TMV166 Linjär algebra för M

Datorlaboration 2: Matrisalgebra och en mekanisk tillämpning

Allmänt

Den här laborationen är mer eller mindre en direkt fortsättning av Laboration 1. Nu utför vi operationer på de matriser vi lärt oss konstruera, och jämför olika sätt att behandla linjära ekvationssystem. Vi går även igenom en vanlig, enkel tillämpning i detalj. Det rekommenderas starkt att läsa den teoretiska beskrivningen av denna innan labtillfället.

Mål

- Addera, multiplicera och invertera matriser
- Tillämpning: bestämma krafter i ett fackverk

Matrisalgebra

Precis som för vektorer är addition och subtraktion av matriser definierad elementvis. I MATLAB utförs dessa operationer med `+` och `-` som vanligt. Till skillnad från vektorer kan vi också multiplicera matriser ifall deras dimensioner passar ihop. Om $A = (a_{ij}) \in \mathbb{R}^{n\times m}$ och $B = (b_{ij}) \in \mathbb{R}^{m\times r}$ så är $AB \in \mathbb{R}^{n\times r}$ och dess element ges av

$$(AB)_{i,j} = \sum_{k=1}^{m} a_{ik} b_{kj}.$$

Detta är precis "rad-kolonn-regeln" och utförs med `*`. Multiplikation med skalär använder också `*`.

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.

a) Kontrollera att dimensionerna är korrekta med kommandot `size`.

b) (Försök) beräkna $A+B$, $-A$, $2A$, $3A-5B$ och $A+C$. Förklara eventuella felmeddelanden.

c) (Försök) beräkna AC, AG och AB. Förklara eventuella felmeddelanden.

d) Jämför DG och GD.

e) Beräkna C^2 och C^{10}.

□

Ursprunglig datorlaboration författad av Carl-Henrik Fant och Lennart Falk, uppdaterad av Tony Stillfjord

Fler matrisoperationer

Transponatet A^T av en matris A skrivs i MATLAB som `A'`. Exempel:

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

>> A'
ans =
     1     4
     2     5
     3     6
```

Detta är särskilt användbart för att se till att alla variabler som är tänkta att representera vektorer tolkas som antingen kolonn- eller rad-vektorer. Börjar man blanda uppstår fort problem. Man kan också snabbt definiera en kolonn-vektor genom t.ex. `x = [1 2 3 4]'` och därmed slippa alla semikolon.

Inversen A^{-1} till en matris A beräknas med hjälp av kommandot `inv` (att föredra) eller helt enkelt `A^-1`. Exempel:

```
>> A = [1, 2; 3, 4];
>> inv(A)
ans =
   -2.0000    1.0000
    1.5000   -0.5000
```

Om matrisen skulle sakna invers får man en varning, men ändå ett svar (som inte betyder något):

```
>> A = [1, 0; 0, 0];
>> inv(A)
Warning: Matrix is singular to working precision.

ans =
   Inf   Inf
   Inf   Inf
```

Uppgift 2. Låt A och G vara matriserna från Uppgift 1.

a) Beräkna A^T och jämför med A.

b) Inför en tredje rad i A genom att sätta den lika med G^T. Låt A vara den nya matrisen.

c) Beräkna A^{-1}. (Om den skulle sakna invers, ändra något av elementen i A.)

d) Testa via definitionen att det som beräknades verkligen är inversen till A.

e) Verifiera räknelagen $(A^T)^{-1} = (A^{-1})^T$ för denna specifika matris A

$\square$

Ekvationssystem igen

Man kan använda matrisinversen för att lösa vissa linjära ekvationssystem. Om $A \in \mathbb{R}^{n \times n}$ (kvadratisk matris) och $Ax = b$ har precis en lösning så ges denna av $x = A^{-1}b$. Finns det flera lösningar eller inga lösningar alls så är inte A^{-1} definierad. Om man inte skall lösa många system med samma matris A men med olika högerled b så är det dock alltid bättre att radreducera (Gausseliminera) istället. Detta kräver mindre beräkningar och introducerar mindre fel (från avrundningar, störningar i högerledet, etc.). Även i fallet med många högerled bör man ofta t.ex. LU-faktorisera A istället.

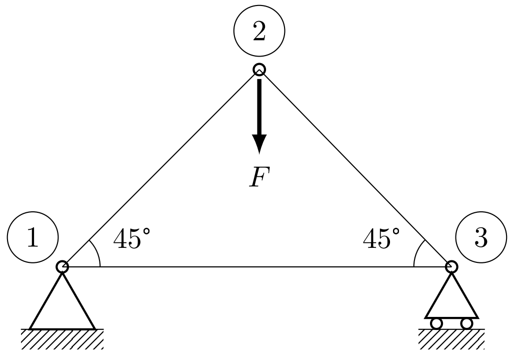
Att lösa $Ax = b$ i MATLAB via radreducering kan göras i flera steg via `rref` som vi såg i Laboration 1, men även direkt via `x = A\b`. Om A inte är kvadratisk så får man istället en så kallad minsta-kvadrat-lösning, vilket vi skall diskutera mer i slutet av kursen. Om systemet inte är lösbart är en sådan "lösning" i en viss mening det bästa man kan åstadkomma. Värt att nämna är även syntaxen `A/B` vilket motsvarar BA^{-1}, dvs. multiplikation med inversen från andra hållet. Liksom för `\` gäller detta bara då A^{-1} existerar.

Uppgift 3. Upprepa Uppgift 2 från Laboration 1, men testa istället `\` och `inv`. Jämför resultaten med det du fick från `rref`. □

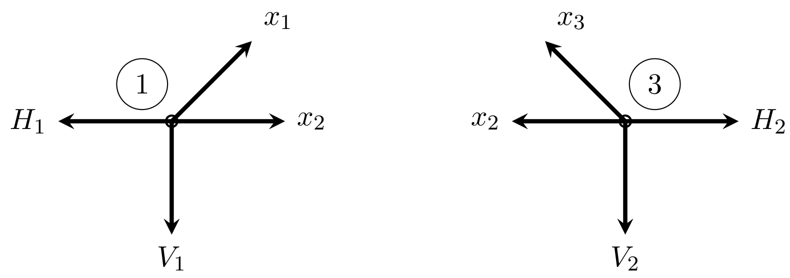
En tillämpning

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.

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 F=8 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 stången, 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 $x_1\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) + x_2(1,0) + H_1(-1,0) + V_1(0,-1) = (0,0)$.

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 totalt 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

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

och totalmatrisen blir

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

Med **`rref`** får vi att lösningen blir $x_1 = -5.66$, $x_2 = 4$, $x_3 = -5.66$, $H_1 = 0$, $V_1 = -4$ och $V_2 = -4$.

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 skrivarbetet 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;` och så vidare.

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. Sedan tilldelar vi följande:

```
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 och med `spy(A)` får man en grafisk bild av var de nollskilda elementen finns. Detta kan ibland räcka som kontroll att man skrivit rätt. Med träning, systematik och eftertanke kan man ange matriserna `R, K, E` utan att skriva upp hela ekvationssystemet. 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 4. Bestäm stång- och stödkrafter i nedanstående plana fackverk. De horisontella och vertikala stängerna är alla lika långa. □