m-filen och redovisa för handledare.

1.5 Grafisk visualisering

Nedan förväntas samtliga lösningsgångar skrivas i script-filer.

1. Låt $y = x \cdot e^{-x}, x \in [0, 5]$ och plotta x mot $\frac{y}{2}$, x mot y, x mot $1,5y$, x mot $2y$, x mot $2,5y$ och x mot $3y$ i samma figur med olika färger.

2. Deklarera variabeln x från -2π till 2π och variabeln $y = \sin(x)$. Jämför figurerna för `plot(x,y)` och `plot(y,x)`. Vad är det som händer och varför?

3. Deklarera variablerna a och b från $-\pi$ till π. Variabeln a ska ha 100 jämnt utspridda värden och variabeln b skall ha 10 000 värden. Jämför plottarna för a mot $\sin(100a)$ och b mot $\sin(100b)$. Ser figurerna likadana ut? Om inte, vilken visar rätt information? Varför?

4. Plotta $y(x) = \frac{1}{x}$ där $-1 \leq x \leq 1$. Vad är det som inte stämmer med figuren? Varför? Hur skulle man kunna lösa problemet?

5. Deklarera variabeln t från 0 till 2π och plotta enhetscirkeln med x och y som funktioner av t.

1.6 Funktionshantering

1. Skapa anonyma funktioner som beräknar:

   (a) absolutbeloppet av sinus för ett värde.

   (b) $\sqrt{1 - \sin^2(x)}$

2. Använd funktionen ifrån 1b för att beräkna funktionsvärdet för $x = \frac{\pi}{2}$.

3. Skapa en funktionsfil som

   (a) beräknar absolutbeloppet av tredjeroten av ett värde.

   (b) omvandlar grader till radianer.

   (c) omvandlar radianer till grader.

1.7 Logik

1. Vad händer om vi jämför två listor med tal? Testa med $A = \begin{bmatrix}1 & 9 & 2 & 13\end{bmatrix}$ och $B = \begin{bmatrix}1 & 5 & 2 & 21\end{bmatrix}$. Hur hanterar MATLAB `A < B`, `A == B` osv?

2. Gör en loop som går igenom en lista med värden. Loopen du skriver ska kontrollera om ett visst värde är större, lika med eller mindre än 13. Om värdet är mindre än 13 skall det multipliceras med sig självt. Är värdet lika med 13 görs ingenting men om det däremot är större än 13 ska det divideras med 2. Det nya värdet ska sparas i listan.
   *Tips: Deklarera en lista med slumpmässigt valda värden genom rand(1,5). Multiplicera denna lista med 13*2 för att få lika stor chans för värden större än 13 som för värden mindre än 13. Slutligen kan ni skriva round(lista) för att få heltal.*

3. Konstruera en loop som räknar ut medelvärdet av alla värden i en lista.

4. Konstruera en funktionsfil *myfactorial.m* som räknar ut *n!*. *'!'* står för fakultet och definieras som

   $$n! \equiv n \cdot (n-1) \cdots 2 \cdot 1$$

   Till exempel så blir $3! = 3 \cdot 2 \cdot 1 = 6$.

5. Beräkna $\sum_{k=1}^{1923912} k$. *Alltså summan av alla tal från 1 till och med 1 923 912.*

2 Lösningsförslag

2.1 Grundläggande hantering av värden

1. Vi skriver följande i *Command Window*

   (a)
   ```
   >> a = 4

   a =
       4
   ```

   (b)
   ```
   >> b = 1

   b =
       1
   ```

   (c)
   ```
   >> c = a-b

   c =
       3
   ```

2. Det finns flera sätt att skapa listor. Kortare listor går till och med att deklarera för hand. Poängen med uppgiften är att vi ska lära oss använda *linspace* och *a:n:b*.

   (a)
   ```
   >> 1:10

   ans =
        0    1    2    3    ...    9    10
   ```

   (b)
   ```
   >> 0:2:10

   ans =
        0    2    4    6    8    10
   ```

   alternativt

   ```
   >> linspace(0,10,6)

   ans =

        0    2    4    6    8    10
   ```

(c)
```
>> linspace(5214,8286,7)

ans =
        5214        5726         ...         8286
```

(d)
```
>> linspace(174417,1529842,26)

ans =
      174417      228634       ...      1529842
```

3.
```
>> d = linspace(0,2*pi,361);
>> sin(d(181))

ans =

   1.2246e-16
```

`d(181)` $\approx \pi$ (på grund av avrundningsfel i programmet) så vi förväntar oss $\sin(\pi) = 0$ och svaret blir $1.22 \cdot 10^{-16} \approx 0$.

4. Till exempel

```
>> a = [1 2; 4 5];
>> b = [3; 6];
>> c = [7 8];
>> d = 9;
>> T = [a b; c d]

T =

     1     2     3
     4     5     6
     7     8     9
```

5. (a)
```
>> x = [1.2 3.2 1.9];
>> y = x+2

y =

    3.2000    5.2000    3.9000
```

(b)
```
>> z = y(2)-3

    z =

        2.2000
```

6.
```
>> A = [1 2; 3 4];
>> A.^2

ans =

     1     4
     9    16
```
Punkten gör att varje värde kvadreras elementvis. Glömmer man punkten blir det någonting helt annat (matrismultiplikation) och vi får vänta på svar om vad detta innebär till kursen i *Linjär Algebra* i läsperiod 3.

7.
```
>> a = [3 1];
>> b = [2 2];
>> a.*b

ans =

     6     2

>> a*b

??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

Återigen så innebär punkten elementvis multiplikation så håll dig till den för nu.

2.2 Hjälpkommandon

1. För att hitta nollställen till polynomet $f(x) = x^3 - 2x^2 - x + 2$ med hjälp av funktionen *roots* måste vi skriva om det som `f = [1 -2 -1 2]`.

```
>> r = roots(f)

r =

   -1.0000
    2.0000
    1.0000
```

2. Funktionen *length* returnerar den största dimensionen variabeln innehåller och *size* returnerar samtliga dimensioner vilket innebär att vi kan lösa problemen på följande sätt:

   (a)
```
>> length([1 3 5 7 9])

ans =

     5
```

   (b)
```
length(primes(37))

ans =

    12
```

   (c)
```
>> size(perms([1 2 3]))

ans =

     6     3
```

   Vilket säger oss att `perms([1 2 3])` har 6 rader och 3 kolumner.

2.3 Tal och matematiska funktioner

1. Funktionen som räknar ut sinus med argumentet uttryckt i grader är *sind* vilket ger oss

```
>> sind(32)

ans =

    0.5299
```

2. (a)
```
>> log(3)

ans =

    1.0986
```
(b)
```
>> log(exp(2))

ans =

     2
```
(c)
```
>> sqrt(log10(100)+sqrt(21+17+11))

ans =

     3
```
(d)
```
>> sin(7*pi/12)

ans =

    0.9659
```
(e)
```
>> nthroot(-8,3)

ans =

    -2
```

(f)
```
>> x=linspace(-1,1);
>> abs(sin(x))

ans =

    0.8415    0.8304    0.8190    ...    0.8304    0.8415
```

(g)
```
>> sqrt(-1)

ans =

         0 + 1.0000i
```

3.
```
>> x = linspace(-5,5);
>> abs(x)

ans =

    5.0000    4.8990    4.7980    ...    4.8990    5.0000
```

4.
```
>> round(rand(10,1)*10+5)

ans =

     9
     5
    14
    14
    10
    10
     8
    14
     9
     6
```

5.
```
>> ceil(rand(1)*5+1)

ans =

     4
```

2.4 Script-filer

1. `>> edit matte.m`

   ```
   x = linspace(-5,5);
   abs(x)

   round(rand(10,1)*10+5)

   ceil(rand(1)*5+1)
   ```

2. ```
   %% Uppgift 1.3.3
   x = linspace(-5,5);

   %Variabeln x deklareras från -5 till 5 i 100 punkter.

   abs(x)

   %Absolutbeloppet av alla värden i x räknas ut.

   %% Uppgift 1.3.4

   round(rand(10,1)*10+5)

   %En lista med 10 slumpmässigt valda heltal
   %mellan 5 och 15 deklareras till ans.

   %% Uppgift 1.3.5

   ceil(rand(1)*5+1)

   %Ett slumpmässigt värde mellan 1 och 6 avrundas uppåt
   %till närmaste heltal och deklareras till variabeln ans.
   ```

3. Publicera koden genom att klicka på *Publish matte* i *File*-menyn.
   ```

2.5 Grafisk visualisering

1. Det finns flera lösningar på problemet.

   ```
   x = linspace(0,5);
   y = x.*exp(-x);

   hold on
   plot(x,y/2)
   plot(x,y,'g')
   plot(x,1.5*y,'r')
   plot(x,2*y,'c')
   plot(x,2.5*y,'m')
   plot(x,3*y,'y')
   hold off
   ```

   innebär att vi plottar en linje i taget i samma figur och vi specificerar varje linjes färg.

   ```
   x = linspace(0,5);
   y = x.*exp(-x);

   plot(x,y/2,x,y,'g',x,1.5*y,'r',x,2*y,'c',x,2.5*y,'m',x,3*y,'y')
   ```

   ger exakt samma resultat som ovan men vi har komprimerat antal rader kod.

   ```
   x = linspace(0,5);
   y = x.*exp(-x);

   plot(x,y/2,x,y,x,1.5*y,x,2*y,x,2.5*y,x,3*y)
   ```

   Vi låter MATLAB välja färger åt oss vilket fungerar så länge alla funktioner ryms inom *plot(...)*.

2. ```
   x = linspace(-2*pi,2*pi);
   y = sin(x);

   hold on
   plot(x,y)
   plot(y,x)
   hold off
   ```

   ritar två funktioner där `plot(y,x)` byter axel vi plottar mot.
   ```

3.
```
a = linspace(-pi,pi);
b = linspace(-pi,pi,10000);

figure(1)
plot(a,sin(100*a))

figure(2)
plot(b,sin(100*b))
```
   Plottarna ser inte alls lika ut. *Figur 1* visar en slät och fin sinuskurva och *Figur 2* visar en kurva med kortare period. *Figur 2* är den som ger oss rätt information eftersom *Figur 1* har för låg upplösning för att få med en såpass hög frekvens.

4. Funktionen $y = \frac{1}{x}$ är som bekant inte definierad för $x = 0$. Om vi skriver

```
x = linspace(-1,1);
y = 1./x;

plot(x,y)
```

   förbinder MATLAB punkterna från $-\infty$ till ∞ eftersom det är så programmet hanterar grafritning. En punkt med en x och y koordinat förbinds med en rät linje till nästa koordinatpar.

   Ett sätt att lösa problemet är att dela upp plotten i två delar. Observera att MATLAB inte indikerar att y går mot $-\infty$ eller ∞ utan plottar ett mycket stort respektive litet tal istället. Detta är typisk problematik som kan uppstå när man räknar med datorer det är därför viktigt att känna till MATLABs begränsningar. Se kapitlet om felsökning i MATLAB kompendiet.

```
x1 = linspace(-1,0);
y1 = 1./x1;

x2 = linspace(0,1);
y2 = 1./x2;

plot(x1,y1,x2,y2,'b')
```

   där vi har specificerat färgen för y_2.

5.
```
t = linspace(0,2*pi);
x = cos(t);
y = sin(t);

plot(x,y), axis equal
```

2.6 Funktionshantering

1. (a) `y = @(x) abs(sin(x));`

   (b) `y = @(x) sqrt(1-(sin(x)^2));`

   Den observante ser att

   $$\sqrt{1-\sin^2(x)} = \sqrt{\sin^2(x)+\cos^2(x)-\sin^2(x)} = \sqrt{cos^2(x)} = |cos(x)|$$

   vilket innebär att det hade gått lika bra att skriva

   `y = @(x) abs(cos(x));`

2.
```
>> y(pi/2)

ans =

     0
```

3. (a) `>> edit myfun.m`

```
function [y] = myfun(x)
y = abs(nthroot(x,3));
end
```

   (b) `>> edit toradians.m`

```
function [y] = toradians(x)
y = x*pi/180;
end
```

   (c) `>> edit todegrees.m`

```
function [y] = todegrees(x)
y = x*180/pi;
end
```

2.7 Logik

1. ```
   A = [1 9 2 13];
   B = [1 5 2 21];

   >> A < B

   ans =

        0     0     0     1

   >> A == B

   ans =

        1     0     1     0
   ```

   Vi ser att programmet testar första elementet i listan A mot första elementet i B och returnerar ett svar för varje element.

2. ```
   x = round(13*2*rand(1,5));

   for i = 1:length(x)
       if x(i) < 13
           x(i) = x(i)^2;
       elseif x(i) > 13
           x(i) = x(i)/2;
       end
   end
   ```

3. ```
   a = rand(1,10);
   sum = 0;

   for i = 1:length(a)
       sum = sum+a(i);
   end

   avg = sum/length(a);
   ```
   ```

4. `>> edit myfactorial.m`

   ```
   function [fac] = myfactorial(n)

   fac = 1;

   while n > 0
       fac = fac*n;
       n = n-1;
   end

   end
   ```

   MATLAB har även en elementär funktion som gör just detta. Hitta denna funktion med hjälp av din kunskap om hjälpfunktioner, jämför din funktion och se så att du får samma svar.

5. ```
   sum = 0;

   for k = 1:1923912
       sum = sum+k;
   end
   ```
   ```