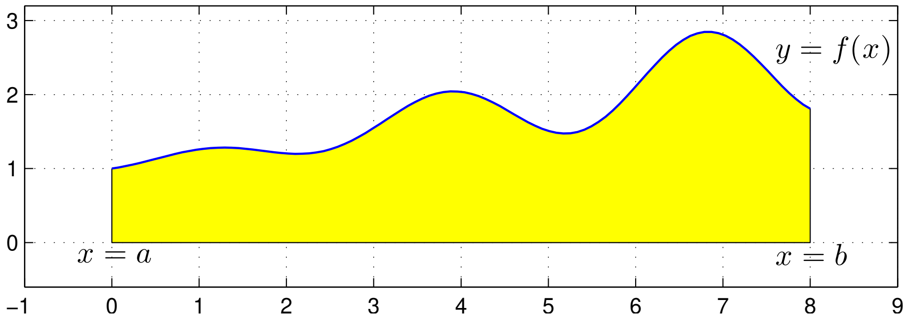
Integraler

Ibland kan man inte bestämma integraler exakt utan man får nöja sig med att beräkna approximationer. T.ex. $\int_0^1 \mathrm{e}^{x^2}\,dx$ 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.

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 $x = b$.



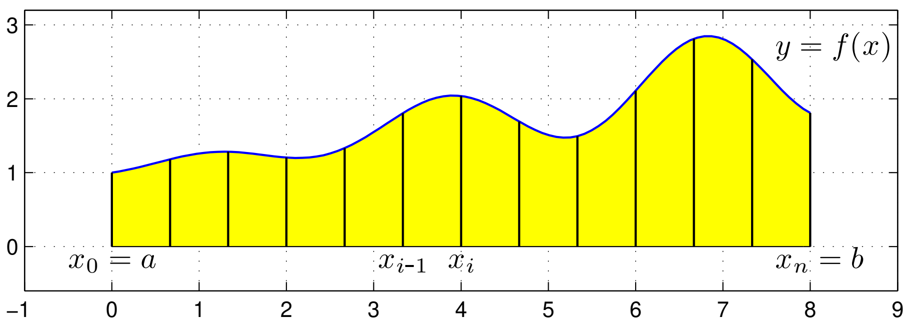
Vi gör en likformig indelning av intervallet $a \le x \le b$

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

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

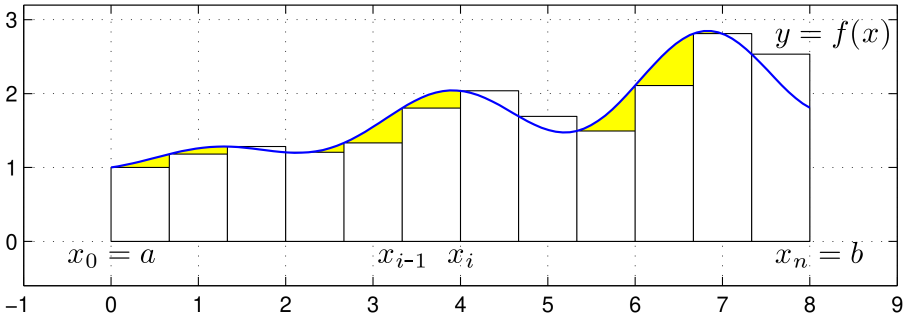
Sedan delar vi upp integralen i en summa av delintegraler över varje delintervall

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



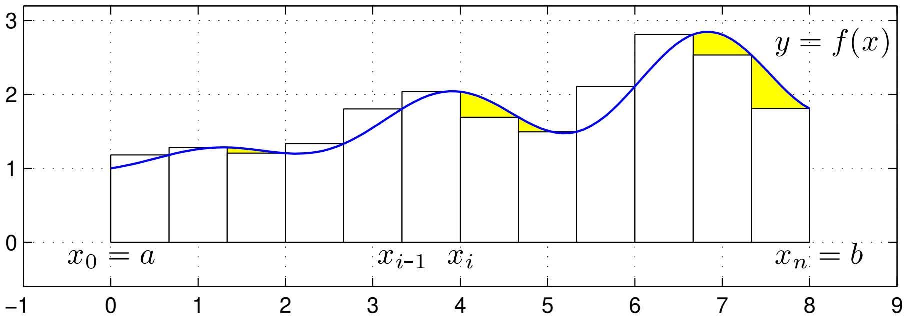
Om vi approximerar $f(x)$ med $f(x_{i-1})$ i intervallen $x_{i-1} \leq x \leq x_i$ får vi **vänster rektangelregel**

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



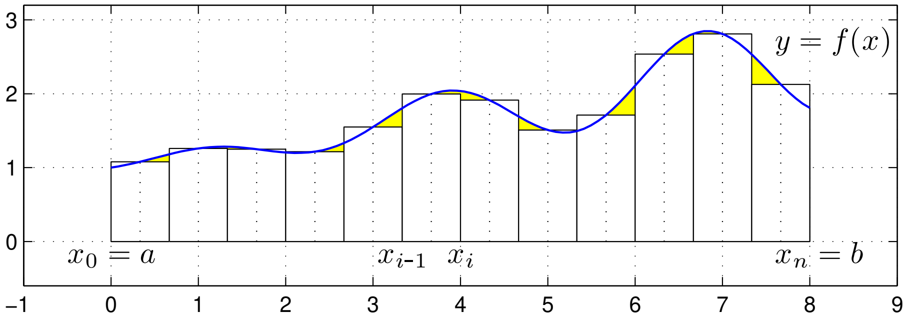
Om vi approximerar $f(x)$ med $f(x_i)$ i intervallen $x_{i-1} \leq x \leq x_i$ får vi **höger rektangelregel**

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



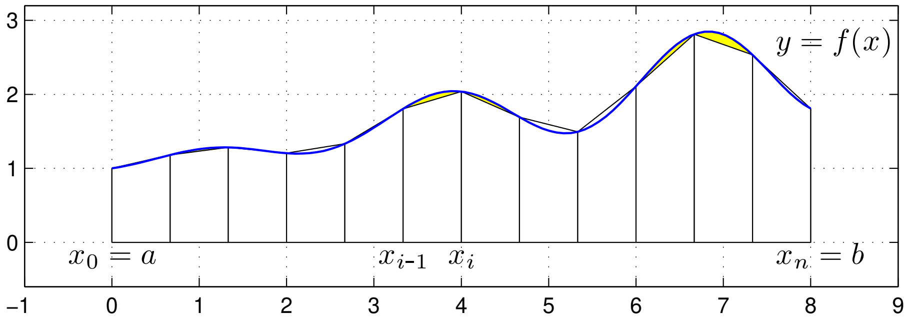
Om vi approximerar $f(x)$ med $f(m_i)$ i intervallen $x_{i-1} \leq x \leq x_i$, där m_i är mittpukterna i intervallen, får vi **mittpunktsmetoden**

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



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

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



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

```
>> n=100;
>> a=0; b=1; f=@(x)x.*cos(x);
>> 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 genererar hellre en vektor av alla funktionsvärdena $f(x_i)$ och sedan summerar dessa enligt

```
>> n=100;
>> a=0; b=1; f=@(x)x.*cos(x);
>> 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 genererar först en eller flera vektorer och utför sedan den önskade beräkningen på dem. De komponentvisa operationerna `.*` `./` `.^` är exempel på vektoriserade operationer. Vi använde funktionen `sum` som snabbt summerar en vektor.

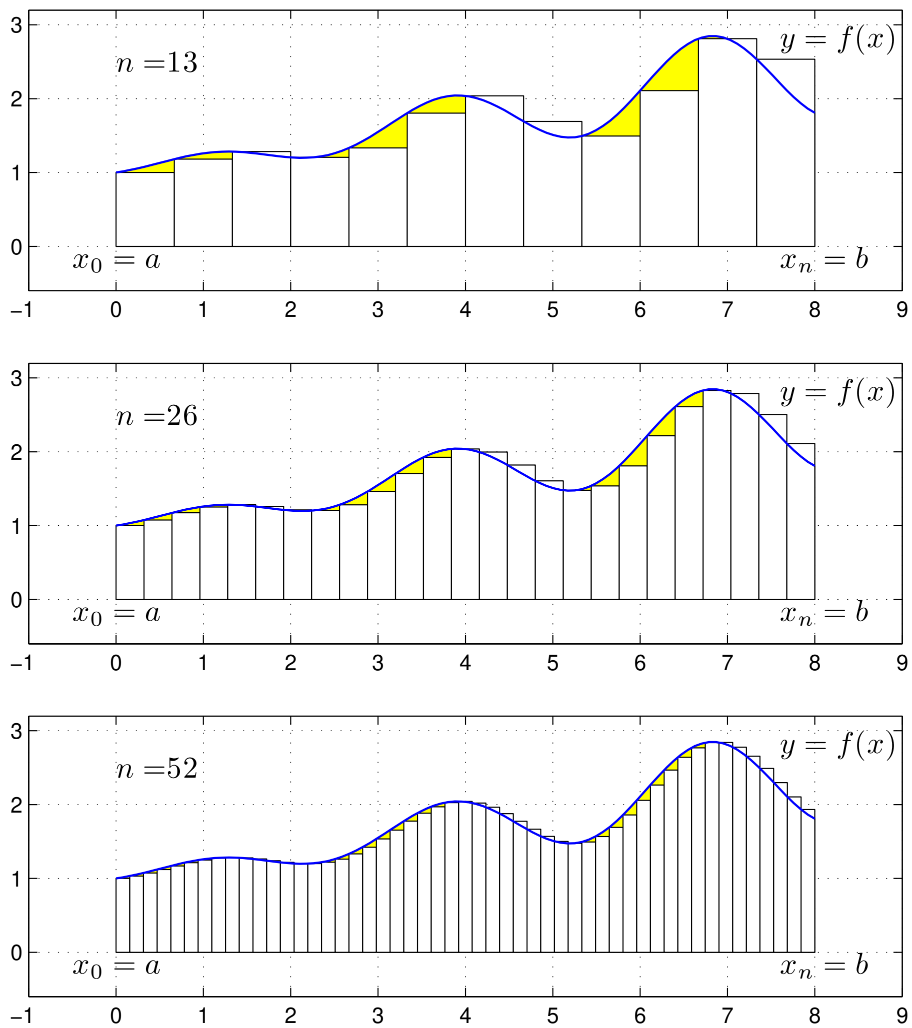
Uppgift 1. Beräkna en approximation av integralen $\int_0^1 x\,\cos(x)\,dx$ med vänster och höger rektangelregel samt mittpunkts- och trapetsreglerna. Använd `sum`.

Uppgift 2. Skriv en function med namnet `min_integral` och anropet `q=min_integral(f,I,n,k)` som beräknar en integral approximativt. På kurshemsidan finner du ett programskal du kan utgå ifrån. In- och ut-variablerna förklaras i programskalet.

Uppgift 3. Testa din funktion `min_integral` på följande integraler. Variera $h = (b - a)/n$.

(a). $\int_0^1 \sin(x^2)\,dx$ **(b).** $\int_{-1}^1 \frac{1}{1+5x^2}\,dx$ **(c).** $\int_0^1 \tan(\sqrt{x})\,dx$

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 $\int_0^1 x\cos(x)\,dx$ 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 felen i approximationerna och se efter hur felen förändras.

Färdiga program i Matlab

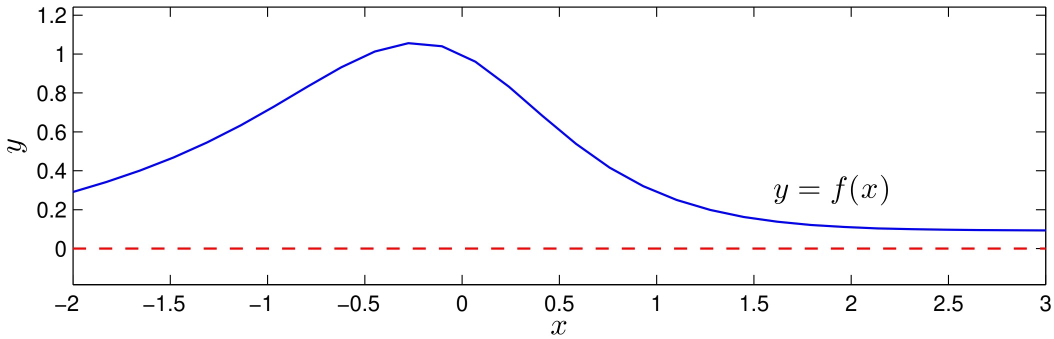
Det finns färdiga funktioner i Matlab för att integrera, både effektivt och noggrant. En sådan funktion är `integral`. Om vi skulle använda `integral` för att beräkna integralen i uppgift 1 skulle det kunna se ut så här

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

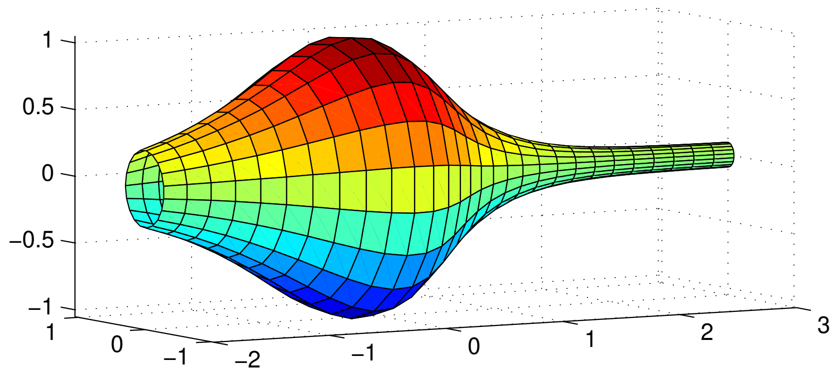
Som ytterligare ett exempel ser vi på en rotationskropp. Betrakta grafen av funktionen

$$f(x) = \frac{1 - 0.5\,\sin(x)}{1 + x^2}$$

över intervallet $-2 \le x \le 3$.



Låter vi grafen rotera runt x-axeln får vi en rotationsyta och vi skall beräkna den inneslutna volymen samt rotationsytans area.



Volymen som begränsas av rotationsytan ges av $V = \int_a^b A(x)\,dx$, där $A(x)$ är tvärsnittsarean (Stewart kapitel 6.2). Vi har

$$A(x) = \pi\,(f(x))^2$$

Arean av rotationsytan ges av (Stewart kapitel 8.2)

$$S = 2\pi \int_a^b |f(x)| \sqrt{1 + (f'(x))^2}\,dx$$

För vårt exempel nöjer vi oss med en numerisk beräkning av volym V och ytarea S enligt

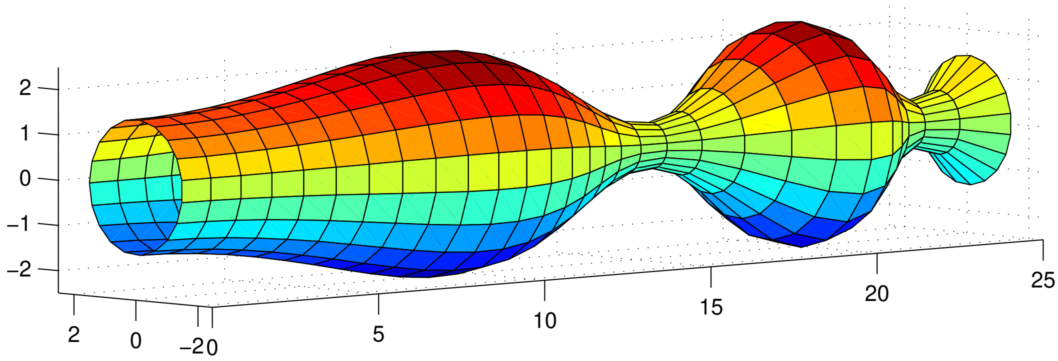
```
>> f=@(x)(1-0.5*sin(x))./(1+x.^2);
>> Df=@(x)-0.5*cos(x)./(1+x.^2)-(1-0.5*sin(x))*2.*x./(1+x.^2).^2;
>> a=-2; b=3;
>> V=pi*integral(@(x)f(x).^2,a,b)
V =
    5.1095

>> S=2*pi*integral(@(x)abs(f(x)).*sqrt(1+Df(x).^2),a,b)
S =
   16.3260
```

Uppgift 5. Beräkna volymen och arean av rotationsytan som bildas då grafen till

$$f(x) = 1.5 + \sin(0.02\,x^2), \quad 0 \le x \le 25$$

roterar runt x-axeln.

Så här ser ytan ut



Om du vill se koden som genererar ytan kan du titta på funktionen `rotationsyta` som ligger på kurshemsidan (förståelsen av hur ytan konstrueras får kanske vänta till kursen i flervariabelanalys.