



29 november 1999

Matematik med Matlab

Z1 1999/2000.

Lab 4: Matriser och ekvationssystem

OBSERVERA :

I vissa av följande uppgifter förekommer variablerna $p_1, \dots, p_{10}$. De avser siffrorna i ditt personnummer $p_1 p_2 p_3 p_4 p_5 p_6 - p_7 p_8 p_9 p_{10}$ (välj ett av era personnummer om ni är två som gör labben tillsammans).

UTVÄRDERING:

Det är bra om du noterar på redovisningsblanketten hur lång tid de olika uppgifterna tar och vad du tycker om dem. Synpunkter som kan förbättra utbildningen är alltid välkomna.

0.1 Matrishantering

Detta avsnitt förutsätter att läsaren är bekant med den linjära algebrans mest grundläggande begrepp (t.ex. matrismultiplikation, system av linjära ekvationer och determinant). Hjälp till detta kapitel hittar du i `helpwin` under `matlab/ops`, `matlab/elmat` och `matlab/matfun`.

0.1.1 Grundläggande matrisoperationer

Man matar in matriser radvis med mellanslag eller kommatecken mellan radelementen och med semi-kolon eller tryck på return/enter-tangenten mellan raderna. Så till exempel ger inmatningen

```
>>A=[1 2 3; 4 5 6; 7 8 0]
```

resultatet

A =

1 2 3

4 5 6

7 8 0

Addition, subtraktion och multiplikation av matriser betecknas med +, - resp. *.

Exempel:

```
>>b= A*x
```

b =

5

8

-7

Uppgift 1: Definiera matriserna

$$A = \begin{bmatrix} 1 & -2 & 6 \\ 7 & -3 & 5 \end{bmatrix}, B = \begin{bmatrix} 4 & -3 & -2 \\ 4 & 4 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, D = \begin{bmatrix} 4 & 0 & -2 \end{bmatrix},$$

$$E = \begin{bmatrix} 2 \\ -1 \\ -3 \end{bmatrix} F = \begin{bmatrix} -1 \\ 2 \\ 5 \end{bmatrix}$$

a. Beräkna $A + B$, $-A$, $2A$ och $3A - 5B$.b. Försök beräkna $A + C$.c. Beräkna AC .d. Beräkna AE .e. Försök beräkna AB .f. Jämför DE och ED .g. Beräkna C^2 och C^{10} .

□

0.1.2 Transponering, inversmatris och determinant

Symbolen för *transponering* är $'$. Transponering av en radvektor ger till exempel kolonnvektor:

```
>> x=[ -1 0 2]'
```

ger alltså resultatet

```
x =
```

```
    -1
```

```
     0 Med A enligt ovan får man:
```

```
     2
```

```
>> C=inv(A)
```

```
C =
```

```
   -1.7778    0.8889   -0.1111
```

```
    1.5556   -0.7778    0.2222
```

```
   -0.1111    0.2222   -0.1111
```

Determinanten av en kvadratisk matris A beräknas enkelt med kommandot `det(A)`. Exempel (med samma matris A som ovan):

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

```
ans =
```

```
    27
```

Uppgift 2: Låt A och E vara matriserna ur uppgift 1.

- Beräkna A^t (transponatet av A). Jämför med A .
- Inför en tredje rad i A genom att sätta den lika med E^t . Låt A vara den nya matrisen.
- Beräkna $\det A$.
- Beräkna A^{-1} .
- Jämför $\det A$ och $\det A^t$.
- Jämför $1/\det A$ och $\det A^{-1}$.
- Jämför $(A^t)^{-1}$ och $(A^{-1})^t$.
- Sätt $Z = \begin{bmatrix} 1+p_7 & 1+p_8 & 1+p_9 & 1+p_{10} \end{bmatrix}$. Beräkna $\det Z^t Z$. Förklara resultatet. Jämför med $\det Z Z^t$.

□

0.1.3 Rader, kolonner och enskilda matriselement. Nya matriser av gamla.

Om M är en given matris betecknar $M(r, :)$ den r :te raden i M , $M(:, k)$ är den k :te kolonnen och $M(r, k)$ är elementet på plats (r, k) . M kan, som nämnts tidigare, uppfattas som en lista med elementen uppräknade kolonnvis. Om $\text{size}(M) = [m, n]$ så innehåller listan $m \cdot n$ element. $M(p)$ är element på plats p i denna lista. Alltså är $M(r, k) = M(p)$ där $p = (k-1)m + r$. Om u och v är två vektorer med heltalskomponenter så betecknar $M(u, v)$ den undermatris av M som består av de rader vars index ges av vektorn u och vars kolonner är kolonnerna i M med index givna av vektorn v .

Låt oss exemplifiera:

```
>> M=[11:15;21:25;31:35]
```

```
M =
```

```
    11    12    13    14    15
```

```
    21    22    23    24    25
```

```
    31    32    33    34    35
```

```
>> M(1, :)
```

```
ans =
```

```
    11    12    13    14    15
```

```
>> M(:, 1)
```

```
ans =
```

```
    11
```

```
    21
```

```
    31
```

```
>> M(3,4)
ans =
    34
>> N=M([2 3],[1 3 5])
N =
    21    23    25
    31    33    35
```

Man kan ändra enstaka element eller hela rader och kolonner genom att direkt tilldela dem nya värden. Exempel:

```
>> M(3,4)=134
M =
    11    12    13    14    15
    21    22    23    24    25
    31    32    33   134    35
>> M(:,5)=[1:3]
M =
    11    12    13    14    1
    21    22    23    24    2
    31    32    33   134    3
```

Man kan ändra storlek på en matris med direkta tilldelningskommandon. Matrisen utökas eventuellt med tillägg av extra nollor. Exempel:

```
>> N(4,:)=ones(1,3)
N =
    21    23    25
    31    33    35
     0     0     0
     1     1     1
```

Man kan även bygga matriser med andra matriser som "block". Exempel:

```
>> P=[M;(-1).^(1:5)] N]
P =
    11    12    13    14     1    21    23    25
    21    22    23    24     2    31    33    35
    31    32    33   134     3     0     0     0
   -1     1    -1     1    -1     1     1     1
```

Man kan också göra om formen på en $m \times n$ -matris med `reshape(M, r, k)`, under förutsättning att $r * k = m * n$. `M(:)` gör om M till en kolonnmatris. Ordningen på elementen i listan M ändras inte av dessa kommandon.

Uppgift 3: Kontrollera beräkningarna i ovanstående avsnitt. □

0.1.4 Speciella matriser

Det finns ett antal kommandon som är avsedda för att bygga upp nya matriser. Här följer ett urval:

<code>zeros(n)</code>	$n \times n$ -matris med bara nollor
<code>zeros(n,m)</code>	$n \times m$ -matris med bara nollor
<code>eye(n)</code>	enhetsmatris av typ $n \times n$
<code>eye(n,m)</code>	$n \times m$ -matris med ettor i diagonalen och nollor för övrigt
<code>ones(n)</code>	$n \times n$ -matris med bara ettor
<code>ones(n,m)</code>	$n \times m$ -matris med bara ettor
<code>rand(n)</code>	$n \times n$ -matris vars element är tal mellan 0 och 1, slumpmässigt utvalda.
<code>rand(n,m)</code>	$n \times m$ -matris vars element är tal mellan 0 och 1, slumpmässigt utvalda.

Ytterligare några användbara kommandon för att bygga upp nya matriser är

<code>tril(A)</code>	ger den undre triangulärmatrisen i den givna matrisen A
<code>triu(A)</code>	ger den övre triangulärmatrisen i A
<code>diag(V)</code>	ger, om V är en rad- eller kolonnmatris, en matris med V på diagonalen och 0 på övriga platser.
<code>diag(A)</code>	ger, om A är en matris, en kolonnvektor med diagonalelementen i A som element.

Det finns även ett antal speciella matriser som den nyfikne läsaren kan inspektera, dessa hittar du i `helpwin` under `matlab/elmat Specialized matrices`

Uppgift 4: Stora matriser byggda av mindre block.

- Skriv in följande: `M= ones(3,3); N=zeros(2,3); P=[2,2,2]; Q=eye(4); R=rand(4,2); X=[M, [N;P];Q,R]`
- Av ovanstående exempel bör du få idéer till hur du kan skapa stora matriser av mindre. Skapa en egen 20×30 -matris, EG.
- För att spara och skriva ut matrisen EG kan du göra så här: Starta en dagbok genom att ge kommandot `diary ovn2.result` Skriv sedan EG och `diary off`. Du har nu en fil, *ovn2.result* i aktuellt bibliotek. Denna kan du skicka till printer med `a2ps ovn2.result |lp` i terminalfönster eller med `!a2ps ovn2.result |lp` i matlabfönstret.

□

0.1.5 Ekvationssystem

Om A är en icke-singulär (d.v.s. inverterbar) kvadratisk matris, kan man entydigt lösa systemet $A * X = b$ för varje matris b som har lika många rader som A .

Lösningen ges av $X = \text{inv}(A) * b$. Analogt om b har lika många kolonner som A så ges lösningen till $X * A = b$ av $X = b * \text{inv}(A)$.

För både inverterbara och icke-inverterbara matriser kan lösningen till $A * X = b$ erhållas med hjälp av Gausseliminering i ekvationssystemet tills koefficientmatrisen är triangulär. Ofta väljer man att presentera räkningarna utan att ta med de obekanta. Man bildar då systemets *totalmatris* eller *utökad koefficientmatris* som består av koefficientmatrisen följt av högerledet, ofta separerade med ett lodrätt streck för att förtydliga att det handlar om en totalmatris till ett ekvationssystem. På denna matris gör man sedan radoperationer tills koefficientmatrisen är triangulär, därefter kan man enkelt lösa ut de obekanta ev. på parameterform, eller se att systemet saknar lösning. I Matlab erhålls trianguleringen genom att man bildar systemets totalmatris $A1 = [A \ b]$ och sedan beräknar $T = \text{rref}(A1)$, (*rref* är förkortning av *row reduced echelon form*). Ur denna kan man bestämma X . Även ekvationen $X * A = b$ kan lösas med hjälp av *rref*, men då måste ekvationen först transponeras, $(X * A)' = b'$, som är samma som $A' * X' = b'$.

Ytterligare ett sätt att beräkna en lösning till ekvationssystemet $A * X = b$ ges av $X = A \setminus b$. Om matrisen A är kvadratisk så erhålls en lösning genom Gausselimination. Även om systemet saknar lösning eller har parameterlösning så får man ett svar dock tillsammans med en varning. Om matrisen är inverterbar så erhålls samma lösning som med matrisinversen ovan, (beräkningsrutinen är dock en annan så det är inte säkert att man får *exakt* samma numeriska resultat).

Om A inte är kvadratisk så är $X = A \setminus b$ en lösning till $A * X = b$ med minstakvadratmetoden. Det innebär att även om systemet saknar lösning så får man alltså ett svar.

Analogt ger kommandot $Y = C / A$ lösningen $Y = C * \text{inv}(A)$ till systemet $Y * A = C$ för varje matris C som har lika många kolonner som A . Kommentarererna ovan gäller även denna operation.

Exempel, med $A =$

```
1 2 3
4 5 6
7 8 0
```

och $b =$

```
5
8
-7
```

ger både $X = \text{inv}(A) * b$ och $X = A \setminus b$

$X =$

```
-1
0
2
```

Vidare ger `rref([A b])` matrisen

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

ur vilken vi direkt kan utläsa samma lösning som ovan.

Låter vi $A =$
 $\begin{bmatrix} 1 & 2 & 3 \\ -2 & 4 & 1 \\ 3 & -2 & 2 \end{bmatrix}$

och $b =$
 $\begin{bmatrix} 4 \\ 5 \\ 2 \end{bmatrix}$

så ger $T = \text{rref}([A \ b])$

$T =$
 $\begin{bmatrix} 1.0000 & 0 & 1.2500 & 0 \\ 0 & 1.0000 & 0.8750 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$

Den sista raden ger oss ekvationen $0 = 1$ vilket visar att ekvationssystemet $AX = b$ saknar lösning.

Däremot ger $X = A \backslash b$

$X =$
 $1.0e + 15 \times$

$\begin{bmatrix} 4.2221 \\ 2.9555 \\ -3.3777 \end{bmatrix}$

Här ges emellertid en varning som säger oss att detta kanske är fel.

Om vi ändrar högerledet till $b =$

$\begin{bmatrix} 4 \\ 5 \\ -1 \end{bmatrix}$

så ger $rT = \text{rref}([A \ b])$

$T =$
 $\begin{bmatrix} 1.0000 & 0 & 1.2500 & 0.7500 \\ 0 & 1.0000 & 0.8750 & 1.6250 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Ur denna matris kan vi se att systemet har parameterlösning.

Nu ger $X = A \backslash b$

$X =$
 $\begin{bmatrix} -0.5000 \\ 0.7500 \\ 1.0000 \end{bmatrix}$

Även nu ges en varning som säger oss att detta kanske är fel.

Uppgift 5: I denna uppgift skall du undersöka några olika ekvationssystem och jämföra lösningarna som erhålls med de olika metoderna som beskrivs ovan.

Matriserna C , D , E och F är samma som i uppgift 1, matrisen A är den du fick i uppgift 2b.

- Lös ekvationssystemen $AX=E$ och $YA=D$ med hjälp av $\text{inv}(A)$. Lös dem sedan även med hjälp av rref (transponera först ekvationen $YA=D$).
- Jämför med resultatet som ges av matlabs operationer $A \backslash E$ och D/A .
- Beräkna $\text{inv}(C)$ och $C * \text{inv}(C)$. Beräkna sedan $\det(C)$. Vilken lösning till ekvationssystemet $CX=E$ ger $X = \text{inv}(C) * E$? Vilken lösning till ekvationssystemet $CX=E$ ger $X = C \backslash E$?
- Bilda totalmatrisen $C1$ till ekvationssystemet ovan, dvs $C1 = [C, E]$. Låt Matlab beräkna trappstegsmatrisen till $C1$ med hjälp av kommandot $\text{rref}(C1)$. Vad kan du nu säga om lösningen till ekvationssystemet.
- Lös $CX=F$ med hjälp av $\text{rref}([C, F])$. Vad ger $X = C \backslash F$? Ser du något samband mellan lösningarna?

□