Rapportexempel, Datorer och datoranvändning

På de följande sidorna finns en (fingerad) laborationsrapport som jag har skrivit i två olika ordbehandlingssystem:

Microsoft Word utan några egna anpassningar. Jag har bara använt de fördefinierade formatmallarna för löpande text och för rubriker. Formlerna är skrivna med Words ekvationseditor.

LaTeX utan några egna anpassningar. Formlerna är skrivna med matematikkommandona i LaTeX.

Allmänt om dokumentens utseende: jag tycker att Word-varianten är ful och LaTeX-varianten snygg. Man kan naturligtvis anpassa Word så att det blir snyggare (andra typsnitt, andra storlekar, och så vidare), men de flesta som använder Word gör inga egna anpassningar.

Om formlerna: LaTeX ger snyggast formler. Dessutom är ekvationseditorer svåra att använda — det gick betydligt fortare för mig att skriva formlerna i LaTeX än i Word trots att jag tidigare använt Word mycket och det var första gången jag använde LaTeX!

Datorer och datoranvändning – laboration 7

Christina Carlsson (d01cc) och David Danielsson (d01dd).

2001–09–11.

1. Inledning

I denna laboration har vi studerat minsta kvadrat-metoden för anpassning av kurvor till givna mätvärden och med hjälp av matematikprogrammet Matlab anpassat en rät linje till ett antal värden.

2. Minsta kvadrat-metoden – teori

Vi studerar här ett enkelt exempel där det gäller att anpassa en rät linje $f^*(x) = c_0 + c_1 x$ till följande mätvärden:

x	f
21	4.63
22	4.78
23	6.61
24	7.21
25	7.78

Vid kurvanpassning gäller det att lägga den givna kurvan så att den ansluter till de givna punkterna så ”bra” som möjligt. I minsta kvadrat-metoden minimerar man summan av kvadraterna på avvikelserna mellan mätvärdena och kurvan i alla mätpunkterna. Vi ska alltså minimera följande summa med avseende på c_0 och c_1:

$$S = \sum_{k=1}^{5} \left[f^*(x_k) - f(x_k) \right]^2 = \sum_{k=1}^{5} \left[c_0 + c_1 x_k - f(x_k) \right]^2$$

Vi gör som vanligt vid minimering, nämligen deriverar (här med avseende på c_0 och c_1) och sätter derivatorna till noll. Detta ger de båda ekvationerna:

$$2\sum_{k=1}^{5} \left[c_0 + c_1 x - f(x_k) \right] = 0$$

$$2\sum_{k=1}^{5} x_k \left[c_0 + c_1 x - f(x_k) \right] = 0$$

Om vi utvecklar summorna får vi:

$$5c_0 + c_1 \sum_{k=1}^{5} x_k = \sum_{k=1}^{5} f(x_k)$$

$$c_0 \sum_{k=1}^{5} x_k + c_1 \sum_{k=1}^{5} x_k^2 = \sum_{k=1}^{5} x_k f(x_k)$$

Nu är det bara att sätta in de kända värdena i ekvationerna. Vi får ekvationssystemet:

$$5c_0 + 115c_1 = 31.01$$

$$115c_0 + 2655c_1 = 721.96$$

Detta ekvationssystem är lätt att lösa. Vi får följande räta linje som bäst anpassar mätvärdena: $f^*(x) = -13.877 + 0.873x$

3. Beräkningar i Matlab

När vi ska anpassa kurvor med hjälp av Matlab kan vi arbeta enligt ovan och utnyttja de inbyggda funktionerna för att lösa ekvationssystem. Men i Matlab finns också speciella funktioner för kurvanpassning med polynom.

Man börjar med att definiera vektorer för mätpunkterna och mätvärdena. Därefter utnyttjar man funktionen `polyfit` som ger koefficienterna i det polynom av grad n som anpassar värdena. Med värden enligt det tidigare exemplet får vi (här är $n = 1$):

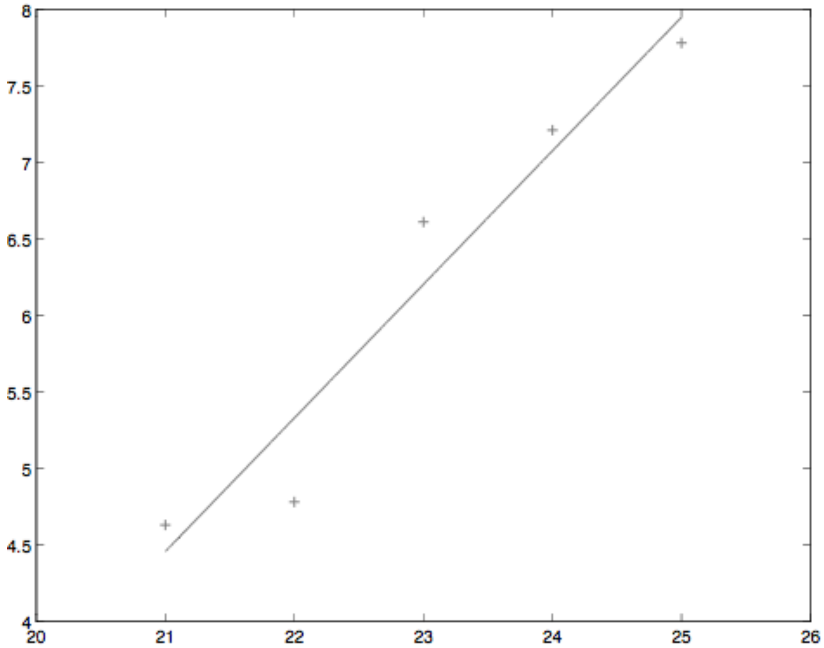
```
>> x = [21 22 23 24 25];
>> f = [4.63 4.78 6.61 7.21 7.78];
>> fstar = polyfit(x,f,1)

ans =
     0.8730 -13.8770
```

Vi kan också plotta resultatet:

```
>> plot(x,f,'+')
>> axis([20 26 4 8])
>> hold on
>> xpoints = x(1) : 0.01 : x(length(x));
>> plot(xpoints,polyval(fstar,xpoints))
```

Vi får en figur med följande utseende:



4. Slutsatser

Vi har sett att Matlab är ett utmärkt verktyg när man med hjälp av datorn ska anpassa en kurva till givna mätvärden. Men detta är inte det enda man kan använda Matlab till. Det finns mycket mera i Matlab, till exempel:

- Ekvationslösning
- Lösning av ekvationssystem
- Integration
- Lösning av differentialekvationer
- Linjär algebra
- …

Om man har ett numeriskt problem bör man alltså inte i första hand skriva ett eget program som löser problemet. I stället bör man börja med att undersöka om Matlab har någon färdig funktion för problemet.

Datorer och datoranvändning – laboration 7

Cecilia Carlsson (d01cc)
och
David Danielsson (d01dd)

2001–09–11

1 Inledning

I denna laboration har vi studerat minsta kvadrat-metoden för anpassning av kurvor till givna mätvärden och med hjälp av matematikprogrammet Matlab anpassat en rät linje till ett antal värden.

2 Minsta kvadrat-metoden – teori

Vi studerar här ett enkelt exempel där det gäller att anpassa en rät linje $f^*(x) = c_0 + c_1 x$ till följande mätvärden:

x	f
21	4.63
22	4.78
23	6.61
24	7.21
25	7.78

Vid kurvanpassning gäller det att lägga den givna kurvan så att den ansluter till de givna punkterna så ”bra” som möjligt. I minsta kvadrat-metoden minimerar man summan av kvadraterna på avvikelserna mellan mätvärdena och kurvan i alla mätpunkterna. Vi ska alltså minimera följande summa med avseende på c_0 och c_1:

$$S = \sum_{k=1}^{5} \left[f^*(x_k) - f(x_k)\right]^2 = \sum_{k=1}^{5} \left[c_0 + c_1 x_k - f(x_k)\right]^2$$

Vi gör som vanligt vid minimering, nämligen deriverar (här med avseende på c_0 och c_1) och sätter derivatorna till noll. Detta ger de båda ekvationerna:

$$\begin{aligned} 2\sum_{k=1}^{5} \left[c_0 + c_1 x_k - f(x_k)\right] &= 0 \\ 2\sum_{k=1}^{5} x_k \left[c_0 + c_1 x_k - f(x_k)\right] &= 0 \end{aligned}$$

Om vi utvecklar summorna får vi:

$$5c_0 + c_1 \sum_{k=1}^{5} x_k = \sum_{k=1}^{5} f(x_k)$$
$$c_0 \sum_{k=1}^{5} x_k + c_1 \sum_{k=1}^{5} x_k^2 = \sum_{k=1}^{5} x_k f(x_k)$$

Nu är det bara att sätta in de kända värdena i ekvationerna. Vi får ekvationssystemet:

$$5c_0 + 115c_1 = 31.01$$
$$115c_0 + 2655c_1 = 721.96$$

Detta ekvationssystem är lätt att lösa. Vi får följande räta linje som bäst anpassar mätvärdena: $f^*(x) = -13.877 + 0.873x$.

3 Beräkningar i Matlab

När vi ska anpassa kurvor med hjälp av Matlab kan vi arbeta enligt ovan och utnyttja de inbyggda funktionerna för att lösa ekvationssystem. Men i Matlab finns också speciella funktioner för kurvanpassning med polynom.

Man börjar med att definiera vektorer för mätpunkterna och mätvärdena. Därefter utnyttjar man funktionen `polyfit` som ger koefficienterna i det polynom av grad n som anpassar värdena. Med värden enligt det tidigare exemplet får vi (här är $n = 1$):

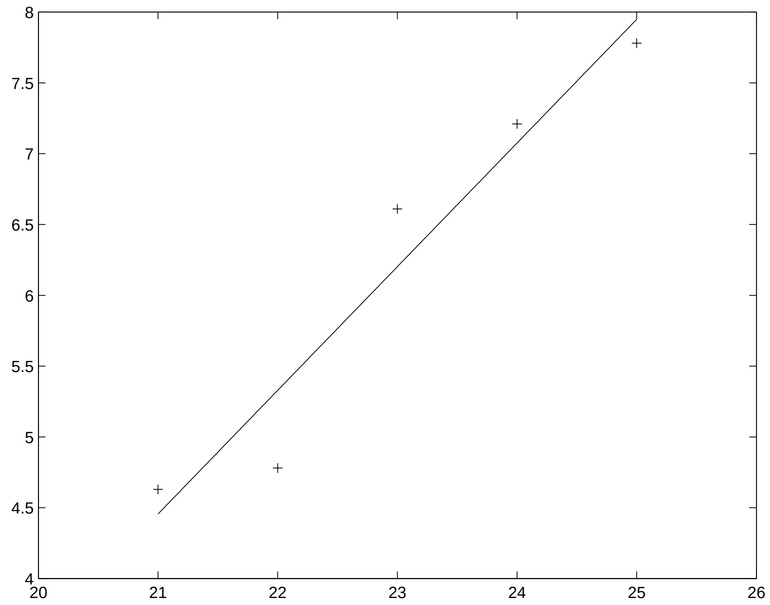
```
>> x = [21 22 23 24 25];
>> f = [4.63 4.78 6.61 7.21 7.78];
>> fstar = polyfit(x,f,1)

ans =
     0.8730 -13.8770
```

Vi kan också plotta resultatet:

```
>> plot(x,f,'+')
>> axis([20 26 4 8])
>> hold on
>> xpoints = x(1) : 0.01 : x(length(x));
>> plot(xpoints,polyval(fstar,xpoints))
```

Vi får en figur med följande utseende:



4 Slutsatser

Vi har sett att Matlab är ett utmärkt verktyg när man med hjälp av datorn ska anpassa en kurva till givna mätvärden. Men detta är inte det enda man kan använda Matlab till. Det finns mycket mera i Matlab, till exempel:

- Ekvationslösning
- Lösning av ekvationssystem
- Integration
- Lösning av differentialekvationer
- Linjär algebra
- ...

Om man har ett numeriskt problem bör man alltså inte i första hand skriva ett eget program som löser problemet. I stället bör man börja med att undersöka om Matlab har någon färdig funktion för problemet.