

TANA17 Matematiska beräkningar med MATLAB för M, DPU

Fredrik Berntsson, Linköpings Universitet

Målet med kursen är att

- Ge grundläggande färdighet i att använda MATLAB för att lösa beräkningsuppgifter samt visualisera beräkningsresultat.
- Introducera programmering, exempelvis villkors- eller repetitionssatser, och ge övning i att skriva MATLAB program.
- Ge exempel på problem från tillämpningar där MATLAB kan användas för att hitta en lösning.

I kursen ingår följande moment:

- **Föreläsningar** Presentation av teori och exempel.
- **Lektioner** Eget arbete med uppgifter i datorsal. Förberedelse för tentamen.
- **Laborationer** Obligatoriska lite större uppgifter.
- **Seminarie** Större uppgift med rapportskrivning och opposition.

Gruppindelning för laborationer Två studenter i varje grupp. Listor finns på första lektionen. Måste vara klart till första laborationsstillfället.

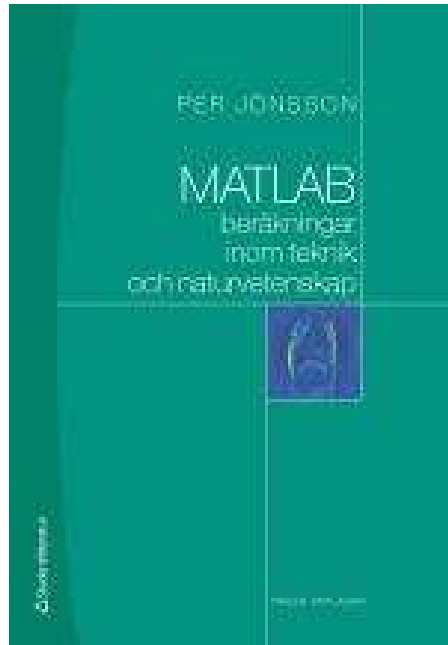
Lärare på kursen är Fredrik Berntsson, Samira Nikkar, Markus Wahlsten, Ingerd Skoglund, Viktor Linders.

Hittar oss i Hus B. Ing 23-25. 2:a vån.

För att bli godkänd på kursen krävs

- **Laborationer** Totalt fyra laborationer att lösa och redovisa skriftligt.
- **Seminarie uppgift** En lite större uppgift presenteras skriftligt. En rapport skall skrivas.
- **Opposition** Muntlig opposition på annan grupps skriftliga rapport.
- **Tenta** Både teoretiska och praktiska uppgifter. Löses i datorsal.

Tips Turas om att skriva på lektionerna. Det är mycket enklare att titta på när någon anna skriver. Även om man förstår måste man kunna många kommandon utantil för att kunna lösa uppgifter effektivt.



P Jönsson, MATLAB beräkningar inom teknik och naturvetenskap, tredje upplagan, Studentlitteratur

Boken innehåller

- Grundläggande programmering i MATLAB.
- Genomgång av MATLAB funktioner för lösning av specifika problem. Exempelvis Interpolation eller ekvationslösning.

Mycket material!

Lektioner och Laborationer Häften med uppgifter finns på kurshemsidan.

Föreläsning 1

- Variabler, Aritmetriska operationer. Prioritet.
- Vektorer och Matriser. Matrisoperationer.
- Att skapa och sätta samman matriser.
- Index och kolonnotation.
- Enkla utskrifter med `disp`.

Variabler

I Matlab skapas en variabel genom att man anger dess namn och ger den ett värde

```
>> x=3.67
```

```
x =  
3.6700
```

Variabeln kan sedan användas i beräkningar

```
>> y=log(1+x)/sin(x)
```

```
y =  
-3.0569
```

Exempel Beräkna uttrycket

$$y = \frac{1 + e^{1+x}}{1 + \sqrt{3}}.$$

I Matlab

```
>> y=( 1+exp(1+x) )/( 1+sqrt(3) );
```

Används ; fås ingen utskrift av beräkningsresultatet. Istället `disp(y)`.

Variabel namn Tillåtna tecken är bokstäver, siffror, och `_`. Måste börja med en bokstav. MATLAB skiljer på stora och små bokstäver.

Tips: Bra valda variabelnamn gör program lätta att förstå!

Exempel Variablerna i, j, k, n, m innehåller heltalsvärden medan variablerna x, y, z innehåller reella värden.

Ofta bra med långa beskrivande namn.

Aritmetriska operationer och standardfunktioner

I MATLAB finns de aritmetiska operationerna $+$ $-$ $*$ $/$ $\backslash$ och $^$ definierade.

Exempel Vad blir resultatet då följande uttryck beräknas:

```
>> x=2.0; y=-1.2;  
>> z=x+y-3*x^2;
```

För att kunna tolka uttryck behöver vi känna till **prioritetsordningen**.

Först $^$, sedan $*$ $\backslash$ $/$ och sist $+$ $-$. Vid lika prioritet beräknas uttryck från vänster till höger.

Exempel Vad blir värdet av

```
>> x=3.0; y=1/2*x^2/3*2
```

För att förtydliga hur uttrycket skall beräknas man parenteser användas.

I MATLAB finns ett flertal standardfunktioner implementerade.
Tillexempel finns `sqrt`, `sin`, `cos`, `exp`, `log`, `atan`,...

Exempel Beräkna π genom

```
>> Pi = 4*atan(1)
Pi =
    3.1416
```

För att se flera decimaler kan man ändra utskriftsformat:

```
>> format long, Pi , format short
Pi =
    3.141592653589793
```

Alla variabler sparas med ungefär 15 decimalers noggrannhet.

Exempel Låt $x = 10^{-5}$ och beräkna

$$y_1 = \frac{1 - \cos(x)}{\sin^2(x)}, \quad \text{och} \quad y_2 = \frac{1}{1 + \cos(x)}.$$

Verifiera att bägge uttrycken ger samma värde. I Matlab

```
>> x=10^-5; y1=(1-cos(x))/sin(x)^2;  
>> y2=1/(1+cos(x))  
y2 =  
    0.5000  
>> abs( y1 - y2 )  
ans =  
    4.1374e-08
```

Funktionen `abs` beräknar *absolut beloppet* av ett tal. Oftast gör beräkningsfel att två tal ej blir exakt lika.

Exempel En 2×3 matris innehåller värdena

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} = \begin{pmatrix} 1 & 3 & 7 \\ 4 & -2 & 1 \end{pmatrix}.$$

I Matlab skapas matrisen med kommandot

```
>> A = [ 1 , 3 , 7 ; 4 -2 1 ];
```

Tecknet ; betyder ny rad och , nytt element. Behöver ej , annat än för tydlighet.

På samma sätt skapas en 1×3 vektor

```
>> x = [ 1  2  4 ];
```

eller en 3×1 vektor

```
>> y = [ 1 ; 2 ; 4 ];
```

Operationen $'$ betyder transponat av en matris eller vektor.

Vi får alltså att

```
>> x - y'  
ans =  
      0      0      0
```

Matrisoperationer

För matriser och vektorer finns operationerna $+$ $-$ $*$ $/$ $\backslash$ och $^$ definierade. Dessa följer de vanliga matematiska räkneregler.

Exempel Beräkna en produkt $A*x$ genom

```
>> A = [ 1 , 3 , 7 ; 4 -2 1 ; 3 1 -4];  
>> x = [ 1 ; 2 ; 4 ];  
>> b = A*x  
b =  
    35  
     4  
    -11
```

Vid matrisberäkningar måste dimensionerna passa. Exempel är att $^$ kräver kvadratiska matriser.

Exempel Skapar vi tre matriser

```
>> A=[ 1 2 -1 ; 2 -3 1];  
>> B=[ 2 -1 ; -1 1 ; 0 3];  
>> C=[ -1 2 ; 0 1 ];
```

så kan vi beräkna

```
>> D = A*B+C
```

```
D =  
-1      0  
7      -1
```

Division mellan matriser skall tolkas som att A/B betyder AB^{-1} och $A \setminus B$ betyder $A^{-1}B$.

Exempel Skriv in matriserna

$$A = \begin{pmatrix} 2 & 1 \\ 4 & -2 \end{pmatrix}, \quad \text{och} \quad B = \begin{pmatrix} 1 & -2 \\ 2 & 4 \end{pmatrix}.$$

Beräkna $A^{-1}B$, BA^{-1} och $A^{-1}A$. I Matlab

```
>> A=[2 1 ; 4 -2 ]; B=[ 1 -2 ; 2 4];  
>> C1 = A\B; C2 = B/A; D = A\A  
D =  
    1    0  
    0    1
```

Skilnadsmatrisen $C1 - C2$ blir noll-skiljd alltså är $A^{-1}B \neq BA^{-1}$.

Att skapa större matriser

Funktionerna `zeros`, `ones`, och `rand` kan användas för att skapa matriser av godtycklig storlek fylla med nollor, ettor, eller slumpstal. Funktionen `eye` ger en enhetsmatris.

Exempel Skapa en 20×2 matris med slumpstal och beräkna summan av elementen i varje kolumn. I MATLAB skriver vi

```
>> A=rand( 20 , 2); B=ones(1,20);  
>> S = B*A  
S =  
    12.8407    9.8512
```

Vi kan sätta samman två matriser till en större. Förutsatt att dimensionerna stämmer.

Exempel Skriver vi

```
>> A=zeros(2,2); B=ones(2,3); C=2*ones(1,5);  
>> D=[ A , B ; C ];
```

så blir resultatet en 3×5 matris.

Exempel Bilda en 7×7 matris som består av en 5×5 enhetsmatris i mitten och ettor på kanten.

Funktionen `diag` skapar en diagonal matris med givna element som sparats i en vektor.

I Matlab

```
>> d = [ 1 3 -2]';
```

```
>> D = diag(d)
```

```
D =
```

```
1      0      0
0      3      0
0      0     -2
```

Exempel Bilda en 4×4 matris med följande utseende

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & -2 & 0 & 2 \\ 0 & 0 & -5 & 2 \\ 1 & 1 & 1 & 2 \end{pmatrix}.$$

Praktiska funktioner

Funktionen `size` talar om hur stor en matris är. För vektorer finns `length`.

Exempel Kontrollera att resultatet efter en matris produkt för rätt storlek.

```
>> A = rand(5,3);  
>> B = ones(4,5);  
>> C = B*A;  
>> [n,m]=size(C)
```

ger $n = 4$ och $m = 3$.

Matriselement och Index

För att manipulera enskilda element i en matris anger man dess *index*.

Exempel Antag att vi har en 2×3 matris A . Elementet a_{23} skrivs ut till skärmen genom:

```
>> A = [ 1 2 3 ; 2 3 4 ];  
>> disp( A(2,3) )
```

Man kan ändra elementet a_{12} genom att skriva

```
>> A(1,2)=5;
```

För att skriva ut en hel rad används `:` notation

```
>> disp( A(2, :) )
```

Hela rader eller kolumner kan även ingå i tilldelningssatser. Skriv

```
>> [n,m]=size(A);  
>> A(2, :) = ones(1,m);
```

så byts hela raden ut mot ettor. Dimensionerna måste stämma.

Exempel Addera första raden i en matris till den tredje raden.

Exempel Vi kan titta på flera index samtidigt i en vektor. I Matlab

```
>> x = [ 1 2 3 4 5 6 ];  
>> y = x( [1 5 2] )  
y =  
    1     5     2
```

Vi får alltså en vektor $y = (x_1, x_5, x_2)$. Detta är ofta användbart.

Exempel Låt A vara en matris. Vi vill byta plats på raderna med index 2 och 3.

I Matlab

```
>> A = [ 1 3 -2 ; 0 2 3 ; 4 -1 5]
```

```
A =
```

```
1      3      -2
```

```
0      2      3
```

```
4     -1      5
```

```
>> A([2 3], :) = A([3 2], :)
```

```
A =
```

```
1      3      -2
```

```
4     -1      5
```

```
0      2      3
```

Utmatning till skärmen

Funktionen `disp` gör utskrifter till skärmen.

Exempel Vi har beräknat ett funktionsvärde $y(x)$ och vill skriva ut värdet. I Matlab

```
>> x = 3; y = exp(-2*x);  
>> disp( y );  
0.0025
```

Vi kan även skriva ut korta texter. Skriv

```
>> disp( 'Vi har beräknat y(x)' );  
Vi har beräknat y(x)
```

Tecknet `'` kallas *sträng parentes* och används för att påbörja respektive avsluta text.

Exempel En text sträng kan ses som en radvektor där varje element är ett tecken. Vi kan sätta sammamn strängar på samma sätt som vanliga vektorer.

I Matlab

```
>> str1='Jag heter ';  
>> str2='Fredrik';  
>> str = [ str1 , str2 ]  
str =  
    Jag heter Fredrik  
>> disp( str );  
    Jag heter Fredrik
```

En funktion `num2str` omvandlar siffer värden till text. Då kan vi kombinera text med beräknade värden i utskrifterna.

Exempel Vi har beräknat ett heltal k och vill göra en utskrift av typen

Vi har $k=3$

Detta åstadkomms med Matlab raden

```
>> disp( [ 'Vi har k=', num2str(k) ] )
```

Exempel Vi vill använda Matlab för att skriva ut texten

`sin(2.3)=0.7457`

Hur skall vi göra?