

I dag

- Matriser och vektorer
- Teckensträngar
- Linjära ekvationssystem

Matriser och vektorer

- *Kom ihåg:* En $m \times n$ -matris A är ett rektangulärt schema av tal med m rader och n kolonner.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- a_{ij} betecknar matriselementet i rad i och kolonn j .
- Om $m = n$ kallas matrisen *kvadratisk* av ordning n .
- Om $n = 1$ fås en kolonnmatris/kolonnvektor.
- Om $m = 1$ fås en radmatris/radvektor.

- *Kom ihåg:* MATLAB-syntax:

Matriser omges av hakparenteser [].

Rader separeras med semikolon.

Kolonner separeras med mellanslag eller komma.

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

```
A =
```

```
     1     2     0
    -4     5     2
```

- Specialfall:
- *Exempel.* Radmatrix/radvektor.

```
>> A = [1 4 7 6]
```

```
A =
```

```
    1    4    7    6
```

- *Exempel.* Kolonnmatrix/kolonnvektor.

```
>> A = [1; 4; 7; 6]
```

```
A =
```

```
    1
```

```
    4
```

```
    7
```

```
    6
```

- Man kan också tilldela elementen värden ett och ett.
- Första index är radindex. Andra index är kolonnindex.
- *Exempel.*

```
>> B(1,1) = 3    % Rad 1. Kolonn 1.
```

```
B =
```

```
    3
```

```
>> B(1,2) = 7    % Rad 1. Kolonn 2.
```

```
B =
```

```
    3    7
```

- *Exempel.* (forts.)

```
>> B(2,1) = -5    % Rad 2. Kolonn 1.
```

```
B =
```

```
     3     7
    -5     0
```

```
>> B(2,2) = 6     % Rad 2. Kolonn 2.
```

```
B =
```

```
     3     7
    -5     6
```

Transponat

- Transponering av en matris A innebär att man skiftar rader och kolonner i A .
- Den resulterande matrisen betecknas A^T (“ A transponat”).
- *Exempel.* Om

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \quad \text{och} \quad B = (1 \ 2 \ 3)$$

så är

$$A^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \quad \text{och} \quad B^T = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

- *Syntax.* För att transponera en matris A i MATLAB skriver man A' .
- *Exempel.*

```
>> A = [1 2; 3 4; 5 6]
```

```
A =
```

```
    1    2  
    3    4  
    5    6
```

```
>> A'
```

```
ans =
```

```
    1    3    5  
    2    4    6
```

Noll- och ettmatriser

- Man kan skapa matriser där samtliga element är 0 med kommandot `zeros`.
- Man kan skapa matriser där samtliga element är 1 med kommandot `ones`.
- *Syntax.*
 - `zeros(n)` Ger en $n \times n$ -matris med bara nollor.
 - `zeros(m,n)` Ger en $m \times n$ -matris med bara nollor.
 - `ones(n)` Ger en $n \times n$ -matris med bara ettor.
 - `ones(m,n)` Ger en $m \times n$ -matris med bara ettor.

- *Exempel.*

```
>> A = zeros(2)
```

```
A =
```

```
    0    0  
    0    0
```

- *Exempel.*

```
>> B = ones(2,3)
```

```
B =
```

```
    1    1    1  
    1    1    1
```

- Man kan skapa *enhetsmatriser* med kommandot `eye`.
- En enhetsmatris är en kvadratisk ($n \times n$ -)matris med ettor på huvuddiagonalen och nollor för övrigt.
- *Syntax.*
`eye(n)` Ger en $n \times n$ -enhetsmatris.
- *Exempel.*

```
>> E = eye(3)
```

```
E =
```

```
    1    0    0
    0    1    0
    0    0    1
```

Delmatriser

- Precis som med vektorer kan man referera till delar av en matris.

- *Syntax.*

$A(j, k)$ Ger elementet på plats (j, k) i matrisen A .

$A(j1:j2, k1:k2)$ Ger delmatrisen till A bestående av elementen i raderna j_1 t.o.m. j_2 och kolonnerna k_1 t.o.m. k_2 .

$A(j1:j2, k)$ Ger delmatrisen till A bestående av elementen i raderna j_1 t.o.m. j_2 i kolonn k .

$A(j, k1:k2)$ Ger delmatrisen till A bestående av elementen i rad j och kolonnerna k_1 t.o.m. k_2 .

- *Notera.* Första, tredje och fjärde fallet är samtliga specialfall av det andra fallet.

- *Syntax.* (forts.)

$A(:, k)$ Ger delmatrisen till A bestående av elementen i *samtliga* rader i kolonn k , d.v.s. kolonn k av A .

$A(:, [k_1 \dots k_p])$ Ger delmatrisen till A bestående av kolonnerna $k_1, \dots, k_p$ av A .

$A(j, :)$ Ger delmatrisen till A bestående av elementen i *samtliga* kolonner i rad j , d.v.s. rad j av A .

$A([j_1 \dots j_p], :)$ Ger delmatrisen till A bestående av raderna $j_1, \dots, j_p$ av A .

- *Notera.* Det första fallet är ett specialfall av det andra fallet. Det tredje fallet är ett specialfall av det fjärde fallet.

- *Exempel.*

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

```
A =
```

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

```
>> B = A(2,3)           % Rad 2. Kolonn 3.
```

```
B =
```

```
     6
```

```
>> C = A(1:2,2:3)      % Rad 1-2. Kolonn 2-3.
```

```
C =
```

```
     2     3
     5     6
```

- *Exempel.* (forts.)

```
>> A
```

```
A =
```

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

```
>> D = A(2,2:3)    % Rad 2. Kolonn 2-3.
```

```
D =
```

```
    5    6
```

```
>> E = A(2,:)      % Rad 2. Samma som A(2,1:3)
```

```
E =
```

```
    4    5    6
```

- Med den tomma matrisen [] kan man ta bort rader eller kolonner.
- *Exempel.*

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

```
A =
```

1	2	3	4	5
6	7	8	9	10

```
>> A(:,2) = [] % Ta bort kolonn 2.
```

```
A =
```

1	3	4	5
6	8	9	10

```
>> A(:, [1 3]) = [] % Ta bort kolonn 1 och 3.
```

```
A =
```

3	5
8	10

Att sätta samman matriser

- Man kan sätta samman matriser till större matriser.
- *Exempel.*

```
>> A = [1 2], B = [3 4]
```

```
A =
```

```
    1    2
```

```
B =
```

```
    3    4
```

```
>> C = [A B]
```

```
C =
```

```
    1    2    3    4
```

```
>> D = [A; B]
```

```
D =
```

```
    1    2
```

```
    3    4
```

```
>> A = [1 2; 3 4], B = [5 6; 7 8]
```

```
A =
```

```
    1    2
    3    4
```

```
B =
```

```
    5    6
    7    8
```

```
>> C = [A B]
```

```
C =
```

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

```
>> D = [A; B(1,:) ]
```

```
D =
```

```
    1    2
    3    4
    5    6
```

Teckensträngar

- En teckensträng är ett antal tecken lagrade som en *radvektor*.
- Man kan referera till delar av en sträng och sätta samman strängar på samma sätt som med vektorer.
- *Exempel.*

```
>> fornamn = 'John'  
fornamn =  
John
```

```
>> efternamn = 'Doe'  
efternamn =  
Doe
```

```
>> namn = [fornamn efternamn]  
namn =  
JohnDoe
```

- *Exempel.*

```
>> namn  
namn =  
JohnDoe
```

```
>> namn(3) = 'a'  
namn =  
JoanDoe
```

```
>> nyttnamn = [namn(1:2) 'e ' namn(5:7)]  
nyttnamn =  
Joe Doe
```

Linjära ekvationssystem

- Ett linjärt ekvationssystem med m ekvationer och n obekanta ges av

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = y_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = y_2 \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = y_m \end{cases}$$

där x_j är de obekanta och a_{ij} och y_i är kända tal.

- Det finns tre huvudfall:
 1. $m = n$ Lika många ekvationer som obekanta.
 2. $m > n$ Fler ekvationer än obekanta. *Överbestämt* system.
 3. $m < n$ Färre ekvationer än obekanta. *Underbestämt* system.

- Det linjära ekvationssystemets koefficientmatris A är en $m \times n$ -matris.
- Vi kommer för att lösa systemet i MATLAB förutom koefficientmatrisen att behöva skapa högerledsvektorn y , vilken är en $m \times 1$ -matris (kolonnvektor).
- Vektorn med de obekanta x , kommer att fås som en $n \times 1$ -matris (kolonnvektor).

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

- I fallet med lika många ekvationer som obekanta ($m = n$) så är koefficientmatrisen A en *kvadratisk* $n \times n$ -matris.
- Ett sådant ekvationssystem har en *entydig* (d.v.s. precis en) lösning precis då determinanten $\det(A) \neq 0$ (läsperiod 3).
- I fallet med en entydig lösning x kan den beräknas med “backslash-operatörn” i MATLAB:

$$x = A \backslash y$$

vilket gör att MATLAB löser systemet med Gauss-elimination.

Exempel. Lös det linjära ekvationssystemet

$$\begin{cases} 2x_1 - 3x_2 + 5x_3 = 0 \\ 4x_1 + x_2 + 3x_3 = 14 \\ 3x_1 - 4x_2 - 2x_3 = -8 \end{cases}$$

```
>> A = [2 -3 5; 4 1 3; 3 -4 -2]; y = [0; 14; -8];
```

```
>> x = A\y
```

```
x =
```

```
    2.0000
```

```
    3.0000
```

```
    1.0000
```

Slutsats: $x_1 = 2$, $x_2 = 3$, $x_3 = 1$.

Exempel. Ekvationssystem utan entydig lösning:

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1 \\ 3x_1 + x_2 + 2x_3 = -2 \\ 4x_1 + 3x_2 + 5x_3 = 4 \end{cases}$$

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

```
>> det(A)
```

```
ans =
```

```
0
```

```
>> x = A\y
```

```
Warning: Matrix is singular to working precision.
```

```
(Type "warning off MATLAB:singularMatrix" to suppress  
this warning.)
```

```
x =
```

```
Inf
```

```
Inf
```

```
Inf
```

- Om lösningen inte är entydig, saknar antingen systemet lösning *eller* så finns det oändligt många lösningar.
- I detta fall kan man ge kommandot `rref([A, y])` som omvandlar totalmatrisen till en radekvivalent totalmatris på trappstegsform:

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

```
>> rref([A, y])
```

```
ans =
```

```
    1.0000         0    0.2000         0
         0    1.0000    1.4000         0
         0         0         0    1.0000
```

Sista ekvationen säger $0 = 1$ alltså saknas lösning i detta fall.

- Jämför med ekvationssystemet

$$\begin{cases} 2x_1 - 3x_2 + 5x_3 = 0 \\ 4x_1 + x_2 + 3x_3 = 14 \\ 3x_1 - 4x_2 - 2x_3 = -8 \end{cases}$$

som har den entydiga lösningen $x_1 = 2$, $x_2 = 3$, $x_3 = 1$.

```
>> A = [2 -3 5; 4 1 3; 3 -4 -2]; y = [0; 14; -8];
```

```
>> rref([A, y])
```

```
ans =
```

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

Övningar

1. Arbeta själv igenom exemplen i texten.
2. Skapa några egna matriser. Prova att sätta samman/plocka ut delmatriser ur dessa.
3. Lös några linjära ekvationssystem från Lay med hjälp av MATLAB. Använd både `\` (när det går) samt `rref`.