Om vi får konvergens gäller, i vårt exempel, att $x = \cos x$, dvs. gränsvärdet är lösningen till en ekvation.

```
>> [x, cos(x), x - cos(x)]
ans = 7.3909e-01   7.3909e-01   0
```

Låt oss trycka x^2-knappen istället. Vi noterar först att om $x_0 \leq 0$ så är alla efterföljande värden ickenegativa. Det räcker att studera ickenegativa värden med andra ord.

Tre olika saker kan inträffa:

1. om $0 \leq x_0 < 1$, så konvergerar värdena mot 0.
   T.ex. $0.1, 0.01, 0.0001, \ldots$
2. $x_0 = 1$ medför att vi stannar i ett
3. $x_0 > 1$ medför att $x_k \to \infty$

Punkten ett är "repulsiv" i den meningen att oavsett hur nära vi startar ett (om vi inte startar precis i ettan) så stöts vi därifrån.

Nollan "attraherar". Om $|x_0| < 1$ så konvergerar följden mot noll.

Vi kommer att studera iterationer av typen $x_{k+1} = g(x_k)$
(inte enbart "knapptryckningsfunktioner").
Om x_k konvergerar, $x_k \to x^*$, gäller att $x^* = g(x^*)$.
Vi kallar g fixpunktsiteration och x^* fixpunkt.
Startar vi i en fixpunkt får vi tillbaks den.

Vi har två syften med de följande sidorna:

- givet en ekvation, $f(x) = 0$, hitta en fixpunktsiteration, g, som har en attraktiv fixpunkt, x^*, sådan att $f(x^*) = 0$.
- vi vill förstå vilka egenskaper hos g som ger konvergens

Newtons metod är en speciell fixpunktsiteration, ty

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad x_{k+1} = g(x_k) \quad \text{med} \quad g(x) = x - \frac{f(x)}{f'(x)}$$

Om Newtons metod konvergerar mot x^* gäller i gränsen att

$$x^* = x^* - \frac{f(x^*)}{f'(x^*)}$$

dvs $f(x^*) = 0$. Så fixpunkten är en lösning till vårt problem.

När konvergerar en fixpunktsiteration?
Dvs om det existerar x^* så att $x^* = g(x^*)$, när gäller att

$$\lim_{k\to\infty} |x_k - x^*| = 0 \ ?$$

Idé: konvergens medför att felet, $|x_k - x^*|$, minskar dvs.
$|x_{k+1} - x^*| < |x_k - x^*|$, så låt oss studera felet.

$$x_{k+1} - x^* = g(x_k) - x^* = g(x^* + x_k - x^*) - x^* =$$
$$g(x^*) + (x_k - x^*)g'(\theta_k) - x^* = g'(\theta_k)(x_k - x^*), \quad \theta_k \in (x_k, x^*)$$

Så,

$$|x_{k+1} - x^*| = |g'(\theta_k)|\ |x_k - x^*|$$

Ett steg till:

$$|x_{k+2} - x^*| = |g'(\theta_{k+1})|\ |x_{k+1} - x^*| = |g'(\theta_{k+1})|\ |g'(\theta_k)|\ |x_k - x^*|$$

Alltså:

$$|x_k - x^*| = |g'(\theta_{k-1})| \cdots |g'(\theta_1)|\ |g'(\theta_0)|\ |x_0 - x^*|$$

Så om det finns ett tal, λ, där alla $|g'(\theta_k)| \leq \lambda < 1$ får vi konvergens.

$$|x_k - x^*| \leq \lambda^k |x_0 - x^*|$$

Följande villkor garanterar konvergens:

- x_0 tillräckligt nära x^*
- g kontinuerligt deriverbar med $|g'(x^*)| < 1$

Den andra punkten medför att det existerar ett intervall, $I = [x^* - \delta, x^* + \delta]$ sådant att $|g'(x)| \leq \lambda < 1, x \in I$.

Om vi ser till att starta tillräckligt nära x^*, så stannar alla x_k kvar i intervallet. Detta medför att alla $\theta_k \in I$.

Första steget: Om $x_0 \in I$ så gäller att $\theta_0 \in I$, varför $|g'(\theta_0)| \leq \lambda$ vilket medför att $x_1 \in I$. Induktion!

Normalt linjär konvergens; ju mindre $|g'(x^*)|$ desto snabbare konvergens:

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|} \to |g'(x^*)|$$

Newton?

$$g(x) = x - \frac{f(x)}{f'(x)}$$

så

$$g'(x^*) = 1 - \frac{(f'(x^*))^2 - f''(x^*)f(x^*)}{(f'(x^*))^2} = 0, \quad \text{om } x^* \text{ enkelrot}$$

Innebär (minst) kvadratisk konvergens (inte att det konvergerar i ett steg).

Några exempel:

$g(x) = x^2$ har vi redan analyserat. $x_{k+1} = g(x_k)$ eller $x_{k+1} = x_k^2$.
Fixpunkter? $g(x^*) = x^*$ eller $(x^*)^2 = x^*$ så $x^* = 0$ eller $x^* = 1$.
Konvergens? $g'(x) = 2x$ och $g'(0) = 0$ så bättre än linjär konvergens $g'(1) = 2$ divergens.

$x_0 = 10^{-1}, x_1 = 10^{-2}, x_2 = 10^{-4}, \ldots$.

$g(x) = x/2$. Fixpunkter: $x^* = x^*/2$ så $x^* = 0$. Konvergens?
$g'(x^*) = 1/2$. Linjär konvergens: $x_0 = 1, x_1 = 1/2, x_2 = 1/4, \ldots$.

$g(x) = \cos x$. Fixpunkter: $x^* = \cos x^*$ så $x^* \approx 0.739$.
Konvergens? $g'(x^*) = -\sin x^*$ och $|-\sin x^*| \approx 0.674 < 1$
så linjär konvergens.

Lös $x^2 - 2 = 0$. Vi kan ju använda Newtons metod, men låt oss testa med omskrivningen $[x^2 - 2]/\alpha + x = x$ och tag $g(x) = [x^2 - 2]/\alpha + x$. Fixpunkterna är rötterna till ekvationen.
Konvergens? $g'(x) = 2x/\alpha + 1$. Tar vi t.ex. $\alpha = -3$ så får vi rätt snabb konvergens ty $|g'(\sqrt{2})| = |-2\sqrt{2}/3 + 1| \approx 0.05719$.

```
>> x = 1; for k=1:9, x(k+1)=x(k)-(x(k)^2 - 2) / 3; end
>> d = x - sqrt(2)  % editerat
d = -4.1e-01 -8.0e-02 -6.8e-03 -4.0e-04 -2.3e-05
    -1.3e-06 -7.5e-08 -4.3e-09 -2.4e-10 -1.4e-11

>> abs(d(2:end) ./ d(1:end-1))
ans = 1.9526e-01  8.4151e-02  5.9460e-02  5.7326e-02
      5.7199e-02  5.7191e-02  5.7191e-02  5.7191e-02
      5.7192e-02
```

Interpolation

Exempel:
Gymnasiet på "den gamla goda tiden", räknesticka och tabeller.

Vi vill beräkna $\sqrt{1.245}$ och har en tabell över
$y = \sqrt{t}$ där $t = 0, 0.01, 0.02, \ldots, 9.99, 10.00$.
y-värdena är givna med fem siffror.

Mer realistiskt, nu för tiden, vore en tabell, $t_k, y_k, k = 1, \ldots, n$, där vi av någon anledning inte kan beräkna $y(t)$ för alla t (mättekniska problem, gamla data).

Hur ska vi gå tillväga. I min skoltabell fanns röda tal mellan y-värdena, differenser, för att underlätta linjär interpolation

t	y
$\ldots$	
1.22	1.1045_{45}
1.23	1.1091_{46}
1.24	1.1136_{45}
1.25	1.1180_{44}
1.26	1.1225_{45}
1.27	1.1269_{44}

44 i 1.1180_{44} ska tolkas som $10^4(1.1180 - 1.1136)$. Så

$$\sqrt{1.245} \approx 1.1136 + \frac{1.245 - 1.24}{1.25 - 1.24} 0.0044 = 1.1158, \text{ fel} \approx -4.3 \cdot 10^{-6}$$

Andra tillämpningar som nyttjar interpolation är kvadratur (integration), lösning av randvärdesproblem, förenkling av funktioner, härledning av metoder (t.ex. sekantmetoden).
Allmänt har vi (t_k, y_k), $k = 1, \ldots, m$ med $t_1 < t_2 < \cdots < t_m$ och vill hitta en funktion (polynom i denna kurs), p, så att $p(t_k) = y_k$, $k = 1, \ldots, m$. Ibland lägger man dessutom krav på derivator, sk Hermite-interpolation.

Låt oss anta att det finns en bakomliggande funktion, f, (i exemplet $\sqrt{\ }$) som vi vill interpolera. Denna funktion är inte alltid känd.

Vi känner y_1, y_2 som är approximationer av f i två punkter $t_1 < t_2$, $y_1 = f(t_1) + \delta_1$ samt $y_2 = f(t_2) + \delta_2$ och vi vill approximera $f(t)$ där $t_1 < t < t_2$.

Vi bestämmer nu interpolationspolynomet, p, som uppfyller interpolationsvillkoren: $p(t_1) = y_1$ samt $p(t_2) = y_2$. Två villkor bestämmer en konstant- eller en linjär funktion så vi kräver att p har grad ≤ 1. Ansätt $p(t) = x_1 + x_2 t$, vilket ger följande linjära ekvationssystem:

$$\begin{cases} x_1 + x_2 t_1 = y_1 \\ x_1 + x_2 t_2 = y_2 \end{cases} \Rightarrow \begin{cases} x_1 = (t_2 y_1 - t_1 y_2)/(t_2 - t_1) \\ x_2 = (y_2 - y_1)/(t_2 - t_1) \end{cases}$$

så att

$$p(t) = \frac{t_2 y_1 - t_1 y_2}{t_2 - t_1} + \frac{y_2 - y_1}{t_2 - t_1} t = y_1 + (t - t_1)\frac{y_2 - y_1}{t_2 - t_1}$$

Observera att den andra omskrivningen direkt svarar mot tabellräkningen. Felet $p(t) - f(t)$ kan skrivas enligt:

$$p(t) - f(t) = f(t_1) + \delta_1 + (t - t_1)\frac{f(t_2) - f(t_1) + \delta_2 - \delta_1}{t_2 - t_1} - f(t) =$$

$$\underbrace{f(t_1) + (t - t_1)\frac{f(t_2) - f(t_1)}{t_2 - t_1}}_{p_f(t)} - f(t) + \underbrace{\delta_1 + (t - t_1)\frac{\delta_2 - \delta_1}{t_2 - t_1}}_{p_\delta(t)}$$

Låt oss införa de två polynomen p_f och p_δ sådana att $p_f(t_1) = f(t_1)$, $p_f(t_2) = f(t_2)$ resp. $p_\delta(t_1) = \delta_1$, $p_\delta(t_2) = \delta_2$. Då är tydligen $p = p_f + p_\delta$. Detta kan man direkt se från det linjära ekvationssystemet, lösningen x beror ju linjärt på högerledet.

$$p(t) - f(t) = p_f(t) + p_\delta(t) - f(t) = p_f(t) - f(t) + p_\delta(t)$$

De två delarna i felet kan tolkas som följer: $p_f(t) - f(t)$ anger hur väl p_f approximerar funktionsvärdena om de hade varit utan fel. $p_\delta(t)$ svarar mot felet i tabellvärden.

Låt oss nu uppskatta felet $e(t) = f(t) - p_f(t)$ (denna härledning kan rätt enkelt generaliseras till polynom av högre gradtal). Vi antar att $t \neq t_1, t_2$ ty $e(t_1) = e(t_2) = 0$. Inför

$$g(z) = e(z) - \frac{(z - t_1)(z - t_2)}{(t - t_1)(t - t_2)} e(t)$$

där vi betraktar t som en fix punkt, g beror alltså av z. Det gäller att $g(t_1) = g(t_2) = 0$ och dessutom är $g(t) = 0$. g har alltså tre distinkta nollställen varför, enligt medelvärdessatsen, $g'(z)$ har två distinkta nollställen. $g''(z)$ har alltså ett nollställe, kalla det $\theta \in (t, t_1, t_2)$ (det minsta intervall som innehåller t, t_1, t_2). Vi deriverar nu g (med avseende på z) och får (ty grad $p_f \leq 1$):

$$g''(z) = e''(z) - \frac{2\ e(t)}{(t - t_1)(t - t_2)} = f''(z) - \frac{2\ e(t)}{(t - t_1)(t - t_2)}$$

Eftersom $g''(\theta) = 0$ kan vi lösa ut $e(t)$ och får

$$e(t) = \frac{f''(\theta)}{2}\ (t - t_1)(t - t_2)$$

Antag att vi tar med fler punkter och interpolerar med ett polynom av högre gradtal. Kommer felet i approximationen att minska?

Vi kan först se på den allmänna satsen: Om p_f interpolerar f för de n t-värdena $t_1 < t_2 < \ldots < t_n$ så gäller att

$$f(t) - p_f(t) = \frac{f^{(n)}(\theta)}{n!}(t - t_1)(t - t_2)\cdots(t - t_n)$$

där $\theta \in (t, t_1, t_2, \ldots, t_n)$.

$n!$ ser lovande ut, men resten är inte så lätt att bedöma (θ känner vi inte t.ex.), så vi ser på vårt exempel i stället.

I exemplet kommer inte felet att minska eftersom det ser ut som:

$$p(t) - f(t) = p_f(t) - f(t) + p_\delta(t)$$

så även om vi kan göra $p_f(t) - f(t)$ mindre, så kommer $p_\delta(t)$, som svarar mot avrundningsfelet i tabellvärdena, att vara tämligen konstant, $10^{-5} - 10^{-6}$.

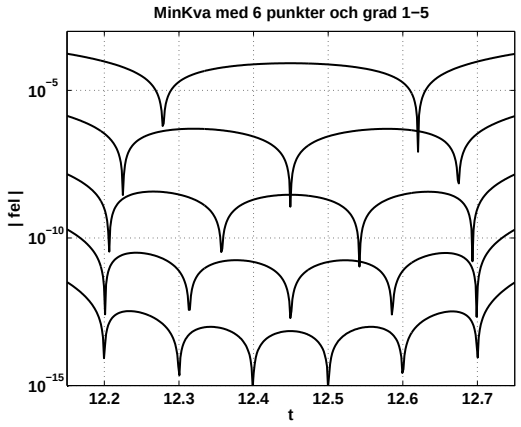
Situationen ändras om tabellvärdena hade givits med mindre fel, anta att $\delta_1 = \delta_2 = 0$. I exemplet hade då felet i approximationen varit 10^{-6} med två punkter (vårt förstagradspolynom), $\approx 10^{-8}$ med tre punkter (andragradspolynom), $\approx 10^{-10}$ med fyra punkter och $\approx 10^{-12}$ med fem punkter.

Observera att felet beror på hur t-punkterna ligger relativt den punkt där vi vill approximera f. I exemplet har jag lagt punkterna symmetriskt kring detta värde.

Så det kan löna sig att höja gradtalet förutsatt att tabellvärdena inte är behäftade med för stora fel. Polynom av höga gradtal är dock inte så lätthanterliga, mer om detta senare.

Kan vi använda polynomet för att extrapolera (gå utför $[t_1, t_n]$)? Vi vet att $|p(t)| \to \infty$ när $|t| \to \infty$ (om inte p är konstant), så det kan vara vanskligt. Polynom kan växa snabbt!

Hur passar minstakvadratanpassning in i detta sammanhang? Antag att vi anpassar mer än två (t_k, y_k)-punkter till ett förstagradspolynom. Kommer vi då att få en bättre approximation av det sökta värdet? Knappast. I bilden nedan har jag anpassat sex punkter, $t_k = 12.2 : 0.1 : 12.7$, ("exakta" y-värden) till polynom av grader 1-5 (när graden är fem har vi interpolation). Den lodräta axeln visar $|p_k(t) - f(t)|, k = 1, \ldots, 5$ (grad $p_k = k$).



Vi noterar att felet ökar utanför intervallet (extrapolation). Dipparna i felen svarar mot att polynomen interpolerar f i vissa punkter. I det sista fallet kräver vi detta, i de övriga fallen "råkar" det bli så. Approximationen ligger knappast helt på ena sidan om f eftersom felen kan minskas om approximationen skär f. Vi kan alltså notera att minstakvadrapolynomet p_1 svarar mot ett interpolationspolynom av grad ett. p_1 interpolerar dock inte f i något av t_j-värdena.

Varför använder man då minstakvadratanpassning? Problemställningen är ofta en annan. Modellen och antalet parametrar är givna och man vill få en så säker bestämning av parametrarna som möjligt. I polynomapproximationen ovan ändrade vi antalet punkter och därmed ändrades även antalet parametrar (koefficienter i polynomet).

Skulle man t.ex. ha två parametrar och endast använda två mätpunkter så får man en mycket osäker bestämning av parametrarna.

Det finns en annan typ av approximation där vi använder polynom men inte kräver interpolation. Säg att vi vill approximera $\sin t$ (används av olika programspråk, Java, Fortran, C/C++ etc.). På en Sundator är approximationskoden skriven i C (och åtkomlig via www) och den kompilerade koden finns i `libm`-biblioteket.

Först gör man argumentreduktion, man reducerar t så att det ligger i ett litet intervall kring origo (sin är ju periodisk). Ju kortare intervall man har desto enklare blir det att approximera.

Man kan också utnyttja att sin är udda. Det visar sig (se min FAQ på www) att det räcker att approximera $\sin t, t \in [0, \pi/4]$.

På detta intervall använder man ett polynom, av grad 13, och med enbart udda potenser. Koefficienterna, x_k, är valda så att

$$\max_{0\le t\le \pi/4} \left| \frac{\sin t}{t} - \left(1 + x_1 t^2 + x_2 t^4 + x_3 t^6 + x_4 t^8 + x_5 t^{10} + x_6 t^{12}\right) \right|$$

minimeras, ett sk minimax-problem. Man vill alltså minimera den maximala relativa avvikelsen.

Till skillnad från Taylorutveckling försöker man sprida ut felet över hela intervallet. Taylorutvecklingar har ett litet fel i den punkt där man gör utvecklingen. För att få ett litet fel över hela intervallet måste man då ta med onödigt många termer.

Några sätt att bestämma interpolationspolynomet

Det står polynomet i bestämd form, detta pga att det alltid existerar och är entydigt.

Interpolationsproblemet: hitta ett polynom p med grad högst $n-1$ sådant att $p(t_k) = y_k, k = 1, \ldots, n$, där alla (t_k, y_k) är givna och $t_1 < t_2 < \cdots < t_n$.

Låt oss anta att existensen är given och studera entydigheten. Antag att det finns ett annat polynom, q, av grad $\le n-1$ som interpolerar data. Det gäller då att $p(t_k) - q(t_k) = 0, k = 1, \ldots, n$, vilket säger att polynomet $p-q$ av grad $\le n-1$ har n distinkta nollställen. $p-q$ måste därför vara nollpolynomet och $p = q$.

Polynomet behöver inte alltid ha grad $n-1$. Om vi t.ex. väljer punkterna $y_k = t_k^2, k = 1, \ldots, 10$ så klarar vi oss med $p(t) = t^2$ fastän $n = 10$.

Nu till existensen. Den går att bevisa på flera olika sätt. Vi kommer att använda ett konstruktivt bevis. Antag att vi har $n = 3$ punkter. Här följer interpolationspolynomet på Lagranges form:

$$p(t) = y_1 \frac{(t-t_2)(t-t_3)}{(t_1-t_2)(t_1-t_3)} + y_2 \frac{(t-t_1)(t-t_3)}{(t_2-t_1)(t_2-t_3)} + y_3 \frac{(t-t_1)(t-t_2)}{(t_3-t_1)(t_3-t_2)}$$

En fördel med denna form på polynomet är att det är lätt att ställa upp och den kan vara användbar vid teoretiskt arbete. Formen lämpar sig dock inte så väl för numeriska beräkningar (många operationer). Det finns också risk för under- overflow om man inte tänker sig för.

Ett annat sätt att konstruera polynomet är att sätta upp ett ekvationssystem som vid gjorde i det linjära fallet. Så vi ansätter $p(t) = x_1 + x_2 t + x_3 t^2$. Interpolationsvillkoren ger då problemet:

$$\begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

I linjäralgebrakursen brukar man visa att en sådan matris, en Vandermonde-matris, är ickesingulär om alla t-värdena är distinkta.

Detta system är lätt att formulera men relativt dyrt att lösa (kubisk kostnad) men det finns snabbare metoder som utnyttjar matrisens speciella utseende. Det är dock billigt och stabilt att beräkna $p(t)$. Man använder normalt Horners metod för detta.

Exempel med $n = 4$:

$$x_1 + x_2 t + x_3 t^2 + x_4 t^3 = x_1 + t(x_2 + t(x_3 + t x_4))$$

Detta skrivs lämpligen i en loop, men har jag använt sekventiell kod: $p = x_4$, $p = x_3 + tp$, $p = x_2 + tp$, $p = x_1 + tp$. Detta kräver $n - 1 +$ respektive *.

Man kan se Vandermonde-härledningen som ett speciallfall av följande. Vi ansätter

$$p(t) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_{n-1}\phi_{n-1}(t) + x_n\phi_n(t)$$

ϕ_k kallas basfunktion och i Vandermonde-matrisen använde vi $\phi_k(t) = t^{k-1}$.

Ett problem med Vandermondematriser är att de kan bli illakonditionerade.

Exempel: Antag att $n = 4$ och tag t-värdena $0.1, 0.2, 0.3, 0.4$. Matrisen kan då skrivas:

$$\begin{bmatrix} 1 & 10^{-1} & 10^{-2} & 10^{-3} \\ 1 & 2\cdot 10^{-1} & 4\cdot 10^{-2} & 8\cdot 10^{-3} \\ 1 & 3\cdot 10^{-1} & 9\cdot 10^{-2} & 27\cdot 10^{-3} \\ 1 & 4\cdot 10^{-1} & 16\cdot 10^{-2} & 64\cdot 10^{-3} \end{bmatrix}$$

Konditionstalet är $\approx 2\cdot 10^3$. Anledningen till det stora konditionstalet är att basfunktionerna liknar varandra (kolonnerna blir nästan linjärt beroende).

Ett sätt att få ner konditionstalet är att använda andra basfunktioner. Låt oss ta bokens förslag.

$$\phi_k(t) = \left(\frac{t-(t_1+t_n)/2}{(t_n-t_1)/2}\right)^{k-1}$$

Den transformerade variabeln ligger i intervallet $[-1,1]$:

$$-1 \le \frac{t-(t_1+t_n)/2}{(t_n-t_1)/2} \le 1, \quad t \in [t_1, t_n]$$

Denna transformation leder till det nya konditionstalet ≈ 8 i vårt exempel.

Det finns ytterligare en vanlig framställning av interpolationspolynomet, nämligen Newtons form. Den är en kompromiss mellan de två tidigare. Det är relativt billigt både att konstruera polynomet och att sedan evaluera det. Dessutom är möjligt att lägga till nya punkter utan att börja om med polynomberäkningen.

Den allmänna formen är;

$p(t) = x_1 + x_2(t-t_1) + x_3(t-t_1)(t-t_2) + \cdots + x_n(t-t_1)(t-t_2)\cdots(t-t_{n-1})$

Låt oss se på specialfallet när $n = 3$.

$$p(t) = x_1 + x_2(t-t_1) + x_3(t-t_1)(t-t_2)$$

Vi får det undertriangulära ekvationssystemet:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & t_2-t_1 & 0 \\ 1 & t_3-t_1 & (t_3-t_1)(t_3-t_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

som ju är enkelt att lösa (framåtsubstitution). Vi ser också att det går att lägga till en punkt (en rad underst i matrisen) och vi behöver inte lösa systemet från början.

Exempel: Finn p som interpolerar $(1,1)$, $(2,4)$ samt $(3,11)$.

1) Vandermondes form. Vi ansätter $p(t) = x_1 + x_2 t + x_3 t^2$.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 11 \end{bmatrix}$$

som har lösning $x = [2, -3, 2]^T$ varför $p(t) = 2 - 3t + 2t^2$ eller $p(t) = 2t^2 - 3t + 2$.

2) Lagranges form:

$$p(t) = 1\frac{(t-2)(t-3)}{(1-2)(1-3)} + 4\frac{(t-1)(t-3)}{(2-1)(2-3)} + 11\frac{(t-1)(t-2)}{(3-1)(3-2)}$$

Förenklar vi detta uttryck får vi (givetvis) $p(t) = 2t^2 - 3t + 2$.

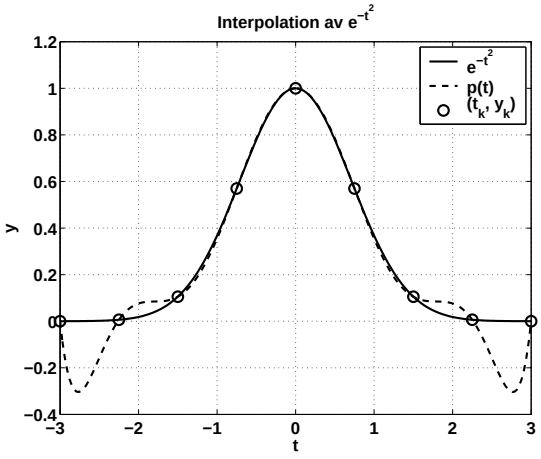
3) Newtons form: $p(t) = x_1 + x_2(t-1) + x_3(t-1)(t-2)$. Lös:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 2-1 & 0 \\ 1 & 3-1 & (3-1)(3-2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 11 \end{bmatrix}$$

så att $x = [1, 3, 2]^T$ varför $p(t) = 1 + 3(t-1) + 2(t-1)(t-2)$ som också kan förenklas till $p(t) = 2t^2 - 3t + 2$.

Problem med interpolation

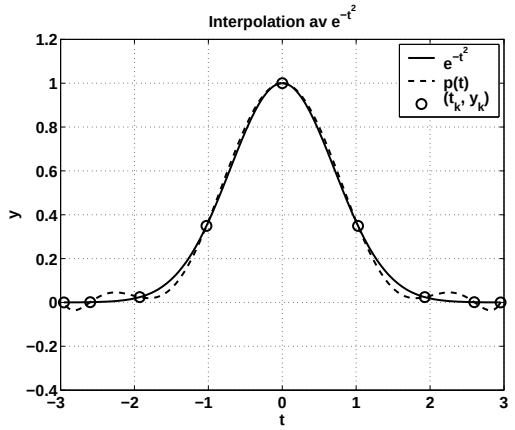
I följande exempel interpolerar $p(t)$, $f(t) = e^{-t^2}$ i nio punkter.



Det stämmer inte bra och det är stora fel i intervallets ändar. För vissa funktioner accentueras detta fenomen (Runges fenomen) när vi ökar antalet punkter (p behöver inte alltid konvergera mot f utan felet kan öka med ökande antal punkter).

Det är inte ovanligt att polynom av högt gradtal svänger kraftigt när man använder ekvidistant interpolation (samma avstånd mellan t_k-värdena).

Vi kan försöka att "hålla nere" polynomet i ändarna genom att lägga punkterna tätare där. Här har jag också använt nio punkter, men de ligger tätare mot intervallets ändpunkter. Polynomet svänger avsevärt mindre.



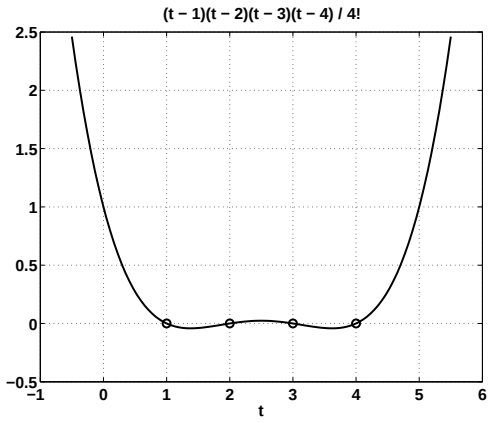
Vad är ett bra sätt att välja punkterna (om vi får välja)? Låt oss studera felets utseende igen (vi kan tänka oss exakta data, så att $p_f = p$):

$$f(t) - p(t) = \frac{f^{(n)}(\theta)}{n!}(t-t_1)(t-t_2)\cdots(t-t_n)$$

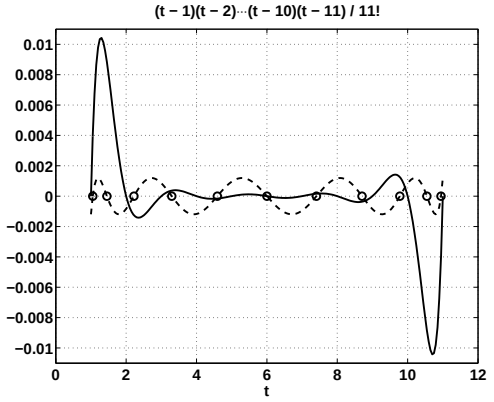
där $\theta \in (t, t_1, t_2, \ldots, t_n)$. Antag att $|f^{(n)}(\theta)| \le M$ för alla $\theta \in (t, t_1, t_2, \ldots, t_n)$. Vi har då

$$|f(t) - p(t)| \le \frac{M}{n!}\,|(t-t_1)(t-t_2)\cdots(t-t_n)|$$

Låt oss specialstudera funktionen $(t-t_1)(t-t_2)\cdots(t-t_n)/n!$. Den växer snabbt utanför $[t_1, t_n]$. I bilden på nästa sida är $n = 4$ och sedan 11. Extrapolation är farligt.



Den kan vara orolig inom intervallet också:



Den heldragna kurvan, i andra bilden på föregående sida, svarar mot ekvidistanta punkter den streckade (bättre) utnyttjar Chebyshevpunkterna. Dessa punkter har egenskapen att göra det maximala värdet av $|(t-t_1)(t-t_2)\cdots(t-t_n)|$ så litet som möjligt.

Sats:

$$\max_{-1\le t\le 1} |(t-t_1)(t-t_2)\cdots(t-t_n)|$$

minimeras då

$$t_k = -\cos\left[\frac{(2k-1)\pi}{2n}\right],\ k=1,2,\ldots,n$$

Det maximala värdet på $|(t-t_1)(t-t_2)\cdots(t-t_n)|$ är då $1/2^{n-1}$.

När t ligger i ett annat intervall, $[\alpha,\beta]$ säg får vi göra en linjär avbildning av Chebyshevpunkterna till detta intervall. Vi ser att

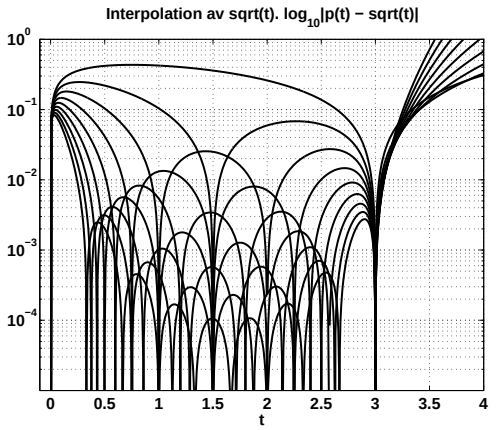
$$\frac{\beta-\alpha}{2}[-1,\ 1]+\frac{\alpha+\beta}{2}=[\alpha,\ \beta]$$

så de transformerade Chebyshevpunkterna blir

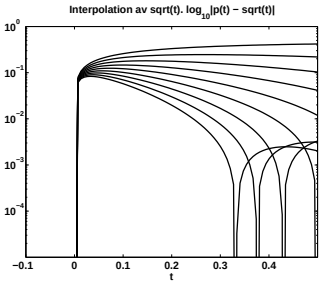
$$-\frac{\beta-\alpha}{2}\cos\left[\frac{(2k-1)\pi}{2n}\right]+\frac{\alpha+\beta}{2}$$

I bland är det ändå problem. Det kan tänka sig att M, begränsningen av $|f^{(n)}(\theta)|$ ej existerar. Exempel: $f(t)=\sqrt{t}$ på intervallet $[0,3]$. Redan $f'(0)$ är ju obegränsad, man säger att derivatan har en singularitet. I vissa fall visar sig singulariteten först i högre derivator (ex $f(t)=t^{5/2}$).

På nästa sida visas felet vid interpolation av $\sqrt{t}, t\in[0,3]$ för ökande n. Chebyshevpunkter ger obetydligt bättre resultat.



Här inzoomat:



Anledningen till att det inte konvergerar vid $t=0$ är att $\sqrt{t}$ där har lodrät tangent, något ett polynom aldrig kan ha.

Om en funktion har $n+1$ antal kontinuerliga derivator så kan den utvecklas i en Taylorutveckling:

$$f(t)=f(a)+\frac{f'(a)}{1!}(t-a)+\frac{f''(a)}{2!}(t-a)^2+\cdots+\frac{f^{(n)}(a)}{n!}(t-a)^n+R(t)$$

där resttermen $R(t)=c(\xi)(t-a)^{n+1}, \xi\in(a,t)$ och $|c(\xi)|$ är uppåt begränsad. Detta innebär att en sådan funktion (som har Taylorutveckling) liknar ett polynom på ett tillräckligt litet intervall.

Om inte alla $f^{(k)}(a)=0, k=0,\ldots,n$ kan vi göra $R(t)$ godtyckligt liten jämfört med resten av Taylorutvecklingen, genom att ta $|t-a|$ tillräckligt litet. På ett stort intervall behöver inte funktionen likna ett polynom.

$\sqrt{t}$ har ingen Taylorutveckling kring $a=0$. Däremot har ju $\sqrt{t}$ en utveckling kring alla $a>0$ och det är inga problem att approximera funktionen för positiva t.

Man kan naturligtvis approximera med annat än polynom. Exempel: Approximera $f(t)=(\sin t-1)/(\cos t-1)$ kring $t=0$. Problem, i detta fall har ju f (inte bara derivatorna) en singularitet. Kan använda rationell approximation (Padé).

$$\frac{\sin t-1}{\cos t-1}=\frac{12-12t+t^2+t^3}{6t^2}+R(t),\quad R(t)=\frac{t^2}{120}+\cdots$$

Så för $t\approx 0$ och med $r(t)=(12-12t+t^2+t^3)/(6t^2)$:

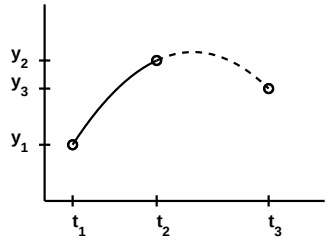
$$\left|\frac{f(t)-r(t)}{f(t)}\right|=\left|\frac{R(t)}{f(t)}\right|\approx\frac{t^4}{240}$$

Ett annat alternativ är att använda en generaliserad potensserie:

$$f(t)=\frac{2}{t^2}-\frac{2}{t}+\frac{1}{6}+\frac{t}{6}+\frac{t^2}{120}+\frac{t^3}{360}+\frac{t^4}{3024}+\cdots$$

Splinefunktioner

Polynom av höga gradtal är svårhanterliga men har samtidigt lokalt goda approximationsegenskaper. Dessutom är ju polynom "trevliga" funktioner. Enkla att beskriva, lagra, beräkna, integrera, derivera etc. Så en vanlig kompromiss är styckvisa polynom av låga gradtal. Man behåller polynomens enkelhet men slipper svängningarna.



I bilden ovan är den heldragna linjen ett polynom och den streckade ett annat. Heldragen plus streckad kurva tillsammans utgör dock inte (nödvändigtvis) ett polynom.

Def: En interpolerande splinefunktion av grad j är en funktion som interpolerar $(t_k, y_k), k = 1, \ldots, n$ och som består av styckvisa polynom, på intervallen $[t_1, t_2], [t_2, t_3], \ldots$
Dessutom är splinefunktionen $j-1$ gånger kontinuerligt deriverbar i knutpunkterna (dvs. i (t_k, y_k)).

Det är inga problem med kontinuiteten av derivatorna av varje enskilt polynom (i varje delintervall).

Om $j = 1$ så har vi ingen kontinuerlig derivata utan bara kontinuitet hos splinefunktionen.
Delpolynomen har högst grad ett.

Om $j = 2$ så är delpolynomen (högst) andragradspolynom. Splinefunktionen är kontinuerlig och är kontinuerligt deriverbar (förstaderivatan är kontinuerlig).

Det vanligaste är dock $j = 3$, kubiska splines, där delpolynomen är kubiska (högst) och splinefunktionen är kontinuerlig liksom dess första- och andraderivator.

Låt oss se varför detta verkar vara möjligt att åstadkomma och varför man inte kan kräva kontinuerlig tredjederivata.

En kubisk spline kan skrivas $p_k(t) = a_k t^3 + b_k t^2 + c_k t + d_k$ på intervallet $[t_k, t_{k+1}]$. Antag att vi har n stycken t-värden. Detta ger $n-1$ intervall (lika många polynom), så antalet obestämda koefficienter är $4(n-1)$. Hur många villkor har vi?

Interpolationskravet ger $2(n-1)$ villkor (ty varje polynom måste interpolera 2 knutpunkter). Detta ger oss kontinuiteten gratis.

Kontinuerlig förstaderivata ger $n-2$ villkor (inre punkter) och lika många för andraderivatan. Så summa $2(n-1)+n-2+n-2 = 4n-6$ villkor.

Det innebär att vi saknar två villkor som måste bestämmas på något sätt. Här är några vanliga tilläggsvillkor
(s är splinefunktionen):

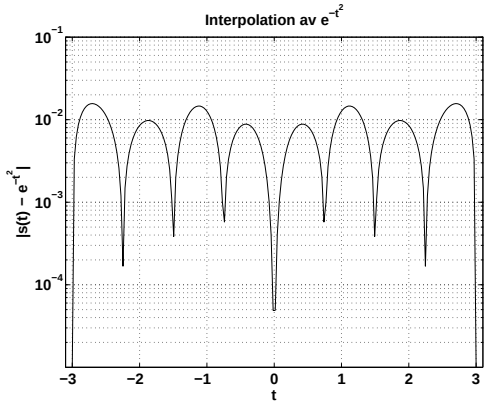
$s''(t_1) = s''(t_n) = 0$ sk naturliga splines (minimerar $\int_{t_1}^{t_n} (s''(t))^2\, dt$)

$s'(t_1) = f'(t_1)$ och $s'(t_n) = f'(t_n)$ komplett spline

$s'(t_1) = s'(t_n)$ samt $s''(t_1) = s''(t_n)$ periodisk första- och andraderivata (kanske rimligt med $y_1 = y_n$ i detta fall).

$p_1(t) = p_2(t), t \in [t_1, t_3]$ och $p_{n-2}(t) = p_{n-1}(t), t \in [t_{n-2}, t_n]$, not-a-knot; medför att s''' kontinuerlig i $t = t_2$ och $t = t_{n-1}$). Det är alltså ett tredjegradspolynom i $[t_1, t_3]$ (och ett (annat) i $[t_{n-2}, t_n]$).

Om vi återvänder till e^{-t^2}-exemplet har vi inget problem att göra en bra approximation med kubiska splines. Jag har ritat felet snarare än de två kurvorna, eftersom de ligger så nära varandra.



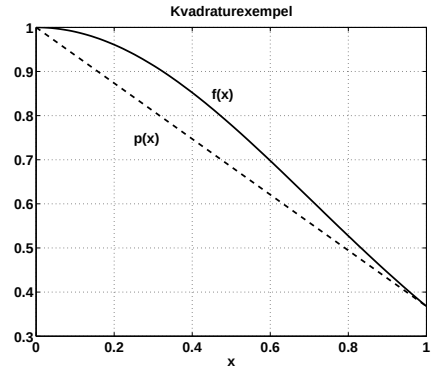
Om $f^{(4)}$ är begränsad (över intervallet) så konvergerar splinefunktionen mot f med hastigheten $\max_k h_k^4$.

Kvadratur - numerisk integration

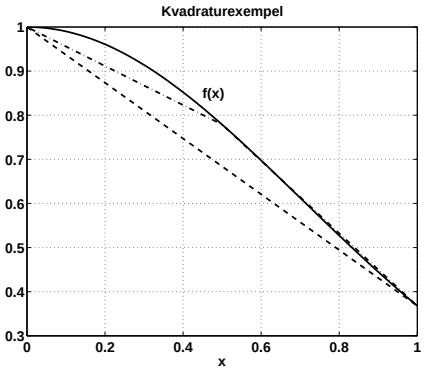
Vill beräkna: $\int_a^b f(x)\, dx$. Inte alltid möjligt att uttrycka en primitiv funktion i elementära funktioner (inte alltid bekvämt heller).

Grundidé: approximera $f(x)$ med en funktion $p(x)$ som har bra approximationsegenskaper, och som är enkel att beräkna och integrera.
Enkelt exempel: vi vill approximera $\int_0^1 e^{-x^2}\, dx$.
Facit: $\int_0^1 e^{-x^2}\, dx \approx 0.74682413281$.

Approximera $f(x)$ med en linjär funktion, $p(x) = 1 + (e^{-2} - 1)x$.



Från bilden framgår att vår approximation måste vara rätt dålig, 0.68394. Låt oss dela upp integrationsintervallet i två delintervall, $[0, 0.5], [0.5, 1]$ och approximera med en linjär funktion på varje delintervall:



Den andra halvan borde stämma rätt bra, absoluta felet är ≈ 0.001. Det är fortfarande rätt stort fel i det vänstra intervallet. Approximationen av integralen är nu: 0.73137. Vi kan fortsätta med att halvera intervallen, men det verkar lite bortkastat att fortsätta med högra halvan. Vi vill ha en adaptiv metod som försöker anpassa sig till felet.

Från bilden ser man att approximationen kommer att konvergera mot det exakta värdet (om vi bortser från avrundningsfel).

Ett annat alternativ är att approximera med ett polynom av högre gradtal. Om vi integrerar interpolationspolynomet, av grad fyra, som interpolerar e^{-x^2} för $x = 0, 0.25, 0.5, 0.72, 1$ blir felet i integralen $\approx 10^{-5}$.

Mer om Trapetsmetoden

Trapetsmetoden: approximation av f med ett linjärt interpolationspolynom på varje delintervall. På intervallet $[a, b]$ approximerar vi integralen med arean av en parallelltrapets (därav namnet):

$$\int_a^b f(x)\ dx \approx \frac{h}{2}(f(a) + f(b)), \quad h = b - a$$

Vi delar nu in $[a, b]$ i $n-1$ lika långa delintervall (en del författare börjar med x_0):

$$x_k = a + (k-1)h, \quad k = 1, \ldots, n, \quad h = (b-a)/(n-1)$$

så att $x_1 = a$ och $x_n = b$.

Beteckna den approximation vi får med $T_n(f)$. Den blir:

$$\frac{h}{2}\left[(f(x_1) + f(x_2)) + (f(x_2) + f(x_3)) + \ldots + (f(x_{n-1}) + f(x_n))\right] =$$

$$h\left[\frac{f(x_1)}{2} + f(x_2) + f(x_3) + \ldots + f(x_{n-1}) + \frac{f(x_n)}{2}\right]$$

Gör man ovanstående i vårt exempel verkar felet ha utseendet ch^2. Kan man bevisa det?

Från interpolationsteorin vet vi att:

$$f(x) - p(x) = \frac{f''(\theta_x)}{2}(x-a)(x-b), \quad \theta_x \in (a, b)$$

med ett intervall. Alltså

$$\int_a^b f(x)\ dx - \int_a^b p(x)\ dx = \int_a^b \frac{f''(\theta_x)}{2}(x-a)(x-b)\ dx =$$

$$\frac{f''(\xi)}{2}\int_a^b (x-a)(x-b)\ dx = -\frac{(b-a)^3\ f''(\xi)}{12}, \quad \xi \in (a, b)$$

Detta följer av integralkalkylens medelvärdessats ($(x-a)(x-b)$ byter inte tecken på $[a, b]$). I det allmänna fallet, med $n-1$, intervall får vi summera felen:

$$\int_a^b f(x)\ dx - T_n(f) = -\sum_{k=1}^{n-1} \frac{(x_{k+1} - x_k)^3\ f''(\xi_k)}{12} = -\frac{h^3}{12}\sum_{k=1}^{n-1} f''(\xi_k)$$

Om vi antar att f'' är kontinuerlig så antar f'' min/max på $[a, b]$ så att

$$\min_{a \le x \le b} f''(x) \le \frac{1}{n-1}\sum_{k=1}^{n-1} f''(\xi_k) \le \max_{a \le x \le b} f''(x)$$

så att (en kontinuerlig funktion antar mellanliggande värden):

$$\frac{1}{n-1}\sum_{k=1}^{n-1} f''(\xi_k) = f''(\xi)$$

Alltså:

$$\int_a^b f(x)\ dx - T_n(f) = -\frac{h^3(n-1)f''(\xi)}{12} = -\frac{(b-a)h^2 f''(\xi)}{12}, \ \xi \in [a, b]$$

ty $h(n-1) = b - a$.

Så om andraderivatan är begränsad i $[a, b]$ och om vi räknar exakt gäller att $T_n(f) \to \int_a^b f(x)\ dx, \ n \to \infty$.

Observera att om man inte vet något om hur f'' ser ut kan man inte garantera konvergens.

Det är enkelt att lura avbrottskriteriet i kvadraturprogram. Det enda vi känner är ju $(x_k, f(x_k)), k = 1, \ldots, n$ men det finns oändligt många funktioner som interpolerar dessa punkter (med olika värden på integralen).
Detta är ett allmänt beräkningsproblem (ändliga punktmängder från oändliga punktmängder).

Newton-Cotes-kvadratur

Man kan generalisera trapetsmetoden. Att integrera interpolationspolynom ger Newton-Cotes metoder. Man skiljer mellan öppna metoder där ändpunkterna ej är med resp. slutna, där ändpunkterna tas med.
Enklaste metoden är mittpunktsmetoden (rektangelmetoden) där vi approximerar $f(x)$ med $f((x_k + x_{k+1})/2)$ i intervallet $[x_k, x_{k+1}]$. Så om vi bara ser på intervallet $[a, b]$ så har vi approximationen:

$$\int_a^b f(x)\ dx \approx (b-a)f\left(\frac{a+b}{2}\right)$$

Vi har tittat på trapetsmetoden där man använder en linjär approximation. Använder man en kvadratisk approximation får man Simpsons formel:

$$\int_a^b f(x)\ dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Om man härleder felen för de sammansatta metoderna (mer än ett intervall) har mittpunktsmetoden felet

$$(b-a)h^2 f''(\xi)/24$$

vilket lustigt nog är mindre än för trapetsmetoden som ju har högre ordning på polynomet.

Dessutom har både mittpunkts- och trapetsmetod polynomiellt gradtal ett (exakt för alla polynom upp till och med grad ett). Detta beror på att vi inte primärt är intresserade av att approximera f (då är normalt en allmän linjär funktion bättre än en konstant) utan att vi vill approximera en integral.

En linjär approximation av t.ex. $f(x) = x$ över $[-1, 1]$ ger felet noll och en exakt integral. Approximationen av samma funktion med $f(0) = 0$ ger stora fel i funktionsanpassningen men en exakt integral pga att approximationsfelen i integralen precis tar ut varandra.

Simpsons formel, som också har ett udda antal punkter (jämn grad på polynomet), har felet $(b-a)h^4 f^{(4)}(\xi)/180$ som också uppvisar mindre fel än först förväntat (tre punkter ger h^4 och $f^{(4)}$).

Allmänt kan en kvadraturmetod skrivas

$$\int_a^b f(x)\ dx \approx \sum_{k=1}^{n} w_k f(x_k)$$

w_k kallas vikter och x_k abscissor.

Hur ser Simpsons formel ut på mer än ett intervall? Dela in $[a,b]$ i sex lika långa delintervall där vi använder metoden på: $[x_1,x_3]$, $[x_3,x_5]$ och $[x_5,x_7]$.

$$\int_{x_1}^{x_3} f(x)\ dx + \int_{x_3}^{x_5} f(x)\ dx + \int_{x_5}^{x_7} f(x)\ dx \approx$$
$$\frac{x_3-x_1}{6}\left[f(x_1)+4f\left(\frac{x_1+x_3}{2}\right)+f(x_3)\right] +$$
$$\frac{x_5-x_3}{6}\left[f(x_3)+4f\left(\frac{x_3+x_5}{2}\right)+f(x_5)\right] +$$
$$\frac{x_7-x_5}{6}\left[f(x_5)+4f\left(\frac{x_5+x_7}{2}\right)+f(x_7)\right]$$

$\frac{x_1+x_3}{2} = x_2$ etc. och $h = x_{k+1} - x_k$ så att approximationen blir:

$$\frac{2h}{6}[f(x_1)+4f(x_2)+2f(x_3)+4f(x_4)+2f(x_5)+4f(x_6)+f(x_7)]$$

eftersom ändpunkterna i delintervallen sammanfaller parvis.

Testar man detta på $f(x)=e^{-x^2}$ och kräver ett absolut fel $\le 1.2\cdot 10^{-9}$ tar trapetsmetoden 7150 funktionsevalueringar, mittpunktsmetoden 5055 och Simpsons formel 52. Matlabs `quadl`, som är adaptiv, tar 18.

Det spelar alltså stor roll vilken metod man använder och h^m-faktorn är mycket viktig. För att exemplifiera, låt oss anta att vi har en uppsättning metoder med feltermer:

$$c\ (b-a)\ h^m\ f^{(m)}(\xi)$$

där c är en konstant och m ett postivt heltal som varierar mellan metoderna. $h = 1/(n-1)$ som vanligt. Om $f^{(m)}(\xi)$ är konstant (inte sannolikt) kan felet skrivas Ch^m för en annan konstant C. För att feltermen skall bli $\approx \tau$, en given tolerans, krävs alltså:

$$Ch^m \approx \tau,\quad n \approx \frac{1}{(\tau/C)^{1/m}},\quad n \propto \frac{1}{\tau^{1/m}}$$

Med $\tau = 10^{-9}$ och $C=1$ så får vi denna tabell:

m	$\propto n$
2	31623
3	1000
4	178
5	63
6	32
7	19

Om någon av f:s lägre derivator har en singularitet i $[a,b]$ kan dock metoderna konvergera avsevärt långsammare.

Exempel:
Trapetsmetoden på $f(x)=x^p$, $0<p<1$, $[a,b]=[0,1]$.

Vi kan ej använda feluppskattningen på hela intervallet eftersom f' och f'' har en singularitet i nollan. Vi kan dock räkna ut skillnaden mellan integral och approximation för $x\in[0,h]$:

$$\int_0^h x^p\ dx - \frac{h\ [0^p+h^p]}{2} = \frac{(1-p)}{2(1+p)}h^{1+p}$$

Man skulle kunna använda feluppskattningen på $[h,1]$ för att visa konvergens (felet går mot noll när $h\to 0$), men det blir ett väldigt svagt resultat.

Använder man uppskattningen på $[h,2h]$, $[2h,3h]$ etc. får man ett bra resultat som visar att felet uppför sig som h^{1+p}.

Det förväntar man sig även för de övriga metoderna. Antag att feltermen över det första intervallet (som innehåller nollan) har utseendet $ch^{m+1}f^{(m)}(\xi)$ där c är en konstant och $\xi > 0$ är en multipel av h, $\xi = \mu h$ säg (†). Med vår funktion så blir

$$ch^{m+1}f^{(m)}(\xi) = c_1(\mu)h^{m+1}h^{p-m} = c_1(\mu)h^{1+p}$$

för någon annan konstant c_1 som beror av μ. Denna konstant är givetvis viktig, så detta resonemang visar bara hur vi förväntar oss att beroendet av h ändras. Vi får alltså bara h^{1+p} som kan kräva många funktionsberäkningar (enligt vår tabell).

Tar vi $p=0.3$ med samma tolerans som i föregående exempel, så kräver Simpson inte 52 funktionsberäkningar utan 1 697 157.

Problemet är väsentligen av samma slag som när vi interpolerade $\sqrt{t}$ kring $t\ge 0$.

Vad kan man göra? I enkla fall kan man kanske byta parametrisering av f och betrakta x som funktion av y (givetvis förutsatt att f^{-1} existerar lokalt) och sedan integrera i y-led (lite mer fixande krävs för att få rätt integral).

$y=\sqrt{x}$ övergår då i triviala $x=y^2$. Man kan skaffa sig ett interpolationspolynom genom att anpassa x-värden till y-värden (sk invers interpolation).

(†) På varje intervall $[\delta,h]$, $\delta>0$ gäller feluppskattningen. Under svaga villkor på f och metod kan skillnaden mellan integralen över $[0,\delta]$ och metoden begränsas av konstant $\cdot\,\delta$, vilket gör att feltermen bestämmer utseendet på felet.

Adaptivitet

Normalt vill vi inte ha ekvidistanta punkter, utan vi vill att metoden automatiskt ska anpassa sig efter funktionens utseende och använda tätare med punkter där så behövs. Vi behöver då en uppskattning av felet.

Att direkt uppskatta feltermen gör man normalt inte.
En vanlig metod är att räkna ut resultatet med två metoder (en med mindre fel) och jämföra resultaten. Kostnaden bör vara som för en metod. Man kan också använda samma metod men med olika antal punkter.

I boken används den senare varianten med trapetsmetoden (Simpson, eller bättre, är vanligare). Här följer en genomgång. Vi börjar med intervallet $[a,b]$, räknar ut trapetsapproximationen med två punkter. Vi lägger sedan till mittpunkten, $m=(a+b)/2$ och räknar ut en ny approximation, nu med tre punkter. Observera att detta kräver ett nytt funktionsvärde, $f(m)$.

Vi fortsätter nu så rekursivt på intervallen $[a,m]$ och $[m,b]$. När felet över ett intervall är tillräckligt halverar vi inte detta intervall vidare.

Antag att vi har kommit ner till ett delintervall av längd h. Approximationerna kan skrivas (I är det exakta värdet av integralen över detta delintervall)

$$I = T_h - h^3 f''(\xi)/12 \quad \text{resp.} \quad I = T_{h/2} - h(h/2)^2 f''(\theta)/12$$

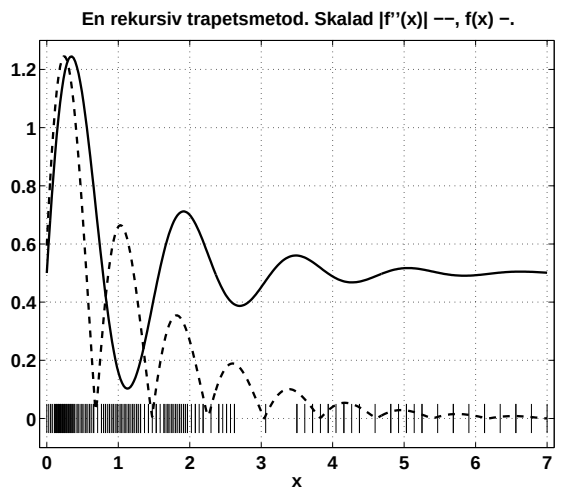
Antag att $c = -f''(\xi) \approx -f''(\theta)$ (behöver inte vara sant).
Då gäller

$$0 \approx T_h - T_{h/2} + ch^3(1-1/4)/12 = T_h - T_{h/2} + 3ch^3/(4\cdot 12)$$

Men felet i $T_{h/2}$ är ju $ch(h/2)^2/12$. Alltså

$$I \approx T_{h/2} + \frac{T_{h/2}-T_h}{3}$$

Så här kan det se ut (med rätt stor tolerans):



Man kan notera att formeln ovan även ger upphov till en ny metod. Om vi lägger till feluppskattningen får vi

$$I \approx (4T_{h/2} - T_h)/3$$

och bakom denna formel döljer sig Simpsons formel.

Man kan betrakta härledningen vi har gjort som ett specialfall av sk Richardsonextrapolation.

Man kan visa att det existerar en serieutveckling av felet

$$\left(\int_a^b f(x)\,dx =\right) I = T_h + a_1h^2 + a_2h^4 + a_3h^6 + \cdots$$

Vi halverar nu h och får

$$I = T_{h/2} + a_1h^2/4 + a_2h^4/16 + a_3h^6/64 + \cdots$$

Genom att kombinera de två uttrycken kan vi bli av med h^2-termen (och därmed minska felet):

$$4I - I = 4T_{h/2} - T_h + (4a_1h^2/4 - a_1h^2) + (4a_2h^4/16 - a_2h^4) + \cdots$$

så att

$$I = \frac{4T_{h/2} - T_h}{3} - \frac{a_2h^4}{4} + \cdots$$

Detta kan man nu upprepa (med $T_{h/4}$) för att bli av med h^4-termen. Denna process (upprepad Richardsonextrapolation) kallas Rombergkvadratur.

Richardsonextrapolation kan användas närhelst man har en utveckling av felet. Exempel, approximation av derivata.

$$\frac{f(x+h) - f(x)}{h} = \frac{1}{h}\sum_{k=0}^{\infty}\frac{h^k f^{(k)}(x)}{k!} - \frac{f(x)}{h} = f'(x) + \sum_{k=2}^{\infty}\frac{h^{k-1}f^{(k)}(x)}{k!}$$

så att

$$f'(x) = \frac{f(x+h) - f(x)}{h} - hf''(x)/2 - h^2f'''(x)/6 - \cdots$$

$$f'(x) = \frac{f(x+h/2) - f(x)}{h/2} - (h/2)f''(x)/2 - (h/2)^2f'''(x)/6 - \cdots$$

och

$$f'(x) = 2\frac{f(x+h/2) - f(x)}{h/2} - \frac{f(x+h) - f(x)}{h} + h^2f'''(x)/12 + \cdots$$

$$f'(x) = \frac{-3f(x) + 4f(x+h/2) - f(x+h)}{h} + h^2f'''(x)/12 + \cdots$$

Gausskvadratur

Antag att vi vill beräkna $\int_a^b f(x)\,dx$ och tillåts göra tre funktionsberäkningar, $f(x_1)$, $f(x_2)$ samt $f(x_3)$. Om vi väljer $x_1 = a$, $x_2 = (a+b)/2$ samt $x_3 = b$ så kommer Simpsons formel att vara optimal när det gäller polynomiellt gradtal. Dvs om vi vill att metoden ska vara exakt för polynom av grad $0, 1, \ldots, m$ för så stort m som möjligt så är Simpsons metod det bästa valet ($m = 3$).

Det visar sig dock att vi kan få större m genom att välja andra x_k-värden. Detta är kärnan i Gausskvadratur, att välja både x_k-värden och vikter för att maximera m.

Låt oss ta intervallet $[-1, 1]$. Vi ska nu välja x_1, x_2, x_3 samt vikter w_1, w_2, w_3 så att

$$\int_{-1}^{1} x^k\,dx = w_1x_1^k + w_2x_2^k + w_3x_3^k, k = 0, 1, \ldots, m$$

för maximalt m. Integralens värde blir 0 om k är udda och $2/(k+1)$ annars. Vi får följande ickelinjära ekvationssystem att lösa:

$$\begin{cases} 2 &= w_1 + w_2 + w_3 & k = 0 \\ 0 &= w_1x_1 + w_2x_2 + w_3x_3 & k = 1 \\ 2/3 &= w_1x_1^2 + w_2x_2^2 + w_3x_3^2 & k = 2 \\ 0 &= w_1x_1^3 + w_2x_2^3 + w_3x_3^3 & k = 3 \\ 2/5 &= w_1x_1^4 + w_2x_2^4 + w_3x_3^4 & k = 4 \\ 0 &= w_1x_1^5 + w_2x_2^5 + w_3x_3^5 & k = 5 \end{cases}$$

Det verkar inte rimligt att ta med en ekvation till. Vi har ju $3+3 = 6$ obekanta och vi kan då kanske satisfiera sex ekvationer. För att lösa systemet kan man använda "brute force", men det verkar rimligt att punkterna måste uppvisa viss symmetri. Vi antar sålunda att $x_1 < x_2 < x_3$ med $x_2 = 0$ och $x_1 = -x_3$.

Detta val leder ($k = 1$) till att $w_1 = w_3$ och satisfiering av fallen $k = 3, 5$. Kvarstår då ekvationerna $2 = 2w_1 + w_2$, $2/3 = 2w_1x_1^2$ samt $2/5 = 2w_1x_1^4$. Vi får $x_1 = -\sqrt{3/5}$ och $w_1 = 5/9$. Metoden blir alltså:

$$\int_{-1}^{1} f(x)\,dx \approx \frac{5}{9}f\left(-\sqrt{3/5}\right) + \frac{8}{9}f\left(0\right) + \frac{5}{9}f\left(\sqrt{3/5}\right)$$

Man ser att metoden inte är exakt för $m = 6$ så det polynomiella gradtalet är 5 (det var 3 för Simpsons metod). Eftersom integration är en linjär operation så är metoden exakt för alla polynom av grad högst 5.

För en Gausskvadraturformel har vi gradtalet $2n - 1$ med n punkter. Vi har dock offrat i enkelhet. Härledningen kan dock förenklas (man använder teorin för ortogonala polynom och kan blanda in egenvärdesproblem för tridiagonala matriser).

En annan nackdel är att värdena måste skrivas in i ett program (stora tabeller). Det allvarligaste problemet är dock att man inte kan återanvända funktionsvärden när man gör adaptiva metoder.

Det finns dock varianter, Gauss-Kronrodkvadratur där man har en kompromiss mellan optimaliteten i Gausskvadratur och kravet på återanvändning av funktionsvärden, se boken.

Hur ser vår metod ut på intervallet $[a, b]$, $\int_a^b f(z)\,dz$? Sätt $z = \alpha x + \beta$ där $\alpha = (b-a)/2$ och $\beta = (a+b)/2$. $z \in [a, b] \to x \in [-1, 1]$. $dz = \alpha\,dx$. Alltså:

$$\int_a^b f(z)\,dz = \int_{-1}^{1} f(\alpha x + \beta)\alpha\,dx \approx \sum_{k=1}^{3}(\alpha w_k)f(\alpha x_k + \beta)$$

Ordinära differentialekvationer

Vi kommer enbart att studera begynnelsevärdesproblem, t.ex.

$$y'(t) = t^2 + \sin y(t), \quad 3 < t \leq 10, \quad y(3) = 4$$

Derivatan, $y'(t)$, är tagen med avseende på t ("tiden").

$3 < t \leq 10$ anger det intervall där vi vill beräkna lösningen (approximativt). $y(3) = 4$ är ett begynnelsevärde som anger y:s värde, 4, vid tiden $t = 3$. Normalt (i övningar och anteckningar) skriver vi aldrig ut t, i $y(t)$. Vi struntar även i intervallet (tiden i begynnelsevärdet är vänster ändpunkt, och Du får anta något lämpligt slutvärde). Problemet kan då formuleras:

$$y' = t^2 + \sin y, \quad y(3) = 4$$

Normalt vill vi studera ett generellt problem, vi skriver:

$$y' = f(t, y), \quad y(t_0) = y_0$$

Så, i exemplet ovan är $f(t, y) = t^2 + \sin y$. Begynnelsetiden är t_0 (3 i exemplet) och y vid detta värde är y_0 (4 i exemplet). Både t_0 och y_0 måste vara kända.

Lösningsmetoderna genererar approximationer till lösningen för en uppsättning tidpunkter: (t_0, y_0), (t_1, y_1), $(t_2, y_2), \ldots, (t_n, y_n)$, där t_n är slut-tiden och $y_k \approx y(t_k)$.

> y_k är en approximation av lösningen vid tiden $t = t_k$. Det exakta värdet är $y(t_k)$.

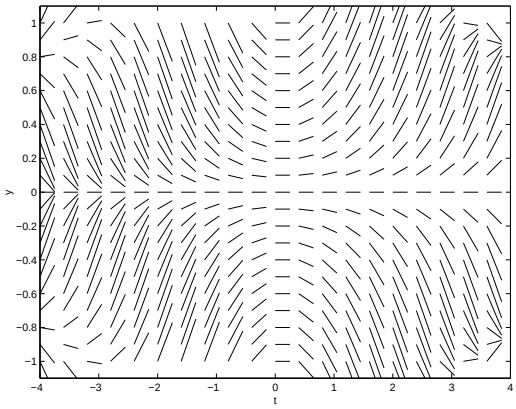
Senare kommer system av ekvationer. Sådana behövs för att vi skall kunna lösa problem som innehåller högre derivator, t.ex.

$$y''' = t + 2y'' + (y')^2 + \sin y, \quad y(0) = 2, \quad y'(0) = -3, \quad y''(0) = 4$$

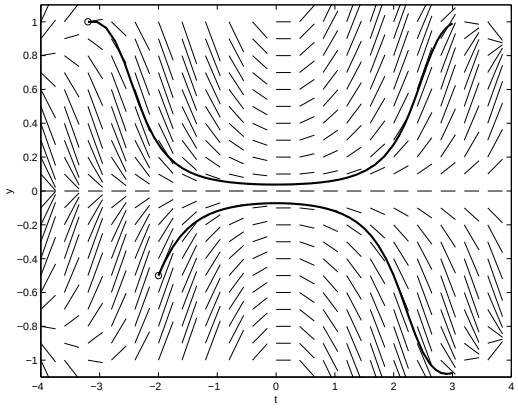
Exempel: låt oss studera problemet: $y' = 1$. Detta är inget begynnelsevärdesproblem (eftersom vi saknar $y(t_0) = y_0$). Ett problem av detta slag har normalt oändligt många lösningar, i detta fall $y(t) = t + c$ där c är ett godtycklig reellt tal. När vi ger ett begynnelsevärde väljer vi ut en av alla dessa oändligt många lösningar. $y(3) = 4$ ger oss lösningen $y(t) = t + 1$.

Med grafiska verktyg kan vi skaffa oss en bild om lösningsmängden även för problemet $y' = f(t, y)$. Låt oss göra detta för problemet $y' = \sin(ty)$.

I bilden nedan har jag skapat ett gitter i (en begränsad del av) (t, y)-planet. I varje gitterpunkt har jag avsatt en pil vars riktning överensstämmer med derivatan av den lösningskurva som går genom punkten. Detta är enkelt eftersom $y' = f(t, y)$, så derivatan är $f(t, y)$.

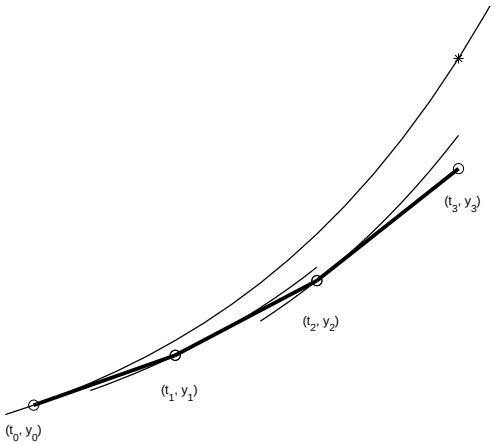


I nästa bild har jag ritat i två lösningskurvor (jag har gett två begynnelsevärden).



Det finns begynnelsevärdesproblem som saknar, eller har flera lösningar. Det kan också vara så att $y(t)$ inte existerar för alla $t > t_0$.

Föregående bild antyder en enkel lösningsmetod. Vi startar i (t_0, y_0) (som vi känner). Vi tar sedan ett litet steg utmed tangenten till lösningen (tangenten kan vi beräkna med hjälp av $f(t, y)$).



Antag att vi stegar med fix steglängd, h, i t så att:

$t_1 = t_0 + h$, $t_2 = t_1 + h$, $t_3 = t_2 + h, \ldots$. Allmänt $t_k = t_0 + kh$.

Vi får Eulers metod:

$$y_{k+1} = y_k + hf(t_k, y_k), \quad k = 0, 1, 2, \ldots$$

eller utskrivet

$y_1 = y_0 + hf(t_0, y_0)$, $y_2 = y_1 + hf(t_1, y_1)$, $y_3 = y_2 + hf(t_2, y_2), \ldots$

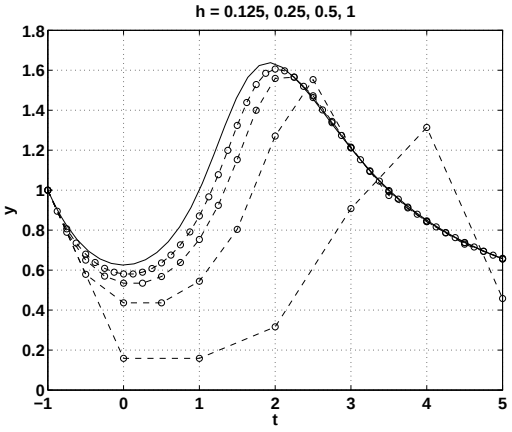
Exempel: $y' = \sin(ty), \quad y(-1) = 1.$
Så $t_0 = -1$, $y_0 = 1$ och $f(t, y) = \sin(ty)$.

Om $h = 0.1$ får vi approximationerna:

$y_1 = y_0 + hf(t_0, y_0) = y_0 + h\sin(t_0 y_0) =$
$1 + 0.1\sin(-1 \cdot 1) \approx 0.9159$

$y_2 = y_1 + hf(t_1, y_1) = y_1 + h\sin(t_1 y_1) \approx$
$0.9159 + 0.1\sin(-0.9 \cdot 0.9159) \approx 0.8425$

$y_3 = y_2 + hf(t_2, y_2) = y_2 + h\sin(t_2 y_2) \approx$
$0.8425 + 0.1\sin(-0.8 \cdot 0.8425) \approx 0.7801$ osv.



Alternativa härledningar av Eulers metod

Taylorutveckling:

$$y(t_k + h) = y(t_k) + h\ y'(t_k) + \frac{h^2}{2}\ y''(t_k) + \cdots$$

Nu är $y'(t_k) = f(t_k, y(t_k))$ och $t_{k+1} = t_k + h$ så att:

$$y(t_{k+1}) \approx y(t_k) + h\ f(t_k, y(t_k))$$

Vi approximerar nu $y_k \approx y(t_k)$, $y_{k+1} \approx y(t_{k+1})$ och får:

$$y_{k+1} = y_k + h\ f(t_k, y_k)$$

Nu till en härledning som använder kvadratur (integration).

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} y'(t)\ dt = \int_{t_k}^{t_{k+1}} f(t, y(t))\ dt$$

Vi approximerar nu integralen med arean av en rektangel;

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(t, y(t))\ dt \approx \underbrace{(t_{k+1} - t_k)}_{h} f(t_k, y(t_k))$$

Så:

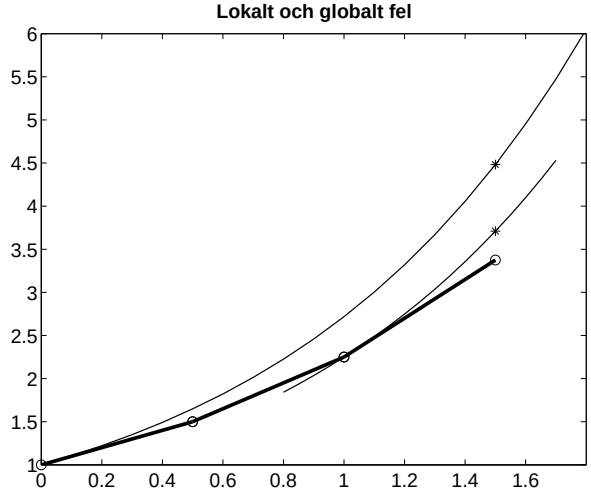
$$y_{k+1} = y_k + h\ f(t_k, y_k)$$

Felkällor

- trunkeringsfel; i Eulers metod trunkerar vi Taylorutvecklingen (approximerar med tangenten)
- avrundningsfel; normalt inte så viktigt

Lokalt fel: felet som uppstår i ett steg när man betraktar startpunkten, (t_{k-1}, y_{k-1}), som exakt. Programvara försöker begränsa detta fel.

Globalt fel: felet mellan approximativ och exakt lösning, $y_k - y(t_k)$



Ordning

Olika metoder har olika ordning: en metod har ordning p om det lokala felet är av storleksordningen h^{p+1} när $h \to 0$. Vi skriver $\mathcal{O}(h^{p+1})$.
Vilken ordning har Eulers metod?

Antag att vi står i punkten (t_{k-1}, y_{k-1}). Vad blir felet i nästa steg förutsatt att (t_{k-1}, y_{k-1}) betraktas som exakt?

Låt oss titta på det speciella problemet $y' = \lambda y, y(0) = y_0$.
Eulers metod ger, som vanligt, approximationerna $y_0, y_1, y_2, \ldots$.

Den exakta lösningen som går genom (t_{k-1}, y_{k-1}) betecknar vi med $z(t)$ och den löser följande problem:

$$z' = \lambda z, \quad z(t_{k-1}) = y_{k-1}$$

Dvs.

$$z(t) = e^{\lambda(t - t_{k-1})} y_{k-1}$$

så när $t = t_k$ är

$$z(t_k) = e^{\lambda(t_k - t_{k-1})} y_{k-1} = e^{\lambda h} y_{k-1}$$

Eulers metod ger:

$$y_k = y_{k-1} + hf(t_{k-1}, y_{k-1}) = (1 + \lambda h) y_{k-1}$$

Det lokala felet blir:

$$y_k - z(t_k) = (1 + \lambda h) y_{k-1} - e^{\lambda h} y_{k-1} =$$

$$\left[1 + \lambda h - \left[1 + \lambda h + \frac{(\lambda h)^2}{2} + \ldots\right]\right] y_{k-1} = -\left[\frac{(\lambda h)^2}{2} + \ldots\right] y_{k-1}$$

som är $\mathcal{O}(h^2)$, så Eulers metod har ordning ett (är en första ordningens metod).

Nu till det globala felet, $y_k - y(t_k)$, där $y(t)$ är den exakta lösningen till $y' = \lambda y, y(0) = y_0$ och y_k är approximationen av $y(t_k)$.

Tydligen är

$$y(t_k) = e^{\lambda t_k} y_0 \quad \text{och} \quad y_k = (1 + \lambda h)^k y_0,$$

Varför?
$y_1 = y_0 + h\lambda y_0 = (1 + h\lambda) y_0$.
$y_2 = y_1 + h\lambda y_1 = (1 + h\lambda) y_1 = (1 + h\lambda)^2 y_0$ etc.

Eftersom $t_k = kh$, får vi följande uttryck för det globala felet:

$$y_k - y(t_k) = (1 + \lambda h)^k y_0 - e^{\lambda t_k} y_0 = (1 + \lambda h)^k y_0 - e^{\lambda k h} y_0 =$$

$$\left[1 + k\lambda h + \frac{k(k-1)}{2}(\lambda h)^2 + \dots\right] y_0 - \left[1 + k\lambda h + \frac{(k\lambda h)^2}{2} + \dots\right] y_0 =$$

$$\frac{-k}{2}(\lambda h)^2 y_0 + \dots = -\frac{1}{2}\lambda^2 (hk) y_0 \; h + \dots = -\frac{1}{2}\lambda^2 t_k y_0 \; h + \dots$$

Så det globala felet uppför sig som h.

Tumregel: det globala felet är $\mathcal{O}(h^p)$.

Vi tappar alltså en potens mellan lokalt och globalt fel.

Vi kan försöka skapa metoder av högre ordning, t.ex. genom att använda tidigare punkter; en så kallad flerstegsmetod.

T.ex.

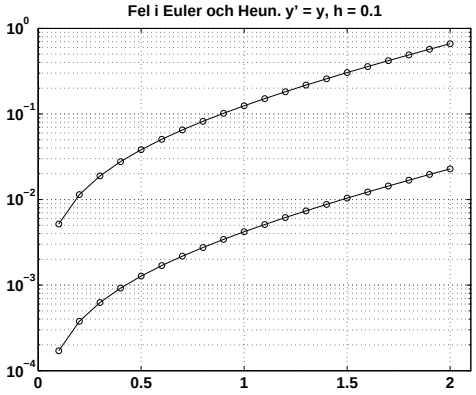
$$y_{k+1} = y_k + \frac{h}{2}\left[3f(t_k, y_k) - f(t_{k-1}, y_{k-1})\right]$$

som har ordning två.
För att starta metoden kan vi ta ett Euler-steg.

En annan metod av andra ordningen är Heuns metod:

$$y_{k+1} = y_k + \frac{h}{2}\left[f(t_k, y_k) + f(t_k + h, y_k + hf(t_k, y_k))\right]$$

Detta är en enstegsmetod.



System av ekvationer

$$u^{(3)} = u'' - 2tu' + u^2 - t + 1, \quad \begin{cases} u(3) = 2 \\ u'(3) = -1 \\ u''(3) = 0 \end{cases}$$

Inför nya funktioner

$$y_1 = u$$
$$y_2 = u' \Rightarrow y_2 = y_1'$$
$$y_3 = u'' \Rightarrow y_3 = y_2'$$

Vi får

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = y_3 - 2ty_2 + y_1^2 - t + 1 \end{cases}, \quad \begin{cases} y_1(3) = 2 \\ y_2(3) = -1 \\ y_3(3) = 0 \end{cases}$$

Detta problem kan fortfarande skrivas, $y' = f(t, y)$, om vi inför vektorerna y och f, dvs.

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix}$$

$$f(t, y) = \begin{bmatrix} y_2 \\ y_3 \\ y_3 - 2ty_2 + y_1^2 - t + 1 \end{bmatrix}, \quad y^{(0)} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}$$

Alla metoder vi har sett kan enkelt generaliseras till systemfallet. Skalära y_k byts mot vektorn $y^{(k)}$. $f(t_k, y_k)$ går över i $f(t_k, y^{(k)})$. Tiden t_k och steglängden h är fortfarande skalärer.

Eulers metod för exemplet ovan blir, med $t_0 = 3$, $h = 0.1$:

$$y^{(0)} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, \quad y^{(1)} = y^{(0)} + hf(t_0, y^{(0)})$$

Dvs.

$$\begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \end{bmatrix} = \begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \end{bmatrix} + h \begin{bmatrix} y_2^{(0)} \\ y_3^{(0)} \\ y_3^{(0)} - 2t_0 y_2^{(0)} + \left(y_1^{(0)}\right)^2 - t_0 + 1 \end{bmatrix}$$

$$\begin{bmatrix} 1.9 \\ -1 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} -1 \\ 0 \\ 0 - 2 \cdot 3 \; (-1) + 2^2 - 3 + 1 \end{bmatrix}$$

osv.

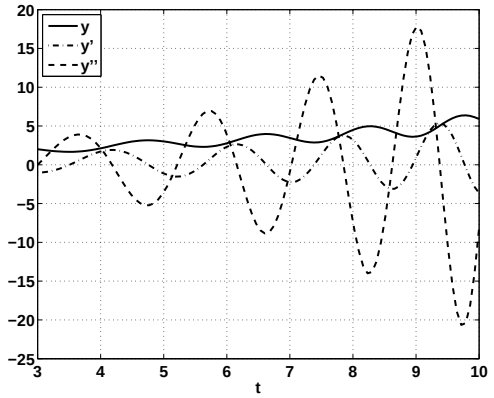
Hur man löser problemet med `ode45`

```
function ode_ex
y0 = [2 -1 0]';    % begynnelsevärden
t0 = 3;            % begynnelsetid
ts = 10;           % slut-tid

[t, y] = ode45(@f, linspace(t0, ts, 100), y0);

figure(1)
hold off
plot(t, y(:, 1), 'k-', t, y(:, 2), 'k-.', ...
     t, y(:, 3), 'k--')
legend({'y', 'y'''', 'y'''''}, 'Location', 'NorthWest')
xlabel('t')
grid on

function yp = f(t, y)
yp = [y(2); y(3); y(3) - 2 * t * y(2) + y(1)^2 - t + 1];
```

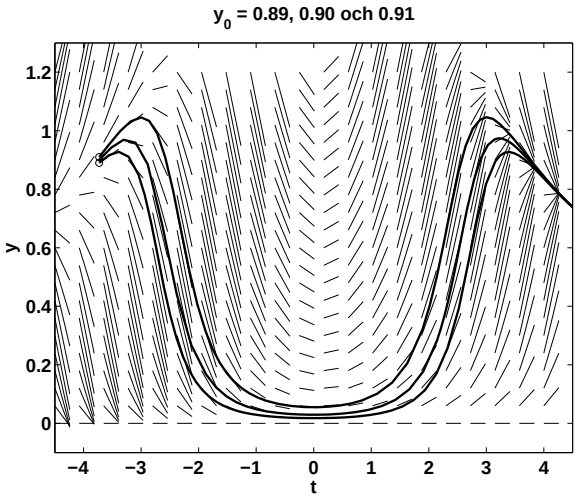


Problemets stabilitet

Hur ändras lösningen vid små ändringar i problemet?

I följande bild visas hur lösningen (till $y' = \sin(ty)$) varierar med $y(t_0)$. $y(t_0) = 0.89, 0.90$ respektive 0.91.

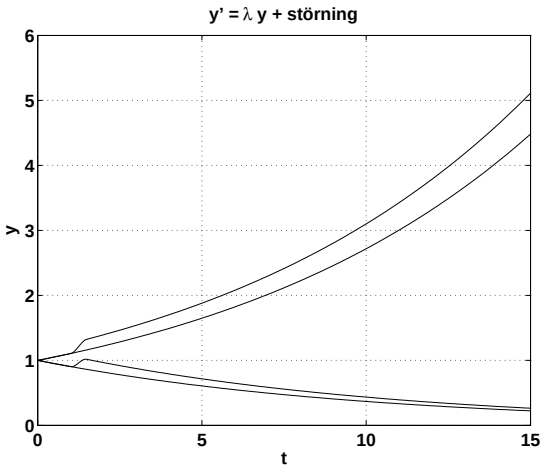
Om lösningskurvorna går ihop eller går isär avgörs av det lokala utseendet på riktningsfältet.



En lösning är stabil om två lösningar kan fås att ligga godtyckligt nära varandra (för $t \geq t_0$) givet att vi stör tillräckligt lite.

I nedanstående bild har jag löst $y' = \lambda y + s(t)$, för ett positivt och ett negativt värde på λ.
$s(t)$ är en liten störning som inträffar omkring $t = 1$.

Den exakta lösningen till $y' = \lambda y$ är $y(t) = e^{\lambda t} y(0)$.



Vi ser att störningen dämpas ut när $\lambda < 0$.

Om λ är komplext med negativ realdel så är differentialekvationen stabil; felen dämpas ut.
Om realdelen är positiv är differentialekvationen instabil.

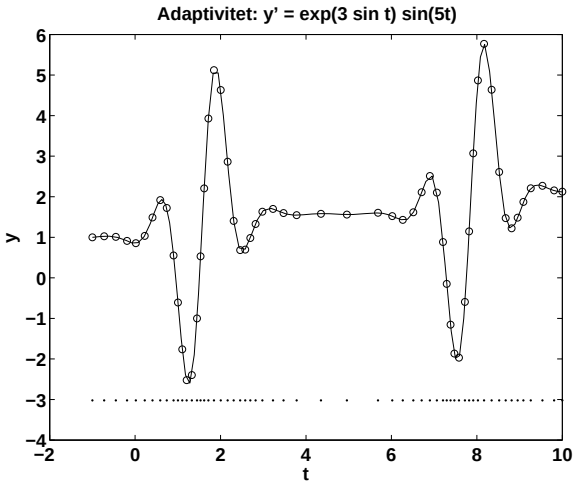
Detta kan generaliseras till ickelinjära problem och system av sådana.

Adaptivitet

De flesta ODE-lösare är adaptiva, dvs. de försöker att anpassa steglängden så att det lokala felet underskrider en tolerans given av programmets användare.

I vissa fall består programmet av en familj av metoder av olika ordning. Programmet kan då även variera ordningen.

I figuren nedan har jag löst ett problem med `ode45` (den heldragna lösningen) och `ode23`, med stor tolerans, (ringarna).



Styva problem och lösarens stabilitet

Det är vanligt med så kallade styva problem (stiff).
Dessa uppkommer t.ex. när man har snabba transienter.

Om vi använder en vanlig ode-lösare på ett styvt problem tvingas lösaren ta mycket korta steg för bibehålla stabiliteten.

Det visar sig att vi kan lära oss mycket om metoders stabilitet genom att studera den skalära testekvationen, $y' = \lambda y$, $y(0) = 1$. Normalt har vi dock styva system (och inte skalära ekvationer).

Antag att $\lambda < 0$, den exakta lösningen är då avtagande.

För vilka h ger Eulers metod $y_k \to 0$ då $k \to \infty$?

$$y_{k+1} = y_k + hf(t_k, y_k) = y_k + h\lambda y_k = (1 + h\lambda)y_k$$

så att

$$y_k = (1 + h\lambda)^k$$

När gäller att $y_k \to 0$? Jo då:

$$|1 + h\lambda| < 1$$

dvs, om $\lambda \in \Re$ (och $\lambda < 0$),

$$0 < h|\lambda| < 2$$

Antag nu att λ är ett mycket negativt tal, säg $\lambda = -20000$. För att vi överhuvudtaget skall få en lösning som går mot noll måste $h < 1/10000$.

Vi noterar att $e^{\lambda t} = \epsilon_{mach}$ om $t = (\log \epsilon_{mach})/\lambda \approx 2 \cdot 10^{-3}$ i vårt exempel.

Vad skall vi göra? Lösningen är implicita metoder.

Bakåt-Euler:

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1})$$

Stabilitet? Testa på $y' = \lambda y$

$$y_{k+1} = y_k + h\lambda y_{k+1}$$

så att

$$y_{k+1} = (1 - h\lambda)^{-1} y_k$$

och

$$y_k = (1 - h\lambda)^{-k} \quad \text{ty} \quad y_0 = 1$$

När är $|(1 - h\lambda)^{-1}| < 1$? Antag $\lambda < 0$ (reellt)
då är $|(1 - h\lambda)^{-1}| < 1$ för alla $h > 0$!

Detta innebär givetvis inte att vi kan ta godtyckligt långa steg. Tar vi för långa steg blir felet för stort.

Implicita metoder har den nackdelen att vi måste lösa en (normalt ickelinjär) ekvation för att bestämma y_{k+1}.

I en explicit metod, som Eulers metod, är detta inte nödvändigt.

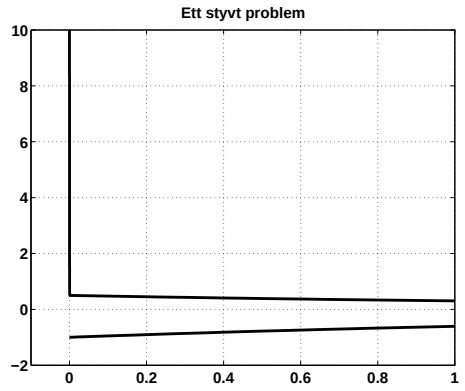
Det finns mer komplicerade implicita metoder, t.ex.

$$y_{k+1} - \frac{4}{3}y_k + \frac{1}{3}y_{k-1} = \frac{2h}{3}f(t_{k+1}, y_{k+1})$$

som är ett exempel på en flerstegsmetod.

Ett exempel:

$$y' = \begin{bmatrix} y_2 \\ -(y_1 + 2y_2)/\epsilon \end{bmatrix}, \quad y(0) = \begin{bmatrix} -1 \\ 10 \end{bmatrix}$$



Med Matlabs `ode23` (en Runge-Kutta-lösare ordning 2 och 3) krävs 11989 steg för att lösa problemet då $\epsilon = 0.0002$. Toleranserna är relativt 10^{-3} och absolut 10^{-6}.

Matlabs ode23s (s för stiff) löser problemet i 192 steg. 140 av dessa steg tas för $t < 0.01$.

Andra problemklasser

Tvåpunkts randvärdesproblem:

$$y'' = f(t, y, y'), \quad \alpha_1 y(a) + \beta_1 y'(a) = \gamma_1, \quad \alpha_2 y(b) + \beta_2 y'(b) = \gamma_2$$

Egenvärdesproblem (vibrerande sträng):

$$(py')' + \lambda \rho y = 0$$

$y(a) = y(b) = 0$, fixerade ändpunkter
$y'(a) = y'(b) = 0$, fria ändpunkter
$y(a) = y(b)$, $y'(a) = y'(b)$, periodiska randvillkor.

Ickelinjärt egenvärdesproblem (bifurkationsproblem).
Knäckning, roterande kedja, Taylor-Couette.

$$y'' + \frac{\lambda y}{\sqrt{y^2 + t^2}} = 0 \quad \text{samt randvillkor}$$

Tidsfördröjningsproblem (delay equations)

$$y'(t) = y(t) - y(t - T) + \ldots$$

Inkubationstid; ändlig utbredningshastighet...

Differentialalgebraiska problem: differentialekvation med algebraiska "bivillkor".
Specialfall, implicita problem: $g(t, y)y' = f(t, y)$.