



**KTH Datavetenskap
och kommunikation**

Staffan Romberger
2008-10-31

DN1212/numpm Numeriska metoder och grundläggande programmering – Laboration 1 – Introduktion

Efter den här laborationen ska du kunna hantera vektorer och matriser, villkorssatser och slingor samt skriva egna program i .m-filer.

Vid redovisningen ska var och en vara beredd att redogöra för allt.

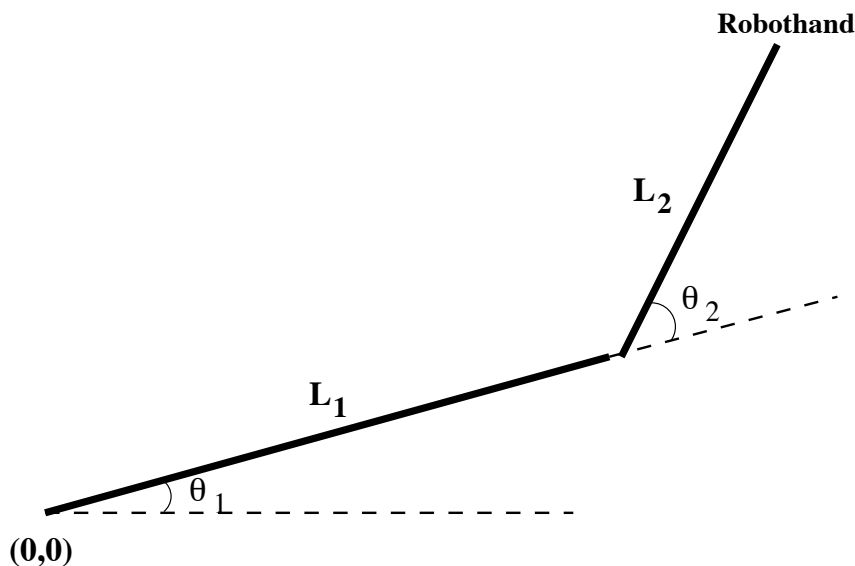
Kursens webbsida

Läs kurswebbsidan. Vi kommer att använda webben för att sprida kursinformation. Börja med att gå till den allmänna kurssidans <http://www.csc.kth.se/utbildning/kth/kurser/DN1212/> och välj sedan din egen kursomgång i menyn. Du bör titta på webbsidan minst en gång i veckan!

1 Lite administration

På kurswebbsidan finns information om kursanmälan. Fyll i webbformuläret och var noggrann med alla personuppgifter (viktigt för att vi ska kunna rapportera in dina kursmoment i LADOK).

2 Linjär algebra: robotarm



Figuren visar en robotarm med två länkar. Ledvinklarna ges av θ_1 och θ_2 . Koordinaterna för robothanden blir

$$\begin{aligned}x &= L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \\y &= L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2)\end{aligned}$$

där L_1 och L_2 är länklängderna.

Vinklarna θ_1 och θ_2 som bestämmer robothandens rörelse kontrolleras tidsmässigt av följande polynomuttryck:

$$\begin{aligned}\theta_1(t) &= \theta_1(0) + a_1 t^3 + a_2 t^4 \\ \theta_2(t) &= \theta_2(0) + b_1 t^3 + b_2 t^4\end{aligned}$$

där $\theta_1(0)$ och $\theta_2(0)$ är startvärdena för vinklarna vid tiden $t = 0$. Vinklarna anges i enheten grader och tiden i sekunder.

a) Ställ upp ett linjärt ekvationssystem för bestämning av parametrarna a_1 , a_2 , b_1 och b_2 , givet startvärdena $\theta_1(0) = 10$, $\theta_2(0) = 20$ samt vinklarnas värden då $t = 3$: $\theta_1(3) = 50.5$, $\theta_2(3) = -28.6$ och då $t = 4$: $\theta_1(4) = 131.6$, $\theta_2(4) = -140.0$. Lös ekvationssystemet i MATLAB.

b) Använd dina resultat i a) för att plotta robothandens svep när tiden t går från 0 till 4 sekunder, med värdena $L_1 = 4$ m och $L_2 = 3$ m.

3 Gissa hur mycket pumpan väger

Skriv ett program som slumpar fram ett heltal mellan ett och hundra och låter den som kör programmet gissa vilket tal det är. För att det inte ska bli för svårt måste ditt program skriva ut meddelanden efter varje gissning, t.ex. så här:

```
Vad heter du? Anna
Hej Anna, hur mycket tror du min pumpa väger? 50
För mycket. Gissa igen: 20
För litet. Gissa igen: 25
För litet. Gissa igen: 42
För mycket. Gissa igen: 36
Rätt gissat. Nästa år ska jag odla en ännu större pumpa.
```

4 Frivillig uppgift: Datorn gissar.

Skriv ett program som gissar vilken pumpavikt du tänker på. Du ska varje gång datorn gissar tala om om gissningen är för liten, för stor eller rätt. Hitta en bra algoritm som ger få gissningar!

Lycka till!



**KTH Datavetenskap
och kommunikation**

Staffan Romberger
2008-10-31

DN1212, Numeriska metoder och grundläggande programmering – Laboration 2 – Polynom Anpassning

Introduktion

Efter den här laborationen ska du kunna skriva lite större program som är uppdelade i egendefinerade funktioner.

Polynom Anpassaren

Du ska skriva ett program som kan användas för att hantera polynom av godtycklig grad. Användaren ska via en meny kunna mata in ett gradtal och ett antal punkter som polynomet ska anpassas till. Punkterna måste vara fler än gradtalet, kontrollera det. Är de en fler ritas polynomet som går exakt genom dessa punkter. Är de ännu fler ritas det bäst anpassade polynomet. Sedan, då användaren ber om det, ska polynomets koefficienter beräknas och polynomet och punkterna ritas upp. Dessutom ska användaren kunna få reda på vilka nollställen polynomet har.

Så här ska det se ut när användaren anger gradtal och punkter:

```
----- Polynom Anpassaren -----
1) Ange gradtal
2) Ange punkter
3) Beräkna koefficienter
4) Plotta polynomet
5) Hitta nollställen
0) Avsluta
Ditt val: 1
Vilken grad ska polynomet ha? 2
----- Polynom Anpassaren -----
1) Ange gradtal
2) Ange punkter
3) Beräkna koefficienter
4) Plotta polynomet
5) Hitta nollställen
0) Avsluta
Ditt val: 2
Ange punkter (retur avslutar inmatningen)
Ange en punkt ([x y]): [1 1]
Ange en punkt ([x y]): [2 1]
Ange en punkt ([x y]): [3 2]
Ange en punkt ([x y]):
```

Du ska själv skriva en funktion för varje menyval utom "Avsluta". Så här ska de anropas från

huvudprogrammet:

```
degree = askdegree();           - Frågar efter, läser in och returnerar gradtal.
points = askpoints();           - Frågar efter, läser in och returnerar punkterna.
coeffs = findcoeffs(degree, points); - Löser det linjära (överbestämda) ekvationssystemet
                                   och returnerar polynomets koefficienter.
                                   Du får inte använda polyfit; använd \-operatör.
plotpoly(coeffs, points);       - Ritar upp polynomets och punkterna.
polyroots(coeffs);              - Bestämmer och skriver ut polynomets nollställen.
```

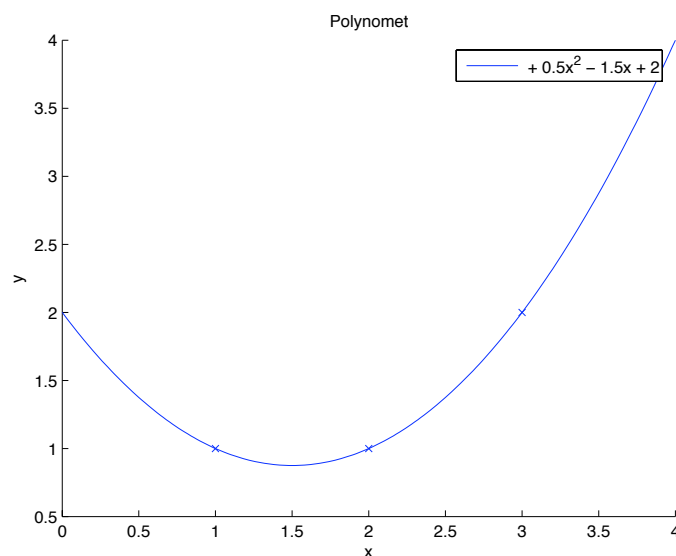
Utöver funktionerna ovan är det vettigt att skriva en funktion som bara skriver ut menyn. Skriver du programmet på det sättet blir koden i huvudprogrammet mer överskådlig. Då du anropar menyfunktionen i huvudprogrammet kan det se ut så här:

```
meny();
choice = input('Ditt val: ');
```

Då användaren ber om att koefficienterna ska beräknas ska de också skrivas ut. Innan dess måste han/hon ha angett gradtal och punkter.

Innan programmet ritar upp grafen måste det ha beräknat koefficienterna.

Grafen för exemplet ovan blir:



Tips, trix och krav

- Förslag på arbetsordning:
 1. Börja med att konstruera tomma funktioner.
 2. Konstruera menyn och anropa för respektive val de tomma funktionerna. Inget ska hända när användaren väljer något.
 3. Gör färdigt funktionerna i tur och ordning.
 4. När du är färdig med de två första funktionerna är det skönt att temporärt lägga in värden på gradtal och punkter direkt i koden, så du slipper skriva in det varje gång du ska prova resten av funktionerna. Kom ihåg att ändra tillbaka sedan!

- Grafen:
 - Använd `plot` för att rita polynomet. Programmet bör rita ut polynomet över ett intervall som täcker in punkterna. Ta fram min och max för punkternas x-värden! Från en vektor av x-värden beräknas y-värdena med `polyval` eller en `for`-slinga.
 - I grafen som ritas ska en »legend» innehålla polynomets grad och koefficienter.
Frivilligt: gör sådan »legend» som syns i exempelgrafan.
- Använd `roots` för att beräkna nollställena.
- Som avbrottsvillkor vid inmatning av punkter kan du använda att det inmatade har en längd som är skild från två (punkterna är ju vektorer med längden två).
- Du får inte använda några globala variabler!

Lycka till!



**KTH Datavetenskap
och kommunikation**

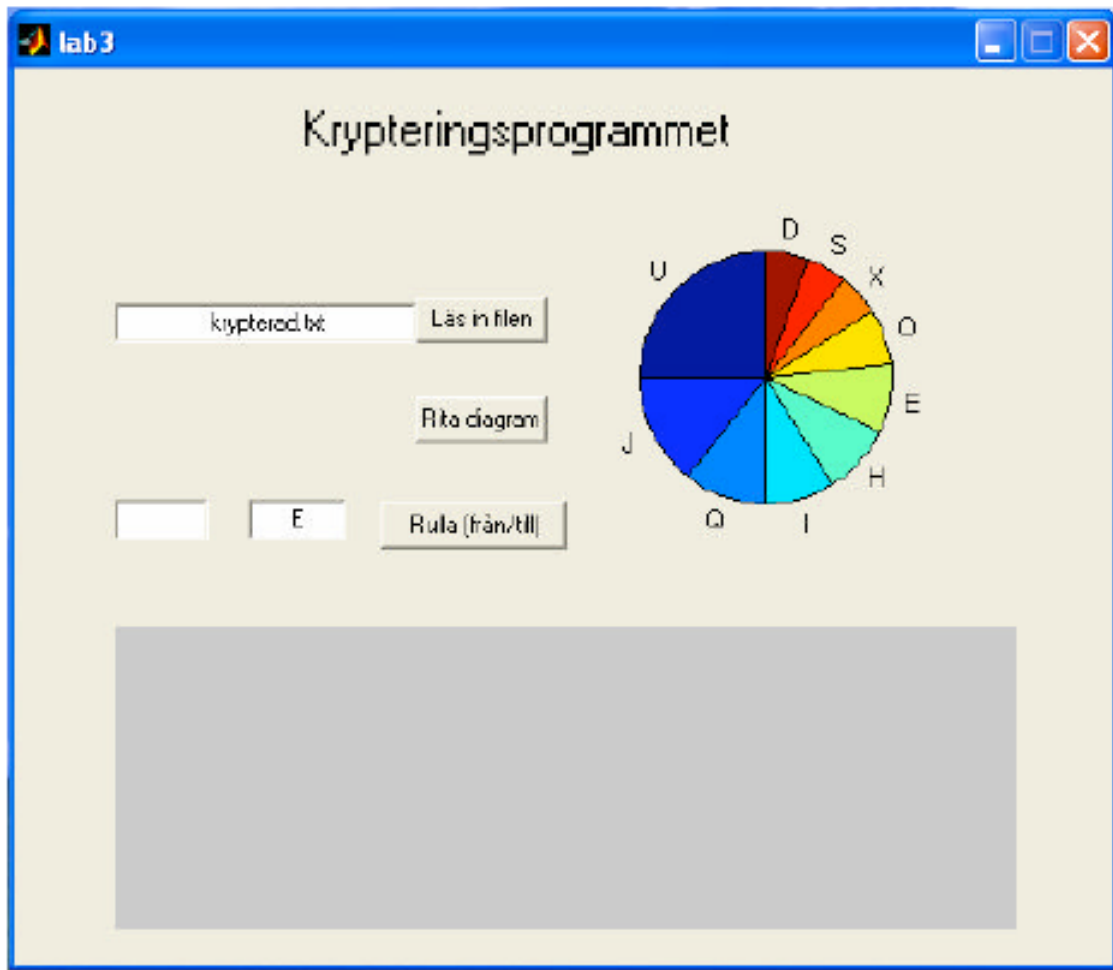
Staffan Romberger
2008-10-31

Numeriska metoder och grundläggande programmering – Laboration 3 – Kryptering

Efter den här laborationen ska du kunna hantera text som data och själv kunna skapa enkla grafiska användargränssnitt (GUI).

Översikt

Kryptering har blivit en del av vår vardag även om vi inte tänker så mycket på det. När du loggar in på din personliga banksida eller pratar i mobiltelefon sker kommunikationen krypterat. I denna laboration ska du skriva ett program som analyserar en krypterad text och sedan dekrypterar den. Ditt program ska läsa och dekryptera en text på engelska som finns på fil. Det finns många olika algoritmer för kryptering men vi förutsätter att texten är krypterad med »caesarrullning» vilket innebär att varje bokstav har bytts mot en bokstav ett visst antal steg framåt i alfabetet. Med 2 positioners förskjutning blir »The zebra has stripes» krypterat till »Vjg bgdte jcu uvtkrgu». Med vetskap om att »e» är den vanligaste bokstaven i engelsk text kan texten dekrypteras med hjälp av ett frekvensdiagram. Om t.ex. bokstaven »g» är den mest frekventa i kryptotexten kan vi gissa att förskjutningen är 2 steg. Gränssnittet för ditt program bör se ut så här:



Ditt program bör lämna andra tecken än bokstäver oförändrade. Tårtdiagrammet ska innehålla de 10 mest frekventa bokstäverna. Det blir dessutom enklare om du gör om alla bokstäver till versaler. På kursens webbplats finns filer med krypterade texter som du kan använda när du testar. Om du vill kan du bygga ut programmet så att det också kan hantera åäö. Dessa bokstäver har inte teckenkoder intill a-z.