Exempel: En olämplig modell. Säg att vi har modellen

$$b = x_1 t + x_2 \sin t$$

Problemet är att $\sin t \approx t$ för $t \approx 0$. $\sin t = t - t^3/6 + \cdots$. Vi har därför nästan modellen $b = (x_1 + x_2)t$ och motsvarande minstakvadratproblem har inte entydig lösning.

Valet av fysikaliska enheter kan orsaka problem. Antag att vi har modellen $b = x_1 + x_2 t + x_3 t^2$ och att $A = QR$ där R är välkonditionerad.

Ett enhetsbyte för t svarar mot en skalning σt. Sambandet $A = QR$ övergår då i $AD = Q(RD)$, $D = \mathrm{diag}(1, \sigma, \sigma^2)$.
Det gäller att $\kappa(A) = \kappa(R) = ||R||_2 \; ||R^{-1}||_2$ varför

$$\kappa(AD) = \kappa(RD) = ||RD||_2 \; ||(RD)^{-1}||_2 \leq$$

$$||R||_2 \; ||D||_2 \; ||R^{-1}||_2 \; ||D^{-1}||_2 = \kappa(R)\kappa(D) = \kappa(R) \;\; \max(\sigma^{-2}, \sigma^2)$$

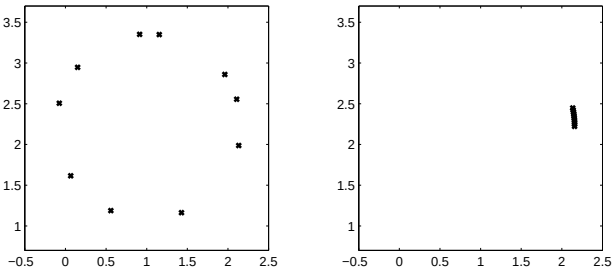
Så att byta från t.ex. sekunder till ms kan öka konditionstalet med 10^6. Lika illa är att byta till ks.

Vissa problem är illa konditionerade i sig (redan innan vi gjort någon olämplig matematik), det gäller många sk inversa problem, t.ex. bildrekonstruktion. Detta används inom bland annat kriminologi och astronomi.

Framåtriktningen är då att vi tar ett fotografi, vi avbildar ett motiv. Den omvända riktningen (inversen) att rekonstruera motivet givet ett fotografi som är suddigt (av rörelseoskärpa t.ex.). Problemet är illa konditionerat pga att flera motiv kan passa ihop med ett suddigt fotografi. En annan viktig tillämpning är datortomografi.

Ett enkelt exempel: vi har punkter i planet som vi tror ligger på en cirkel och vi vill bestämma cirkelns centrum och radie.
Efter variabeltransformation kan problemet skrivas som ett linjärt minstakvadratproblem.

Vi får en säker bestämning om vi har mätpunkter fördelade över cirkeln. Om punkterna grupperar ihop sig utmed en liten del av cirkeln, är problemet att bestämma den illa konditionerat. Om punkterna sammanfaller finns oändligt många lösningar (cirklar).



Ibland kan man inte välja hur man ska mäta, utan man får nöja sig med de data man har. För att kunna få ett vettigt svar måste lösningsprocessen stabiliseras med sk regularisering.

Antag att $||R||_2 \approx 1$ och $||b||_2 \approx 1$. När R är illakonditionerad så är $||R^{-1}||_2$ stor vilket medför att $||x||_2$ blir stor. Ett indirekt sätt att minska inflytandet av de små osäkra värdena i R är att begränsa längden på lösningen, en enkel variant är:

$$\min_x \; ||Ax - b||_2 \;\; \text{med} \;\; ||x||_2 \leq \alpha$$

ett annat sätt är att ändra direkt i R. α måste sättas av den som löser problemet. Hur man gör detta på ett vettigt sätt tar jag inte upp.

Ett fysikproblem

En kväll fick jag ett mail från Peter Berntsen, kondenserade materiens fysik, om ett besvärligt optimeringsproblem. Följande sidor är resultatet av mina försök att förstå problemets natur.

Svante Arrhenius (1859-1927) är en av grundarna av den fysikaliska kemin. Han undersökte (bland annat) hur hastigheten hos kemiska reaktioner beror av temperaturen. Om t.ex. ämnena α och β reagerar och producerar ämnet γ, så gäller (ofta) att:

$$\frac{d\,[\gamma]}{dt} = k(T)\;[\alpha]^m\,[\beta]^n$$

där [ ] betecknar koncentrationen, t är tiden och T är absoluta temperaturen (i Kelvin). m och n kallas ordningar (båda kan vara ett t.ex).

Arrhenius ekvation (1889) är en modell för utseendet på $k(T)$:

$$k(T) = Ae^{-E/RT}.$$

A kallas den pre-exponentiella faktorn, E (ofta skriven E_a) är aktiveringsenergin och R är den allmänna gaskonstanten.

Arrhenius resonerade så här: För att en kemisk reaktion, mellan två molekyler, skall inträffa, så måste rörelseenergin hos molekylerna uppnå en viss nivå, aktiveringsenergin E.

Enligt Ludwig Boltzmanns (1844-1906) arbeten (statistisk mekanik och termodynamik) följer att antalet kollisioner med energi $\geq E$ är $e^{-E/RT}$, så $k(T)$ bör vara proportionell mot denna faktor. Om temperaturen ökar, så blir sannolikheten större att molekyler uppnår E varför $k(T)$ ökar.

Arrhenius formel passar till flera andra situationer. Min bok i fysikalisk kemi nämner frekvensen av syrsors spelande (som funktion av T), myrors krypande, åldrandets hastighet, eldflugors lysande, och hur snabbt man glömmer.
Anledningen att Arrhenius formel passar in, är att ovanstående processer är kemiska.

Nu till tillverkning av glas. Det är intressant att ha en modell för beroendet mellan viskositet, b, (av en glas-smälta) och temperatur. Arrhenius modell stämmer inte så bra. Man noterade att $\log b$ inte var linjär i $1/T$:

$$b = Ae^{-E/RT} \;\Leftrightarrow\; \log b = \log A - \frac{E}{R} \cdot \frac{1}{T}$$

Gordon Fulcher (Corning Glass Works, NY) listade, i en artikel från 1925, följande modeller

$$\begin{aligned}
\log b &= A - B/T + C/T^2 \\
\log b &= -A + B/T + C/T^2 \\
\log b &= -A + B\log T + C/T^2 \\
\log b &= -A + B/(T - T_0)^2 \\
\log b &= -A + B/(T + 273)^{2.33} \\
\log b &= -A + 10^3 \cdot B/(T - T_0)
\end{aligned}$$

T ges i $^\circ C$ och $\log = \log_{10}$. Den sista ekvationen fungerade rätt väl. Vogel (1925) och Tammann (1926) publicerade samma formel, som nu kallas: Vogel-Fulcher-Tammanns modell (VFT), här skriven på en vanlig form:

$$b = A\; e^{E/(T - T_0)} \qquad \text{VFT}$$

$T_0 = 0$ ger Arrhenius modell. Vi har mätt b vid olika temperaturer, T, och vill bestämma parametrarna A, E samt T_0. Vi har tydligen en ickelinjär modell i parametrarna.

Fulcher använde en grafisk teknik. Först bestämde han T_0 från tre mätvärden. Han plottade sedan $\log b$ som funktion av $1/(T - T_0)$ och anpassade en rät linje till mätpunkterna.

Låt oss nu attackera problemet med moderna hjälpmedel.
Första idén: formulera problemet som ett ickelinjärt minstakvadratproblem (jag har tagit bort $\sqrt{\;}$):

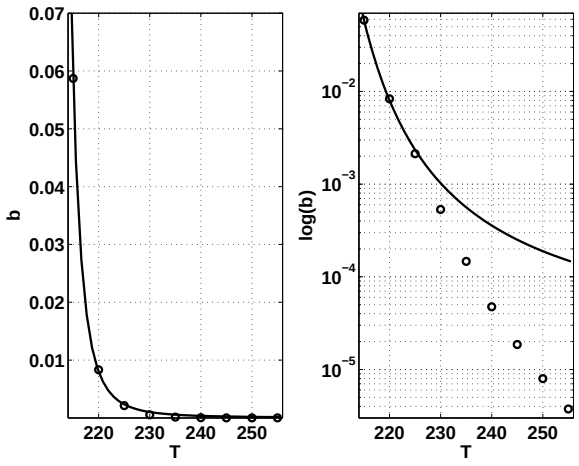
$$\min_{A,\; E,\; T_0} \; \sum_{k=1}^{n} \left[b_k - A\; e^{E/(T_k - T_0)} \right]^2$$

De lösare som används är *iterativa* och kräver en startapproximation och producerar, förhoppningsvis, en serie approximationer som konvergerar mot ett *lokalt* minimum.

Lösaren stannar när ett avbrottskriterium är uppfyllt. Detta kriterium baseras normalt på förändringen av approximationerna, förändringen av funktionen som skall minimeras (objektfunktion, målfunktion) och på normen av gradienten.

Det är *viktigt* med bra startapproximationer. En dålig approximation kan ge divergens eller konvergens mot ett lokalt minimum med större minimivärde.

Låt oss ignorera dessa råd och starta med en *slumpvektor*. Vi använder Matlabs `lsqnonlin` för det ickelinjära problemet. Så här ser plottarna av mätdata och lösning ut:



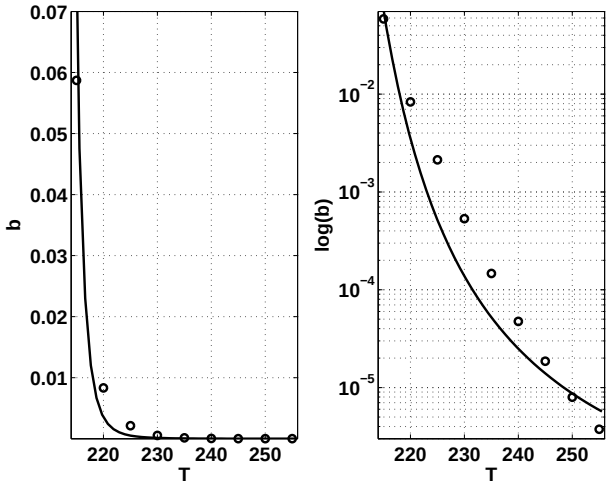
Den dåliga anpassningen beror *inte* för stora toleranser i avbrottskriteriet (man kan skärpa dessa).

Residualvektorn, med element $b_k - Ae^{E/(T_k - T_0)}$, har element av samma storleksordning, $\approx 10^{-4}$. De *relativa* avvikelserna är dock enormt stora för de små värdena.

Om vi tror (vet) att alla b-värden är givna med samma *relativa* fel kan man använda *vikter*, så att alla mätvärden får samma inflytande. Om vi viktar med $1/b_k$ får vi problemet:

$$\min_{A,\ E,\ T_0} \sum_{k=1}^{n} \left[\frac{b_k - A\ e^{E/(T_k - T_0)}}{b_k} \right]^2$$

Detta fungerar nu inte så bra, men de små värdena kommer i alla fall med:



Felet är startgissningen. Att gissa går inte så bra, vi behöver bättre värden.

För att bestämma startapproximationer av parametrarna, skriver vi om det ickelinjära problemet som ett linjärt problem. Detta går givetvis inte alltid.

Logaritmera VFT:

$$\log b = \frac{E}{T - T_0} + \log A$$

Multiplicera upp $T - T_0$ och samla ihop termerna:

$$T \log b = T_0 \log b + T \log A + E - T_0 \log A$$

Låt $x_1 = T_0$, $x_2 = \log A$ och $x_3 = E - T_0 \log A$. Det linjära problemet kan då skrivas:

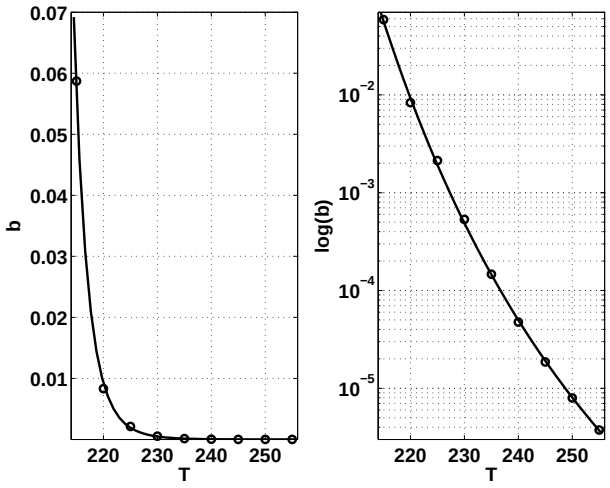
$$\min_x \left\| \begin{bmatrix} \log b_1 & T_1 & 1 \\ \log b_2 & T_2 & 1 \\ \vdots & \vdots & \vdots \\ \log b_n & T_n & 1 \end{bmatrix} x - \begin{bmatrix} T_1 \log b_1 \\ T_2 \log b_2 \\ \vdots \\ T_n \log b_n \end{bmatrix} \right\|_2$$

Här är Matlabkoden:

```
n = length(T);
x = [log(b), T, ones(n, 1)] \ (T .* log(b));

T0 = x(1);
A  = exp(x(2));
E  = x(3) + T0 * x(2);
```

Vi använder nu dessa värden som startapproximation, till `lsqnonlin`, och får en nästan perfekt plot:



Ett alternativ till vikter, i detta fall, är att logaritmera ekvationen. Logaritmen av b-värdena är:

```
-2.8350e+00
-4.7892e+00
-6.1521e+00
-7.5369e+00
-8.8262e+00
-9.9547e+00
-1.0892e+01
-1.1740e+01
-1.2494e+01
```

som är av liknande storleksordning.

Man löser sedan det ickelinjära problemet, där det är lämpligt att ta $\log(A)$ som en parameter. Det blir då mindre spridning på parametrarnas storleksordningar.

$$\min_{\log A,\ E,\ T_0} \sum_{k=1}^{n} \left[\log b_k - \log A - \frac{E}{T_k - T_0}\right]^2$$

Värdena skiljer sig dock inte från vad det viktade problemet ger. Här är en sammanställning:

	Original	Linjärt	Viktat	Symb
A	6.6872e-13	7.0725e-13	7.2491e-13	4.1547e-13
E	1.6081e+03	1.6080e+03	1.6079e+03	1.6973e+03
T0	1.5118e+02	1.5102e+02	1.5090e+02	1.4880e+02
res	1.9417e-01	1.8218e-01	1.8363e-01	1.7873e-01

där `res` är roten ur summan ovan. `Symb`-värdena har jag räknat fram med en egen rutin i Maple (via Matlab). Jag har då kunnat räkna med ett litet ϵ_{mach}.

När Peter fick se detta, sa han ungefär "plottarna ser bra ut, men jag hade inte väntat mig de parametervärdena."

Detta kan bero på att parametrarna är dåligt bestämda av målfunktionen. Två enkla exempel:

Exempel: minimera $f(x) = x^2$ och $g(y) = y^4$. Antag vi accepterar x som minimum om $f(x) \leq 10^{-8}$, $g(y) \leq 10^{-8}$. Vi får intervallen $-10^{-4} \leq x \leq 10^{-4}$ respektive $-10^{-2} \leq y \leq 10^{-2}$.

Exempel:

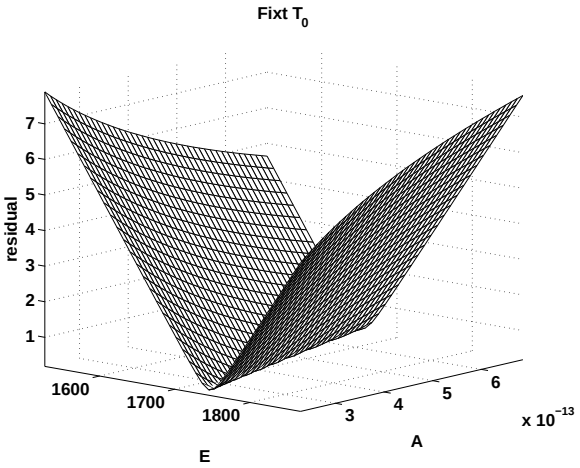
$$\min_{x_1,\ x_2} (x_1 + x_2)^2$$

Minvärdet är noll, som antas för alla x_1 och x_2 där $x_1 + x_2 = 0$. Vi har inte entydigt minimum. Målfunktionen ser ut som ett dike (ränna). Om vi rör oss utmed dikets botten ändras inte funktionen värde. Hessianen, H, är en 2×2-matris av tvåor, så H är positivt semidefinit (singulär).

En annan orsak kan vara att vi har få mätpunkter, nio, i förhållande till antalet, tre, parametrar. Det hade varit trevligt med, säg 30, mätpunkter. Tyvärr tar redan nio mätpunkter ett dygn att producera. Mätfel påverkar också resultatet.

Här kommer en bild som visar varför parametrarna är illa bestämda. Låt oss plotta följande funktion för, *fixt* T_0 (det optimala värdet), som funktion av A och E.

$$f(A, E) = \left\{ \sum_{k=1}^{n} \left[\log b_k - \log A - \frac{E}{T_k - T_0}\right]^2 \right\}^{1/2}$$



Grafen ser ut som ett dike. Övriga kombinationer, fixt A respektive fixt E ger liknande plottar.

Ett annat sätt att förstå problemen är att studera

Taylorutvecklingen av kvadratsumman, residualfunktionen r:

$$r(\log A, E, T_0) = \sum_{k=1}^{n} \left[\log b_k - \log A - \frac{E}{T_k - T_0}\right]^2$$

Om $(\log A, E, T_0)$ är optimal ($\nabla r(\log A, E, T_0) = 0$) så gäller att:

$$r(\log A + \delta_{\log A}, E + \delta_E, T_0 + \delta_{T_0}) = r(\delta_{\log A}, E, T_0) + \frac{d^T H d}{2} + \cdots$$

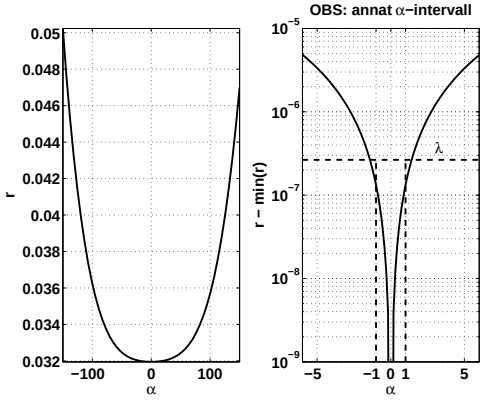
där $d^T = [\delta_{\log A}, \delta_E, \delta_{T_0}]$ och H är Hessianen av r.
H:s minsta egenvärde är $\lambda \approx 2.6 \cdot 10^{-7}$, vilket innebär att om vi går ut från optimum i egenvärdesriktningen, d (dvs. $Hd = \lambda d$), så ändrar sig knappt funktionen, dvs.

$$d^T H d = d^T \lambda d = \lambda d^T d \approx 2.6 \cdot 10^{-7}\, d^T d.$$

Jämför med $r(\log A, E, T_0) \approx 3.2 \cdot 10^{-2}$. Här är d^T:

```
-5.9815e-03    9.9969e-01   -2.4039e-02
```

Följande plot visar $r(\log(A) + \alpha d_1, E + \alpha d_2, T_0 + \alpha d_3)$ som funktion av α. Vänstra bilden är funktionsvärdet och högra avvikelsen från det minsta värdet. Observera att det är olika α-intervall. De streckade linjerna markerar $\alpha = 1$ och λ.



Egenvektorn är tangent till den blå kurvan i den plot TE kommer att visa nu (den tar så stor plats att den inte är inkluderad i dessa sidor).

Nu några ord om störningsteori. Antag att vi har *en* parameter, x, och modellen $b_k = f(x, t_k)$. Antag att vi endast har osäkerhet i b_k. Vilken osäkerhet ger det i x? Definiera

$$r(x) = \frac{1}{2} \sum_{k=1}^{n} (f(x, t_k) - b_k)^2$$

Villkor för minimum är att $r'(x) = 0$, dvs. (f' är derivatan med avseende på x):

$$0 = \sum_{k=1}^{n} (f(x, t_k) - b_k) f'(x, t_k) \qquad (\dagger)$$

Vi behöver r'' senare, och den blir (jag slutar skriva ut (x, t_k)):

$$r'' = \sum_{k=1}^{n} (f')^2 + f f'' - b_k f''$$

Antag att x beror av b, så modellen ser ut som $b = f(x(b), t_k)$ och vi är intresserade av x'_{b_j}. Vi deriverar $(\dagger)$ med avseende på b_j och får (δ_{jk} är Kroneckers delta):

$$0 = \sum_{k=1}^{n} f' x'_{b_j} f' + f f'' x'_{b_j} - \delta_{jk} f' - b_k f'' x'_{b_j} = r''(x)\, x'_{b_j} - f'(x, t_j)$$

så

$$x'_{b_j} = \frac{f'(x, t_j)}{r''(x)}$$

Antag att b_j har osäkerheten δ_{b_j}. Osäkerheten i x får vi med Taylorutveckling:

$$\Delta x = x(b_1 + \delta_{b_1}, \ldots, b_n + \delta_{b_n}) - x(b) \approx x'_{b_1} \delta_{b_1} + \ldots + x'_{b_n} \delta_{b_n} =$$

$$\frac{f'(x, t_1)\, \delta_{b_1} + \ldots + f'(x, t_n)\, \delta_{b_n}}{r''(x)} = \frac{\sum_{j=1}^{n} f'(x, t_j)\, \delta_{b_j}}{r''(x)}$$

I vårt fall har vi tre parametrar, x, y och z, säg och man kan då visa (övning) att med

$$r(x) = \frac{1}{2}\sum_{k=1}^{n} (f(x,y,z,t_k) - b_k)^2$$

så blir

$$\begin{bmatrix} r''_{xx} & r''_{xy} & r''_{xz} \\ r''_{yx} & r''_{yy} & r''_{yz} \\ r''_{zx} & r''_{zy} & r''_{zz} \end{bmatrix} \begin{bmatrix} x'_{b_j} \\ y'_{b_j} \\ z'_{b_j} \end{bmatrix} = \begin{bmatrix} f'_x(x,y,z,t_j) \\ f'_y(x,y,z,t_j) \\ f'_z(x,y,z,t_j) \end{bmatrix}$$

där andraderivatsmatrisen är Hessianen, $\boldsymbol{H}$. Alltså gäller att osäkerheten i parametrarna ges av:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \approx \begin{bmatrix} r''_{xx} & r''_{xy} & r''_{xz} \\ r''_{yx} & r''_{yy} & r''_{yz} \\ r''_{zx} & r''_{zy} & r''_{zz} \end{bmatrix}^{-1} \sum_{j=1}^{n} \begin{bmatrix} f'_x(x,y,z,t_j) \\ f'_y(x,y,z,t_j) \\ f'_z(x,y,z,t_j) \end{bmatrix} \delta_{b_j} \quad (\ddagger)$$

Övningar: vad ger ovanstående för en *linjär* modell?

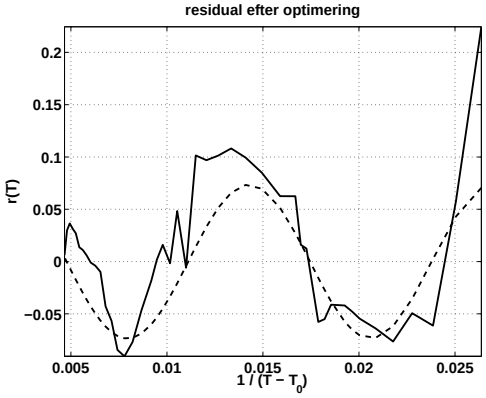
Om man anpassar ($\ddagger$) till vårt problem (med den logaritmerade modellen) så stämmer formeln utmärkt. Låt $\boldsymbol{W}$ vara matrisen

$$\begin{bmatrix} 8.0 & -0.8 & -5.1 & -6.3 & -5.4 & -3.1 & 0.3 & 4.4 & 8.9 \\ -1439 & 79 & 835 & 1068 & 940 & 562 & 11 & -656 & -1400 \\ 37.4 & -0.2 & -19.2 & -25.4 & -22.9 & -14.3 & -1.5 & 14.2 & 31.8 \end{bmatrix}$$

då är

$$\begin{bmatrix} \Delta A/A \\ \Delta E \\ \Delta T0 \end{bmatrix} \approx \boldsymbol{W} \begin{bmatrix} \delta_{b_1}/b_1 \\ \vdots \\ \delta_{b_9}/b_9 \end{bmatrix}$$

Man kan säga mer om detta problem, med det kräver matematik och matematisk statistik som vi inte har gått igenom. Frågan är dock om man inte borde arbeta vidare med modellen. Residualvektorn ser inte speciellt slumpmässig ut. I följande bild har jag ritat $\log b_k - \log A - E/(T_k - T_0)$ som funktion av $1/(T_k - T_0)$. Mätdata kommer från en annan mätserie. Den steckade linjen är en anpassad sinuskurva.



Avslutningsvis: detta är ett vanligt problem. Här resultat från Google:

```
arrhenius "least squares" 160,000  träffar
"arrhenius equation"      371,000
"vogel-fulcher-tammann"     3,340
"vogel-fulcher-tamman"      1,610
```


Ickelinjära ekvationer

$$f(x) = 0, \quad f : \Re \to \Re$$

Vi kan också ha system av ekvationer:

$$\begin{cases} f(x,y,z) = 0 \\ g(x,y,z) = 0 \\ h(x,y,z) = 0 \end{cases}$$

Exempel:

$$\begin{cases} x^2 + y^2 - 2 = 0 \\ x - y = 0 \end{cases}$$

med rötter $(1,1)$ och $(-1,-1)$.

En ickelinjär ekvation kan ha $0, 1, 2, 3, \ldots, \infty$ lösningar.

Ett linjärt problem ($Ax = b$) har $0, 1$ eller ∞ många.

Det kan tänkas att f är definierad via en procedur.

$$f(x) = \int_{-4}^{x} (1+t)e^{-t^2} \sin t \; dt$$

Flertalet metoder:

- Startas med en (eller flera) approximation(er).
- Skapar en sekvens av approximationer som förhoppningsvis konvergerar mot nollstället.
- Kan divergera.
- Försöker att hitta ett nollställe åt gången.

Halveringsmetoden (bisektionsmetoden)

Givet en kontinuerlig funktion f och $p, n \in \Re$ med $f(n) < 0$, $f(p) > 0$.

```
while |n − p| > tol do
    m = (n + p)/2
    if f(m) < 0 then    ! borde ta hand om exakt likhet också
        n = m
    else
        p = m
    endif
end
```

Om begynnelseintervallet har längden τ har intervallet längden

$$\frac{\tau}{2^k}$$

efter k iterationer.

Halveringsmetodens fördelar

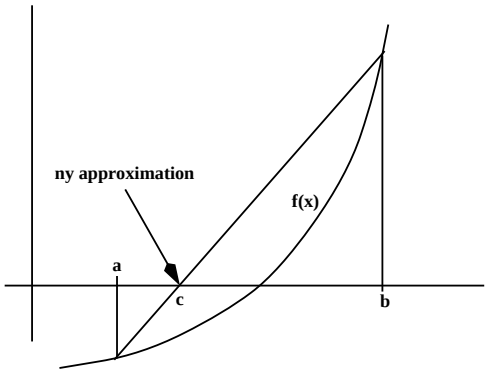
- konvergerar alltid
- räcker att f är kontinuerlig
- får ett intervall där roten ligger
- deterministisk i antal steg

och nackdelar

- kan ej generaliseras till system
- långsam
- kan vara svårt att hitta p och n

"långsam men säker"

Snabbare metoder: lös ett svårt problem genom att lösa en sekvens av enklare problem.
Linjärisering, approximera f med en linjär funktion.



Sekanten (den räta linjen) har ekvationen

$$y(x) = \frac{f(b) - f(a)}{b - a}(x - a) + f(a)$$

varför c ges av

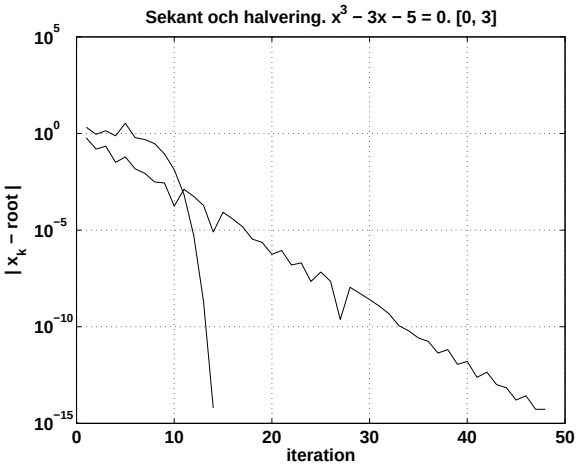
$$c = a - f(a)\frac{b - a}{f(b) - f(a)} = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

Iterera: givet två startvärden x_0, x_1

$$x_{k+1} = x_k - f(x_k)\frac{x_{k-1} - x_k}{f(x_{k-1}) - f(x_k)}, \quad k = 1, 2, \ldots$$

Om f är linjär ger detta nollstället i ett steg.

Exempel: $f(x) = x^3 - 2x - 5$.



Sekantmetoden konvergerar i en iteration om funktion är linjär, ett faktum som använts i flera hundra år.

'A discussion of sheep' is a problem taken from Robert Record's "Grounde of Artes" (London, 1542?).

There is supposed a lawe made that (for furtheryng of tyllage) every man that doth kepe shepe, shall for every 10 shepe eare and sowe one acre of grounde, and for his allowance in sheepe pasture there is appointed for every 4 shepe one acre of pasture. Nowe is there a ryche shepemaister whyche hath 7000 akers of grounde, and would gladlye kepe as manye sheepe as he myght by that statute. I demaunde howe many shepe shall he kepe?

Fyrste I suppose he maye kepe 500 shepe, and for them he shall have in pasture, after the rate of 4 shepe to an acre, 125 akers, and in earable grounde 50 acres that is 175 in all, but this errour is to litell by 6825. Therefore I gesse agayn that he maye kepe 1000 shepe, that is in pasture 250 akers, and in tyllage 100 akers, which maketh 350, that is to lytle by 6650.

These bothe erroures with theyr positions I sette downe as you see, and multiply in crosse 6825 by 1000, and it maketh 6825000. Then I multiply 6650 by 500, and it doth amount to 3325000, which sum I do subtract out of the fyrst, & there remaineth 3500000, as the dividende. Also I doo subtract the lesser errour out of the greater, and so remayneth 175, by which I divide the said dividende, and the quotient will be 20000, so that I see that by this rate he that hathe 7000 acres of ground may keepe 20000 shepe: & therby I conjecture that many menne may kepe so many shepe: for many men (as the common talke is) have so many acres of grounde.

The 'equals' symbol '=' appears in Recorde's book "The Whetstone of Witte" published in 1557. He justifies using two parallel line segments "bicause noe 2 thynges can be moare equalle".

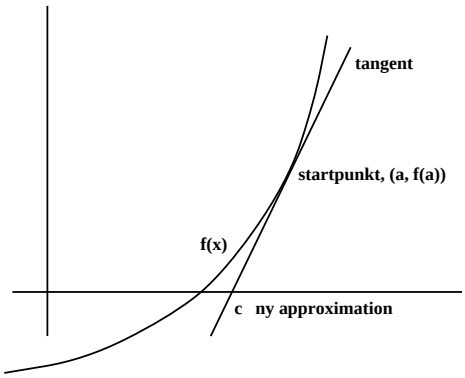
Problem 12, kapitel 7 i "Nio kapitel om matematik".
(1 cùn = $\frac{1}{10}$ chī)

Two rats gnawing towards each other

Jīn yǒu héng hòu wǔ chī,	Now let there be a wall of thickness 5 chī,
liǎng shǔ duì chuān.	two rats gnawing towards (each other).
Dà shǔ rì yī chī,	The big rat gnaws on the first day 1 chī,
xiǎo shǔ yì rì yī chī,	the small rat also on the first day 1 chī.
Dà shǔ rì zì bèi,	The big rat gnaws everyday twice itself,
xiǎo shǔ rì zì bàn,	The small rat gnaws everyday half itself.
Wèn	Question:
jǐ hé rì xiāng féng,	(In) how many days (do they) meet,
gè chuāng jǐ hé.	how much has each one gnawn?
Dá yuē:	The answer says:
èr rì shí qī fēn rì zhī èr.	(In) $2\frac{2}{17}$ days.
Dà shǔ chuān	The big rat has gnawn
sān chī sì cùn shí qī fēn cùn zhī shí èr,	3 chī $4\frac{12}{17}$ cùn,
xiǎo shǔ chuān	the small rat has gnawn
yī chī wǔ cùn shí qī fēn cùn zhī wǔ.	1 chī $1\frac{5}{17}$ cùn.
Shù yuē:	The rule says:
Jiǎ lìng èr rì,	Assuming 2 days,
bù zú wǔ cùn.	the deficit is 5 cùn.
Lìng zhī sān rì,	Let it (be) 3 days,
yǒu yú sān chī qī cùn bàn.	there is 3 chī $7\frac{1}{2}$ cùn in excess.

Newtons metod

Kan approximera med tangenten i stället för med sekanten (Newton-Raphson, 1690).



Tangenten har ekvationen:

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

När $x = c$ är $y = 0$

$$c = a - \frac{f(a)}{f'(a)}$$

Iterera

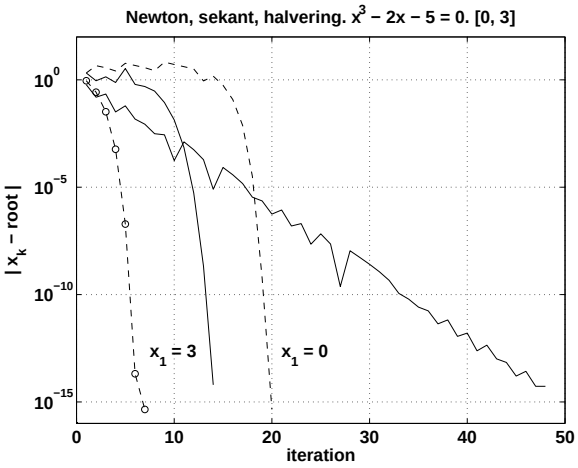
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Kräver endast ett startvärde, men måste ha derivatan.

Exempel: Newtons eget exempel $x^3 - 2x - 5 = 0$.
Iterationen blir

$$x_{k+1} = x_k - \frac{x_k^3 - 2x_k - 5}{3x_k^2 - 2}$$

I bilden nedan har vi testat med de två startvärdena $x_0 = 0$ respektive $x_0 = 3$.



Startvärdet viktigt! (roten = 2.0946)

Hur skall vi karakterisera de olika konvergenshastigheterna?

Om $f(x^*) = 0$ och $\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^r} = C$ konstant $< \infty$

så säger vi att metoden har konvergensordning r

- om $r = 1$ och $C < 1$ så har vi linjär konvergens
- om $r > 1$ så har vi superlinjär konvergens
- om $r = 2$ så har vi kvadratisk konvergens

Vad innebär $r = 1$? Om x_0 ligger tillräckligt när x^* så gäller att:

$$|x_1 - x^*| \approx C|x_0 - x^*|, \quad |x_2 - x^*| \approx C|x_1 - x^*| \approx C^2|x_0 - x^*|$$

Dvs. $|x_k - x^*| \approx C^k|x_0 - x^*|$

Exempel: $|x_{k+1} - x^*| \approx C|x_k - x^*|^r$. Antag att $|x_0 - x^*| = 0.1$ och $C = 0.1$, då är $|x_k - x^*|$:

k	linjär r = 1	superlinjär r = 1.618	kvadratisk r = 2
0	1e-1	1e-1	1e-1
1	1e-2	2e-3	1e-3
2	1e-3	6e-6	1e-7
3	1e-4	3e-10	1e-15
4	1e-5	5e-17	1e-31

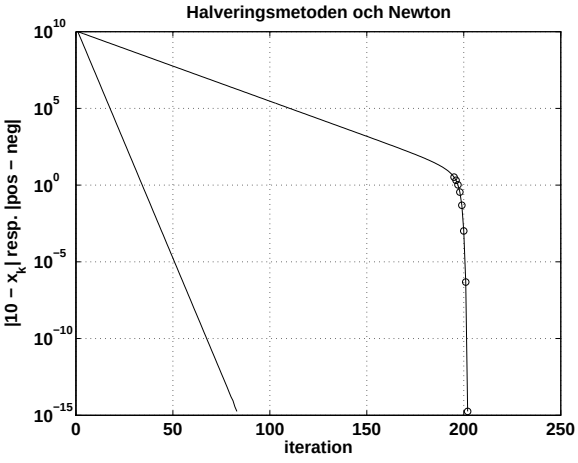
Normalt (nära lösningen för sekant och Newton) är:

- Halveringsmetoden linjär med $C = 0.5$.
- Sekantmetoden superlinjär med $r = (1 + \sqrt{5})/2 \approx 1.618$
- Newtons metod kvadratiskt konvergent (om enkelrot)

Exempel: lös med Newtons metod och halveringsmetoden

$$x^{10} - a = 0, \quad a = 10^{10}$$

Använd det urusla startvärdet a ($[0, a]$ för halveringsmetoden). Uruselt eftersom $x^* = 10$.



Konvergensordningen är definierad som ett gränsvärde. Det kan krävas många steg innan x_k ligger så nära x^* att den kvadratiska konvergensen sätter igång.

Om den kommer igång. Både Newtons metod och sekantmetoden kan divergera. "Snabba men osäkra."

Hybridmetoder: använd "dyra" Newton där det lönar sig och en billig metod för övrigt.

Vilken metod är billigare, Newton eller sekant?

Sekantmetoden kräver ett funktionsvärde i varje steg.
Newton kräver både ett funktionsvärde och en derivata men konvergerar snabbare (nära nollstället).

Vi är normalt inte intresserade av att minimera antalet iterationer. Det viktiga är det totala körtiden och minnesbehovet.

- få komplexa iterationer
- många enkla iterationer

Den metodoberoende feluppskattningen

Givet approximationen $\hat{x}$ och det exakta värdet x^* hur skall vi uppskatta $|\hat{x} - x^*|$?
Det vi kan beräkna är "residualen" $f(\hat{x})$.

Medelvärdessatsen:

$$f(\hat{x}) = f(x^*) + (\hat{x} - x^*)f'(\xi), \quad \xi \in (\hat{x}, x^*)$$

Antag att $f'(\xi)$ är kontinuerlig med $M = \min |f'(\xi)|, \xi \in [\hat{x}, x^*]$.

Om då $M > 0$ gäller att:

$$|\hat{x} - x^*| \leq |f(\hat{x})|/M$$

M kan vara noll. Tag $f(x) = x^2$ (så nollan är dubbelrot).
Då är både $f(0) = 0$ och $f'(0) = 0$.

Exempel:

$$f(x) = 1/x - 1/10, \quad \hat{x} = 11, \quad f(\hat{x}) = 1/11 - 1/10 = -1/110,$$

$$|f'(\xi)| = 1/\xi^2, \quad |\hat{x} - x^*| \leq \frac{|1/11 - 1/10|}{1/11^2} = 1.1$$

f är strängt avtagande med $f(9) > 0$ och $f(11) < 0$ varför $(9, 11)$ innehåller precis en rot. Beloppet av derivatan är $1/x^2$ som är strängt avtagande. Det minsta värdet på derivatan, i intervallet, är därför $1/11^2$.

Avbrottskriterium

I sekantmetoden får vi inte en sekvens av intervall som innehåller roten. Metoden kan ju till och med divergera. Så, hur vet vi när vi skall avsluta iterationen? Vi har ett avbrottskriterium som kontrollerar:

- k, för att undvika oändliga loopar (divergens, eller för små toleranser)
- $|x_k - x_{k-1}|$, borde gå mot noll vid konvergens, men litet värde kan betyda att det går långsamt
- $|f(x_k)|$, noll i lösningen (tänk också på $|f(x_k)|/M$)

Första försöket: avsluta om ($\vee$ = eller):

$$k > maxit \quad \vee \quad |x_k - x_{k-1}| \leq tol_x \quad \vee \quad |f(x_k)| \leq tol_f$$

$maxit$ (max antal iterationer), tol_x och tol_f ges av användaren. Man kan givetvis kräva att $|x_k - x_{k-1}| \leq tol_x$ & $|f(x_k)| \leq tol_f$. Inte skalningsoberoende: $10^5 f(x) = 0$ borde helst fungera lika bra som $f(x) = 0$. Motsvarande för $f(10^5 x) = 0$. Toleranserna måste skalas efter problemet.

Andra försöket: avsluta om:

$$k > maxit \quad \vee \quad |x_k - x_{k-1}| \leq tol_x|x_0| \quad \vee \quad |f(x_k)| \leq tol_f|f(x_0)|$$

Fungerar inte om $x_0 = 0$. Vi skulle kanske kunna uppskatta derivatan för att få något i stil med $|\hat{x} - x^*| \leq |f(\hat{x})|/M$.

Det är inte enkelt att utforma ett säkert och effektivt kriterium. Ett kriterium går alltid att lura eftersom vi endast känner funktionen (och kanske derivatan) i ett ändligt antal punkter. Det finns oändligt många funktioner som går genom dessa punkter.

Newton för system

Repetition av Taylors formel.

$$\begin{bmatrix} f(a+h, b+k) \\ g(a+h, b+k) \end{bmatrix} = \begin{bmatrix} f(a,b) \\ g(a,b) \end{bmatrix} + \begin{bmatrix} \frac{\partial f(a,b)}{\partial x}h + \frac{\partial f(a,b)}{\partial y}k \\ \frac{\partial g(a,b)}{\partial x}h + \frac{\partial g(a,b)}{\partial y}k \end{bmatrix} + \dots =$$

$$\begin{bmatrix} f(a,b) \\ g(a,b) \end{bmatrix} + \begin{bmatrix} \frac{\partial f(a,b)}{\partial x} & \frac{\partial f(a,b)}{\partial y} \\ \frac{\partial g(a,b)}{\partial x} & \frac{\partial g(a,b)}{\partial y} \end{bmatrix} \begin{bmatrix} h \\ k \end{bmatrix} + \dots$$

Matrisen av partiella derivator kallas Jacobianen.

Vi står i (x_j, y_j) och vill hitta korrektioner, (h, k), så att $f(x_j + h, y_j + k) = 0$ och $g(x_j + h, y_j + k) = 0$.

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f(x_j + h, y_j + k) \\ g(x_j + h, y_j + k) \end{bmatrix} \approx \begin{bmatrix} f(x_j, y_j) \\ g(x_j, y_j) \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\partial f(x_j,y_j)}{\partial x} & \frac{\partial f(x_j,y_j)}{\partial y} \\ \frac{\partial g(x_j,y_j)}{\partial x} & \frac{\partial g(x_j,y_j)}{\partial y} \end{bmatrix}}_{J(x_j,y_j)} \begin{bmatrix} h \\ k \end{bmatrix}$$

Om Jacobianen, J, är ickesingulär kan vi få de approximativa korrektionerna:

$$\begin{bmatrix} h \\ k \end{bmatrix} \approx -J^{-1}(x_j, y_j) \begin{bmatrix} f(x_j, y_j) \\ g(x_j, y_j) \end{bmatrix}$$

Iterera!

$$\begin{bmatrix} x_{j+1} \\ y_{j+1} \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} - J^{-1}(x_j, y_j) \begin{bmatrix} f(x_j, y_j) \\ g(x_j, y_j) \end{bmatrix}$$

Jämför med det skalära fallet:

$$x_{j+1} = x_j - f(x_j)/f'(x_j)$$

Vi räknar naturligtvis inte ut inversen utan löser systemet:

$$J(x_j, y_j) \begin{bmatrix} h \\ k \end{bmatrix} = - \begin{bmatrix} f(x_j, y_j) \\ g(x_j, y_j) \end{bmatrix}$$

Exempel : $\begin{cases} x^2+y^2-2=0 \\ xy-\frac{1}{2}=0 \end{cases}$

Våra funktioner är alltså: $\begin{cases} f(x,y)=x^2+y^2-2 \\ g(x,y)=xy-\frac{1}{2} \end{cases}$

Newtons metod blir:

$$\begin{bmatrix} x_{j+1} \\ y_{j+1} \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} 2x_j & 2y_j \\ y_j & x_j \end{bmatrix}^{-1} \begin{bmatrix} x_j^2+y_j^2-2 \\ x_jy_j-\frac{1}{2} \end{bmatrix}$$

Om vi startar med $x_0=-3$ och $y_0=10$ så får vi följande approximationer:

```
  -3.0000e+00  -1.4121e+00  -5.4236e-01  -1.4188e-02
   1.0000e+01   5.1264e+00   2.8033e+00   1.8081e+00

   2.7380e-01   3.5877e-01   3.6597e-01   3.6603e-01
   1.4593e+00   1.3733e+00   1.3661e+00   1.3660e+00

   3.6603e-01   3.6603e-01   3.6603e-01
   1.3660e+00   1.3660e+00   1.3660e+00

>> fel =
  -3.3660e+00  -1.7781e+00  -9.0838e-01  -3.8021e-01
   8.6340e+00   3.7603e+00   1.4373e+00   4.4208e-01

  -9.2230e-02  -7.2583e-03  -5.1931e-05  -2.6966e-09
   9.3297e-02   7.2586e-03   5.1931e-05   2.6966e-09

            0            0            0
            0            0            0
```


Om man arbetar med stora system kan man inte ha variabler $x, y, z, w, \ldots$ utan vi får använda vektorer, analogt för funktionerna.

Exemplet kan skrivas på följande vis. x och y får vara elementen x_1 respektive x_2 i vektorn (kolonnmatrisen) x.

$$\begin{cases} x_1^2+x_2^2-2=0 \\ x_1x_2-\frac{1}{2}=0 \end{cases}$$

Vår funktion, f, med två komponenter är:

$$\begin{cases} f_1(x_1,x_2)=x_1^2+x_2^2-2 \\ f_2(x_1,x_2)=x_1x_2-\frac{1}{2} \end{cases}$$

Normalt skriver vi bara $f(x)=0$ där f, x och 0 är vektorer. f är alltså en vektorvärd funktion som beror av en vektor.

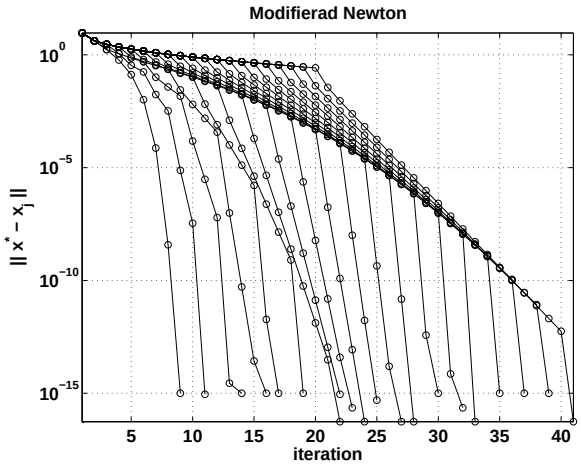
Newtons metod blir (notera placeringen av iterationsindex):

$$\begin{bmatrix} x_1^{(j+1)} \\ x_2^{(j+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(j)} \\ x_2^{(j)} \end{bmatrix} - \begin{bmatrix} 2x_1^{(j)} & 2x_2^{(j)} \\ x_2^{(j)} & x_1^{(j)} \end{bmatrix}^{-1} \begin{bmatrix} (x_1^{(j)})^2+(x_2^{(j)})^2-2 \\ x_1^{(j)}x_2^{(j)}-\frac{1}{2} \end{bmatrix}$$

Allmänt:

$$x^{(j+1)} = x^{(j)} - J^{-1}(x^{(j)})f(x^{(j)})$$

Dyrt och komplicerat att beräkna $J(x)$. Alternativ? Modifierad Newton: Beräkna $J(x^{(j)})$ då och då (inte i varje iteration).



Differensapproximation av J; slipper beräkna de n^2 derivatorna explicit. Om f är given via en algoritm kanske det inte är möjligt att beräkna derivatorna. Välj ett lämpligt tal δ (se övning):

$$f(x+\delta e_i) \approx f(x) + \delta J e_i$$

eller

$$Je_i \approx \frac{f(x+\delta e_i)-f(x)}{\delta}$$

Fixpunkter och lite teori

Upprepade tryckningar på `cos`-knappen. Tre olika startvärden.

```
  -5.0000e+00            0   2.0000e+01
   2.8366e-01   1.0000e+00   4.0808e-01
   9.6004e-01   5.4030e-01   9.1788e-01
   5.7349e-01   8.5755e-01   6.0750e-01
   8.4001e-01   6.5429e-01   8.2108e-01
   6.6745e-01   7.9348e-01   6.8143e-01
   7.8540e-01   7.0137e-01   7.7667e-01
   7.0711e-01   7.6396e-01   7.1325e-01
   7.6025e-01   7.2210e-01   7.5624e-01
   7.2467e-01   7.5042e-01   7.2742e-01
   7.4872e-01   7.3140e-01   7.4689e-01
   7.3256e-01   7.4424e-01   7.3380e-01
   7.4346e-01   7.3560e-01   7.4263e-01
   7.3613e-01   7.4143e-01   7.3669e-01
   7.4107e-01   7.3751e-01   7.4070e-01
   7.3774e-01   7.4015e-01   7.3800e-01
   7.3999e-01   7.3837e-01   7.3982e-01
   7.3848e-01   7.3957e-01   7.3859e-01
   7.3949e-01   7.3876e-01   7.3942e-01
   7.3881e-01   7.3930e-01   7.3886e-01
```

Så, i varje kolumn har vi $\cos(\cos(\cos(\cos(\ldots\cos(x_0)\ldots))))$. Detta kan skrivas på formen $x_{k+1}=\cos x_k$.

- Iterationen verkar konvergera
- Gör den alltid det?
- Hur snabbt konvergerar det?
- Kan vi använda detta till något?