Linjär algebra

Analys och Linjär Algebra, del B, K1/Kf1/Bt1

1 Inledning

Vi fortsätter även denna läsperiod att arbete med MATLAB i matematikkurserna. Dessutom kommer vi göra en projektuppgift tillsammans med kemikursen.

För att handledning och redovisning skall fungera effektivt kräver vi att all redovisning görs via sammanhållande skriptfil tillsammans med nödvändiga funktionsfiler. Vi kräver också att ni har en MATLAB desktop layout av det slag vi använde förra läsperioden.

Denna studio-övning börjar med att vi påminner oss om matriser i MATLAB samtidigt som vi börjar se på matriser i matematiken. Sedan ser vi på martismultiplication för att avslutningsvis se på linjära ekvationssystem.

2 Matriser

En matris är som ni vet ett rektangulärt talschema:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \tag{1}$$

Matrisen ovan har m rader och n kolonner, vi säger att den är av typ $m \times n$. Ett matriselement i rad nr i, kolonn nr j tecknas a_{ij}, där i är radindex och j är kolonnindex. I MATLAB skrivs detta `A(i,j)` och `[m,n]=size(A)` ger matrisens typ.

Indexeringen i MATLAB är alltid som i (1), dvs. rad- och kolonnindex börjar alltid på ett och vi kan inte ändra på det.

En matris av typ $m \times 1$ kallas kolonnmatris (kolonnvektor) och en matris av typ $1 \times n$ kallas radmatris (radvektor):

$$\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix} \tag{2}$$

Du kommer att se att vi använder oftast kolonnvektorer för att representera kvantiteter som vi beräknar. Element nr i ges i MATLAB av `b(i)` och antalet element ges av `m=length(b)`. Även för vektorer gäller att indexeringen alltid börjar på ett. Motsvarande gäller för radvektorn $\mathbf{c}$.

Som exempel tar vi

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 0 & 2 & 4 \end{bmatrix}$$

Vi skriver in detta i MATLAB enligt

```
>> A=[1 4 7 10; 2 5 8 11; 3 6 9 12]
>> b=[1; 3; 5]
>> c=[0 2 4]
```

och ser på typerna och några element med

```
>> [m,n]=size(A)
m =
     3
n =
     4

>> A(2,3)
ans =
     8
```

Prova gärna `length` och `size` på `b` och `c`. Någon skillnad? Skriv ut något element också.

En matris kan betraktas som en kollektion av kolonner:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mj} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_j & \cdots & \mathbf{a}_n \end{bmatrix}$$

med kolonnerna

$$\mathbf{a}_1 = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix}, \quad \mathbf{a}_j = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{bmatrix}, \quad \mathbf{a}_n = \begin{bmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

Man kan även betrakta den som en kollektion av rader, men vi använder oftast kolonnrepresentationen. I MATLAB plockar man ut kolonn nr j med `A(:,j)`. Här är j kolonnindex medan radindex $i = 1, \ldots, m$ representeras av tecknet kolon `:`. På liknande vis ges rad nr i av `A(i,:)`.

```
>> a1=A(:,1)
a1 =
     1
     2
     3

>> A2=A(2,:)
A2 =
     2     5     8    11
```

Det är läge att repetera Moore avsnitt 4.1 nu!

Uppgift 1. Skriv in följande matriser i MATLAB.

$$\mathbf{A} = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 4 & 5 & 6 \\ 3 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

(a). Skriv ut matriselementen a_{23}, b_{23}, x_2. Prova `size` och `length`. Ändra b_{23} genom att skriva `B(2,3)=5`.

(b). Skriv ut kolonn nr 1, 2 och 3 ur matrisen $\mathbf{A}$. Sätt in kolonnvektorn $\mathbf{x}$ som första kolonn i $\mathbf{B}$ genom att skriva `B(:,1)=x`.

(c). Radera matrisen $\mathbf{B}$ (`clear B`) och skriv in den igen genom att först bilda kolonnerna

$$\mathbf{b}_1 = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 6 \\ 1 \\ 1 \end{bmatrix}$$

och sedan sätta in dem i matrisen $\mathbf{B} = [\mathbf{b}_1 \; \mathbf{b}_2 \; \mathbf{b}_3]$.

3 Matris-vektor-multiplikation

Vi definierar produkten av en radmatris och en kolonnmatris (med samma antal element) enligt:

$$\mathbf{ax} = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = a_1x_1 + a_2x_2 + \cdots + a_nx_n.$$

Observera att detta är samma formel som för skalärprodukt. Produkten $\mathbf{y} = \mathbf{Ax}$ av en matris av typen $m \times n$ och en kolonnvektor av typen $n \times 1$ definieras på liknande vis genom att vi multiplicerar matrisens rader i tur och ordning med kolonnvektorn. Vi får en kolonnvektor av typen $m \times 1$ och den ges av

$$\mathbf{y} = \mathbf{Ax} \tag{3}$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}, \tag{4}$$

$$y_i = \sum_{j=1}^{n} a_{ij}x_j = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n.$$

Observera att typerna måste stämma överens för att produkten ska vara definierad enligt regeln $m \times 1 = (m \times n)(n \times 1)$.

Ett alternativt sätt att introducera matris-vektor-multiplikation är att definiera $\mathbf{Ax}$ som en linjärkombination av kolonnerna i $\mathbf{A}$, (se Lay avsnitt 1.4)

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{a_1} & \cdots & \mathbf{a_n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{a_1} + x_2\mathbf{a_2} + \cdots + x_n\mathbf{a_n}.$$

Detta leder förstås till samma uttryck som i (4),

$$x_1\mathbf{a_1}+x_2\mathbf{a_2}+\cdots+x_n\mathbf{a_n} = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ \vdots \\ a_{m2} \end{bmatrix} x_2 + \cdots + \begin{bmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix} x_n = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}.$$

I MATLAB skriver man helt enkelt `y=A*x`.

Uppgift 2. Skriv in följande matriser i MATLAB.

$$\mathbf{A} = \begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 4 & 5 & 6 \\ 3 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

Beräkna följande produkter, först för hand sedan med MATLAB.

$$\mathbf{Ax}, \quad \mathbf{Bx}, \quad \mathbf{ax}, \quad \mathbf{Aa}.$$

4 Linjärt ekvationssystem

Matriser används bland annat för att skriva ned linjära ekvationssystem. Exempel: ekvationssystemet

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 14 \\ 3x_1 + 2x_2 + x_3 = 10 \\ 7x_1 + 8x_2 = 23 \end{cases}$$

kan skrivas på matrisform

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 10 \\ 23 \end{bmatrix},$$

dvs.

$$\mathbf{Ax} = \mathbf{b}, \quad \text{med } \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 7 & 8 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 14 \\ 10 \\ 23 \end{bmatrix}.$$

Vi ska lära oss hur man löser sådana ekvationssystem. I MATLAB finns backslash-kommandot (\) eller alternativt kommandot `rref` (row-reduced-echelon form) som löser systemet, $\mathbf{Ax} = \mathbf{b}$:

```
>> x=A\b
>> rref([A b])
```

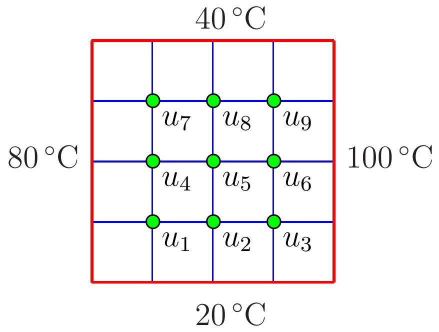
I det första fallet fungerar det bra om lösningen är entydig men sämre om det finns fria variabler eller inga lösningar alls. I det andra fallet reducerar MATLAB den utökade matrisen $[\mathbf{A}\ \mathbf{b}]$ till reducerad trappstegsform.

Uppgift 3. Skriv följande ekvationssystem på matrisform (utökad matris!) och lös dom sedan med \ respektive `rref`.

$$\begin{cases} x_1 + 5x_2 + 9x_3 = 29 \\ 2x_1 + 5x_3 = 26 \\ 3x_1 + 7x_2 + 11x_3 = 39 \end{cases} \qquad \begin{cases} x_1 + x_2 + 3x_3 + 4x_4 = 2 \\ -2x_1 + 2x_2 + 2x_3 = -4 \\ x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ x_1 - x_2 - 2x_3 - x_4 = 1 \end{cases}$$

Uppgift 4. Läs "Balancing Chemical Equations" i Lay sid 51. Lös sedan stökiometriuppgiften Lay 1.6: 11 (sid 54). Be MATLAB räkna med rationella tal med kommandot `format rat` så blir det något enklare att tolka svaret. Med `format short` får vi sedan tillbaka standard formatet.

Uppgift 5. Vi skall beräkna temperaturen på en stålplatta där plattans kanter hålls vid temperaturer enligt figuren. Detta är en fortsättning på uppgiften Lay 1.1: 33 (sid 11).



Antag att temperaturen i en nodpunkt är medelvärdet av temperaturena i de närmsta nodpunkterna i *väster*, *öster*, *söder* och *norr*. Låt $u_1, u_2, \cdots, u_9$ beteckna temperaturerna i de olika nodpunkterna. Sätt upp de ekvationer som ger temperaturen i de olika nodpunkterna. Skriv det linjära ekvationssystemet på matrisform $\mathbf{Au} = \mathbf{b}$ och lös detta i MATLAB.

5 Speciella matriser

MATLAB har funktionerna `zeros, ones, eye` för att bilda speciella matriser med nollor, ettor och enhetsmatris.

Med `zeros(2,5)` får vi en matris av typen 2×5 fylld med nollor och med `ones(2,5)` får vi en matris av samma typ, men fylld med ettor. För att få en enhetsmatris av typen 5×5 ger vi `eye(5,5)`. Med `ones(size(A))` får vi en matris fylld med ettor av samma typ som `A`. Motsvarande gäller för `zeros` och `eye`. Se Moore 10.3.

Det finns funktioner `rand` och `randn` för att bilda matriser fyllda med slumptal. Se Moore 3.6.

6 Redovisning

Denna vecka skall uppgifterna 1-5 redovisas för handledaren.

7 Inför nästa veckas studio-övning

Inför nästa veckas studio-övning är det viktigt att man i förväg läser igenom texten för studio-övningen. Vi skall då se på integraler, vilket är utgångspunkt för differentialekvationer (det som kemiprojektet bygger på).