1 Inledning

Vi skall se på numerisk beräkning av nollställen, extrempunkter, integraler och derivator. En del är säkert repetition, men en del är nytt. Det gäller att förstå vad vi gör för vi kommer bygga vidare på resultaten i senare laborationer.

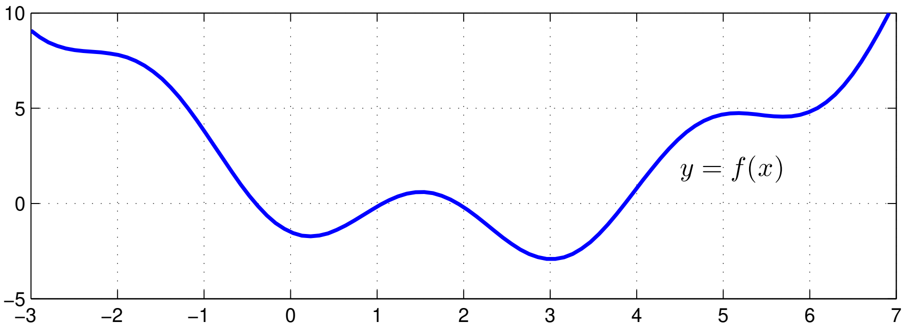
2 Nollställen

Låt $f : \mathbb{R} \to \mathbb{R}$ vara en deriverbar funktion. Vi skall lösa ekvationen $f(x) = 0$ med Newtons metod. Som exempel kan vi ta

$$f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5 = 0$$

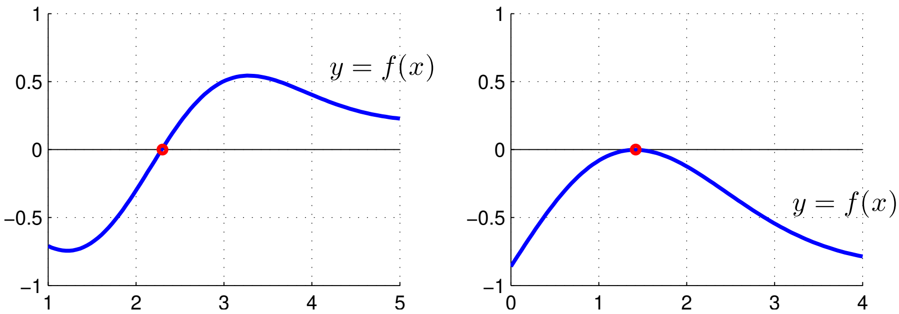
Vi börjar med att rita grafen till f för att få en uppfattning om hur många nollställen vi har och ungefär var de ligger.

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> x=linspace(-3,7);
>> plot(x,f(x))
```



Vi ser en lösning till $f(x) = 0$ som en punkt där grafen skär x-axeln. Vi kan grafiskt läsa av en första approximation av en lösning för att sedan förbättra denna med Newtons metod.

En lösning $\hat{x}$ till en ekvation $f(x) = 0$ kallas också ett *nollställe* till f. Ett nollställe $\hat{x}$ till $f(x) = 0$ kallas *enkelt* om $f'(\hat{x}) \neq 0$ annars kallas det *multipelt*.



Innan vi kan komma till Newtons metod måste vi dock först se på linjäriseringar av funktioner.

2.1 Linjärisering

Vi skall se på linjärisering av en differentierbar (deriverbar) funktion i en variabel $f : \mathbb{R} \to \mathbb{R}$. Linjäriseringen av f runt punkten a ges av

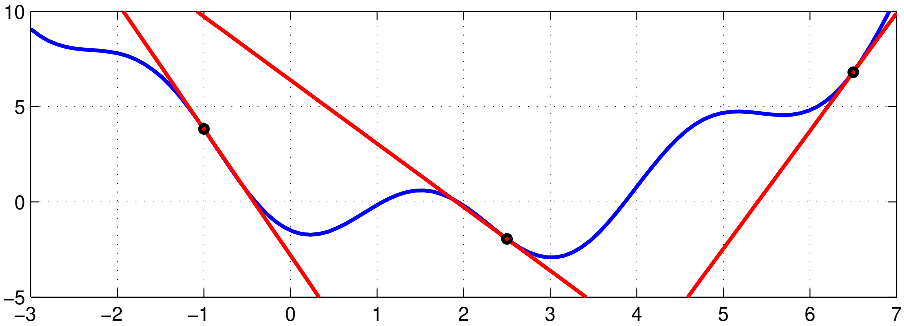
$$L(x) = f(a) + f'(a)(x - a)$$

Nära punkten a har vi $f(x) \approx L(x)$. Vi påminner oss om att den räta linjen $y = L(x)$ är tangenten till kurvan $y = f(x)$ vid a.

Som exempel linjäriserar vi $f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5$ runt $a = -1$, $a = 2.5$ och $a = 6.5$.

Vi har derivatan $f'(x) = x - 2 + 4\sin(2x)$ och vi ritar en figur

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> Df=@(x)x-2+4*sin(2*x);
>> L=@(x,a)f(a)+Df(a)*(x-a);
>> a=2.5;
>> plot(x,f(x),'b',x,L(x,a),'r')
```



Vi ser funktionskurvan tillsammans med tangenterna i a för de olika a-värdena.

2.2 Newtons metod

Antag att vi har en approximativ lösning x_k och vi vill hitta en bättre approximation x_{k+1}. Newtons metod går ut på att bilda linjäriseringen av f i x_k:

$$L(x) = f(x_k) + f'(x_k)(x - x_k)$$

och lösa $L(x) = 0$ istället för $f(x) = 0$,

$$f(x_k) + f'(x_k)(x - x_k) = 0 \tag{1}$$

Lösningen blir vår nästa approximation:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

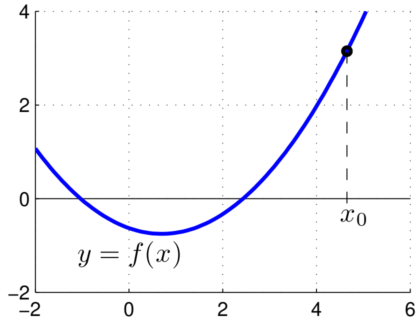
Som stoppvillkor för iterationen tar vi

$$|x_{k+1} - x_k| \leq tol$$

Det betyder att vi accepterar x_{k+1} om ändringen i sista iterationen är mindre än toleransen. Vi tillåter maximalt k_{max} iterationer ($k_{max} = 10$ är rimligt).

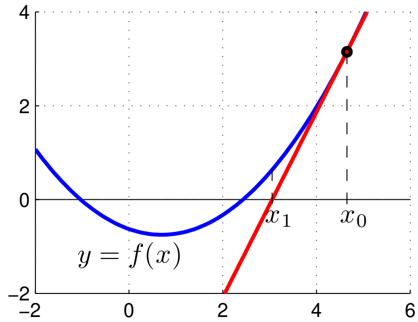
Geometriskt betyder (1) att vi följer tangenten och hittar x_{k+1} där den skär x-axeln. Vi ser på några steg med metoden:

Starta med en approximation x_0 av ett nollställe till $f(x) = 0$.



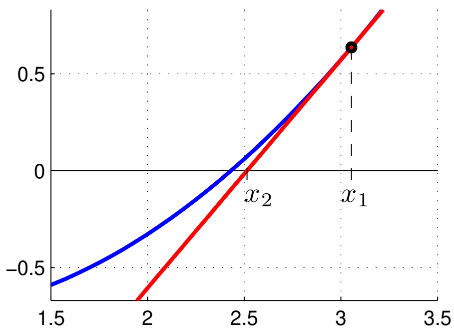
Bilda tangenten $y = f(x_0) + f'(x_0)(x - x_0)$ till f i $x = x_0$ och tag dess skärningspunkt med x-axeln som en bättre approximation

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$



Bilda tangenten $y = f(x_1) + f'(x_1)(x - x_1)$ i $x = x_1$ och tag dess skärningspunkt med x-axeln som en ännu bättre approximation

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$



Som exempel tar vi: Lös ekvationen $f(x) = 0$ där $f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5$. Grafen som vi redan ritat visar att vi har ett nollställe nära $x_0 = 4$ som vi tar som startapproximation.

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> Df=@(x)x-2+4*sin(2*x);
>> x=4;
>> kmax=10; tol=0.5e-8;
```

```
>> for k=1:kmax
       d=-f(x)/Df(x);
       x=x+d;
       disp([x d])
       if abs(d)<tol, break, end
   end
```

Utskriften av x_k och $d_k = x_{k+1} - x_k$ ger

```
   3.867224680594720  -0.132775319405280
   3.866407999346416  -0.000816681248304
   3.866407887464428  -0.000000111881988
   3.866407887464427  -0.000000000000002
```

För en tillräckligt bra startapproximationen x_0 konvergerar Newtons metod snabbt.

Uppgift 1. Låt $f(x) = x^3 - \cos(4x)$. Lös ekvationen $f(x) = 0$. Rita upp grafen av f för att se var ungefär lösningarna ligger. Hur många lösningar finns det? Läs av i grafiken en första approximation av en lösning för att sedan förbättra denna med Newtons metod. Rita ut lösningen med en liten ring. Upprepa tills du beräknat alla lösningar till ekvationen.

2.3 Konvergens

Nära ett nollställe $\hat{x}$ får vi ungefär en fördubbling av antalet korrekta decimaler i varje iteration med Newtons metod. Vi har s.k. *kvadratisk* konvergens.

Sats: Antag $\hat{x}$ är en lösning till ekvationen $f(x) = 0$ med $f'(\hat{x}) \neq 0$. Om Newtons metod

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \ k = 0, 1, \cdots$$

konvergerar mot $\hat{x}$ så gäller att

$$|x_{k+1} - \hat{x}| = \frac{|f''(\xi_k)|}{|2f'(x_k)|}|x_k - \hat{x}|^2$$

dvs. vi har kvadratisk konvergens.

Bevis: Taylorutveckling runt x_k ger att

$$f(\hat{x}) = f(x_k) + f'(x_k)(\hat{x} - x_k) + \frac{1}{2}f''(\xi_k)(\hat{x} - x_k)^2$$

Eftersom $f(\hat{x}) = 0$ så gäller

$$0 = f(x_k) + f'(x_k)(\hat{x} - x_k) + \frac{1}{2}f''(\xi_k)(\hat{x} - x_k)^2 \tag{1}$$

Newtons metod kan skrivas

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0 \tag{2}$$

Addition av (1) och (2) ger

$$f'(x_k)(x_{k+1} - \hat{x}) = \frac{1}{2}f''(\xi_k)(\hat{x} - x_k)^2$$

Om $f'(x_k) \neq 0$ så gäller

$$|x_{k+1} - \hat{x}| = \frac{|f''(\xi_k)|}{|2f'(x_k)|}|x_k - \hat{x}|^2$$

och vi har visat resultatet.

2.4 Färdiga program i Matlab

Verktygslådan Optimization Toolbox i Matlab har en funktion `fzero` som finner nollställen. Metoden i `fzero` är en modifiering av Newtons metod och används enligt något av alternativen

```
x=fzero(fun,x0)      x=fzero(fun,x0,opts)
```

där `fun` beskriver funktionen vi skall finna nollstället till, `x0` är en första approximation av nollstället eller ett intervall med teckenväxling som omsluter nollstället vi söker. I senare fallet kommer `fzero` garanterat finna en approximation av ett nollställe.

Alternativet med `opts` använder vi då vi t.ex. vill ange hur noggrant lösningen skall beräknas. Man skapar vektorn `opts` med funktionen `optimset`, se hjälptexten.

Vi använder `fzero` för att beräkna ett nollställe till $f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5$ för en sista gång.

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> x0=4;
>> x=fzero(f,x0)
x =
   3.866407887464427
```

Uppgift 2. Rita den slutna kurva som i polära koordinater ges av

$$r(\varphi) = \frac{2+\sin(3\varphi)}{\sqrt{1+\exp(\cos(\varphi))}}$$

För vilka vinklar skär kurvan enhetscirkeln? Använd `fzero`.

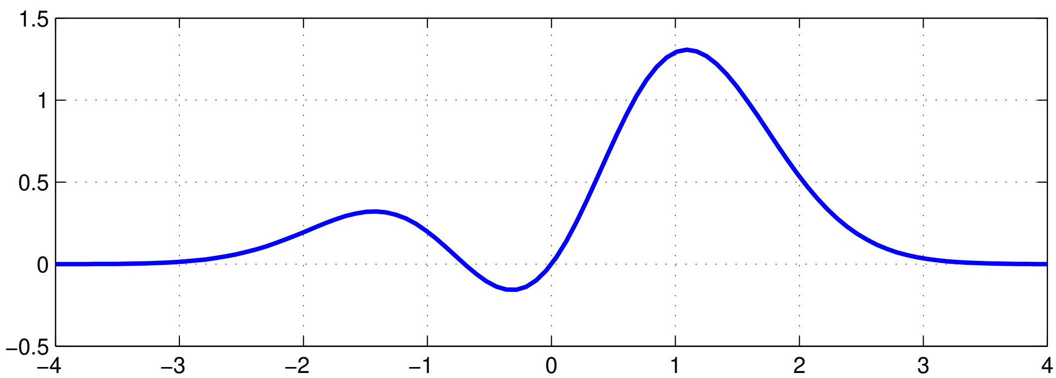
3 Extrempunkter

Vi skall bestämma max- och minpunkter till en funktion $f\colon \mathbb{R} \to \mathbb{R}$. Som exempel tar vi

$$f(x) = (x + 1.5x^2 + 0.1x^3)\exp(-0.7x^2)$$

Vi ritar upp grafen för att se var de stationära punkterna finns och av vilken typ de är.

```
>> f=@(x)(x+1.5*x.^2+0.1*x.^3).*exp(-0.7*x.^2)
>> x=linspace(-4,4);
>> plot(x,f(x))
```



Det ser ut som vi har tre stationära punkter, en lokal minpunkt (global minpunkt) och två lokala maxpunkter (varav en är en global maxpunkt)

En stationär punkt till $f(x)$ är en punkt a för vilken $f'(a) = 0$. Bestämningen av vilken typ av stationär punkt det är kan vi göra med hjälp av tecken på andraderivatan $f''(a)$.

I förra avsnittet linjäriserade vi en funktion $f(x)$ runt a för att beskriva grafens lutning, dvs. vi gjorde en linjär modell av funktionen runt a med

$$f(x) \approx L(x) = f(a) + f'(a)(x - a)$$

Nu vill vi ha en modell av $f(x)$ runt den stationära punkten a som beskriver hur grafen böjer sig, och då ger den linjära modellen inte tillräckligt med information.

En kvadratisk modell av funktionen $f(x)$ runt a kommer däremot att beskriva hur grafen böjer sig och den modellen ges av (Taylorutveckling runt a)

$$f(x) \approx f(a) + f'(a)(x - a) + \frac{1}{2}f''(a)(x - a)^2$$

Eftersom a är en stationär punkt så är $f'(a) = 0$ och vi får att

$$f(x) \approx f(a) + \frac{1}{2}f''(a)(x - a)^2$$

och vi kan avgöra vilken typ av stationär punkt vi har, genom att se på tecknet hos $f''(a)$.

Vill vi minimera eller maximera en funktion $f(x)$ som är deriverbar så söker vi lokala minpunkter eller maxpunkter, t.ex. genom att lösa ekvationen $g(x) = f'(x) = 0$. Sedan får vi kontrollera om dessa är lokala eller globala extrempunkter.

Det finns också andra metoder som söker lokalt minimum eller maximum till en funktionen. Vi beskriver inte dessa nu utan nöjer oss med att se vad MATLAB har för färdiga program.

3.1 Färdiga program i MATLAB

Verktygslådan OPTIMIZATION TOOLBOX i MATLAB har en funktion `fminbnd` som finner lokal minpunkt till en given funktion och används enligt något av alternativen

`x=fminbnd(fun,x1,x2)`   `x=fminbnd(fun,x1,x2,opts)`

där `fun` beskriver funktionen vi skall minimera, `x1` och `x2` ger ett intervall som omsluter minpunkten vi söker.

Alternativet med `opts` använder vi då vi t.ex. vill ange hur noggrant lösningen skall beräknas. Man skapar vektorn `opts` med funktionen `optimset`, se hjälptexten.

Som exempel ser vi på funktionen från förra exemplet. Från grafen till funktionen ovan ser vi att det finns en lokal minpunkt i intervallet $-0.5 \le x \le 0$ och lokala maxpunkter i intervallen $-1.6 \le x \le -1.2$ och $1 \le x \le 1.5$.

Vi bestämmer den lokala minpunkten med

```
>> x=fminbnd(f,-0.5,0)
x =
   -0.3183
```

Vi bestämmer nu de lokala maxpunkterna, genom att minimera $h(x) = -f(x)$, med

```
>> h=@(x)-f(x);
>> x=fminbnd(h,-1.6,-1.2)                >> x=fminbnd(h,1,1.5)
x =                                      x =
   -1.4268                                   1.0960
```

Uppgift 3. Bestäm samtliga extrempunkter till funktionen

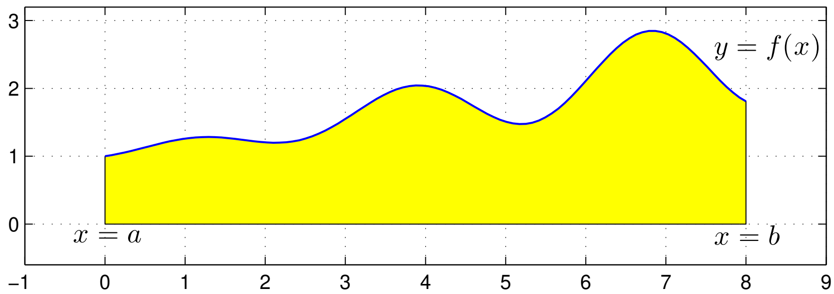
$$f(x) = \frac{1 + x^2 - 1.5x^3 + 0.5x^4}{1 + x^4}$$

Använd `fminbnd` på lämpligt sätt. Ange av vilken typ extrempunkterna är och beräkna eventuella max- och minvärden.

4 Integraler

Ibland kan man inte bestämma integraler exakt utan man får nöja sig med att beräkna approximationer. T.ex. $\int_0^1 \exp(x^2)\,dx$ 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. att 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 $x = b$.



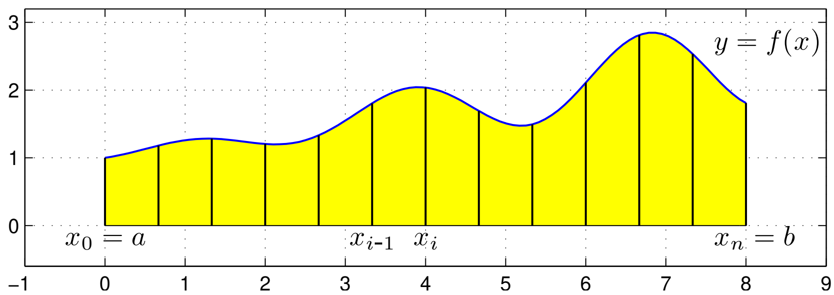
Vi gör en likformig indelning av intervallet $a \le x \le b$

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

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

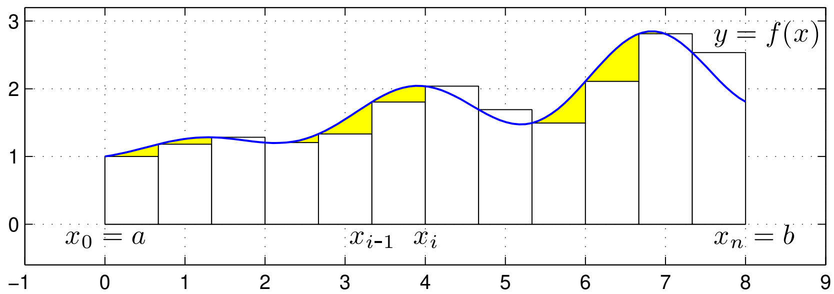
Sedan delar vi upp integralen i en summa av integraler över varje delintervall

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



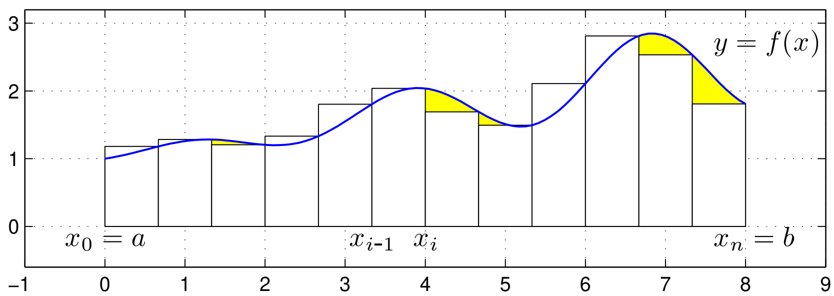
Om vi approximerar $f(x)$ med $f(x_{i-1})$ i intervallen $x_{i-1} \leq x \leq x_i$, dvs. beräknar $f(x)$ i vänster intervallgräns, får vi **vänster rektangelregel**

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



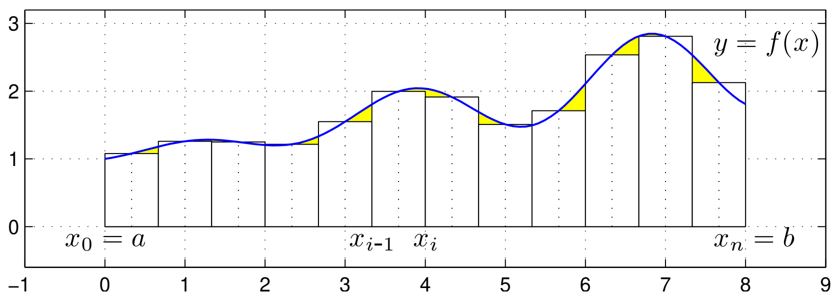
Om vi approximerar $f(x)$ med $f(x_i)$ i intervallen $x_{i-1} \leq x \leq x_i$, dvs. beräknar $f(x)$ i höger intervallgräns, får vi **höger rektangelregel**

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



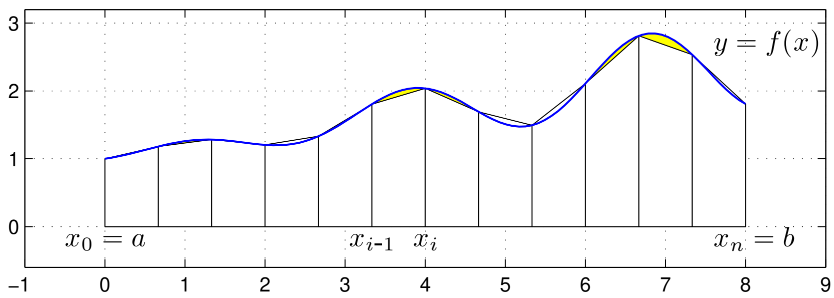
Om vi approximerar $f(x)$ med $f(m_i)$ i intervallen $x_{i-1} \leq x \leq x_i$, där m_i är mittpukterna i intervallen, får vi **mittpunktsmetoden**

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



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

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



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

Att använda en `for`-sats är oftast inte effektivt i MATLAB. Vi genererar hellre en vektor av alla funktionsvärden $f(x_i)$ och sedan summerar dessa enligt

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

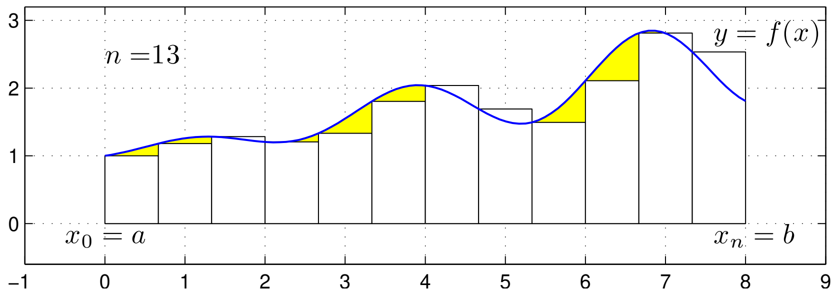
Detta sätt att organisera beräkningar kallas *vektorisering*, dvs. man genererar först en eller flera vektorer och utför sedan den önskade beräkningen på dem. De elementvisa operationerna `.*` `./` `.^` är exempel på vektoriserade operationer. Vi använde funktionen `sum` som snabbt summerar en vektor.

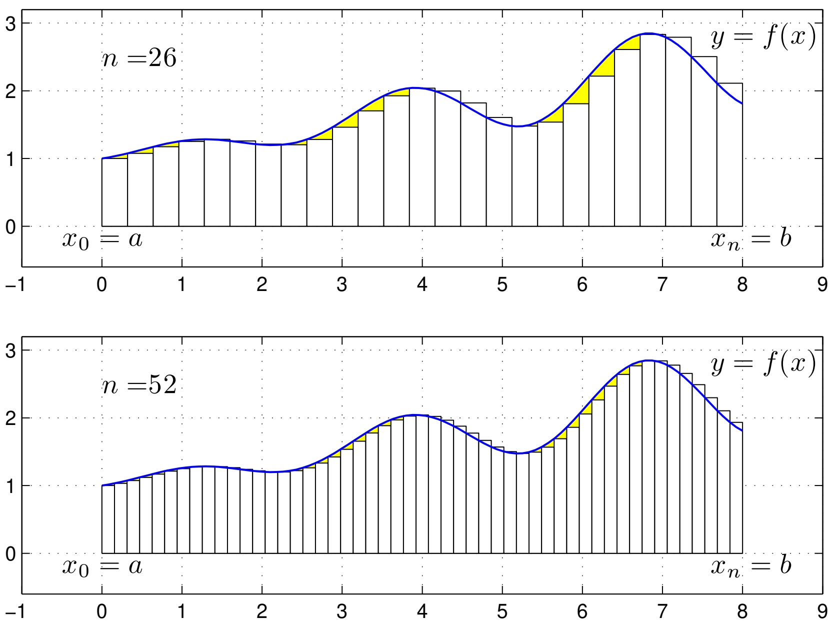
Uppgift 4. Beräkna en approximation av integralen $\int_0^1 x\ \sin(x)\,dx$ med vänster och höger rektangelregel samt mittpunkts- och trapetsreglerna. Använd `sum`.

4.1 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 däremot gäller att om vi fördubblar antal delintervall så delas felet med fyra, dvs. mycket bättre utdelning.

Sats: Antag $f(x)$ har kontinuerlig första derivata på intervallet $a \leq x \leq b$ med $|f'(x)| \leq K$. Givet en likformig indelning av intervallet: $a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$ med delintervallängd $\Delta x = x_i - x_{i-1} = \frac{b-a}{n}$. För vänster rektangelregel

$$L_n = \sum_{i=1}^{n} f(x_{i-1})\,\Delta x$$

gäller då att

$$\left| \int_a^b f(x)\,dx - L_n \right| \leq \frac{K(b-a)}{2}\,\Delta x$$

Bevis: Taylorutveckling runt $x = x_{i-1}$ ger

$$f(x) = f(x_{i-1}) + f'(\xi_i)(x - x_{i-1})$$

Integration ger

$$\int_{x_{i-1}}^{x_i} f(x)\,dx - f(x_{i-1})\Delta x = \int_{x_{i-1}}^{x_i} f'(\xi_i)(x - x_{i-1})\,dx$$

och efter summering har vi

$$\left| \int_a^b f(x)\,dx - L_n \right| = \left| \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f'(\xi_i)(x - x_{i-1})\,dx \right| \leq \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} |f'(\xi_i)(x - x_{i-1})|\,dx \leq$$

$$\leq \sum_{i=1}^{n} K \int_{x_{i-1}}^{x_i} |\, x - x_{i-1} \,|\, dx = n\, K\, \tfrac{1}{2}(\Delta x)^2 = \frac{K(b-a)}{2}\, \Delta x$$

Sats: Antag $f(x)$ har kontinuerlig andra derivata på intervallet $a \leq x \leq b$ med $|f''(x)| \leq K$. Givet en likformig indelning av intervallet: $a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$ med delintervallängd $\Delta x = x_i - x_{i-1} = \frac{b-a}{n}$. För mittpunktsmetoden

$$M_n = \sum_{i=1}^{n} f(m_i)\, \Delta x, \quad m_i = \frac{x_{i-1} + x_i}{2}$$

gäller då att

$$\left| \int_a^b f(x)\, dx - M_n \right| \leq \frac{K(b-a)}{24}\, (\Delta x)^2$$

Bevis: Taylorutveckling runt $x = m_i$ ger

$$f(x) = f(m_i) + f'(m_i)(x - m_i) + \frac{f''(\xi_i)}{2}(x - m_i)^2$$

Integration ger

$$\int_{x_{i-1}}^{x_i} f(x)\, dx - f(m_i)\Delta x = f'(m_i) \int_{x_{i-1}}^{x_i} (x - m_i)\, dx + \int_{x_{i-1}}^{x_i} \frac{f''(\xi_i)}{2}(x - m_i)^2\, dx =$$

$$= \tfrac{1}{2} \int_{x_{i-1}}^{x_i} f''(\xi_i)(x - m_i)^2\, dx$$

eftersom $\int_{x_{i-1}}^{x_i} (x - m_i)\, dx = 0$.

Efter summation har vi

$$\left| \int_a^b f(x)\, dx - M_n \right| = \left| \tfrac{1}{2} \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f''(\xi_i)(x - m_i)^2\, dx \right| \leq \tfrac{1}{2} \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} |f''(\xi_i)(x - m_i)^2|\, dx \leq$$

$$\leq n\, K\, \tfrac{1}{24}(\Delta x)^3 = \frac{K(b-a)}{24}\, (\Delta x)^2$$

där vi utnyttjat att $\int_{x_{i-1}}^{x_i} (x - m_i)^2\, dx = \frac{1}{12}\, (\Delta x)^3$.

Vi ser att felet i rektangelregeln är av storleksordningen Δx medan felet hos mittpunktmetoden är av storleksordningen $(\Delta x)^2$. Den senare är alltså mycket noggrannare och därmed effektivare.

4.2 Färdiga program i Matlab

Det finns funktioner i Matlab för att beräkna bestämda integraler både effektivt och noggrant, t.ex. finner vi `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.

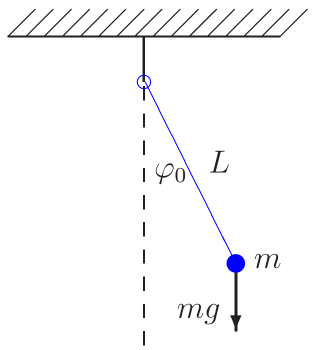
Genom att ge `name` som `'AbsTol'` eller `'RelTol'` kan vi ange önskad absolut respektive relativ noggrannhet i beräkningen och med `value` ger vi värdet på noggrannheten. Standardvärden är `1e-10` resp. `1e-6`, dvs. 1×10^{-10} resp. 1×10^{-6}.

Integranden skall beskrivas med elementvisa operationer, precis som om man skulle rita dess graf, `integral` kommer nämligen beräkna integranden för en vektor av x-värden.

Om vi skulle 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)
```

Uppgift 5. Den matematiska pendeln. En masspunkt med massan m hänger i en viktlös smal stav av längden L. Vi vill för olika begynnelseutslag φ_0 bestämma pendelns periodlängd.



Periodlängden ges av formeln

$$T(\varphi_0) = 4\sqrt{\frac{L}{g}} \int_0^{\pi/2} \frac{d\varphi}{\sqrt{1 - \sin^2(\varphi_0/2)\sin^2(\varphi)}}$$

där integralen är en s.k. elliptisk integral som inte kan beräknas exakt.

Låt $L = 0.1$ m och tag begynnelseutslagen $\varphi_0 = 10°, 30°, \cdots, 170°$. Beräkna en approximation av periodlängden för de olika begynnelseutslagen. Rita en graf av T som funktion av φ_0. Använd `integral`.

Lite ledning: Ibland vill man rita grafen av en funktion definierad genom $f(x) = \int_c^d h(t,x)\, dt$ över ett intervall $a \le x \le b$. Om det inte finns någon användbar primitiv funktion till integranden h är detta en relativt krävande uppgift, för varje x-värde som behövs för grafen måste vi beräkna $f(x)$ genom att beräkna en integral.

Man kan i MATLAB beräkna funktionen $f(x) = \int_c^d h(t,x)\, dt$ för en parameter x, som vi kan ge olika värden, enligt

```
>> f=integral(@(t)h(t,x),c,d)
```

Här förutsätts att `h` är en funktionsbeskrivning (funktionsfil eller anonym funktion med funktionshandtag) i de två variablerna `t` och `x`.

Skall vi nu rita en graf av $f(x)$ över $a \le x \le b$, så är här strukturen på en skriptfil för detta.

```
h=@(t,x)...;              % Integranden, om vi gör ett funktionshandtag
c=...; d...;              % Integrationsgränserna
a=...; b=...; n=...;      % Intervallgränser för grafen samt antal punkter
x=linspace(a,b,n);        % x-värdena
f=zeros(size(x));         % Fördimensionering. Skall fylla på integralvärdena

for i=1:length(x)
    f(i)=integral(@(t)h(t,x(i)),c,d);   % f(x(i))-värdena
end
plot(x,f)                 % Ritar grafen
```

Vi tittar närmare på `f=zeros(size(x))`. Här ger `size(x)` storleken på radvektorn `x`. Därmed ger `zeros(size(x))` en radvektor, lika stor som `x`, fylld med nollor. Nu kommer `f` vara en radvektor av rätt storlek och vi fyller på rätt $f(x_i)$-värden i `for`-satsen.

Dessa värden, dvs. $f(x_i) = \int_c^d h(t, x_i)\, dt$, beräknas med `f(i)=integral(@(t)h(t,x(i)),c,d)` där `@(t)h(t,x(i))` en anonym funktion med ett funktionshandtag. Här är t variabeln och $h(t, x_i)$ är funktionens värde, där x_i är ett konstant värde. Vi har alltså en funktion i en variabel `t` som `integral` kommer integrera.

5 Derivator

Vi kommer bl.a. i samband med lösning av differentialekvationer behöva approximera derivator $u'(x)$, $u''(x)$, $\cdots$, genom att använda flera funktionsvärden i närheten av x.

Inför *framåt-* och *bakåtdifferenskvoterna*

$$D_+u(x) = \frac{u(x+h) - u(x)}{h}, \quad D_-u(x) = \frac{u(x) - u(x-h)}{h}$$

Dessa kvoter användes för definition av derivator. Då lät vi $h \to 0$, nu kommer vi ta h som ett litet positivt tal.

Vi skall visa att

$$u'(x) = D_+u(x) + \mathcal{O}(h), \quad u'(x) = D_-u(x) + \mathcal{O}(h)$$

Taylorutveckling av $u(x)$ ger

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u^{(3)}(x) + \frac{h^4}{24}u^{(4)}(x) + \cdots \tag{1}$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u^{(3)}(x) + \frac{h^4}{24}u^{(4)}(x) - \cdots \tag{2}$$

Från (1) får vi direkt

$$D_+u(x) = \frac{u(x+h) - u(x)}{h} = u'(x) + \frac{h}{2}u''(x) + \cdots = u'(x) + \mathcal{O}(h)$$

och från (2) får vi

$$D_-u(x) = \frac{u(x) - u(x-h)}{h} = u'(x) - \frac{h}{2}u''(x) + \cdots = u'(x) + \mathcal{O}(h)$$

Även *centraldifferenskvoten*

$$D_0u(x) = \frac{u(x+h) - u(x-h)}{2h}$$

är en approximation av derivatan $u'(x)$.

Genom subtraktion av (1) med (2) kan man se att

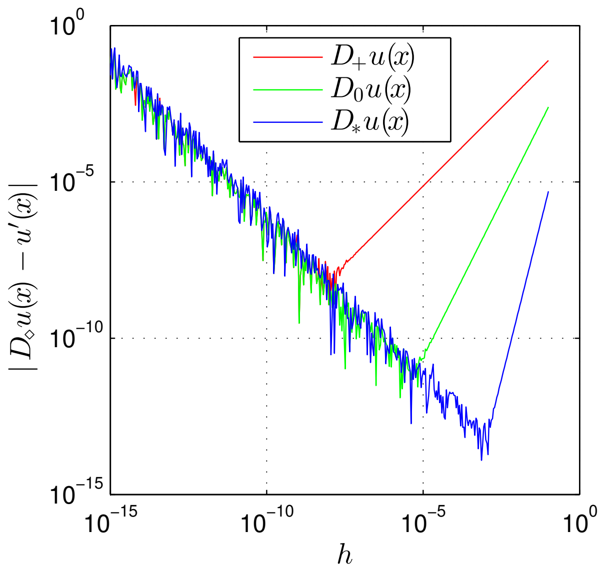
$$u'(x) = D_0u(x) + \mathcal{O}(h^2)$$

Vi ser att $D_0u(x)$ är noggrannare än $D_+u(x)$ och $D_-u(x)$.

Vid behov kan man med Taylorutveckling ta fram ännu noggrannare differenskvoter. t.ex.

$$D_*u(x) = \frac{-u(x+2h) + 8u(x+h) - 8u(x-h) + u(x-2h)}{12h}$$

Låt oss för $u(x) = \exp(x)$ approximera $u'(x)$ med differenskvoterna $D_+u(x)$, $D_0u(x)$ och $D_*u(x)$ för att jämföra approximationernas olika noggrannhetsordning. Vi ser på en följd av allt mindre h-värden och ser vilken noggrannhet vi får vid beräkning på dator.



Genom att rita i ett loglog-diagram kan vi se approximationsordningen. Vi har $\mathcal{O}(h)$, $\mathcal{O}(h^2)$ respektive $\mathcal{O}(h^4)$. Varför blir graferna ryckiga då h blir mycket litet?

Så här ritade vi

```
>> u=@(x)exp(x); uprim=@(x)exp(x); x=0.4;
>> h=logspace(-15,-1,400);
>> loglog(h,abs((u(x+h)-u(x))./h-uprim(x)),'r'), hold on
>> loglog(h,abs((u(x+h)-u(x-h))./(2*h)-uprim(x)),'g')
>> loglog(h,abs((-u(x+2*h)+8*u(x+h)-8*u(x-h)+u(x-2*h))./(12*h)-uprim(x)),'b')
>> axis square, grid on
```

5.1 Högre ordningens derivator

Vi kan approximera andra derivatan $u''(x)$ med

$$D_+D_-u(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

Addition av (1) med (2) ger

$$u''(x) = D_+D_-u(x) + \mathcal{O}(h^2)$$

Vidare kan vi approximera $u'''(x) = u^{(3)}(x)$ med

$$\frac{u(x+2h) - 2u(x+h) + 2u(x-h) - u(x-2h)}{2h^3}$$

samt $u''''(x) = u^{(4)}(x)$ med

$$\frac{u(x+2h) - 4u(x+h) + 6u(x) - 4u(x-h) + u(x-2h)}{h^4}$$

och får noggrannheten $\mathcal{O}(h^2)$ även i dessa fall.

Uppgift 6. Funktionen $f(x)$ ges av en tabell med mätdata (x-värdena är ekvidistanta, dvs. avståndet mellan på varandra följande x-värden är konstant)

x	0	0.0707	0.1414	$\cdots$
y	-1.5000	-1.6189	-1.6934	$\cdots$

som finns lagrade på data-filen `labtabell.mat` på kurshemsidan.

Hämta data-filen och ladda in i MATLAB med `load('labtabell')`. Rita en graf av funktionen.

Bestäm krökningen

$$\kappa(x) = \frac{f''(x)}{(1 + f'(x)^2)^{3/2}}$$

hos grafen till $f(x)$. Rita även grafen av krökningen.