

Linjär algebra V, vt 2012.

LAB 1: MATRISER OCH EKVATIONSSYSTEM

1 Matrishantering

Detta avsnitt förutsätter att läsaren är bekant med den linjära algebrans mest grundläggande begrepp (t.ex. matrismultiplikation, matrisinvers och system av linjära ekvationer).

För ytterligare hjälp, se till exempel Matlabs hjälpsystem, kapitel 9 i Jönssons bok Matlab-beräkningar, eller ytterligare material som finns tillgängligt via hemsidan.

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

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

resultatet

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

En radvektor (radmatris) med konstant differens d mellan elementen kan inmatas enligt mönstret: $R=[a:d:b]$.

Exempel 1:

```
>>R=[3:2:15]
```

ger resultatet

```
R=  
    3    5    7    9   11   13   15
```

Om ingen differens anges, sätts den till 1:

```
>>S=[3:9]
```

ger resultatet

```
S=  
    3    4    5    6    7    8    9
```

□

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

Exempel 2:

Med $x=[-1;0;2]$ och M som ovan,

```
>>b= M*x
```

```
b =  
    5  
    8  
   -7
```

□

Uppgift 1: Definiera egna matriser A och B av typ 2×3 , C av typ 3×3 , D av typ 1×3 samt F och G av typ 3×1 .

- Testa kommandot *size* på några av matriserna. (*size(A)*)
- Beräkna $A + B$, $-A$, $2A$ och $3A - 5B$.
- Försök beräkna $A + C$.
- Beräkna AC och AG .
- Försök beräkna AB .

f. Jämför DG och GD .

g. Beräkna C^2 och C^{10} .

□

1.2 Transponering och inversmatris

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

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

ger alltså resultatet

```
x =
    -1
     0
     2
```

Med M enligt ovan får man dess invers $N = M^{-1}$:

```
>> N=inv(M)
```

```
N =
   -1.7778    0.8889   -0.1111
    1.5556   -0.7778    0.2222
   -0.1111    0.2222   -0.1111
```

Man kan också skriva A^{-1} .

Man kan dividera med matriser från vänster och höger, men måste vara försiktig med hur resultatet skall tolkas. Om A har invers så är M/A samma som $M * inv(A)$ och $A \backslash M$ samma som $inv(A) * M$. Men om A inte har invers så ger t.ex $A \backslash N$ det man kallar lösning till $AX = N$ enligt "minstakvadratmetoden" (se Lay, §6.5).

Uppgift 2: Låt A och G 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 G^T . Låt A vara den nya matrisen.
- Beräkna A^{-1} . Om den skulle sakna invers, ändra något av elementen i A !
- Testa om vi verkligen har fått inversen till A .
- Jämför $(A^T)^{-1}$ och $(A^{-1})^T$.
- Testa M/A och $A \backslash N$, där M och N är lämpligt valda matriser. Jämför med $M * inv(A)$ respektive $inv(A) * N$.

□

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 `size(M) = [m,n]` så innehåller listan `m*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:

Exempel 3:

```
>> 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 4:

```
>> 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 5:

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

□

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

Exempel 6: >> P=[M;2*[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
     2     4     6     8    10     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.

1.4 Speciella matriser

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

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

Ytterligare några användbara kommandon för att bygga upp nya matriser är

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

Det finns flera speciella matriser som den nyfikne läsaren kan inspektera, dessa hittar du med hjälpsystemet om du letar efter Specialized matrices.

Uppgift 3: Stora matriser byggda av mindre block.

Bygg en valfri 6×8 -matris med hjälp av de tidigare matriserna **A–G**, specialmatriser ur detta avsnitt och de flesta av metoderna som ges i föregående avsnitt.

□

2 Ekvationssystem

Betrakta ett linjärt ekvationssystem $\mathbf{A} \cdot \mathbf{X} = \mathbf{b}$, där **A** är en $m \times n$ -matris, **X** en $n \times 1$ -matris (n obekanta) och **b** en $m \times 1$ -matris (högerled, m ekvationer). Systemet kan alltid lösas med hjälp av Gauss-elimination, oftast genom radoperationer på systemets *totalmatris* eller *utökade koefficientmatris*. I Matlab erhålls den radreducerade trappstegsformen genom att man först bildar systemets totalmatris $\mathbf{A1} = [\mathbf{A} \ \mathbf{b}]$ och sedan använder kommandot $\mathbf{T} = \mathbf{rref}(\mathbf{A1})$ (**rref** är förkortning av *row reduced echelon form*).

MATLAB utför så k pivotering: före varje eliminationssteg ser man till genom erforderliga radbyten, att det pivotelement som står på tur har maximalt belopp. Därigenom hålls beräkningsfelen nere.

Om matrisen **A** är kvadratisk och inverterbar, så har systemet entydig lösning, vilken kan erhållas med $\mathbf{X} = \mathbf{A} \backslash \mathbf{b}$. Som redan nämnts ger $\mathbf{X} = \mathbf{A} \backslash \mathbf{b}$ alltid ett unikt svar även i de fall systemet har oändligt många, eller ingen, lösning. Man får emellertid en varning — Warning: Matrix is singular to working precision. Då måste man övergå till **rref**.

Exempel 7:

Med **A** =

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

och **b** =

```
5
8
-7
```

ger både $\mathbf{X} = \mathbf{inv}(\mathbf{A}) \cdot \mathbf{b}$ och $\mathbf{X} = \mathbf{A} \backslash \mathbf{b}$

X =

```
-1
0
2
```

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

```
1  0  0  -1
0  1  0   0
0  0  1   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.

Om vi ändrar högerledet till $b =$

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

så ger $T = \text{rref}([A \ b])$

$T =$

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

Ur denna matris kan vi se att systemet har oändligt många lösningar.

Här ger $X = A \backslash b$

$X =$

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

en lösning bland oändligt många. □

Uppgift 4: I denna uppgift skall du undersöka några olika ekvationssystem och jämföra lösningarna som erhålls med de olika metoderna som beskrivs ovan. Definiera först följande matriser (I står här för enhetsmatrisen):

$$A = \begin{bmatrix} 3 & -3 & 1 & 0 & 4 \\ 6 & 0 & 2 & 5 & 5 \\ -7 & 7 & 0 & 4 & 1 \\ 4 & 2 & 1 & 6 & 1 \\ -3 & 1 & 1 & 4 & 10 \end{bmatrix},$$

$$B = A - I, \quad C = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}, \quad D = B * \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix}.$$

Talen $p_1 - p_5$ och $q_1 - q_5$ är de 5 första siffrorna i ditt respektive din labkompis/grannes personnummer.

I varje deluppgift skall du testa alla tre metoderna: **rref**, användning av **inv(A)** och användning av bakstreck (backslash). Ange också lösningen/lösningarna där sådana finns. Behandla följande ekvationssystem:

a. $AX=C$.

b. $BX=C$.

c. $BX=D$.

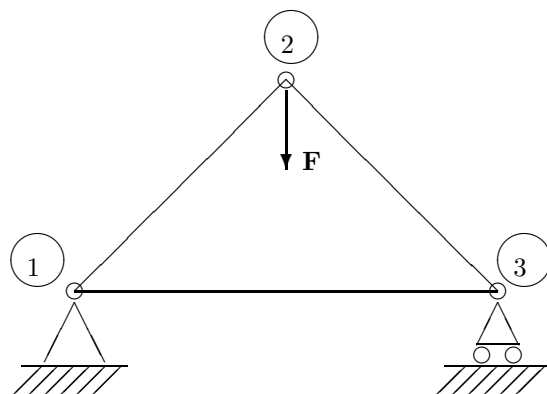
□

2.1 En tillämpning av ekvationssystem

Ett fackverk är en konstruktion där stänger/balkar sätts samman för att ge en lätt men ändå stabil konstruktion. Normalt handlar det om ett tredimensionellt bygge, men för enkelhets skull studerar vi här endast plana fackverk. För att enkelt få stabilitet sätter man samman stängerna så att fackverket består av ett antal trianglar. Punkterna där stängerna sätts samman kallas knutpunkter eller noder.

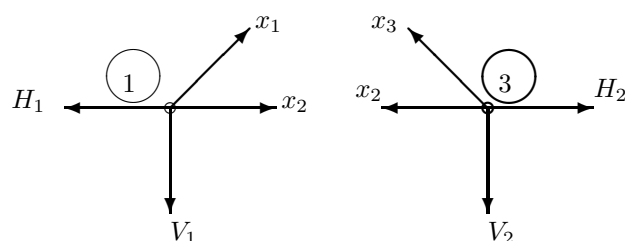
Exempel 8:

Det enklaste fackverket består av en enda triangel. Vi skall nu se hur man kan bestämma krafterna i fackverkets stänger då det belastas av en yttre kraft $\mathbf{F}=8 \text{ kN}$ (samma enhet i fortsättningen).



Alla stångkrafter betraktas som dragkrafter. En tryckkraft ses som en negativ dragkraft.

Vid friläggning av fackverket kommer alla stångkrafter att verka ut från noden i riktning längs stängen, mot den motsatta noden. Stödkrafterna väljer vi här också utåtriktade.



I varje nod måste kraftsumman vara lika med nollvektorn, $\mathbf{0}$.

I nod 1 innebär det att $x_1(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}) + x_2(1, 0) + H_1(-1, 0) + V_1(0, -1) = (0, 0)$

Denna vektorekvation motsvarar två koordinatekvationer:

$$\begin{cases} \frac{1}{\sqrt{2}}x_1 + x_2 - H_1 + 0V_1 = 0 \\ \frac{1}{\sqrt{2}}x_1 + 0x_2 + 0H_1 - V_1 = 0 \end{cases}$$

I nod 2 får vi ekvationerna:

$$\begin{cases} -\frac{1}{\sqrt{2}}x_1 + \frac{1}{\sqrt{2}}x_3 = 0 \\ -\frac{1}{\sqrt{2}}x_1 - \frac{1}{\sqrt{2}}x_3 - 8 = 0 \end{cases}$$

I nod 3 får vi ekvationerna:

$$\begin{cases} -x_2 - \frac{1}{\sqrt{2}}x_3 + H_2 + 0V_2 = 0 \\ 0x_2 + \frac{1}{\sqrt{2}}x_3 + 0H_2 - V_2 = 0 \end{cases}$$

Det rörliga stödet i nod 3 innebär att $H_2 = 0$.

Vi har nu sex ekvationer och sex obekanta x_1, x_2, x_3, H_1, V_1 och V_2 .

$$\begin{cases} \frac{1}{\sqrt{2}}x_1 + x_2 + 0x_3 - H_1 + 0V_1 + 0V_2 = 0 \\ \frac{1}{\sqrt{2}}x_1 + 0x_2 + 0x_3 + 0H_1 - V_1 + 0V_2 = 0 \\ -\frac{1}{\sqrt{2}}x_1 + 0x_2 + \frac{1}{\sqrt{2}}x_3 + 0H_1 + 0V_1 + 0V_2 = 0 \\ -\frac{1}{\sqrt{2}}x_1 + 0x_2 - \frac{1}{\sqrt{2}}x_3 + 0H_1 + 0V_1 + 0V_2 = 8 \\ 0x_1 - x_2 - \frac{1}{\sqrt{2}}x_3 + 0H_1 + 0V_1 + 0V_2 = 0 \\ 0x_1 + 0x_2 + \frac{1}{\sqrt{2}}x_3 + 0H_1 + 0V_1 - V_2 = 0 \end{cases}$$

Detta systems koefficientmatris är

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 1 & 0 & -1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & -1 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -1 \end{bmatrix}$$

Den utökade matrisen är

$$\left[\begin{array}{cccccc|c} \frac{1}{\sqrt{2}} & 1 & 0 & -1 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & -1 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 8 \\ 0 & -1 & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -1 & 0 \end{array} \right]$$

Med `rref` får vi lösningen till $x_1 = -5.66$, $x_2 = 4$, $x_3 = -5.66$, $H_1 = 0$, $V_1 = -4$, $V_2 = -4$

□

I problem av den här typen kommer koefficientmatrisen alltid att ha många nollor. Det blir då lite trist att skriva in alla dessa. Ett sätt att minska skrivearbetet är att börja med att låta alla element i matrisen vara nollor, `A=zeros(6,6)`, och sedan ändra de element som inte är noll, `A(1,1) = 1/sqrt(2)`; `A(1,2)=1`; `A(1,4)=-1`; osv.

Också detta skrivsätt blir trist om matrisen är stor. Då utnyttjar man begreppet *gles*, *sparse* matris.

Med kommandot `A=sparse(R,K,E,6,6)` där `R`, `K`, `E` är radmatriser som innehåller radindex, kolonnindex och element för alla nollskilda positioner i `A` skapas en 6×6 -matris med nollor på alla andra platser. I exemplet ovan har `A` 13 nollskilda element. Radmatriserna `R`, `K`, `E` skall alltså innehålla 13 element som uppfyller att `E(i)= A(R(i), K(i))`. För att förenkla inleder vi med `u=1/sqrt(2)`; så att vi slipper skriva in hela kvoten gång på gång.

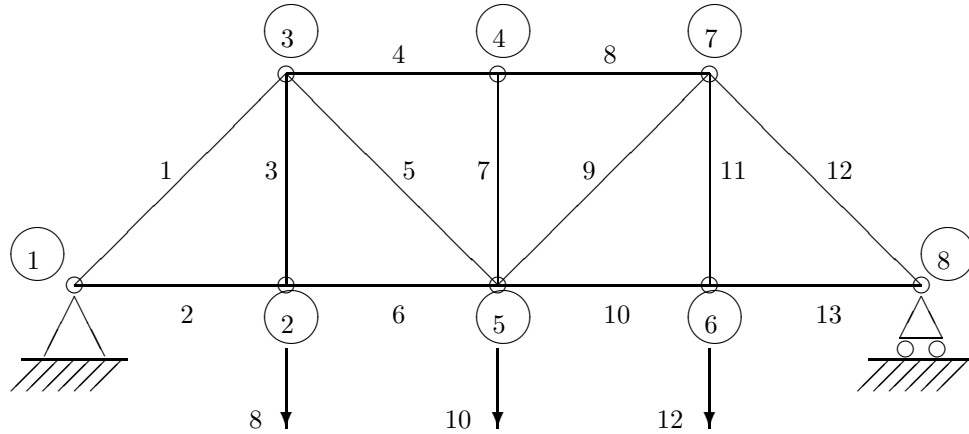
```
R = [ 1 1 1 2 2 3 3 4 4 5 5 6 6]; K = [ 1 2 4 1 5 1 3 1 3 2 3 3 6];
E =[u 1 -1 u -1 -u u -u -u -1 -u u -1];
```

Man får emellertid inte se matrisen `A` om man definierar den med `A=sparse(R,K,E,6,6)` och utelämnar semikolon. Man ser bara en lista över positioner och nollskilda element. Med kommandot `full(A)` får man se matrisen på vanligt sätt, med `spy(A)` får man en grafisk bild av var de nollskilda elementen finns, det kan ibland räcka som kontroll att man skrivit rätt. Med `spy` får man något helt annat, som kanske också kan vara värt att titta på.

Med träning, systematik och eftertanke kan man ange matriserna `R`, `K`, `E` utan att skriva upp hela ekvations-systemet. Varje rad i `A` svarar ju mot en nod, antingen de horisontella eller de vertikala krafterna i noden. Varje kolonn svarar mot en bestämd stångkraft eller stödkraft.

Uppgift 5:

Bestäm stång- och stödkrafter i nedanstående plana fackverk. De horisontella och vertikala stängerna är alla lika långa.



□