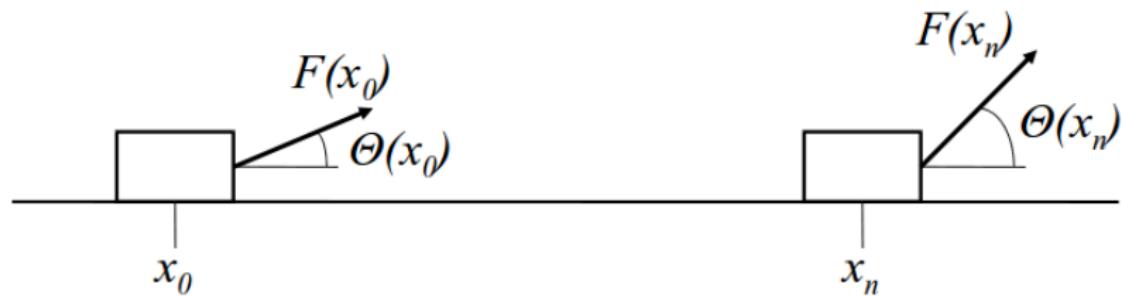


Fallstudie: numerisk integration
Baserad på läroboken, Case Study 19.9
Beräkningsvetenskap DV

Institutionen för Informationsteknologi, Uppsala Universitet

30 september, 2013

Att beräkna arbete



Problem: Beräkna arbetet för att förflytta blocket från x_0 till x_n givet kraften $F(x)$ och vinkeln $\theta(x)$.

Matematisk modell

$$W = \int_{x_0}^{x_n} F(x) \cos(\theta(x)) dx$$

Beteckningar:

W : arbetet (Joule)

x : position i rummet (meter)

$F(x)$: kraftens storlek i punkten x (Newton)

$\theta(x)$: vinkel i punkten x (radianer)

Angreppssätt

Olika angreppssätt för olika fall:

- $F(x)$ och $\theta(x)$ givna som formler och enkla: analytisk integration
- $F(x)$ och $\theta(x)$ givna som formler men svåra: numerisk integration
- $F(x)$ och $\theta(x)$ givna som mätvärden: numerisk integration

Angreppssätt (forts.)

På föreläsningarna har vi lärt oss:
Steg i numerisk lösning av problemet:

- Diskretisering:
 - a. Diskreta punkter i integrationsintervallet
 - b. Approximation av integranden med styckvis polynom
 - c. Bestämning av integralen av det styckvisa polynomet
- Den formel som blir resultatet av a,b,c tillämpas på våra aktuella data; det beräknade värdet blir vårt approximativa värde på W .

Angreppssätt (forts.)

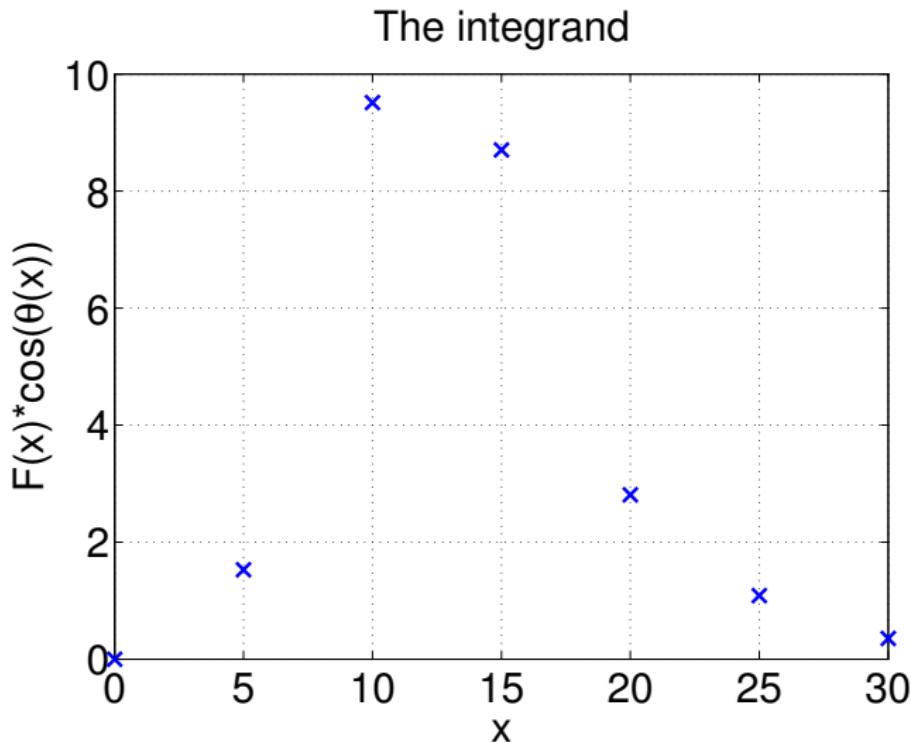
Viktigt att notera:

- När steg 1 a,b,c utförs på en allmän integral, så leder det till en generell formel för numerisk integrering. Så har exempelvis trapetsformeln och Simpsons formel härletts. Normalt går man alltså direkt till steg 2.
- Diskretiseringfelet kallas också trunkeringsfel och minskar med minskande steglängd (ökande antal punkter).

I vårt fall: $F(x)$ och $\theta(x)$ är givna i form av mätvärden

x	$F(x)$	$\theta(x)$
0	0.0	0.50
5	9.0	1.40
10	13.0	0.75
15	14.0	0.90
20	10.5	1.30
25	12.0	1.48
30	5.0	1.50

Våra indata: integranden $F(x) * \cos(\theta(x))$

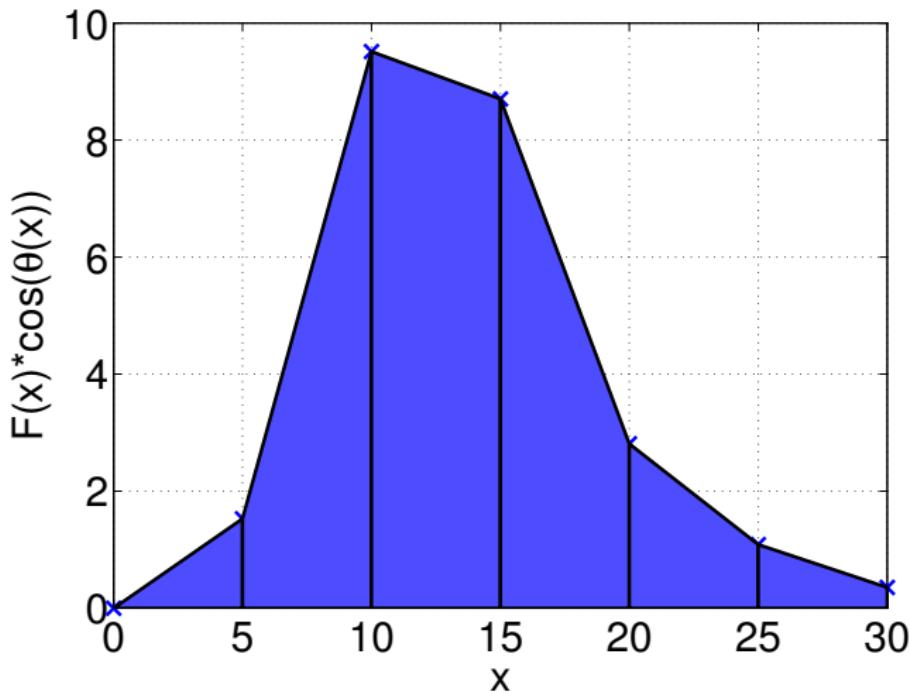


Lösning med trapetsformeln

Skriv ett program som löser vårt problem med hjälp av trapetsformeln. Använd Matlabs inbyggda funktion trapz. Använd samtliga tillgängliga punkter och uppskatta felet med tredjedelsregeln.

Trapetsformeln

The composite trapezoidal formula



Algoritmskiss

- Lagra våra x -värden i en variabel x
- Lagra våra $F(x)$ -värden i en variabel F
- Lagra våra $\theta(x)$ -värden i en variabel $theta$
- Beräkna $g(x)$ för våra givna x -värden och lagra resultatet i en variabel g
- Beräkna $T(h)$, lagra i variabeln Th
- Beräkna $T(2h)$, lagra i variabeln $T2h$
- Beräkna uppskattat fel $(T(h) - T(2h))/3$ och lagra värdet i variabeln $trunc_err$
- Skriv en resultatutskrift (med Th och $trunc_err$)

Implementering i Matlab

```
x      = [0 5 10 15 20 25 30];
F      = [0.0 9.0 13.0 14.0 10.5 12.0 5.0];
theta = [0.50 1.40 0.75 0.90 1.30 1.48 1.50];
g      = F.*cos(theta);
Th     = trapz(x,g);
T2h    = trapz(x(1:2:7),g(1:2:7));
trunc_err = (Th-T2h)/3;
disp(['Approximate value W: ' num2str(Th)])
disp(['Estimated truncation error: ' num2str(trunc_err)])
```

Resultat från körning av programmet

Approximativt värde på W : 119.0892

Uppskattat diskretiseringfel: -1.9621

Hur noggrant kan vi bedöma att det beräknade W -värdet är?

$F(x)$ -värdena (och därmed $g(x)$ -värdena) är givna med en korrekt decimal. Funktionsfelet i beräkningen av W ovan kan då till absolutbeloppet vara så stort som:

$$(30 - 0) \cdot 0.5 \cdot 10^{-1}$$

Totalt kan felet till absolutbeloppet uppskattas vara högst ca $1.5 + |-1.9621| \approx 3.5$.

Slutsats: $W \approx 119 \pm 3.5$

OBS! Detta är den uppskattning av felet som våra data medger.
Det verkliga felet kan vara större.

Med experimentdata från .MAT-fil

Fallet då experimentdata finns lagrade i .MAT-fil:

```
fil = input('Input data filename (without .MAT): ', 's');
load(fil)
np  = length(x); % Number of points
g   = F.*cos(theta);
Th  = trapz(x,g);
T2h = trapz(x(1:2:np),g(1:2:np));
trunc_err = (Th-T2h)/3;
```

etc.

Med Matlab-funktion för $g(x)$

Fallet när en formel för $g(x)$ är given som en Matlab-funktion $g(x)$.

```
v = input('Integration interval [a b]: ')
n = input('Desired number of subintervals: ')
x = linspace(v(1),v(2),n+1)
gvalues = g(x);
Th = trapz(x,gvalues);
T2h = trapz(x(1:2:n+1),gvalues(1:2:n+1));
trunc_err = (Th-T2h)/3;
```

Att fundera på

Vad skulle behövas för att programmet automatiskt skulle kunna beräkna integralen med en viss av användaren önskad noggrannhet?

(I vilka fall är det realistiskt? Hur skulle en algoritm kunna se ut?)