* **Idag:** • Flyttal
  • Avrundningsfel, (feluppskattningar)

$\boxed{F20}$ (orientering)

---

* Datorrepresentation av reella tal: "Flyttal"

- Reella tal: oändligt många decimaler
- Flyttal: ändligt många decimaler

⇒ Flyttal är alltid rationella tal!

⇒ Avrundningsfel

Viktigt att känna till vid numeriska beräkningar!

---

* Representation av flyttal

Skrivs på <u>exponentform</u>:

$$\boxed{x = \underset{\substack{\uparrow \\ \text{mantissa}}}{m} \cdot \underset{\substack{\uparrow \\ \text{bas}}}{b}^{\,e \,\leftarrow\, \text{exponent}}}$$

Normaliserad mantissa:

$$1 \leq |m| < b$$
$$\cap$$
$$\mathbb{Q}$$

Bas:

$b = 10$ (decimalt talsystem)

$b = 2$ (binärt talsystem)

$b = 16$ (hexadecimalt talsystem)

Exponent:

$$e_{min} \leq e \leq e_{max}$$
$$\cap$$
$$\mathbb{Z}$$

Exempel: $(b = 10)$

$$x = 3.1415$$
$$= \underbrace{3.1415}_{1 \le |m| < 10} \cdot \underbrace{10}_{b}^{0 \leftarrow e}$$

$$x = 31415.9265$$
$$= \underbrace{3.14159265}_{1 \le |m| < 10} \cdot \underbrace{10}_{b}^{4 \leftarrow e}$$

Exempel: $(b = 2)$

$$x = 1.011001$$
$$= \underbrace{1.011001}_{1 \le |m| < 2} \cdot \underbrace{2}_{b}^{0 \leftarrow e}$$

$$x = 1011001.11011$$
$$= \underbrace{1.01100111011}_{1 \le |m| < 2} \cdot \underbrace{2}_{b}^{6 \leftarrow e}$$

Notera: De enda siffror som finns är $0, \ldots, b-1$, dvs 0 och 1

* Något om talsystem

På samma sätt som

$$255_{10} = 2 \cdot 100 + 5 \cdot 10 + 5 \cdot 1$$
$$= 2 \cdot 10^2 + 5 \cdot 10^1 + 5 \cdot 10^0$$

har vi

$$11111111_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$
$$= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$
$$= 255$$

$$\therefore \quad 255_{10} = 11111111_2$$

Allmänt:

$$(a_n a_{n-1} a_{n-2} \cdots a_1 a_0)_b = \sum_{k=0}^{n} a_k \cdot b^k.$$

Notera att speciellt gäller

$$(\underbrace{(b-1)(b-1)(b-1)\cdots(b-1)}_{n \text{ siffror}})_b = b^n - 1$$

n siffror $(k = 0, 1, 2, \ldots, n-1)$

<u>Övning:</u> Visa detta!

Ledning: Använd välbekant lemma från F14...

<u>Exempel:</u>

$$11111111_2 = 2^8 - 1 = 256 - 1 = 255$$

$$999_{10} = 10^3 - 1 = 1000 - 1 = 999$$

* Andra talsystem:

trinära $(b=3)$

kvartenära $(b=4)$

$\vdots$

hexadecimala $(b=16)$

$0, 1, 2, 3, \ldots, 9, A, B, C, D, E, F$

* <u>Exempel:</u> RGB (röd-grön-blå)

(Färgpalett för HTML och grafikprogram)

$$\text{"}\underbrace{FF}_{R}\underbrace{FF}_{G}\underbrace{FF}_{B}\text{"} = \begin{cases} \text{Röd} & FF_{16} = 15 \cdot 16 + 15 \cdot 1 = 255 \\ \text{Grön} & FF_{16} = 255 \\ \text{Blå} & FF_{16} = 255 \end{cases}$$

$= $ "vit"

Varje färg får variera mellan 0 och $FF_{16} = 255$. Ger

$$256 \cdot 256 \cdot 256 = (2^8)^3 = 2^{24}$$

$$= 2^{10+10+4} = \underbrace{2^{10}}_{=1024} \cdot 2^{10} \cdot 2^4$$

$\approx 1000 \cdot 1000 \cdot 16 = 16$ miljoner olika färger.

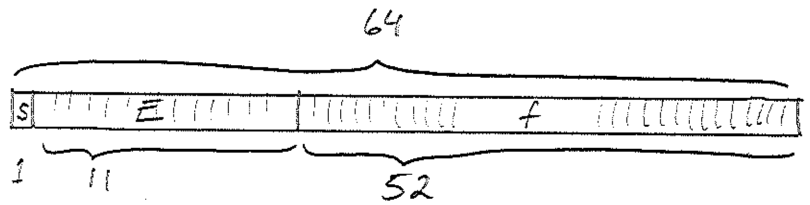
* Flyttalsstandarden IEEE 754

Två versioner:

Binary32 ← "float" i C/C++

Binary64 ← "double" i C/C++, MATLAB

- Bas 2
- 64 bitar / tal (Binary64), dvs 8 bytes (1 byte = 8 bitar)
- $m = 1.f$, 1 lagras ej
- I stället för e lagras $E = e + 1027$
- En bit används för att lagra tecknet
- 11 bitar används för att lagra E
- 52 bitar används för att lagra f



Största tal: $\approx \pm 2 \cdot 10^{308}$

Minsta tal: $\approx \pm 2 \cdot 10^{-308}$

$\varepsilon_{mach} \approx 2 \cdot 10^{-16}$

"Maskinprecision"

* Maskinprecision, avrundningsfel

- Alla tal representeras med ett relativt fel som är som störst ε_{mach}:

$$fl(x) = x \cdot (1 + \Delta)$$

$$|\Delta| \leq \varepsilon_{mach}$$

- Alla operationer har ett relativt fel som är som störst ε_{mach}:

$$fl(fl(x) \cdot fl(y)) = fl(x) \cdot fl(y) \cdot (1 + \Delta)$$

$$|\Delta| \leq \varepsilon_{mach}$$

* **Exempel:**

Beräkna $f(x) = \dfrac{4x}{(1+x)^2 - (1-x)^2}$

Notera först att

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

MATLAB ger

$f(1) = 1$

$f(0.001) = 1.00\ldots08$

$f(10^{-10}) = 0.99\ldots172$

$f(10^{-16}) = 1.8014\ldots$

$f(10^{-17}) = \mathrm{Inf}$

Om $|x|$ är litet gäller att

$$\frac{4x}{(1+x)^2-(1-x)^2} \approx \frac{4x}{1+2x-(1-2x)}$$

MATLAB (IEEE 754) räknar

$$\frac{4x}{(1+2x)\cdot(1+\Delta_1) - (1-2x)\cdot(1+\Delta_2)}$$

$$= \frac{4x}{\cancel{1}+\Delta_1+2x+\underbrace{2x\Delta_1}_{\approx 0}-(\cancel{1}+\Delta_2-2x-\underbrace{2x\Delta_2}_{\approx 0})}$$

$$\approx \frac{4x}{4x+\Delta_1+\Delta_2}$$

Om $x \sim \Delta_1 \sim \Delta_2$ så kan resultatet bli nästan vad som helst...

Demo: (MATLAB)

- Beräkning av $f(x) = \frac{4x}{(1+x)^2 - (1-x)^2}$

- Beräkning av största/minsta tal (notera "subnormala" tal)
- Beräkning av ε_{mach}

- Mandelbrot / fixpunktsiteration