Mariana Dalarsson, ME1 & Johan Svenonius, IT1

mdtoppen@yahoo.se / jonius@gmail.com

Algebra och geometri 5B1146 - Matlablaboration

2006-12-03

Kurs: 5B1146
Handledare: Karim Daho

Uppgift 1

Enligt uppgiften gäller följande ekvationer:

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \tag{1.1}$$

$$f(x) = x^7 \tag{1.2}$$

$$p'(x) = a_1 + 2a_2 x + 3a_3 x^2 \tag{1.3}$$

$$f'(x) = 7x^6 \tag{1.4}$$

Målet är att bestämma a_0, a_1, a_2 och a_3 så att följande villkor är uppfyllda:

$$p(1) = f(1), \tag{1.5}$$

$$p(-1) = f(-1), \tag{1.6}$$

$$p'(1) = f'(1) \tag{1.7}$$

$$p'(-1) = f'(-1). \tag{1.8}$$

Dessa villkor ger efter insättning följande linjära ekvationer:

$$p(1) = f(1) \Rightarrow a_0 + a_1 + a_2 + a_3 = 1 \tag{1.9}$$

$$p(-1) = f(-1) \Rightarrow a_0 - a_1 + a_2 - a_3 = -1 \tag{1.10}$$

$$p'(1) = f'(1) \Rightarrow a_1 + 2a_2 + 3a_3 = 7 \tag{1.11}$$

$$p'(-1) = f'(-1) \Rightarrow a_1 - 2a_2 + 3a_3 = 7 \tag{1.12}$$

eller

$$\begin{cases} a_0 + a_1 + a_2 + a_3 = 1 \\ a_0 - a_1 + a_2 - a_3 = -1 \\ a_1 + 2a_2 + 3a_3 = 7 \\ a_1 - 2a_2 + 3a_3 = 7 \end{cases} \tag{1.13}$$

Detta ekvationssystem kan skrivas på följande matrisform:

$$A\,x = b \tag{1.14}$$

där

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & -2 & 3 \end{pmatrix} \quad , \quad x = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \text{och} \quad b = \begin{pmatrix} 1 \\ -1 \\ 7 \\ 7 \end{pmatrix} \tag{1.15}$$

Det är således värdena i vektorn *x* vi ska bestämma. Vi matar därför in data för *A* och *b* enligt följande i Matlab:

```
A = [1 1 1 1; 1 -1 1 -1; 0 1 2 3; 0 1 -2 3]        (1.16)
b = [1 -1 7 7]'                                      (1.17)
```

Tilldelningen görs med hjälp av likhetstecken och värdena inom klamrar. Värdena är separerade med blanktecken horisontellt och semikolon vertikalt. Vid tilldelningen av *b* används apostrof för att transponera. På så vis sparades några knapptryckningar relativt att mata in värdena separerade med semikolon.

Vi kan nu mycket enkelt bestämma värdena i vektor *x*, med hjälp av vänsterdivision. Detta eftersom:

$$Ax = b \Rightarrow A^{-1} \cdot A \cdot x = A^{-1}b \tag{1.18}$$

Eftersom $A^{-1} \cdot A = I$, enhetsmatrisen, blir det kvar:

$$x = A^{-1}b \tag{1.19}$$

där A^{-1} är multiplicerad från vänster. Skulle man göra högerdivision istället, skulle följande hända:

$$Ax = b \Rightarrow A^{-1} \cdot x \cdot A = bA^{-1} \tag{1.20}$$

$A^{-1} \cdot x \cdot A$ leder inte till någonting, eftersom det inte kan förkortas till enhetsmatrisen. Rent allmänt är matrismultiplikationen inte en kommutativ operation. Med andra ord AB ≠ BA, och därmed $A^{-1}b \neq bA^{-1}$.

Matlabs funktion för vänsterdivision (lösning av problemet Ax = b, där x söks) lyder enligt följande[1]:

```
x = A\b                                              (1.21)
```

För detta returnerar Matlab:

```
x =

0
-2                                                   (1.22)
0
3
```

[1] http://www.cs.ubc.ca/spider/cavers/MatlabGuide/node7.html

Vilket innebär att matris x är som följer:

$$\begin{pmatrix} 0 \\ -2 \\ 0 \\ 3 \end{pmatrix} \tag{1.23}$$

Och de koefficienter vi sökte blir:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ 0 \\ 3 \end{pmatrix}$$

<u>SVAR</u>: Polynomet vi sökte är $p(x) = -2x + 3x^3$.

Uppgift 1 - Grafer

Vi ska göra två plottningar, en med grafer för f(x) och p(x) och en med f(x)-p(x). För båda gäller intervallet $-1 \leq x \leq 1$.

Först skapar vi en vektor med de värden för *x* som vi ska plotta för:

```
x = [-1:0.001:1]
```
(1.24)

Det som tilldelas ovan är värden mellan -1 och 1 för x, med intervallet 0,001.

Därefter måste funktionerna f(x) och p(x) matas in:

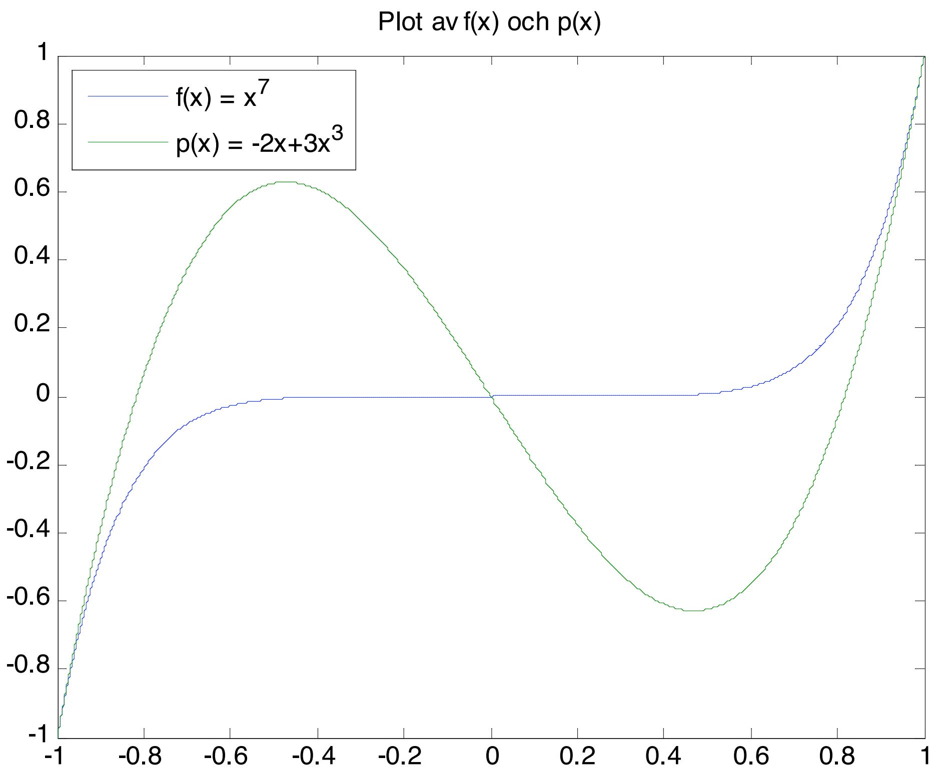
```
f = x.^7                    (1.25)
p = -2*x + 3*x.^3           (1.26)
```

Eftersom upphöjt till ska göras för varje element i vektor *x* måste operatorn ”.” (punkt) användas. Vidare kan det nämnas att Matlab inte tillåter underförstådda multiplikationstecken, vilket betyder att ”*” måste skrivas ut.

Vi utför plottningen för f(x) och p(x), resultat framgår i Figur 1:

```
plot(x,f,x,p), legend ('f(x) = x^7', 'p(x) = -2x+3x^3',
'Location', 'NW'), title('Plot av f(x) och p(x)')
```
(1.27)

Kommandot `plot` tar parametrar på formen (x_1, y_1, x_2, y_2, [.......]) och ritar in graferna för de olika paren av *x* och *y*-värden. Kommandot `legend` skapar en ruta med namn på de olika graferna så man kan hålla isär dem, och `title` sätter en titel.



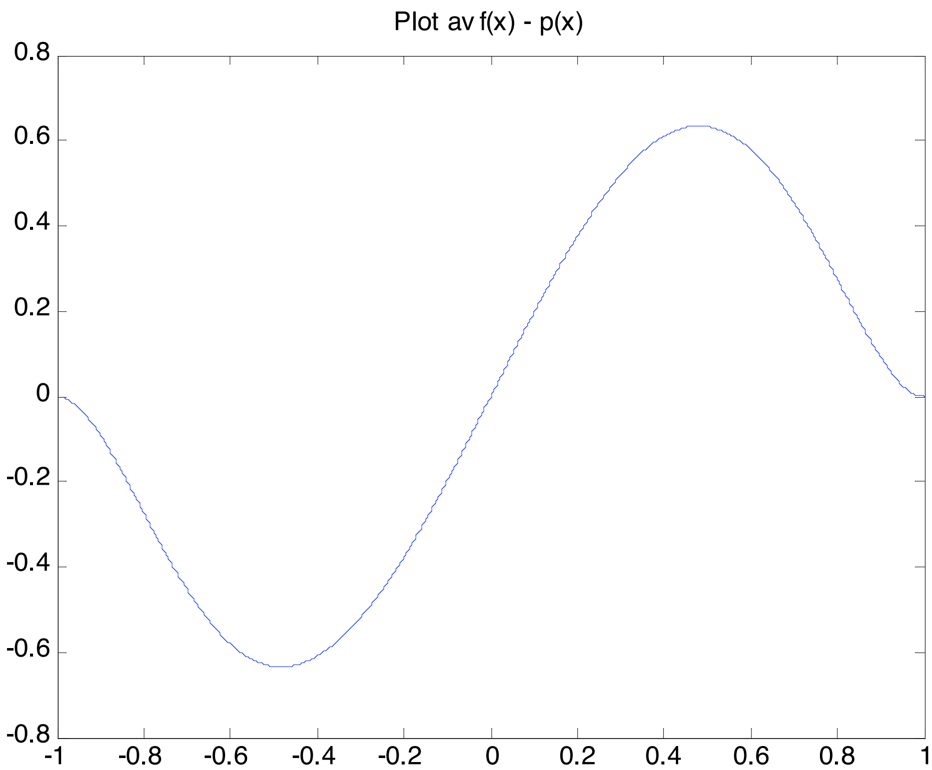
Figur 1

Återstår då att göra plottningen för f(x) – p(x). Vi beräknar först differensen:

```
d = f – p
```
(1.28)

Nu kan vi plotta, resultat se Figur 2:

```
plot (x, d), title('Plot av f(x) – p(x)')
```
(1.29)



Figur 2

Uppgift 2

Enligt uppgiften, har vi en iterativ process i flera steg. Det vi vill producera är:

$$\begin{bmatrix} k_0 \\ b_0 \\ e_o \end{bmatrix} = \begin{bmatrix} 1350 \\ 630 \\ 960 \end{bmatrix} \tag{2.1}$$

där *k* är kol, *b* är bensin, *e* är energi. För att producera så många andelar kol, bensin och energi behöver vi dock dessutom:

$$\begin{bmatrix} k_1 \\ b_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.1 \\ 0.1 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} 1350 \\ 630 \\ 960 \end{bmatrix} = \begin{bmatrix} 96 \\ 390 \\ 357 \end{bmatrix} \tag{2.2}$$

dvs ytterligare 96 enheter kol, 390 enheter bensin och 357 enheter energi. Men för att producera:

$$\begin{bmatrix} k_1 \\ b_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} 96 \\ 390 \\ 357 \end{bmatrix} \tag{2.3}$$

behöver vi:

$$\begin{bmatrix} k_2 \\ b_2 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.1 \\ 0.1 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.1 \end{bmatrix} \begin{bmatrix} 96 \\ 390 \\ 357 \end{bmatrix} = \begin{bmatrix} 35.7 \\ 120 \\ 123.3 \end{bmatrix} \tag{2.4}$$

osv. Denna process fortsätter medan andelarna i $\begin{bmatrix} k_n \\ b_n \\ e_n \end{bmatrix}$ går mot noll.

Den totala mängden energislag som behövs blir därför:

$$\begin{bmatrix} k_0 \\ b_0 \\ e_o \end{bmatrix} + \begin{bmatrix} k_1 \\ b_1 \\ e_1 \end{bmatrix} + \begin{bmatrix} k_2 \\ b_2 \\ e_2 \end{bmatrix} + \begin{bmatrix} k_3 \\ b_3 \\ e_3 \end{bmatrix} + \ldots \tag{2.5}$$

där

$$\begin{bmatrix} k_1 \\ b_1 \\ e_1 \end{bmatrix} = A \cdot \begin{bmatrix} k_0 \\ b_0 \\ e_o \end{bmatrix} \quad , \quad A = \begin{bmatrix} 0 & 0 & 0.1 \\ 0.1 & 0.1 & 0.2 \\ 0.1 & 0.2 & 0.1 \end{bmatrix} \tag{2.6}$$

Från detta kan vi dra slutsatsen att:

$$\begin{bmatrix} k_2 \\ b_2 \\ e_2 \end{bmatrix} = A \cdot A \cdot \begin{bmatrix} k_0 \\ b_0 \\ e_o \end{bmatrix} = A^2 \cdot \begin{bmatrix} k_0 \\ b_0 \\ e_o \end{bmatrix} \tag{2.7}$$

och

$$\begin{bmatrix} k_3 \\ b_3 \\ e_3 \end{bmatrix} = A^3 \cdot \begin{bmatrix} k_0 \\ b_0 \\ e_o \end{bmatrix} \tag{2.8}$$

osv.

Därmed får vi slutsumman:

$$\begin{bmatrix} k \\ b \\ e \end{bmatrix} = \left(1 + A + A^2 + A^3 + A^4 \ldots A^n\right) \cdot \begin{bmatrix} 1350 \\ 630 \\ 960 \end{bmatrix} \qquad n \to \infty \tag{2.9}$$

Vi vet att för vanliga tal vars absolutvärde är mindre än 1 ($|a| < 1$), så gäller:

$$1 + a + a^2 + a^3 + ... = \frac{1}{1-a} = (1-a)^{-1} \tag{2.10}$$

För matriser vars norm (längd) är mindre än 1 ($\|A\| < 1$), gäller den liknande formeln:

$$1 + A + A^2 + A^3 + ... = (I - A)^{-1} \tag{2.11}$$

Därför behöver vi kontrollera att normen av vår matris A är mindre än 1. Vi vet att normen för A är det största av dess egenvärden. Egenvärdena beräknar vi i Matlab enligt följande:

```
A = [0 0 0.1; 0.1 0.1 0.2; 0.1 0.2 0.1]
eigenvalues = eig(A)
```
(2.12)

Där kommandot `eig` tar fram egenvärdena till matrisen. I vårt fall har vi en 3×3-matris, så vi vet att vi ska förvänta oss 3 egenvärden. Matlab returnerar också detta, enligt:

```
eigenvalues =

   -0.0303
    0.3303
   -0.1000
```
(2.13)

Vi ser att det största av egenvärderna, 0.3303, är < 1. Alltså betyder det att vi kan använda formel (2.11) för att lösa denna uppgift. Därmed kan vi skriva om (2.9) till:

$$\begin{bmatrix} k \\ b \\ e \end{bmatrix} = (I - A)^{-1} \cdot \begin{bmatrix} 1350 \\ 630 \\ 960 \end{bmatrix} \tag{2.14}$$

Detta kan man med fördel räkna ut med Matlab. Vi matar i Matlab in 3x3-enhetsmatrisen *I*, matrisen *A*, och vektorn som vi kallar *v*:

```
I = eye(3)
A = [0 0 0.1; 0.1 0.1 0.2; 0.1 0.2 0.1]
v = [1350; 630; 960]
```
(2.15)

Enhetsmatrisen skapades med kortkommandot `eye` som tar parametern *n* som står för antalet önskade rader och kolonner.

Nu är det enkelt att utföra beräkningen:

```
((I – A)^(-1)) * v
```
(2.16)

Vi får:

$$\begin{bmatrix} k \\ b \\ e \end{bmatrix} = \begin{bmatrix} 1500 \\ 1200 \\ 1500 \end{bmatrix}$$

SVAR: Vi behöver 1500 enheter kol, 1200 enheter bensin och 1500 enheter el.

Uppgift 3

Enligt uppgiften har vi att 100 gram deg $\Rightarrow$ 24.1 g fett
55 g kolhydrater
7.5 g protein
500 kcal

Dessutom gäller:

$$\left.\begin{aligned} &\frac{smör}{\deg} = x \\ &\frac{soc\ker}{\deg} = y \\ &\frac{vetemjöl}{\deg} = z \\ &\frac{skummjölk}{\deg} = w \end{aligned}\right\} \quad x + y + z + w = 1 \tag{3.1}$$

Från tabellen given i uppgiften kan vi då ställa upp:

$$80x + 0y + 0z + 1w = 24.1 \tag{3.2}$$

$$0x + 100y + 75z + 50w = 55 \tag{3.3}$$

$$0x + 0y + 10z + 35w = 7.5 \tag{3.4}$$

$$800x + 400y + 350z + 400w = 500 \tag{3.5}$$

för att beskriva vilka andelar från var och en av ingredienserna som ger den totala mängden fett, socker, mjöl, och skummjölspulver.

Detta kan skrivas som matrisen:

$$\begin{bmatrix} 80 & 0 & 0 & 1 \\ 0 & 100 & 75 & 50 \\ 0 & 0 & 10 & 35 \\ 800 & 400 & 350 & 400 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 24.1 \\ 55 \\ 7.5 \\ 500 \end{bmatrix} \tag{3.6}$$

Vi kan lätt beräkna värdena på x, y, z och w med hjälp av Matlab. Vi kallar de olika matriserna från vänster till höger för *A*, *v* och *b*.

Vi matar in matris *A* och vektor *v* i Matlab:

```
A = [80 0 0 1; 0 100 75 50; 0 0 10 35; 800 400 350 400]
b = [24.1; 55; 7.5; 500]
```
(3.7)

Sedan utför vi vänsterdivision och får som resultat en vektor med proportioner enligt (1.21):

```
v = A\b
```

$$v = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.4 \\ 0.1 \end{bmatrix} \tag{3.8}$$

Vi vill ha 500 g deg, varför vi multiplicerar varje element i vektor *v* med 500:

```
v.*500
```
(3.9)

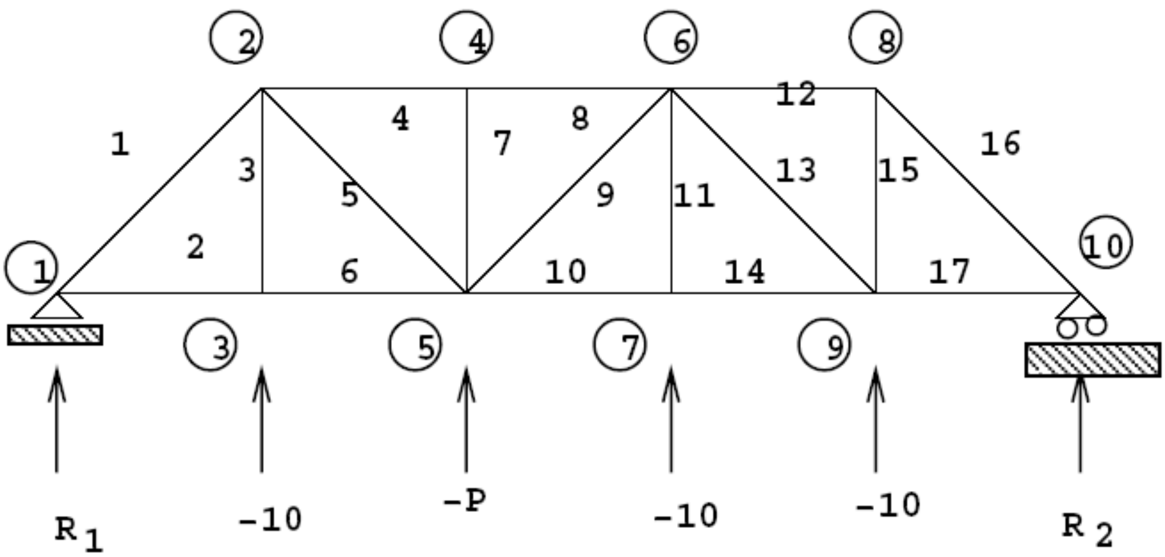
Vilket ger som resultat:

$$\begin{bmatrix} 150 \\ 100 \\ 200 \\ 50 \end{bmatrix} \tag{3.10}$$

<u>SVAR:</u> Vi behöver 150 g socker, 100 g smör, 200 g vetemjöl och 50 g skummjölkspulver för att laga ”mormors smörringar”.

Uppgift 4

Vi har ett plant fackverk av följande utseende:



Figur 3

Lasterna i fackverket i figur består av nedåtriktade krafter P i noderna 3,5,7 och 9. I övriga noder är lasterna noll. Vår uppgift är att variera ett okänt P i vertikalled i nod 5, så att den maximala stångkraften blir 200. I avsnitt 1.7.1 av pdf-filen på hemsidan[2] får vi en ganska detaljerad ledning för hur man löser ett liknande problem.

Uppgift 4 a)

Vi ska avgöra om maxkraften är linjärt beroende av den vertikala kraften P i nod 5 (b(8) i vår matris).

Matriserna *A* och *b* läses in från filen `fack.m`:

```
fack
```
(4.1)

Belastningar i olika noder avgörs med hjälp av vänsterdivision av *A* och *b*. Vi testar att lista belastningarna i ursprungsläget jämte index (för indexet är intervallet 1 underförstått):

```
index=[1:17]';
[index A\b]
```
(4.2)

Vi får en tabell med belastningen i de olika noderna. Enligt tabellen är det högsta absolutbeloppet av belastningen i nod 16; värdet där är 47.1405. Detta ser vi i tabellen, men vi kan även använda följande kommando för att Matlab ska hitta värdet åt oss. (Returvärden anges som kommentarer efter procenttecken).

```
max(abs(A\b))            % 47.1405
```
(4.3)

Först sker vänsterdivision. Kommandot `abs` ger oss en matris med absolutbeloppen av resulterande matris. Därefter hittar `max` det största beloppet.

Vi varierar värdet på b(8), dvs nod 5, och ser vilka resultat vi får:

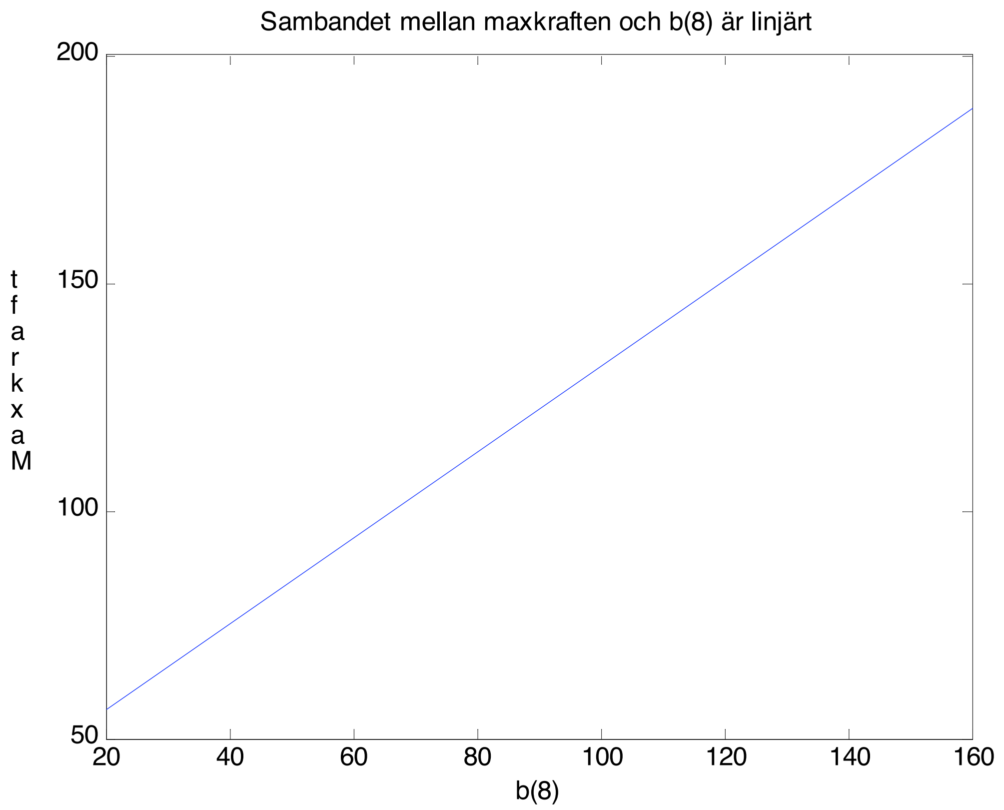
```
b(8) = 20;  max(abs(A\b)) % 56.5685
b(8) = 60;  max(abs(A\b)) % 94.2809
b(8) = 80;  max(abs(A\b)) % 113.1371
b(8) = 120; max(abs(A\b)) % 150.8494
b(8) = 160; max(abs(A\b)) % 188.5618
```
(4.4)

Vi lägger in resultaten i vektorer som vi sedan plottar (se Figur 4):

```
x = [20 60 80 120 160];
y = [56.5685 94.2809 113.1371 150.8494 188.5618];

plot (x, y), ylabel('Maxkraft'), xlabel('b(8)'),
title('Sambandet mellan maxkraften och b(8) är linjärt');
```
(4.5)

[2] http://www.math.kth.se/math/student/courses/5B1146/ME/200607/matlab/5B1146MATLAB.pdf



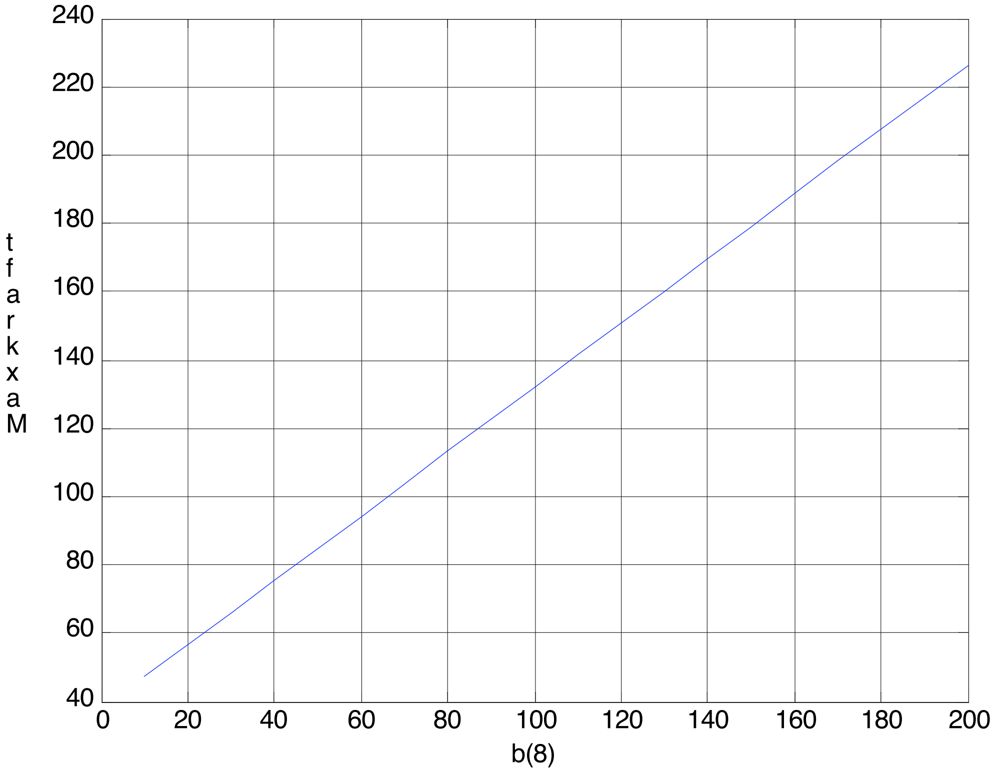
Figur 4

Vi ser att vi har ett linjärt samband. En alternativ lösning är att plotta med hjälp av ett program. Resultatet av detta visas i Figur 5.

```
fack
p=[];
kraft=[];
hl = b;
for P=10:10:200
  hl(8) = P;
  x=A\hl;
  p=[p P];
  maxk=max(abs(x));
  kraft=[kraft maxk];
end
plot(p,kraft)
grid
```

(4.6)

Skillnaden här är att jobbet utförs automatiskt, men principen är helt den samma. Vi varierar värden för högerledet, löser ut x, sätter in värdena för maxkraft och b(8) i vektorer och konstaterar att vi har ett linjärt samband.



Figur 5

Vi kan även konstatera att vi grafiskt kan bestämma värdet för b(8) sådant att maxkraft blir 200 till omkring 170. Vi kan med hjälp av Basic Fitting på Tools-menyn (alternativ linear) bestämma vår linjes ekvation. Vi får att $k = p1 = 0.94281$ och $m = p2 = 37.712$. Eller så använder vi `polyfit`, enligt nedan:

```
polyfit (p, kraft, 1)
```
(4.7)

Inparametrar var (x-värden, y-värden, vilken grad vi anpassar till). Returvärden är samma som de vi fick från Basic Fitting.

Härifrån kan vi enkelt bestämma ett bättre värde för b(8), enligt räta linjens ekvation:

$$y = kx + m \Leftrightarrow x = \frac{(y-m)}{k} = \frac{200-37.712}{0.94281} = 172.13 \qquad (4.8)$$

Uppgift 4 b)

Enligt data från filen fack.m för godtyckligt P, kan vektor **b** skrivas som följer

$$\mathbf{b} = [0 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0 \quad 0 \quad P \quad 0 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0 \quad 0 \quad 10 \quad 0]^T \qquad (4.9)$$

eller

$$\mathbf{b} = [0 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0 \quad 0 \quad (10+P-10) \quad 0 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0 \quad 0 \quad 10 \quad 0]^T \qquad (4.10)$$

eller

$$\mathbf{b} = [0\ \ 0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 0]^T$$

$$+ (P-10)[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T \qquad (4.11)$$

Ekvationen (4.10) kan skrivas kortare som

$$\mathbf{b} = \mathbf{b}_{10} + (P - 10) \cdot \mathbf{e}_8 \qquad (4.12)$$

där

$$\mathbf{b}_{10} = [0\ \ 0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0\ \ 10\ \ 0]^T\ , \qquad (4.13)$$

är den högerledsvektor som erhålls för P = 10, och

$$\mathbf{e}_8 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T\ , \qquad (4.14)$$

är den åttonde enhetsvektorn av dimension 17. Därmed har vi bevisat likheten (4.12).

Uppgift 4 c)

Med hjälp av likheten (4.11) kan vi skriva om matrisekvationen $Ax = b$, enligt följande:

$$A\, \mathbf{x} = \mathbf{b} = \mathbf{b}_{10} + (P - 10) \cdot \mathbf{e}_8 \qquad (4.15)$$

Vi kan nu dela upp vektor x i en summa av två vektorer y och z, d.v.s.

$$\mathbf{x} = \mathbf{y} + \mathbf{z} \qquad (4.16)$$

så att

$$A\, (\mathbf{y} + \mathbf{z}) = \mathbf{b}_{10} + (P - 10) \cdot \mathbf{e}_8 \qquad (4.17)$$

eller

$$A\, \mathbf{y} = \mathbf{b}_{10} \qquad (4.18)$$

$$A\, \mathbf{z} = (P - 10) \cdot \mathbf{e}_8 \qquad (4.19)$$

Då vår matrisekvation är en linjär ekvation, ser vi att lösningen för vektor **y** är just den lösningen vi tidigare fått för P = 10. Koefficientmatrisen A är i båda ekvationerna (4.18) och (4.19) densamma.

Uppgift 4 d)

Med hjälp av ekvationen (4.14) kan vi skriva följande:

$$A\, \mathbf{x} = \mathbf{b}_{10} + (P - 10) \cdot \mathbf{e}_8 \qquad (4.20)$$

eller

$\mathbf{x} = A^{-1}\, \mathbf{b}_{10} + (P - 10) \cdot A^{-1}\, \mathbf{e}_8$ (4.21)

Då den maximala stångkraften finns i stången 16, beaktar vi elementet x_{16} inom vektorn **x** och skriver:

$x_{16} = A^{-1}(16)\, \mathbf{b}_{10} + (P - 10) \cdot A^{-1}(16)\, \mathbf{e}_8$ (4.22)

där $A^{-1}(16)$ är den sextonde raden i matrisen A^{-1}. Då vektorn $\mathbf{e}_8$ har formen

$\mathbf{e}_8$ `= [0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0]`^{T} (4.23)

är det bara det 8:e elementet i radvektorn $A^{-1}(16)$ som ger en produkt som är skild från noll. Det 8:e elementet i radvektorn $A^{-1}(16)$, betecknar vi med $A^{-1}(16,8)$. Därmed får vi från (4.21):

$x_{max} = x_{16} = A^{-1}(16)\, \mathbf{b}_{10} + (P - 10) \cdot A^{-1}(16,8)$ (4.24)

eller

$P – 10 = [\, x_{max} - A^{-1}(16)\, \mathbf{b}_{10}]/ A^{-1}(16,8)$ (4.25)

eller slutligen

$P = [\, x_{max} - A^{-1}(16)\, \mathbf{b}_{10}]/ A^{-1}(16,8) + 10$ (4.26)

För att nu kunna räkna fram P för ett givet värde av x_{max}, behöver vi bara räkna fram matrisen A^{-1} och sedan extrahera radvektorn $A^{-1}(16)$, samt räkna värdet på produkten $A^{-1}(16)\, \mathbf{b}_{10}$, som är ett vanligt tal. Sedan behöver vi extrahera elementet i sextonde raden och åttonde kolumnen av matrisen A^{-1} , d.v.s. $A^{-1}(16,8)$, som också är ett vanligt tal. Detta ger oss parametrarna av den linjära ekvationen:

$P = k\, x_{max} + m$ (4.27)

som vi annars fått fram numeriskt också. Genom analys av ekvationerna (4.26) och (4.27) ser vi att k och m anges av:

$k = 1/ A^{-1}(16,8)$ (4.28)

$m = - A^{-1}(16)\, \mathbf{b}_{10}/ A^{-1}(16,8) + 10$ (4.29)

Värdena för *A* läses in med hjälp av filen `fack.m`, och vi matar in värdena för b_{10} (4.13):

```
fack
b10 = [0  0  0 10  0  0  0  10  0  0  0 10  0  0  0 10  0]'
```
(4.30)

Vi vill att Matlab ska returnera fler värdesiffror än vad den normalt gör (14 istället för 4). Vi utnyttjar kommandot `format` för detta:

```
format long
```
(4.31)

A^{-1} beräknas och resultatet lagras i matrisen *B*.

```
B = A^(-1)
```
(4.32)

Rad 16 i matris *B* läggs in i en ny vektor *C*.

```
C = B(16,:)
```
(4.33)

Värdet för *k* i ekvation (4.28) beräknas genom att ta 1/*C(8)*:

```
k = 1/C(8)                          % ger 1.06066...
```
(4.34)

Värdet för *m* i ekvation (4.29) ekvation hämtas från kolumn 8 i vektor *C*

```
m = -(C*b10)/C(8)+10                % ger -39.9999...
```
(4.35)

Detta ger alltså:

$$k \approx 1.0607 \tag{4.36}$$

$$m \approx 40 \tag{4.37}$$

Nu kan P enkelt beräknas. För $x_{max} = 200$ blir det:

$$P = kx_{max} + m = 1{,}06066 \cdot 200 - 39{,}9999 = 172.132 \tag{4.38}$$

Detta stämmer också överens med det värde vi fick fram numeriskt tidigare.

Alltså har vi visat hur man kan räkna ut P för ett godtyckligt x_{max} både analytiskt och numeriskt, och att sambandet mellan P och x_{max} är linjärt.

Referenser:

1. http://www.math.kth.se/math/student/courses/5B1146/ME/200607/matlab/5B1146MATLAB.pdf ”Vektorer, matriser och linjära ekvationssystem”, KTH, 5B1146 Geometri och Algebra för IT och ME’s hemsida.
2. http://www.cs.ubc.ca/spider/cavers/MatlabGuide/node7.html Matlab Guide - Matrix Operations