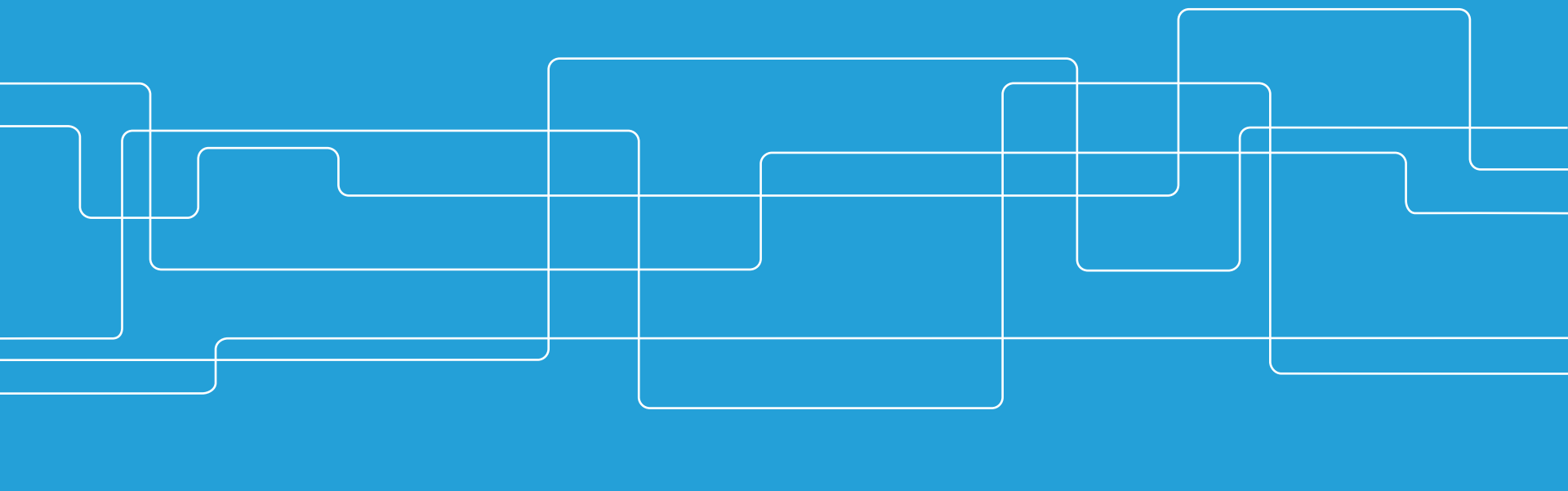
Matematmatisk Modellering

Anders Västberg
08-7904455, vastberg@kth.se



Innehåll

- Matematisk modellering
- Simulering
- Analytiska modeller
- Numeriska modeller
- Talrepresentation i en dator
- Exempel på matematisk modellering

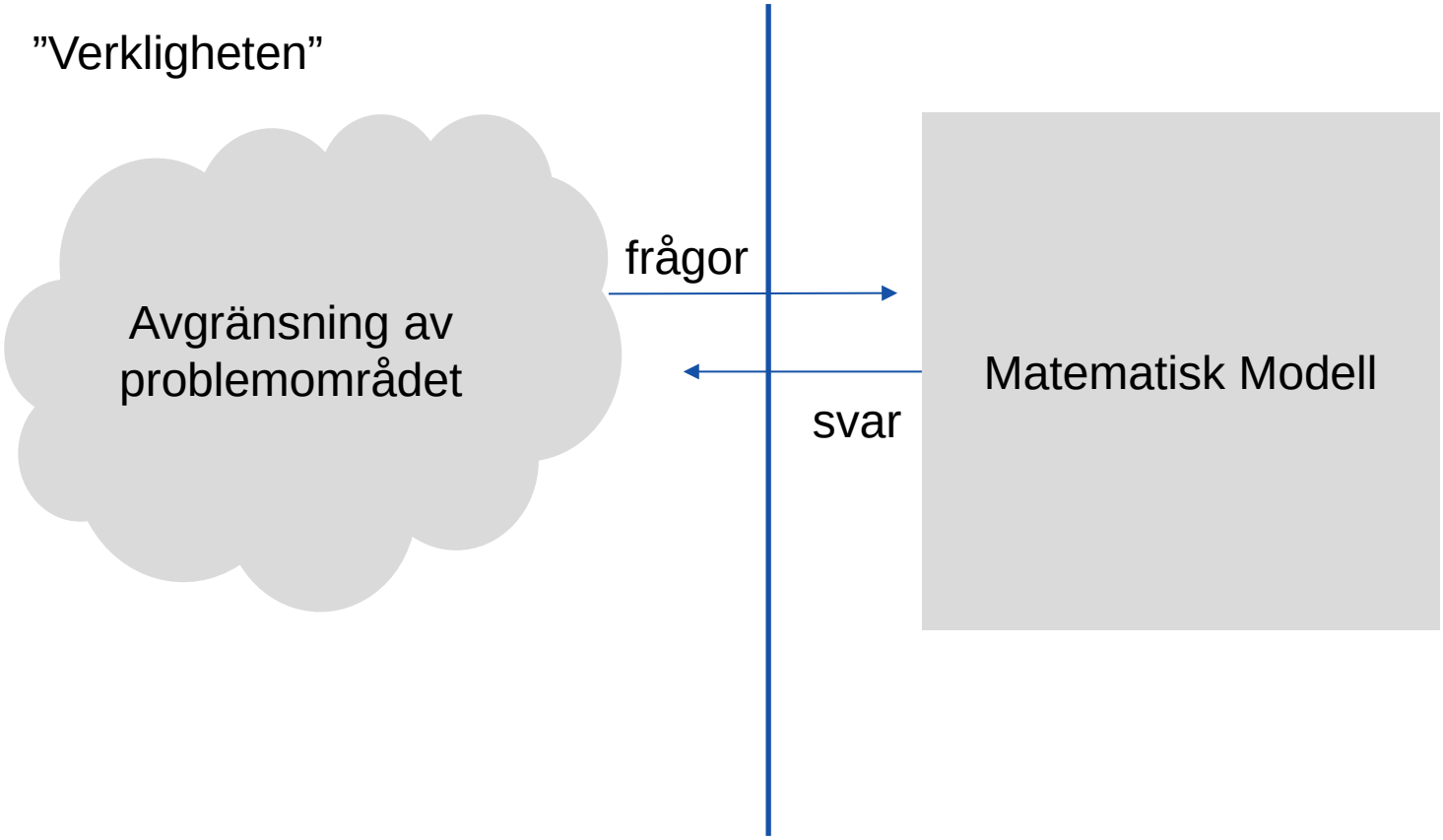
Varför matematisk modellering

Man kan analysera ett system genom att bygga det och direkt undersöka systemet, men:

- Det är för dyrt att bygga ett riktigt system och mäta på det
- Det är för riskfyllt att bygga ett riktigt system och undersöka detta
- Systemet är otillgängligt (solsystemet) eller existerar (ännu) inte (ny bil eller nytt flygplan).

[Ljung och Glad, 2004]

Matematisk modellering



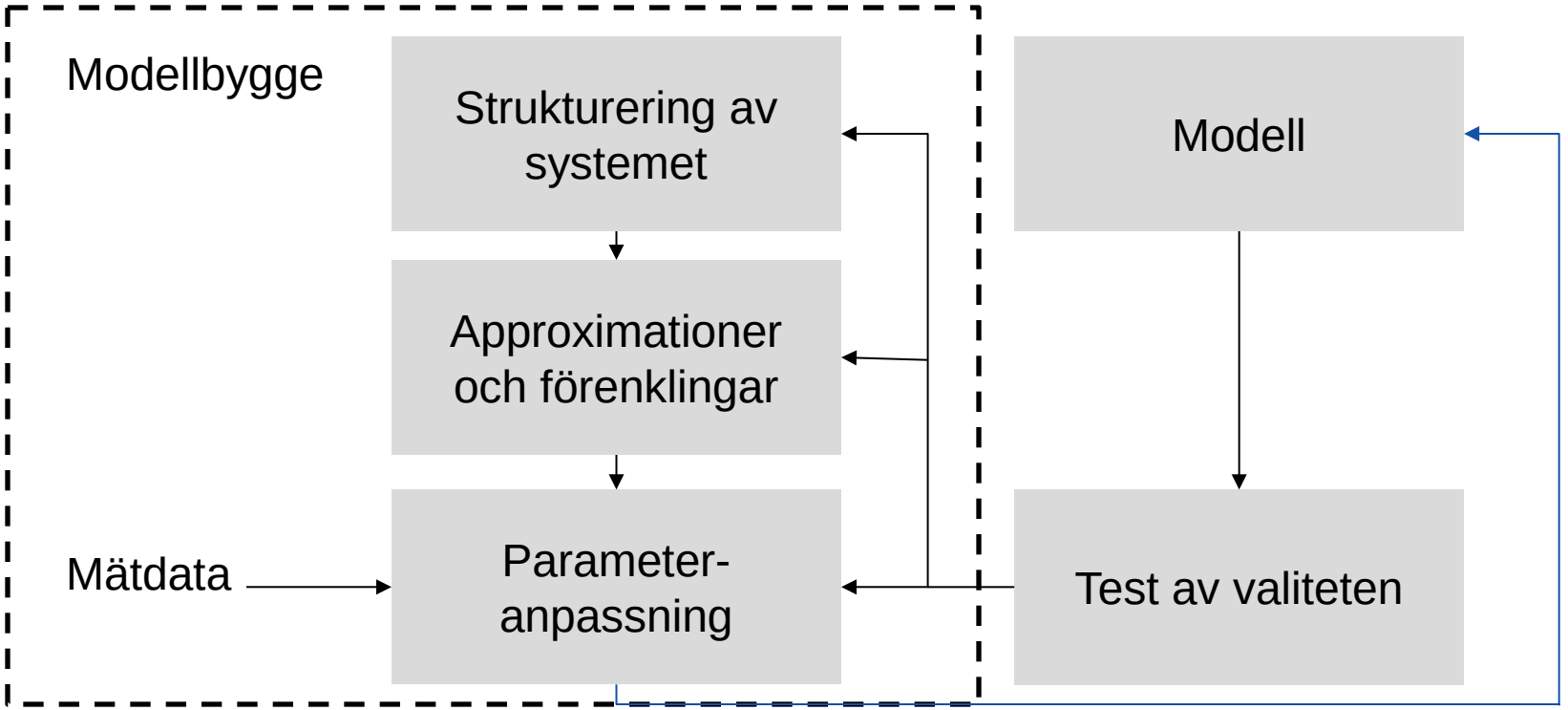
[Eldén och Wittmeyer-Koch, 1987]

Simulering

- Det går inte att göra ett experiment på systemet, men en matematisk modell existerar
- Det går då att använda modellen för att undersöka systemet
  - Analytiska modeller
  - Numeriska modeller
- Datorer kan simulera komplexa system som är svåra och /eller dyra att realisera i verkligheten
- Simuleringarna används för att förstå hur systemet fungerar
- Värdet av resultaten beror på hur bra modellen är
- Modellen måste verifieras

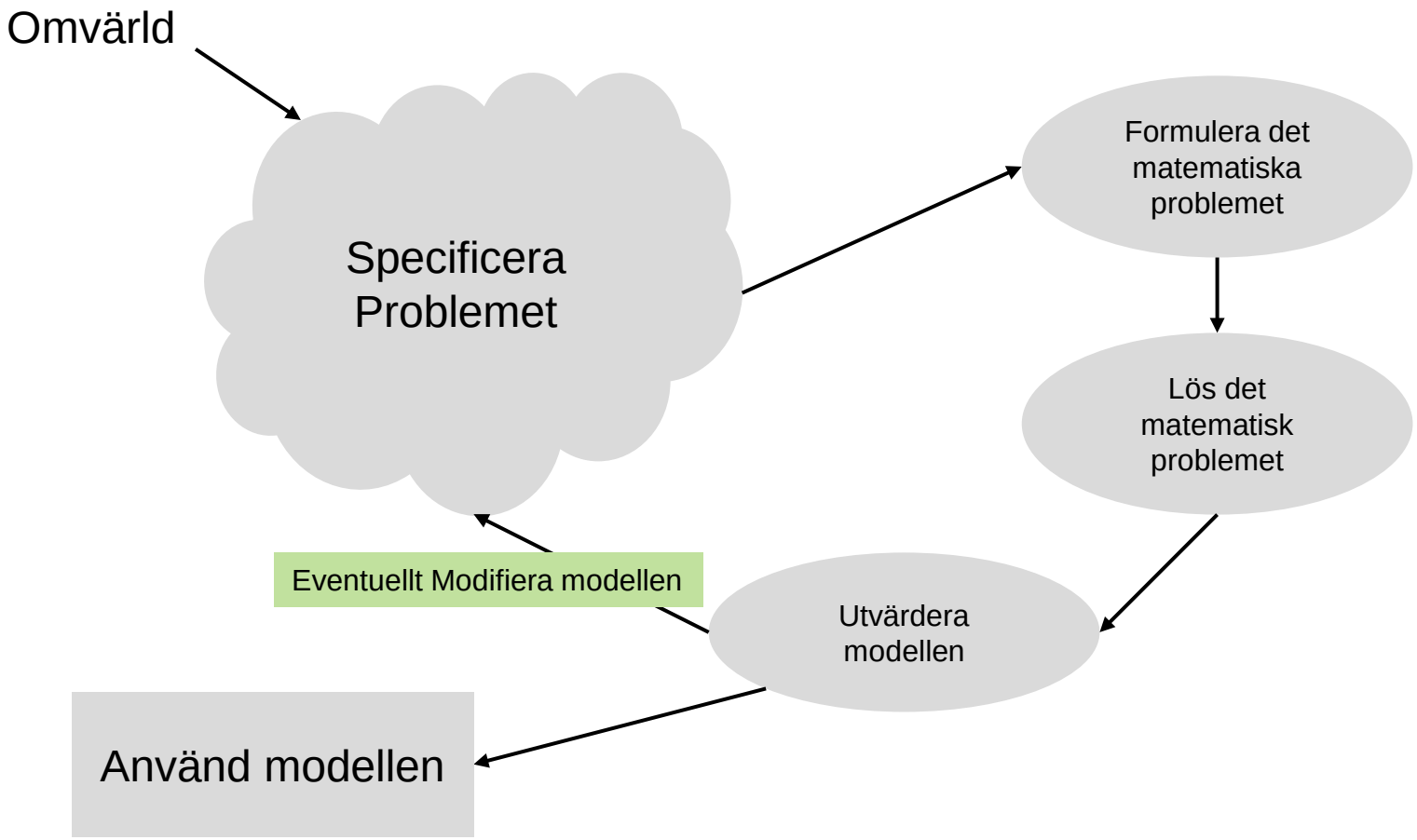
Giltighetsområde och valideringsprocess

- En modell har ett begränsat giltighetsområde
  - Inom det område som modellen har validerats

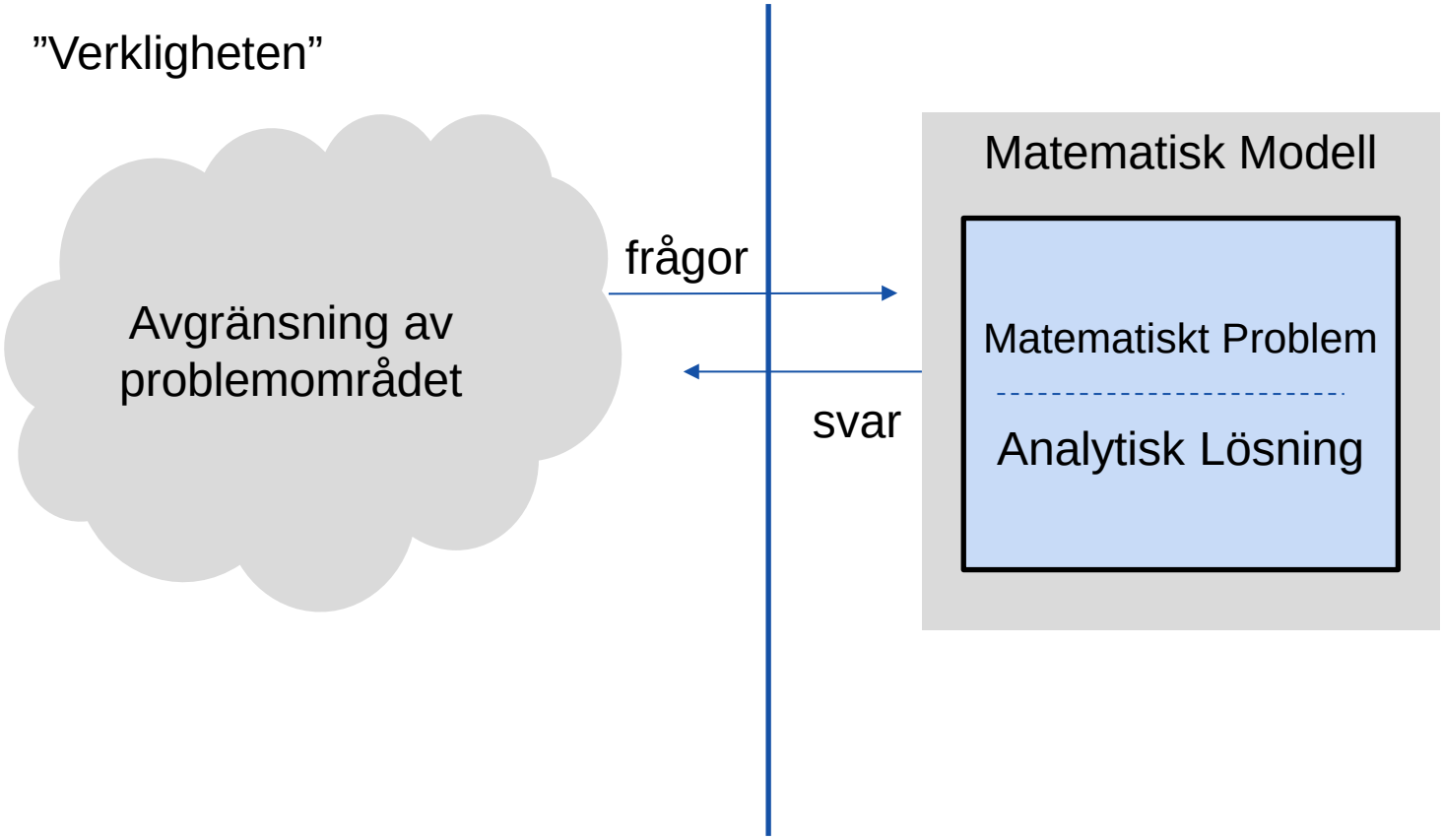


[Ljung och Glad, 2004]

Verifiering av matematiska modeller

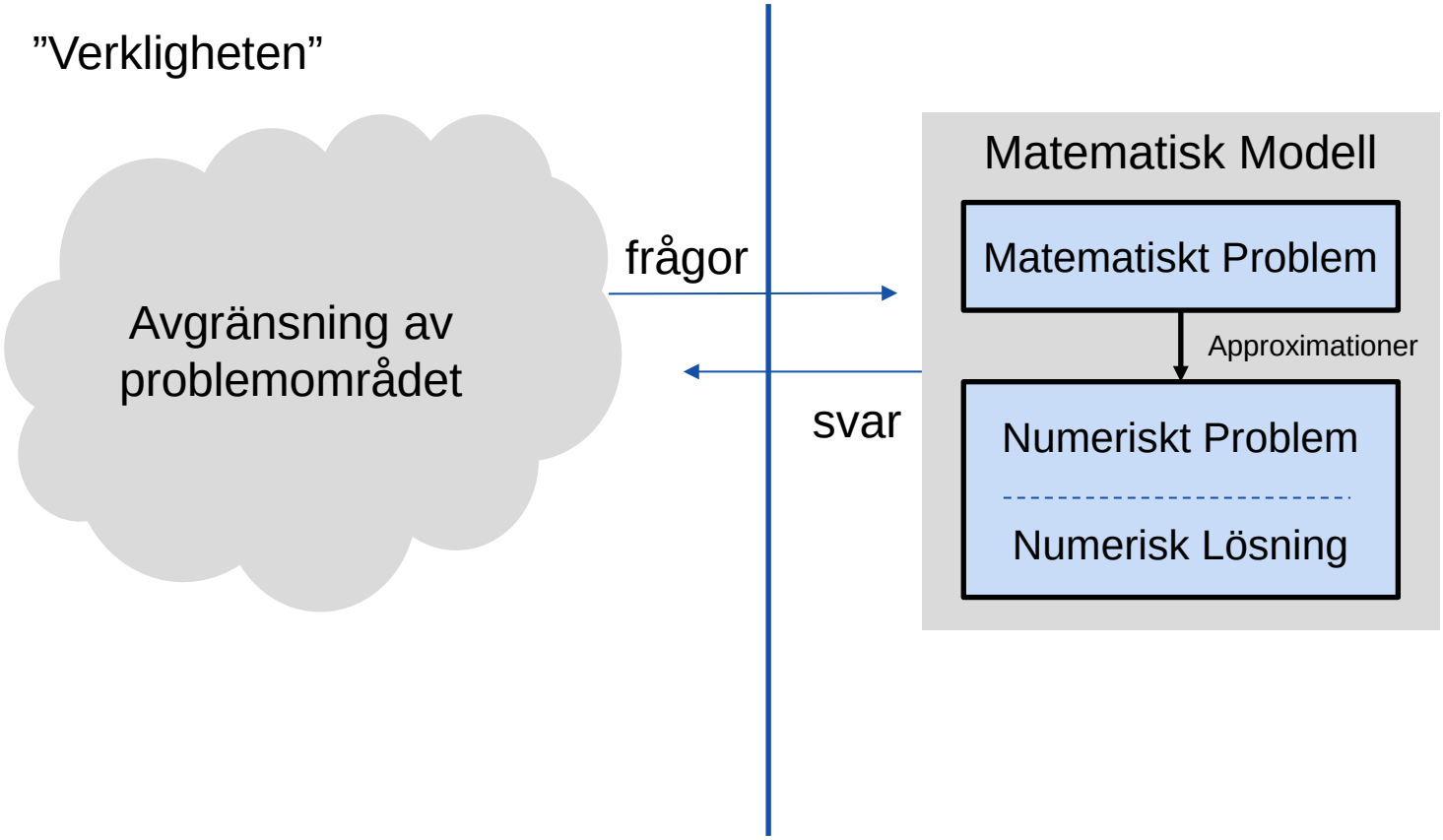


Analytiska modeller



[Eldén och Wittmeyer-Koch, 1987]

Numeriska modeller



[Eldén och Wittmeyer-Koch, 1987]

Några exempel

- Trafikflöde
- Fallande mynt från Eiffeltornet
- Kurvanpassning till data

Talrepresentation i en dator

- Vanliga talsystemet har basen 10
- Datorer använder binära tal:
  - Tal med basen 2
- Heltal kan identiskt representeras i båda baserna.
- Exempel talet $14 = (14)_{10} = (1110)_2$:

$$14 = 1 \cdot 10^1 + 4 \cdot 10^0$$

$$1110 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

- Reella tal kan inte alltid representeras exakt med binära tal
- Exempel talet $0{,}2 = (0{,}2)_{10} = (0.00110011\ldots)_2$

$$2 \cdot 10^{-1}$$

$$0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + 0 \cdot 2^{-6} + 1 \cdot 2^{-7} + \cdots$$

Heltalsrepresentation i en dator

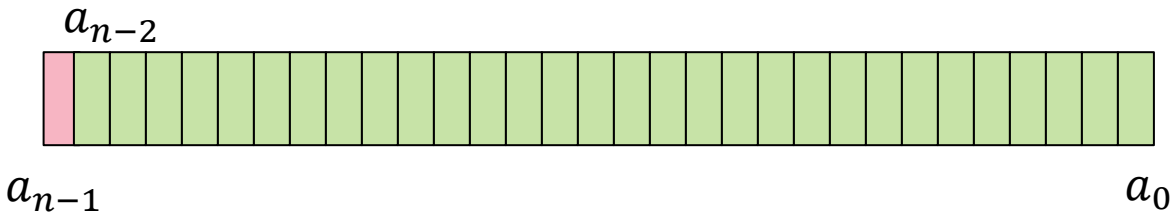
- Heltal representeras av ett antal bitar
- Antalet bitar sätter gränsen för största och minsta heltalet
- 16 bit = $2^{16-1} - 1 = \pm 32767$
- 32 bit = $2^{32-1} - 1 = \pm 2147483647$
- 64 bit = $2^{64-1} - 1 = \pm 9223372036854775807$
- Det finns flera nackdelar med denna representation.
  - En är att addition och subtraktion kräver att man håller reda på tecken för båda talen.
  - En annan nackdel är att vi har två nollor $+0$ och -0.
    - Det innebär att tester som involverar 0 tar något längre tid att utföra.

Tvåkomplements representation

- Normalt används tvåkomplement för att representera heltal, vilket ger följande minsta och största tal
- Representationen förenklar beräkningar i datorer
- 16 bit = $[-2^{16-1}, 2^{16-1} - 1] = [-32768, 32767]$
- 32 bit =$[-2^{32-1}, 2^{32-1} - 1] = [-2147483648,2147483647]$
- 64 bit =$[-2^{64-1}, 2^{64-1} - 1]$
  $= [-9223372036854775808, 9223372036854775807\ ]$

Tvåkomplements representation

$$-2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$



Om a_{n-1} är noll är talet positivt, om den är ett är talet negativt.
Exempel för fyra bitar:

$$0111 = -0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^0 = +7$$
$$1001 = -1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^0 + 1 \cdot 2^0 = -7$$
$$0000 = -0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^0 + 0 \cdot 2^0 = 0$$
$$1000 = -1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^0 + 0 \cdot 2^0 = -8$$

För att konvertera ett positivt tal till motsvarande negativt,
ta fram komplementet till bit-mönstret och lägg till 1

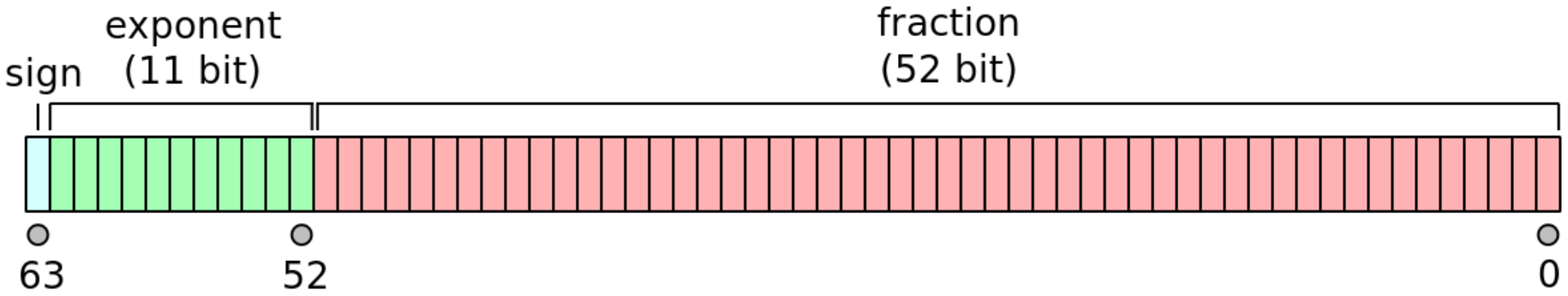
[Stallings, 2003]

Flyttalsrepresentation i en dator

Double precision numbers – 64 bit representation av flyttal.

- 1 bit för $\pm$ tecken
- 11 bit för exponent
- 52 bit för mantissan - signifikant precision

$$(-1)^t \left(1 \cdot 2^0 + \sum_{i=1}^{52} b_{52-i} 2^i \right) \cdot 10^{e-1023}$$



[https://en.wikipedia.org/wiki/Double-precision_floating-point_format]

Flyttalsrepresentation i en dator

- Kan representera tal mellan 10^{-308} och 10^{308} med $53\ \log_{10} 2 \approx 15.995$ eller nästan 16 siffors noggrannhet
- Standarden är definierad av IEEE 754 som också definierar bland annat:
- Single (32 bit flyttal)
- Och motsvarande talrepresentation för finansiell sektor:
  - Decimal64
  - Decimal32
  - Dessa talrepresentationer emulerar decimal-avrundning exakt (basen 10).

Felkällor

- Fel i indata
- Avrundningsfel
- Trunkeringsfel
- Absolut fel i $\bar{a}$: $|\Delta a| = |\bar{a} - a\,|$
- Relativt fel i $|\bar{a}|$: $\left|\frac{\Delta a}{a}\right|, \quad (a \neq 0)$
  - Exempelvis relativt och absolut fel att använda 1,414 istället för $\sqrt{2}$
  - $|\Delta a| = \left|1{,}1414 - \sqrt{2}\right| = 0{,}0002135\ldots$
  - $\left|\frac{\Delta a}{a}\right| = \frac{0.0002135\ldots}{\sqrt{2}} \approx 0{,}015\%$
  - Tänk på skillnaden mellan avrundning och avhuggning

[Eldén och Wittmeyer-Koch, 1987]

Felkällor

- Om $\Delta a \leq 0{,}5 \cdot 10^{-t}$ sägs närmevärdet för $\bar{a}$ ha t korrekta decimaler
- I ett närmevärde med t korrekta decimaler är alla siffror i positioner med enhet $\geq 10^{-t}$ vara signifikanta siffror, utom inledande nollor
- Exempel

Närmevärde	Korrekta decimaler	Signifikanta siffor
$0{,}001234 \pm 0{,}5 \cdot 10^{-5}$	5	3
$4{,}567 \pm 0{,}5 \cdot 10^{-3}$	3	4
330000 ± 5000	-	2

[Eldén och Wittmeyer-Koch, 1987]