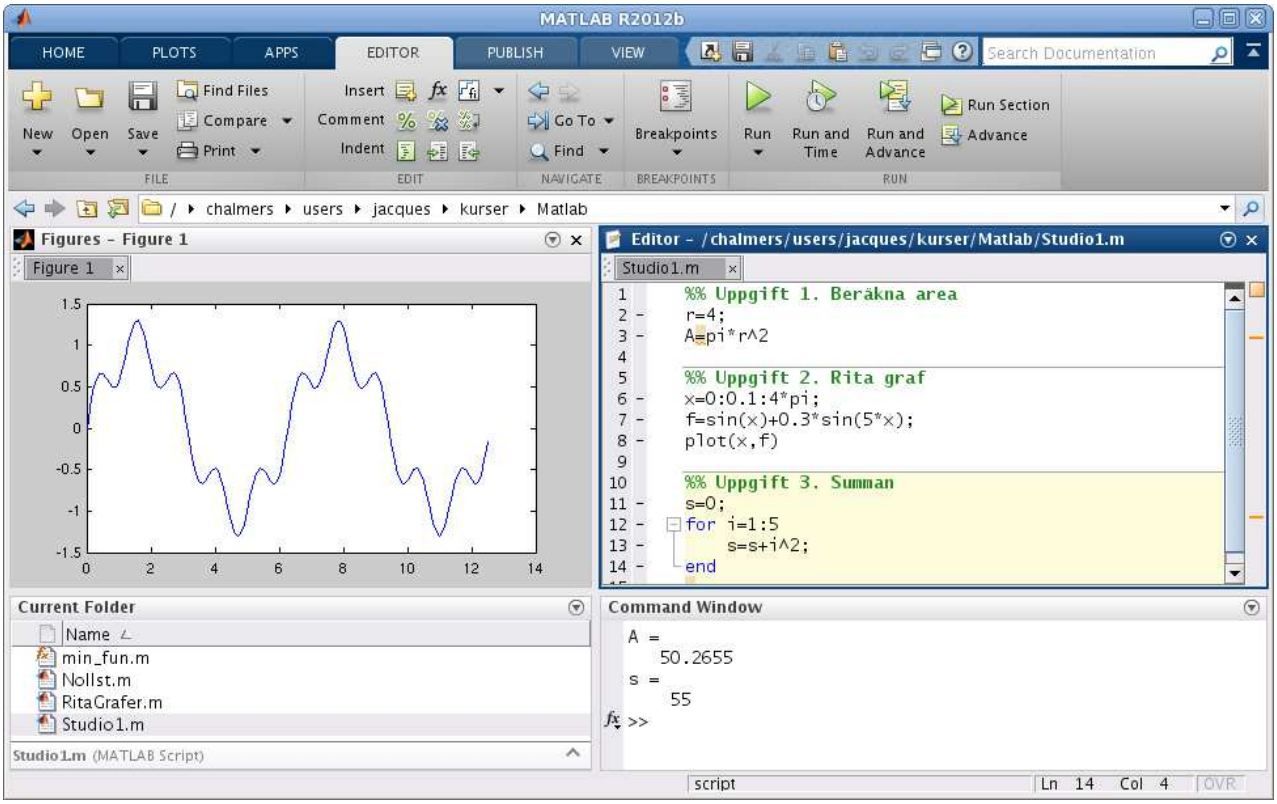
Matriser och vektorer i MATLAB

1 Inledning

Först skall vi se lite på matriser, vilket är den grundläggande datatypen i MATLAB, sedan skall vi beskriva hur vi kan lösa linjära ekvationssystem på ett effektivt sätt i MATLAB. Avslutningsvis ser vi på några användbara inbyggda matris- och vektorfunktioner.

Men allra först: För att arbeta på ett överskådligt och effektivt sätt, dokumentera arbetet och redovisa för handledare snabbt och smidigt skall ni använda er av script (som vi såg på i förra laborationen).

Så här kunde det sett ut efter det att vi löst de tre första uppgifterna i första laborationen.



Editorn i MATLAB har använts i Cell Mode (cell-läge). Skriver man en kommentar som börjar med två procent-tecken, så avgränsar det en cell. Poängen är att man kan låta MATLAB utföra kommandona från en cell, istället för hela filen.

På så sätt kan man dela upp ett stort script (för en hel laboration) i flera delar (varje deluppgift). En cell kan utföras utan att textfilen är sparad, så gör Save då och då. Har du glömt bort script och cell-läge repetera i så fall laboration 1, avsnitt 4.

Vi har också gjort en egen desktop layout så att Figures (figurfönster), Editor samt Command Window syns samtidigt. Läs i texten för laboration 1, avsnitt 8 hur man gör.

2 Något om matriser

En matris är ett rektangulärt talschema

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

Matrisen ovan har m rader och n kolonner, vi säger att den har storleken $m \times n$. Ett matriselement på rad nr i, kolonn nr j skrivs a_{ij}, där i är radindex och j är kolonnindex.

En matris av storleken $m \times 1$ kallas kolonnmatris (kolonnvektor) och en matris av storleken $1 \times n$ kallas radmatris (radvektor):

$$\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}$$

Som exempel tar vi

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 0 & 2 & 4 & 6 & 8 \end{bmatrix}$$

Vi beskriver matrisen i MATLAB enligt

```
>> A=[1 4 7 10; 2 5 8 11; 3 6 9 12]
```

och som svar får vi i Command Window utskriften

```
A =
     1     4     7    10
     2     5     8    11
     3     6     9    12
```

Man använder hakparanteser (`[ ]`) för att bygga upp matriserna. Semikolon (`;`) innanför hakparanteserna betyder radbyte.

Så här beskriver vi kolonnvektorn

```
>> b=[1; 3; 5]
b =
     1
     3
     5
```

och så här radvektorn

```
>> c=[0 2 4 6 8]
c =
     0     2     4     6     8
```

Ett matriselement a_{ij}, dvs. elementet på rad nr i, kolonn nr j, skrivs i MATLAB med `A(i,j)` och ett vektorelement b_i skrivs med `b(i)`.

Indexeringen i matriser och vektorer i MATLAB börjar alltid på `1`, vi kan inte påverka detta. Sista index för en rad eller kolonn ges av `end`. T.ex. `b(1)` är första elementet i vektorn `b` och `b(end)` är sista elementet.

Vi låter s få tredje värdet c_3, dvs. $s = c_3$ med

```
>> s=c(3)
s =
     4
```

och bildar vektorn $\mathbf{v}$ av andra och femte värdet, dvs. $\mathbf{v} = (c_2, c_5)$, med

```
>> v=c([2,5])        % v=[c(2) c(5)] går också
v =
     2     8
```

Vi kan ändra ett element i $\mathbf{v}$, t.ex. låta $v_2 = 0$, med

```
>>  v(2)=0
v =
     2     0
```

Vi låter s få värdet av a_{23}, dvs. elementet på rad 2, kolonn 3 i matrisen $\mathbf{A}$ med

```
>> s=A(2,3)
s =
     8
```

och vi bildar en radvektor $\mathbf{v}$ av rad 3, alla kolonner med

```
>> v=A(3,:)         % v=A(3,1:end) eller v=A(3,1:4) går också
v =
    3     6     9    12
```

samt en kolonnvektor $\mathbf{u}$ av rad 2-3, kolonn 2 med

```
>> u=A(2:3,2)
u =
    5
    6
```

Vi bildar en matris $\mathbf{V}$ av blocket rad 1-2, kolonn 2-3

```
>> V=A(1:2,2:3)
V =
    4     7
    5     8
```

Uppgift 1. Skriv in matriserna $\mathbf{A}$, $\mathbf{b}$ och $\mathbf{c}$ i MATLAB. Skriv sedan ut matriselementen a_{23}, b_2, c_3. Ändra a_{23} genom att skriva `A(2,3)=15`. Gör ett script och använd cell-läge så att ni kan bygga på med kommande uppgifter.

3 Linjära ekvationssystem

Linjära ekvationssystem kan vi lösa med MATLAB om vi först skriver dem på matrisform. Vi tar som exempel: Ekvationssystemet

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 14 \\ 3x_1 + 2x_2 + x_3 = 10 \\ 7x_1 + 8x_2 = 23 \end{cases}$$

skrivs på matrisform

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 10 \\ 23 \end{bmatrix}$$

dvs.

$$\mathbf{Ax} = \mathbf{b}, \quad \text{med } \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 0 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{och} \quad \mathbf{b} = \begin{bmatrix} 14 \\ 10 \\ 23 \end{bmatrix}$$

Vi bildar koefficientmatrisen $\mathbf{A}$ och högerledsvektorn $\mathbf{b}$ med

```
>> A=[1 2 3;3 2 1;7 8 0]
A =
     1     2     3
     3     2     1
     7     8     0

>> b=[14;10;23]
b =
    14
    10
    23
```

Med kommandot `rref` kommer vi till radreducerad trappstegsform (row-reduced-echelon form) så att vi kan läsa av lösningen till $\mathbf{Ax} = \mathbf{b}$.

Först bildar vi den utökade matrisen $\mathbf{E} = [\,\mathbf{A}\ \mathbf{b}\,]$ med

```
>> E=[A b]
E =
     1     2     3    14
     3     2     1    10
     7     8     0    23
```

och sedan får vi den reducerade matrisen med

```
>> R=rref(E)
R =
     1     0     0     1
     0     1     0     2
     0     0     1     3
```

Lösningen ser vi i sista kolonnen i `R` och vi har

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Som ytterligare ett exempel ser vi på följande ekvationssystem med oändligt många lösningar

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 10 \\ 3x_1 + 2x_2 + x_3 = 14 \\ 7x_1 + 8x_2 + 9x_3 = 46 \end{cases}$$

eller på matrisform

$$\mathbf{Ax} = \mathbf{b} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \\ 46 \end{bmatrix}$$

```
>> A=[1 2 3;3 2 1;7 8 9]
A =
     1     2     3
     3     2     1
     7     8     9

>> b=[10;14;46]
b =
    10
    14
    46
```

Vi reducerar utökande matrisen med

```
>> R=rref([A b])
R =
     1     0    -1     2
     0     1     2     4
     0     0     0     0
```

Vi har en fri variabel. Om vi sätter $x_3 = t$ får vi

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 + t \\ 4 - 2t \\ t \end{bmatrix}$$

där t är ett godtyckligt reellt tal.

Uppgift 2. Skriv följande ekvationssystem på matrisform och lös dem sedan med `rref`.

$$\begin{cases} x_1 + 5x_2 + 9x_3 = 29 \\ 2x_1 + 5x_3 = 26 \\ 3x_1 + 7x_2 + 11x_3 = 39 \end{cases} \qquad \begin{cases} x_1 + x_2 + 3x_3 + 4x_4 = 2 \\ -2x_1 + 2x_2 + 2x_3 = -4 \\ x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ x_1 - x_2 - 2x_3 - x_4 = 1 \end{cases}$$

Skulle det finnas oändligt många lösningar skriv upp en formel för samtliga lösningar.

4 Matris- och vektorfunktioner

Vi ser nu på några användbara inbyggda funktioner som tar matriser eller vektorer som argument. För exemplen använder vi följande matris och vektorer.

$$\mathbf{A} = \begin{bmatrix} 11 & 4 & 3 & 7 \\ 2 & 6 & 8 & 5 \\ 9 & 12 & 1 & 10 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 4 & 2 & 8 & 0 & 6 \end{bmatrix}$$

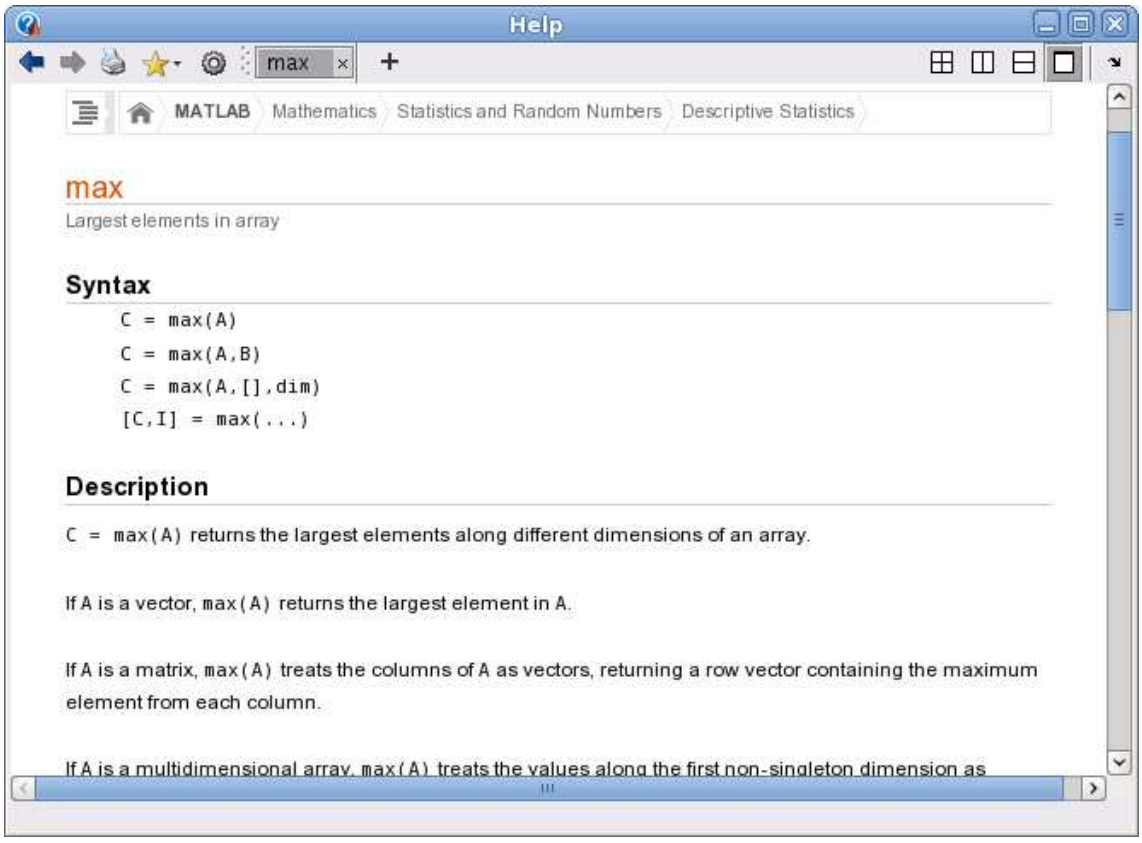
Antalet rader och kolonner i $\mathbf{A}$ får vi med

```
>> [m,n]=size(A)
m =
     3
n =
     4
```

och antal element i vektorn $\mathbf{c}$ ges av

```
>> l=length(c)
l =
    5
```

Största eller minsta elementet i en vektor eller en matris får man med funktionerna `max` och `min`. Här är hjälptexten till `max`.



Vi ser hur vi får största elementet i en vektor och hur vi får de största elementen i varje kolonn för en matris.

```
>> v=max(c)
v =
    8

>> v=max(A)
v =
    11    12     8    10
```

Vi ser också att vi med `[v,i]=max(c)` kan få reda på var det maximala värdet finns någonstans.

Uppgift 3. Skriv in matrisen **A** samt vektorerna **b** och **c** vi använt som exempel. Pröva `size` på vektorerna **b** och **c**. Hur ser man att den ena är en kolonnvektor och att den andre är en radvektor? Bestäm största och minsta elementet i matrisen **A** med hjälp av funktionerna `max` och `min`. Vad har dessa element för rad- respektive kolonnindex?

Summan och produkten av elementen i vektorn fås med `sum` och `prod`. För en matris blir det summan eller produkten av varje kolonn.

```
>> s=sum(b)
s =
     9

>> s=sum(A)
s =
    22    22    12    22
```

I förra laborationen beräknade vi en summa $s = 3 + 4 + 5 + \cdots + 52$ med en `for`-sats, vi skulle även kunna beräkna den med `sum` enligt

```
>> s=sum(3:52)
s =
        1375
```

Detta kallas att vektorisera beräkningen.

Vi kan också dela upp i två satser

```
>> t=3:52;      % Bildar en vektor t med 3, 4, ..., 52
>> s=sum(t)     % Summerar alla element i vektorn t
s =
        1375
```

Först bildar vi vektorn `t` med elementen $3, 4, \cdots, 52$ och sedan summerar vi elementen i vektorn.

Uppgift 4. Beräkna summan $s = 1^2 + 2^2 + 3^2 + 4^2 + 5^2$ med `sum` och komponentvis kvadrering.

Vill vi sortera en vektor i stigande ordning gör vi det med funktionen `sort`. För en matris blir det varje kolonn som sorteras. För att sortera i avtagande ordning se hjälptexten för `sort`.

Funktionerna `zeros` och `ones` används för att bygga upp matriser fylla med nollor respektive ettor. Med t.ex. `B=ones(size(A))` får vi en matris `B` med samma storlek som `A` fast fylld med ettor. Vidare ger funktionerna `rand` och `randn` en matris fylld med slumptal (se hjälptexterna).

1 Målsättning

Avsikten med laborationen är dels att komma igång med att arbeta effektivt i MATLAB och dokumentera arbetet, dels att vi i MATLAB skall kunna beskriva och lösa ett linjärt ekvationssystem. Vi skall också kunna modifiera en matris eller vektor samt ta ut delar av eller enskilda element. Vidare skall vi bekanta oss med inbyggda funktioner som opererar på matriser och vektorer.

2 Kommentarer och förklaringar

Här följer kommentarer till några avsnitt i laborationstexten.

I avsnittet "Något om matriser" ser vi på hur vi bygger upp matriser och vektorer. Vi ser också på hur vi kan modifiera sådana samt ta ut delar av en matris eller vektor. Det senare kommer vi få stor nytta av då vi arbetar med problemlösning av t.ex. teknisk natur. Då är det inte ett ekvationssystem man löser utan kanske en följd av sådana där ingående matriser och vektorer successivt skall förändras.

Dessa färdigheter kommer vi få mycket träning på i samband med kursen i linjär algebra som ni läser till våren.

Avslutningsvis beskriver avsnittet "Matris- och vektorfunktioner" hur vi lätt tar reda på storleken på en matris. Ibland låter man matriser eller vektorer växa upp medan man löser ett problem och det är då bra att efteråt kunna bestämma storleken (`size`, `length`).

Att enkelt kunna summera alla element i en matris eller vektor behövs då och då, precis som att ta produkt, minsta eller största värde samt sortera element (`sum`, `prod`, `min`, `max`, `sort`).

Funktionerna `zeros` och `ones` kommer vi använda ofta, och ibland vill vi ha slumptal med `rand`.

3 Lärandemål

Efter denna laboration skall du i MATLAB

- kunna bygga upp och modifiera matriser och vektorer
- kunna lösa ekvationssystem $\mathbf{Ax} = \mathbf{b}$ med `rref`
- kunna bestämma storlek av en matris eller vektor med `size` och `length`
- kunna bestämma största och minsta värde i en matris eller vektor samt bilda summa och produkt med `min`, `max`, `sum` och `prod`