

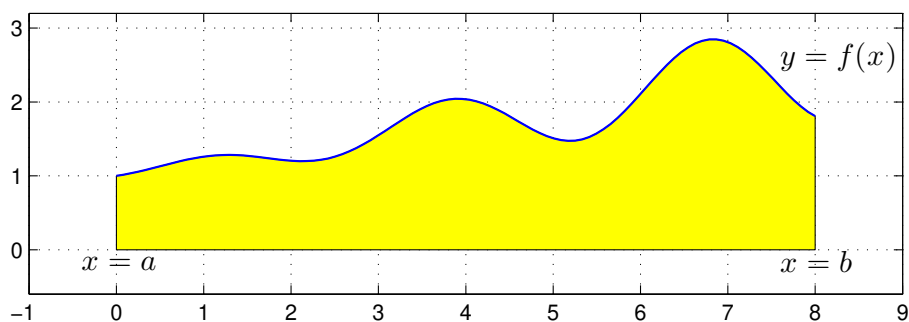
# Numerisk beräkning av integraler

## 1 Inledning

Ibland kan man inte bestämma integraler exakt utan man får nöja sig med att beräkna approximationer. T.ex.  $\int_0^1 e^{x^2} dx$  kan inte beräknas exakt, eftersom det inte finns någon användbar primitiv funktion. Det kan också vara så att integranden bara är känd i vissa punkter, t.ex. vi har en serie med mätdata.

## 2 Beräkningsmetoder

Den geometriska tolkningen av integralen  $\int_a^b f(x) dx$  är arean av ytan mellan grafen av integranden  $y = f(x)$  och  $x$ -axeln, dvs.  $y = 0$ , mellan  $x = a$  och  $x = b$ .



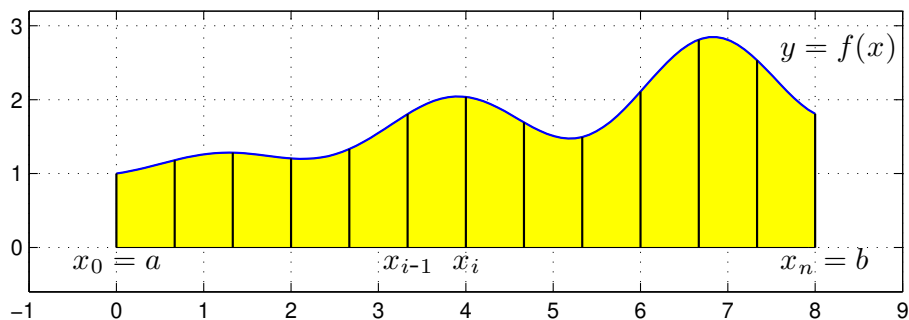
Vi gör en likformig indelning av intervallet  $a \leq x \leq b$

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b$$

så att vi får  $n$  lika långa delintervall  $x_{i-1} \leq x \leq x_i$  med bredden  $h = \frac{b-a}{n}$ .

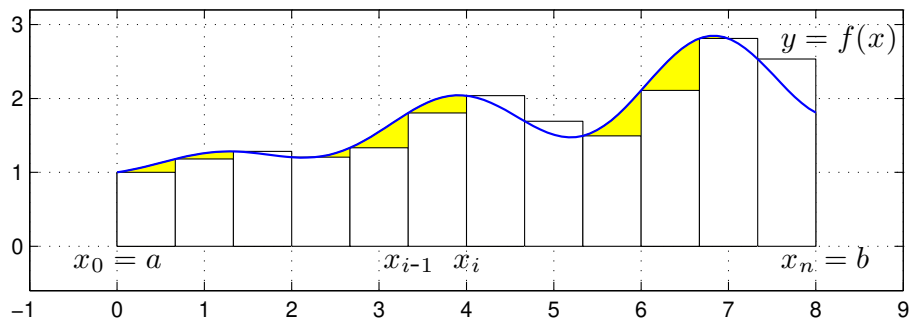
Sedan delar vi upp integralen i en summa av delintegraler över varje delintervall

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx$$



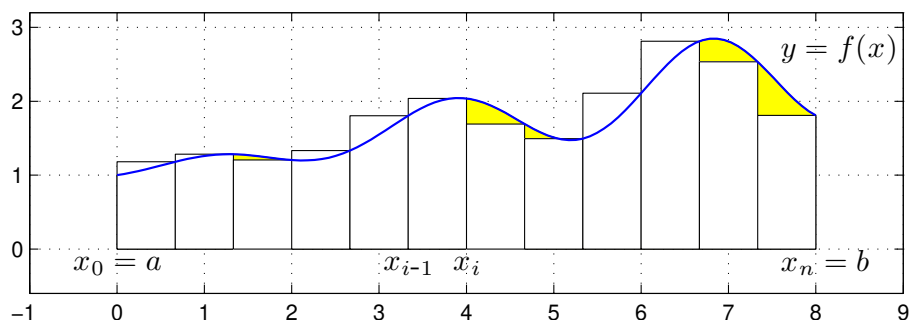
Om vi approximerar  $f(x)$  med  $f(x_{i-1})$  i intervallen  $x_{i-1} \leq x \leq x_i$  får vi **vänster rektangelregel**

$$\int_a^b f(x) dx \approx \sum_{i=1}^n h f(x_{i-1})$$



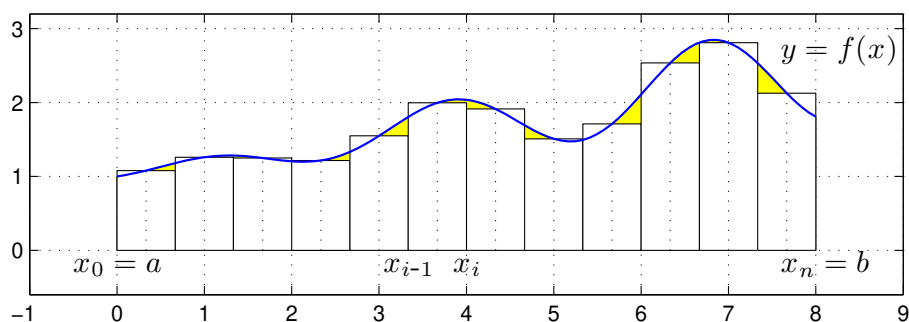
Om vi approximerar  $f(x)$  med  $f(x_i)$  i intervallen  $x_{i-1} \leq x \leq x_i$  får vi **höger rektangelregel**

$$\int_a^b f(x) dx \approx \sum_{i=1}^n h f(x_i)$$



Om vi approximerar  $f(x)$  med  $f(m_i)$  i intervallen  $x_{i-1} \leq x \leq x_i$ , där  $m_i$  är mittpunkterna i intervallen, får vi **mittpunktsmetoden**

$$\int_a^b f(x) dx \approx M_n = \sum_{i=1}^n h f\left(\frac{x_{i-1} + x_i}{2}\right)$$



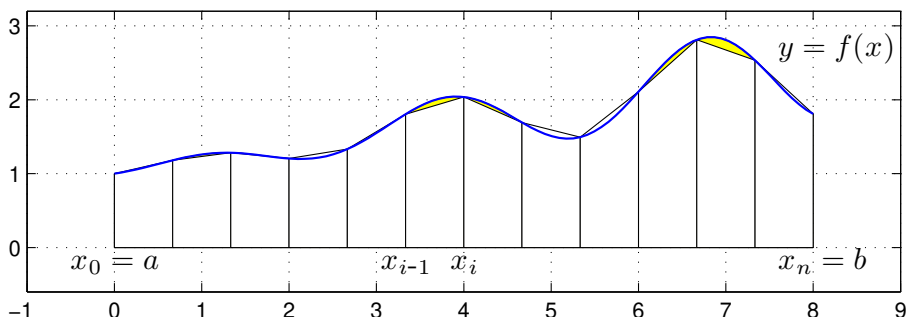
I Persson-Böiers kapitel 6.2 definieras (konstrueras) integralen  $\int_a^b f(x) dx$  med hjälp av Riemannsumman

$$\sum_{i=1}^n f(c_i) h_i$$

Metoderna ovan är olika varianter av Riemannsummor, med  $c_i = x_{i-1}$ ,  $c_i = x_i$  respektive  $c_i = m_i$ , och  $h_i = h$ .

Vi kan också approximera integralen med medelvärde av vänster och höger rektangelregel och får då **trapetsmetoden**

$$\int_a^b f(x) dx \approx T_n = \sum_{i=1}^n \frac{h}{2} (f(x_{i-1}) + f(x_i))$$



Antag att vi vill beräkna  $\int_0^1 x \sin(x) dx$  med vänster rektangelregel med  $n = 100$ . Vi skulle kunna göra så här

```
>> n=100;
>> f=@(x)x.*sin(x); a=0; b=1;
>> h=(b-a)/n
>> q=0;
>> for i=0:n-1
    x=a+i*h;
    q=q+h*f(x);
end
>> q
```

Att använda en for-sats är oftast inte effektivt i MATLAB. Vi bildar hellre en radvektor av alla funktionsvärdena  $f(x_i)$  och sedan summerar dessa enligt

```
>> n=100;
>> f=@(x)x.*sin(x); a=0; b=1;
>> x=linspace(a,b,n+1);
>> h=(b-a)/n;
>> q=sum(h*f(x(1:n)))
```

Detta sätt att organisera en beräkning kallas att vektorisera den, dvs. man bildar först en eller flera vektorer och utför sedan den önskade beräkningen på dem. De elementvisa operationerna `.*` `./` `.^` är exempel på vektoriserade operationer. Vi använde funktionen `sum` som snabbt summerar en vektor.

**Uppgift 1.** Beräkna en approximation av integralen  $\int_0^1 x \sin(x) dx$  med vänster och höger rektangelregel samt mittpunkts- och trapetsmetoderna. Använd `sum`.

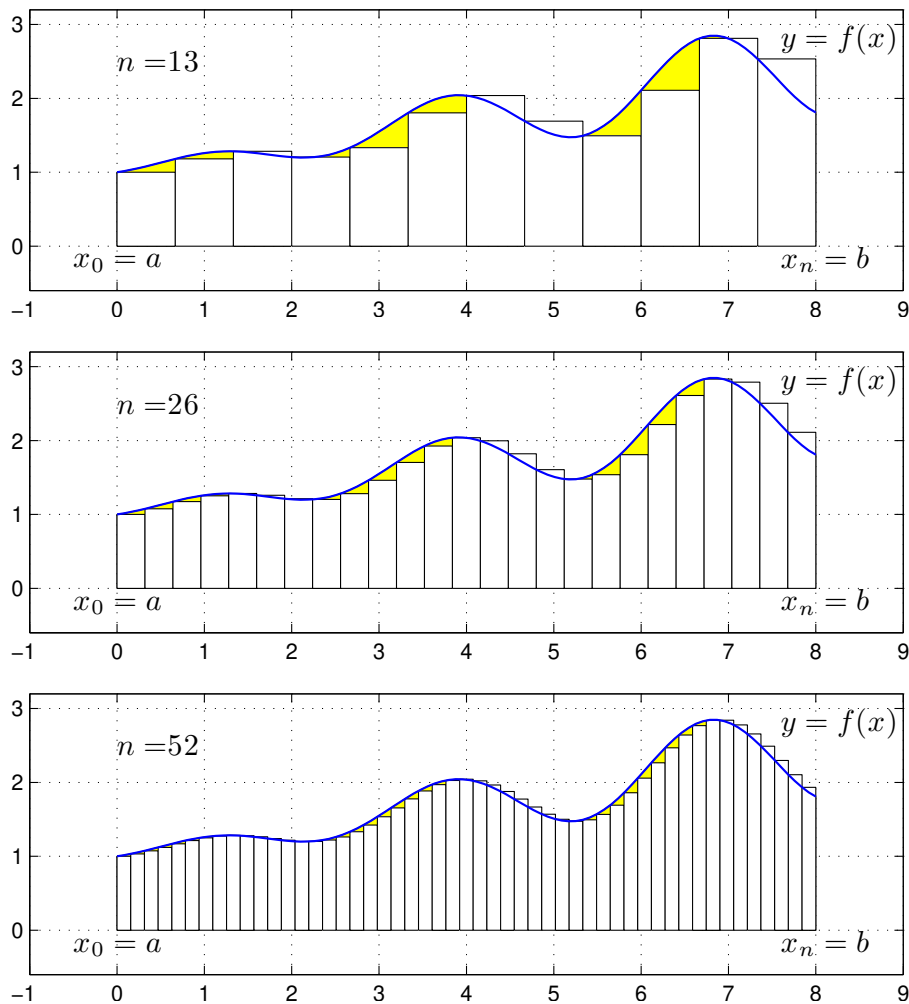
**Uppgift 2.** Skriv en funktion med namnet `min_integral` som beräknar integralen approximativt. Funktionen skall ha anropet `q=min_integral(f,I,n,k)`, där `f` är ett funktionshandtag till funktionen som beräknar  $f(x)$ , `I` är en radvektor som ger intervallet  $a \leq x \leq b$ , `n` ger antal delintervall (steglängden ges av  $h = (b - a)/n$ ) och `k` ger metoden som skall användas ( $k = 1, 2, 3, 4$  ger vänster och höger rektangelregel, mittpunkts- och trapetsmetoderna respektive).

**Uppgift 3.** Testa ditt program på följande integraler. Variera metodval och antal delintervall  $n$ .

(a).  $\int_0^1 e^{-x^2} dx$       (b).  $\int_{-1}^1 \frac{1}{1+x^2} dx$       (c).  $\int_0^1 \tan(\sqrt{x}) dx$

### 3 Konvergens

För metoderna ovan gäller att samtliga är konvergenta, dvs. låter vi antal delintervall  $n$  gå mot oändligheten så går approximationerna mot integralens värde. Vi ser på några bilder för vänster rektangelregel där  $n$  blir allt större



Vi ser att vi allt bättre täcker upp ytan under grafen med allt fler och smalare staplar.

Nu räcker det i praktiken inte med konvergens. Vi måste få en bra approximation på en kort tid, dvs. inte behöva ta  $n$  alltför stort.

För vänster och höger rektangelregel gäller att om vi fördubblar antal delintervall så halveras felet i approximationen av integralen. För mittpunkts- och trapetsmetoderna gäller vid samma fördubbling att felet delas med fyra, dvs. mycket bättre utdelning.

**Uppgift 4.** Vi ser på integralen  $\int_0^1 x \sin(x) dx$  igen. Beräkna integralen exakt (för hand). Jämför exakt värde med de approximationer vi får med metoderna ovan för olika antal delintervall  $n$ . Hur stort blir felet? Tag t.ex först  $n = 50$  och sedan  $n = 100$ , beräkna felet i approximationerna och se efter hur felet förändras.

## 4 Färdiga program i MATLAB

Det finns färdiga funktioner i MATLAB för att integrera. En sådan funktion är `integral` som används enligt

```
q=integral(fun,a,b)      q=integral(fun,a,b,name,value)
```

där `fun` är en funktion som beskriver integranden, `a` och `b` ger integrationsintervallet.

Funktionen `fun` måste kunna beräkna funktionsvärden för en radvektor `x`, dvs. elementvisa operationer skall användas.

Om vi vill använda `integral` för att beräkna integralen  $\int_0^1 x \sin(x) dx$  så skulle det se ut så här

```
>> f=@(x)x.*sin(x); a=0; b=1;
>> q=integral(f,a,b)
```

Genom att ge `name` som `'AbsTol'` eller `'RelTol'` kan vi ange önskad absolut respektive relativ noggrannhet i beräkningen och med `value` ger vi värdet på noggrannheten.

**Uppgift 5.** Beräkna arean av det slutna området mellan graferna till funktionerna

$$g(x) = e^{-\frac{x^2}{2}} \quad \text{och} \quad h(x) = x^2 - 3x + 2.$$

Rita en bild av området. Använd `fzero`, `fill` och `integral`. Tänk på att använda elementvisa operationer.