

Datoraritmetik

Beräkningsvetenskap DV

Institutionen för Informationsteknologi, Uppsala Universitet

5 september, 2012

Från labben

- Två huvudtyper av fel: *diskretiseringfel* och *avrundningsfel*
- Olika sätt att mäta fel: *relativt fel*, *absolut fel*
- Begreppen ε_M (maskinepsilon), **Inf**, **NaN**, overflow, underflow, diskretisering
- Beräkningen $A^{-1}A$ blev inte riktigt enhetsmatrisen

Observationer från labben

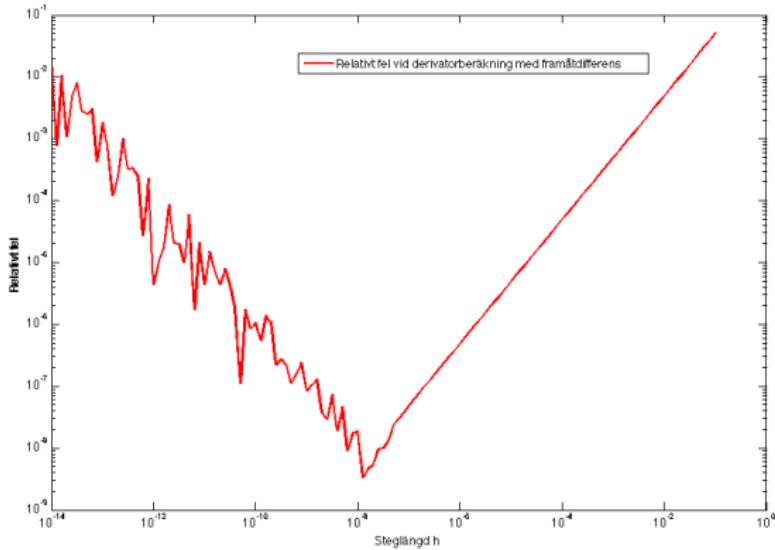
Representation av reella tal i dator

- Reella tal kan representeras med en viss relativ noggrannhet. (ε_M)
- Det finns gränser för hur små och stora tal som kan representeras. (`realmin`, `realmax`)
- Mindre tal kan representeras men då får vi sämre relativ noggrannhet.

Från labben

- Numerisk beräkning av derivata med

$$y' \approx \frac{f(x+h) - f(x)}{h}$$





Några exempel

Uttryck	exakt	i Matlab
$\cos\left(\frac{\pi}{2}\right)$	0	6.1232e-017
$0.42 - 0.5 + 0.08$	0	-1.3878e-017
$0.08 + 0.42 - 0.5$	0	0
$A^{-1} \cdot A$	I	Se lab

Hur mäter man fel?

- Det exakta talet betecknas x
Samma tal men som innehåller fel betecknas \hat{x}
- Felen kan t ex vara avrundningsfel eller mätfel
- Felet kan mätas

Absolut fel: $|x - \hat{x}|$ Relativt fel: $\frac{|x - \hat{x}|}{|x|}$

- Om x är en vektor blir det istället

Absolut fel: $\|x - \hat{x}\|$ Relativt fel: $\frac{\|x - \hat{x}\|}{\|x\|}$
 $\| \quad \|$ kallas för norm. Mer om detta senare.

Hur mäter man fel?

Exempel från labben:

- Du köper varmkorv en lördagkväll. Den kostar 15 kr, men av misstag betalar du 20 kr.
Absolut fel: $|15 - 20| = 5$
Relativt fel: $\frac{|15 - 20|}{15} \approx 0.333 = 33.3\%$
- Du köper en ny bil för 299995 kr, men betalar 300000 kr och bryr dig inte om växeln.
Absolut fel: $|299995 - 300000| = 5$
Relativt fel: $\frac{5}{299995} \approx 0.0000166 \approx 0.0017\%$
- Man förlorar lika mycket, men det känns sannolikt mindre i det andra fallet. Relativt fel upplevs ofta som mer korrekt än absolut fel

Representation av tal i dator ...

IEEE flyttalsrepresentation

- Under 60- och 70-talen hade varje datortillverkare sitt eget flyttalssystem
- En flyttalsstandard utvecklades under tidigt 80-tal och följdes av tillverkare som Intel och Motorola
- Utvecklat av arbetsgrupp hos Institute for Electrical and Electronics Engineering, IEEE (uttalas "I-triple-E")
- IEEE-standarden har tre viktiga krav:
 - Konsistent flyttalsrepresentation
 - Korrekt avrundningsaritmetik
 - Konsistent hantering av exceptionella situationer

IEEE flyttalsstandard

Single precision (32 bitar): $(\beta, p, L, U) = (2, 23 + 1, -126, 127)$

S	EEEEEEEEE	FFFFFFFFFFFFFFF	FFFF
0	1	8	9
			31

Double precision (64 bitar): $(\beta, p, L, U) = (2, 52 + 1, -1022, 1023)$

S	EEEEEEEEEEE	FFFFFFFFFFFFFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	
0	1	11	12						64

IEEE flyttalsrepresentation

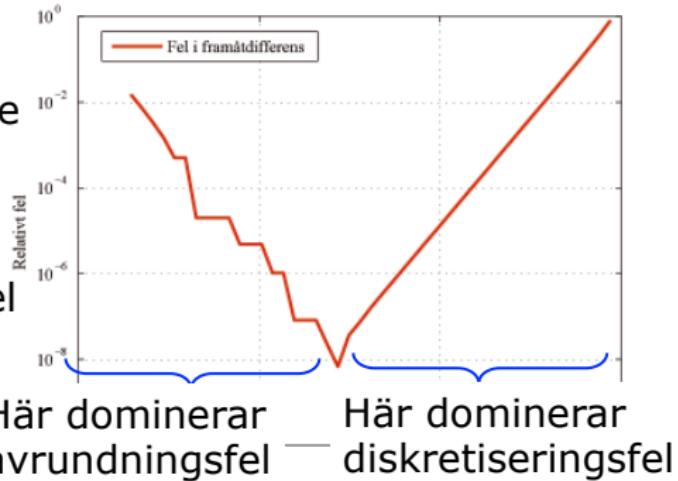
- IEEE definierar fem olika "exceptions"
 - "Invalid operation", t ex $\frac{0}{0}$, $0 \cdot \infty$ => ges värdet **NaN** (Not a Number)
 - Division med 0 => sätt till $\pm\infty$ (dvs **Inf** i Matlab)
 - overflow => sätt till $\pm\infty$ eller största flyttal
 - underflow => sätt till 0 (eller "subnormalt" tal)
 - Korrekt avrundning av reella tal (inte exceptionell situation egentligen)
- För den som vill veta mer
 - http://en.wikipedia.org/wiki/IEEE_floating-point_standard
 - <http://www.validlab.com/goldberg/paper.pdf>

Diskretiseringfel

- Förutom avrundningsfel finns även diskretiseringfel
Exempel) Numerisk derivering från lab

$$y' \approx \frac{f(x+h) - f(x)}{h}$$

När h blir mindre borde approximationen bli bättre – mindre diskretiseringfel



Diskretiseringsfel och avrundningsfel

Sammanfattning

- Diskretiseringsfelet spelar vanligen den dominerande rollen. Vanligt att man helt kan bortse från avrundningsfelen.
- Avrundningsfelet får konsekvenser i vissa fall, t ex vid kancellation.
- Exakt noll existerar inte i praktiken för flyttal. Diverse avrundningar gör att tal som kan anses vara lika ändå skiljer sig ute i decimalerna
- Ett relativt fel i sorleksordningen ε_M efter beräkning med flyttal är enbart slumpmässigt "skräp"