

TANA17 Matematiska beräkningar med Matlab

Laboration 4. Data analys

Namn: \_\_\_\_\_

Personnummer: \_\_\_\_\_

Epost: \_\_\_\_\_

Namn: \_\_\_\_\_

Personnummer: \_\_\_\_\_

Epost: \_\_\_\_\_

Godkänd den: \_\_\_\_\_

Sign: \_\_\_\_\_

Retur: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

# 1 Beräkning av Integraler

Integralen

$$I = \int_a^b f(x)dx,$$

kan approximeras med *Trapetsregeln*. Vi väljer då  $(N + 1)$ st jämnt utspridda  $x$ -värden

$$a = x_1 < x_2 < \dots < x_N < x_{N+1} = b$$

så att avståndet mellan punkterna blir  $h = (b - a)/N$ . Integralens värde approximeras med

$$I_h = h \left( \frac{f(x_1)}{2} + f(x_2) + \dots + f(x_N) + \frac{f(x_{N+1})}{2} \right).$$

Givet att ett stort antal  $x$ -värden används så bör vi få en bra approximation av integralens rätta värde.

**Uppgift 1.1** Skriv en funktion `Trapets` som beräknar en integral med trapetsmetoden. Funktionen skall kunna anropas enligt

```
>> Ih = Trapets( @funk , a , b , N );
```

där  $[a, b]$  är intervallet, `funk` är den funktion som skall integreras, och  $N$  är antalet  $x$ -värden som skall användas.

Vill du beräkna funktionen  $f(x)$  inne i din `Trapets` funktion så använd `feval`.

**Uppgift 1.2** Använd funktionen `Trapets` för att beräkna en approximation  $I_h$  av

$$I = \int_0^\pi \sin(x^2)dx$$

Du kan antingen skriva en ny funktionsfil som implementerar  $f(x) = \sin(x^2)$  eller skapa ett funktionshandtag. Vilket värde får du om du använder  $N = 100$ ?

Svar: \_\_\_\_\_

**Tips** Du kan använda `integral` för att verifiera att din funktion ger rätt resultat.

**Uppgift 1.3** Vi väljer nu att beräkna

$$\int_0^1 \frac{4}{1+x^2} dx = \pi.$$

med hjälp av Trapetsregeln och vill undersöka hur felet i den beräknade approximationen av  $\pi$  beror på antalet delintervall som används. Skriv ett program som beräknar felet  $\epsilon(h) = |I - I_h|$  i Trapetsapproximationen för  $N = 10, 20, 30, \dots, 200$ .

Vi antar att felet kan uttryckas som  $C(h)^p$  för något heltal  $p$ . Utnyttja att

$$\log(\epsilon(h)) = \log(C) + p \log(h) = c_1 + c_2 y = p_1(y), \quad y = \log(h).$$

för att hitta de bästa parametrarna  $C$  och  $p$  med hjälp av minsta kvadratmetoden. Vilka värden på  $C$  och  $p$  får du? Parametern  $p$  borde bli ett heltal. Stämmer det med dina resultat?

Svar: \_\_\_\_\_

Plotta även de observerade felen (med röda +) och de fel som förutsägs av modellen med dina värden för  $C$  och  $p$ . Använd `loglog` när du plottar.

**Tips** Här kan du antingen använda `polyfit` eller formulera ditt eget minsta kvadratproblem.

**Redovisa** Skriv ut din funktion `Trapets.m`, huvudprogrammet som undersöker felets beroende på  $h$ , samt grafen som illustrerar felen.

## 2 En Komet

Kometer rör sig enligt ellipsformade banor. När man först ser en ny komet så vill man uppskatta dess bana. Det görs genom att man samlar in observerade positioner för kometen. Man kan sedan anpassa en ellips så bra som möjligt till de observerade positionerna med hjälp av minsta kvadrat metoden.

En punkt  $(x, y)$  ligger på en ellips om ekvationen

$$c_1 x^2 + c_2 xy + c_3 y^2 + c_4 x + c_5 y + 1 = 0.$$

är uppfylld för en specifik uppsättning parametrar  $c = (c_1, c_2, c_3, c_4, c_5)^T$ . Det är inte säkert att en uppsättning parametrar ger en ellips. Man kan även få parabler eller hyperbler.

**Uppgift 2.1** Antag att ett antal observationer  $(x_k, y_k)$ ,  $k = 1, \dots, n$ , för kometen finns. Skriv ner det överbestämda ekvationssystem  $Ac = b$  som uppstår då vi försöker bestämma ellipsen med minsta kvadrat metoden.

Svar: \_\_\_\_\_

**Uppgift 2.2** Programmet `CometObservation` simulerar observationer av kometen. Använd programmet genom att skriva

```
>> Times = 0:0.05:0.5;
>> PositionData=CometObservation(Times);
```

så skapas en matris `PositionData` med observerade positioner för kometen vid tiderna som sparats i vektorn `Times`. Första raden i matrisen, dvs `PositionData(1,:)`, innehåller  $x$ -koordinater och andra raden innehåller  $y$ -koordinater.

Använd de observationer som finns i `PositionData` matrisen för att hitta parametrarna som bestämmer ellipsen.

Använd slutligen programmet `DisplayComet` för att rita upp ellipsen och de observationer som användes. Programmet anropas enligt

```
>> DisplayComet( Coefs , PositionData )
```

där `Coefs` är en vektor som innehåller coefficienterna  $c_1, c_2, c_3, c_4$  och  $c_5$  ovan. Blev resultatet bra?

Svar: \_\_\_\_\_

**Kommentar** Med `Times=0:0.05:0.5` använder vi  $n = 11$  punkter som endast täcker en liten del av ellipsen. Det gör att det är svårt att hitta bra värden på koefficienterna.

**Uppgift 2.3** Försök förbättra ellipsen genom att lägga till fler observationer. Ändra tiderna då observationerna görs till

```
>> Times = 0:0.05:3.0;
```

Hur många observationer används nu för att bestämma ellipsen? Blev resultatet bra?

Svar: \_\_\_\_\_

**Uppgift 2.4** Ett alternativ till fler observationer är att sprida ut observationerna bättre. Använd observationstider

```
>> Times = 0:0.5:5.0;
```

Hur många observationer används nu? Blev resultatet bra?

Svar: \_\_\_\_\_

**Uppgift 2.5** Vad är viktigast: Att ha många observationer eller att observationerna är väl utspridda?

Svar: \_\_\_\_\_

**Redovisa** Skriv ut ditt program samt den sista grafen för det sista fallet, dvs `Times = 0:0.5:5.0;`.