TANA23 Föreläsning 2

- Talrepresentation i datorer. Flyttalssystem.
- Datoraritmetik, Beräkningsfelsanalys och Kancellation.
- Icke-linjära ekvationer, Fixpunktsiteration.

Talrepresentation i Datorer

Tal kan skrivas på *exponentform* eller som *flyttal*. Exempelvis är

$$763.45 = 7.6345 \cdot 10^2.$$

Ett *flyttal* är *normaliserat* om det endast finns en siffra framför decimalpunkten. Talet har *heltalsdelen* 7 och *bråkdelen* 0.6345.

Detta betyder egentligen

$$(7 \cdot 10^0 + 6 \cdot 10^{-1} + 3 \cdot 10^{-2} + 4 \cdot 10^{-3} + 5 \cdot 10^{-4})10^2.$$

Vi har alltså ett *positionssystem* med basen 10.

Hur ser datorn flyttalssystem ut?

Definition Ett *flyttalssystem* karakteriseras av parametrar (β, t, L, U), där β är talsystemets *bas*, t är antalet siffror i bråkdelen, och L och U är systemets minsta respektive största exponent.

Exempel Talsystemet $(10, 3, -9, 9)$ innehåller exempelvis talen

$$4.562, \quad 123.7, \quad \text{och } 0.006532.$$

Talet 0 kan inte skrivas som ett normaliserat flyttal.

IEEE 754 Enkel Precision $(2, 23, -126, 127)$

I datorn lagras talet som ett *ord* (32 bitar). Bitarna fördelas som

s (1 bit)	e (8 bitar)	f (23 bitar)

I *Normalfallet*, $1 \leq e \leq 254$, gäller att flyttalet skall tolkas som,

$$x = (-1)^s (1.f)_2 \cdot 2^{e-127}.$$

Undntagsfallen $e = 0$ eller $e = 255$ ger möjlighet att definiera $x = 0$, $x = \pm\infty$, och $x = \text{NaN}$.

Exempel Hur lagras talet 13.25 i datorn? Vilket är det största talet som kan lagras?

Observation Då vi lagrar x=0.1 i flyttalssystemet $(2, 23, -126, 127)$ fås

$$x = (0.1)_{10} = (0.0001100110011\ldots)_2 = (1.1001100110011\ldots)\cdot 2^{-4}$$

Med 23 bitar i bråkdelen blir inte $x = 0.1$ lagrat exakt på datorn. Ett avrundningsfel $|x - xr| \leq 2^{-27} = 7.45 \cdot 10^{-9}$ görs. Är det viktigt?

> Ett tal som kan lagras exakt i det decimala talsystemet kan inte säkert lagras exakt i det binära.

Felen är små men datorer kan göra *många* beräkningar *snabbt*.

Example Låt $h = 0.1$ och räkna upp x tills den når 1. I Python

```
h=0.1
x=0.0
while x<1.0:
    x=x+h
```

programmet stannar då $x = 1.0999999999999999$. Skriv istället

```
while abs(x-1)<h/2:
```

eller använd en `for`-loop.

> Testa inte likhet mellan reella tal!

Datoraritmetik och Beräkningsfel

Exempel Då vi lagrar tal i ett flyttalssystem gör vi ett *avrundningsfel*. Antag att talet $x = 573.672$ skall lagras i talsystemet $(10, 3, -9, 9)$.

Hur stort fel kommer vi att göra?

Exempel Antag att vi vill addera $x = 34.23$ och $y = 85.28$. Vad är den bästa tänkbara gränsen för beräkningsfelet om vi räknar i talsystemet $(10, 3, -9, 9)$?

Avrundningsfel i Flyttalssystem

> **Sats** Då ett tal x lagras i flyttalssystemet (β, t, L, U) görs ett *relativt fel* högst
>
> $$\frac{|x - x_r|}{|x|} \leq \frac{1}{2}\beta^{-t},$$
>
> där x_r är det tal i talsystemet som ligger närmast x.

> **Definition** Konstanten $\mu = \frac{1}{2}\beta^{-t}$ kallas talsystemets *avrundningsenhet*.

Aritmetriska Operationer i Flyttalssystem

Antag att vi räknar i talsystemet (β, t, L, U). Då gäller

Sats Då en aritmetrisk operation $x \odot y$ utförs gäller att

$$\frac{|x \odot y - \mathrm{fl}[x \odot y]|}{|x \odot y|} \leq \mu$$

där $\mathrm{fl}[x \odot y]$ resultatet beräknat inom talsystemet, μ är avrundningsenheten, och $\odot$ betyder $+$, $-$, $*$, eller $/$.

Tolkning Räkna först *exakt* och avrunda svaret till flyttalssystemet.

Det går att implementera standard funktioner $\exp(x)$, $\log(x)$, $\sqrt{x}$,..., så att de beräknas med relativt fel högst μ.

Exempel Vi vill beräkna $a + b + c$ i talsystemet $(10, 3, -9, 9)$ då $a = 9.876 \cdot 10^4$, $b = -9.880 \cdot 10^4$, och $c = 3.456 \cdot 10^1$.

Vi kan välja mellan alternativen

$$\begin{aligned} \mathrm{fl}[\mathrm{fl}[a+b]+c] &= \mathrm{fl}[\mathrm{fl}[-0.004 \cdot 10^4] + 3.456 \cdot 10^0] \\ &= \mathrm{fl}[-4.000 \cdot 10^1 + 3.456 \cdot 10^1] = -5.440 \cdot 10^0. \end{aligned}$$

eller

$$\mathrm{fl}[a + \mathrm{fl}[b+c]] = \mathrm{fl}[9.876 \cdot 10^4 - 9.877 \cdot 10^4] = -1.000 \cdot 10^1.$$

Gör en beräkningsfelsanalys och avgör vilket alternativ som är bäst.

Förutsättning Alla beräkningar inom flyttalssystemet utförs med relativt fel högst avrundningsenheten $\mu = 0.5 \cdot 10^{-3}$.

Beräkningsfelsanalys

Exempel Beräkna $f(x) = (1 - x^2)\mathrm{e}^{-x/2}$, för $x = 0.37$, på en dator och uppskatta samtidigt beräkningsfelet. Avgör även om det finns risk för kancellation.

Lösning Inför beräkningsordningen

$$f(x) = (1 - x^2)\mathrm{e}^{-x/2} = (1 - a)b = cb = e,$$

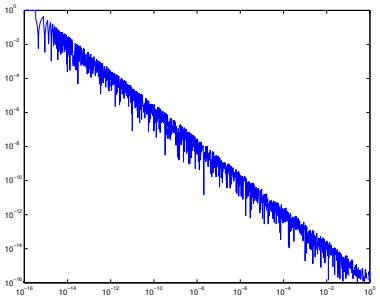
och antag att de relativa felen i mellanresultaten är mindre än μ, eller $|\Delta a|/|a| \leq \mu$, etc.

Använd sedan felfortplantningsformeln för att uppskatta $|\Delta f|$.

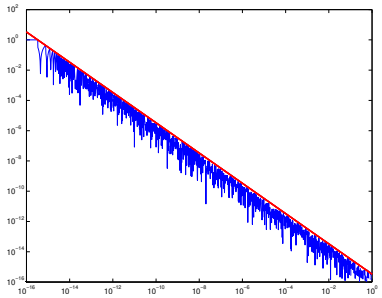
Kommentar Det är viktigt att beräkningsordningen överensstämmer med hur datorn räknar.

Exempel Vi vill beräkna $f(x) = \sqrt{1+x} - 1$, för små x. I Matlab

```
>> x=10.^-(0:0.01:16); f=sqrt(1+x)-1;
>> loglog(x,abs(f-f_ex)./f_ex);
```



Gör en beräkningsfelsanalys som förklarar resultatet.



Relativa felet,

$$\frac{|\mathrm{fl}[f(x)] - f(x)|}{|f(x)|}, \qquad f(x) = \sqrt{1+x} - 1.$$

och felgränsen $$\frac{|\Delta f|}{|f|} \leq \frac{3\mu}{x}.$$

Kancellation gör att relativa felet *växer* då x minskar! Hur kan vi åtgärda problemet? Lämplig omskrivning?

Flyttalsaritmetik och Beräkningsfelsanalys

Viktiga saker att komma ihåg är

- Kan anta att alla beräkningar utförs med ett *relativt fel* högst μ.
- Beräkningsordningen är viktig. Alltså gäller

$$\mathrm{fl}[a + (b + c)] \neq \mathrm{fl}[(a + b) + c].$$

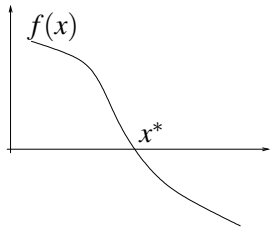
- Matematiskt ekvivalenta uttryck kan ge väldigt olika resultat. Omskrivningen

$$\sqrt{1+x} - 1 = \frac{x}{\sqrt{1+x}+1},$$

undviker kancellationen.
- Gör man inga misstag brukar beräkningsfel orsakade av flyttalssystemet vara försumbara jämfört med andra fel.

Icke-Linjära Ekvationer

Vi vill lösa en icke-linjär ekvation $f(x) = 0$.

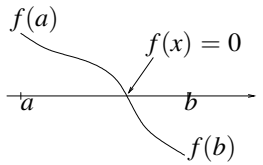


Frågor Existens och Entydighet? Bra numeriska metoder? Feluppskattning?

Vi antar att vi kan beräkna $f(x)$ och eventuellt $f'(x)$.

Existens

Sats Om $f(x)$ är kontinuerlig på intervallet $[a,b]$ och $c \in [f(a), f(b)]$ då finns det ett $x \in [a,b]$ sådant att $f(x) = c$.



Lemma Antag att $f(x)$ är kontinuerlig. Om $f(a)f(b) < 0$ så finns det en rot x^* till ekvationen $f(x) = 0$ i intervallet (a,b).

Detta ger ett bra kriterium för existens av lösning till $f(x) = 0$.

Ekvationslösning och Iterationsmetoder

Metod Givet en *start approximation* x_0 konstruerar vi en följd $\{x_k\}_{k=1}^{\infty}$. Metoden är *konvergent* om

$$x_k \to x^*, \quad \text{as } k \to \infty,$$

där x^* är en *rot* till ekvationen $f(x) = 0$.

En *start approximation* fås genom *grovlokalisering*.

Andragrads ekvationer $x^2 + ax + c = 0$ kan lösas med en *explicit formel*. Det går inte för de allra flesta ekvationer.

Fixpuntsiteration

Definition En *fixpunktsiteration* kan skrivas på formen

$$x_{k+1} = \varphi(x_k), \qquad k = 0, 1, 2, \ldots$$

En *fixpunkt* x^* till *iterationsfunktionen* $\varphi(x)$ satisfierar $x^* = \varphi(x^*)$.

Helst skall en fixpunkt x^* vara en rot till ekvationen $f(x) = 0$.

Exempel Vi vill lösa $f(x) = x - \mathrm{e}^{-x} = 0$ och prövar metoderna

(i) $x_{k+1} = \mathrm{e}^{-x_k} = \varphi_1(x_k)$ $\qquad$ (ii) $x_{k+1} = -\log(x_k) = \varphi_2(x_k)$.

Vad händer?

(i) Metod: $x_{k+1} = \mathrm{e}^{-x_k}$

k	x_k
1	0.57694981
2	0.56160877
3	0.57029086
4	0.56536097
5	0.56815502
10	0.56708395
20	0.56714309
30	0.56714329

Vi får en rot $x^* \approx 0.56714329$.

(ii) Metod: $x_{k+1} = -\log(x_k)$

k	x_k
1	0.59783700
2	0.51443714
3	0.66468192
4	0.40844668
5	0.89539391
6	0.11049153
7	2.20281638
8	-0.78973671

Vi får divergens!

Hur skall vi avgöra om en metod konvergerar eller ej?
Konvergenshastigheten?

Fixpunktsteori

Sats Antag att $\varphi(x)$ har en reell fixpunkt x^* samt att $|\varphi'(x)| \le m < 1$ i en omgivning av x^*. Då gäller, om x_0 väljs tillräckligt nära x^*, att

$$\lim_{k\to\infty} x_k = x^*.$$

Exempel Vi vill hitta den positiva roten x^* till ekvationen,

$$f(x) = x - \mathrm{e}^{x} = 0.$$

med hjälp fixpunktsiteration. Avgör om metoderna

(i) $x_{k+1} = \mathrm{e}^{-x_k} = \varphi_1(x_k)$, $\qquad$ (ii) $x_{k+1} = -\log(x_k) = \varphi_2(x_k)$,

är konvergenta eller inte.

Sats För en konvergent fixpunktsiteration $x_{k+1} = \varphi(x_k)$ gäller att

$$|x_{k+1} - x^*| \leq m|x_k - x^*|$$

där x^* är fixpunkten.

Detta kallas *linjär konvergens*. Bättre med en mindre konstant m.

Exempel Vi vill lösa ett ekvationssystem

$$f(x) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 - 1 \\ (x_1 - 0.5)^2 + x_2^2 - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

och prövar följande fixpunktsiteration

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}), \quad \text{med } x_0 = (0.4, 1.2)^T.$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$\|x^{(k)} - x^*\|$
0	0.4000	1.2000	$2.76 \cdot 10^{-1}$
1	-0.2000	0.7500	$5.00 \cdot 10^{-1}$
5	0.2661	0.8528	$1.17 \cdot 10^{-1}$
10	0.2034	0.9652	$4.67 \cdot 10^{-2}$
15	0.2581	0.9686	$8.10 \cdot 10^{-3}$
20	0.2486	0.9682	$1.43 \cdot 10^{-3}$

Oftast enkelt att hitta fixpunktsiterationer som konvergerar.