

CCSE

Center for Computing
in Science Education



Programmering for å styrke matematikken, er det mulig?

Knut Mørken

Matematisk institutt

Centre of Computing in Science
Education

Det matematisk naturvitenskapelige fakultet
Universitetet i Oslo

Lærerkonferanse i programmering
NTNU, 11. mars 2019

Min bakgrunn

Fagområde matematikk/informatikk, numerisk analyse

Arbeidet med å fornye grunnutdanningene i matematikk og naturvitenskap med et integrert beregningsperspektiv siden 2000

Ta vare på det beste i fagtradisjonene og samtidig se framover

Grunnleggende spørsmål: Hva skal en matematiker være?

Medlem av kjerneelementgruppa i matematikk, utredning om teknologi

Tenketanken for Skaperskolen



34 000 000 000 000 000
operasjoner pr. sekund



Mer enn 10^{12} operasjoner pr.
sekund
Ville vært blant verdens raskeste
datamaskiner rundt år 2000

Hva betyr dette for utdanning?

Skal datamaskinen kontrollere oss?

Eller skal vi kontrollere den?

Hva betyr i så fall dette i praksis?

I skolen?

Programmering i skolen

Før 2020:

- Programmering for programmering
- Programmering i teknologi
- Programmering på fritid
- Fra 2017: Programmering og modellering X

Programmering for de ivrige

Fra 2020:

- Programmering **for alle**, med matematikk som vertskap
- Programmering utnyttes i andre fag der det er naturlig
- Matematikkfaget og matematikklærere får et stort ansvar
- **Hvordan nå alle?**

Overordnede rammebetingelser

Premisser

- Matematikkfaget er vertskap for programmering
- Svært mange matematikklærere trenger å lære å programmere
- Lærerne er stolte fagpersoner innen matematikk og didaktikk

Derfor

- Lærerne må oppleve at programmering styrker matematikkfaget
- Syntax og andre tekniske ting må reduseres til et minimum i starten
- Didaktiske og andre perspektiver må tas på alvor
- Hvordan?

ProFag ved UiO

1. Koding (syntaks)
2. Algoritmisk tenkning
3. Didaktikk
4. Programmering **for** matematikken (og andre disipliner)
5. Programmering påvirker matematikk (og andre disipliner)

**Hva betyr dette i praksis i
matematikk?**

Nye muligheter med datamaskin

1 000 000 000 000 000 operasjoner pr. sekund

- Gjør det mulig å regne på store datamengder
- Løsning av matematiske problemer ved iterasjon (gjentagelse)

To muligheter:

1. Kalkulator og ferdige programmer (blir lett passivt)
- 2. Utforskende (aktiv tilnærming, krever programmering)**

**Hva betyr det når vi løser
ligninger?**

Løse ligninger

Hva vil det si å løse en ligning $f(x) = 0$?

Finne et eksakt uttrykk (tall) for x :

$$x^2 + px + q = 0$$

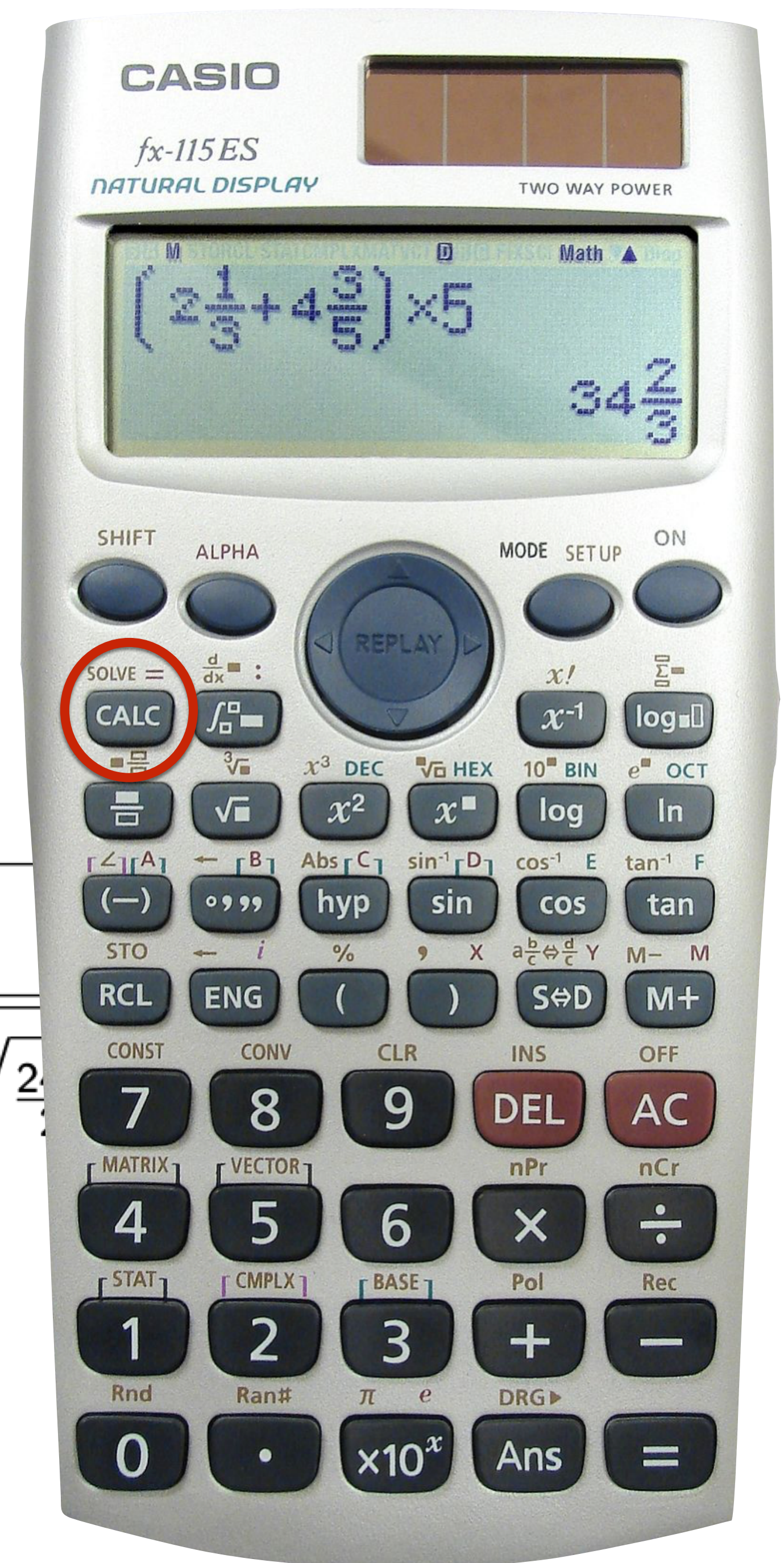
$$x = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

Generalisere

$$x^4 + 3x + 1 = 0$$

$$x = \frac{1}{2} \sqrt{\frac{1}{3} \sqrt[3]{\frac{243}{2} - \frac{3\sqrt{5793}}{2}} + \frac{\sqrt[3]{\frac{1}{2}(81 + \sqrt{5793})}}{3^{2/3}}} - \frac{1}{2} \sqrt{-\frac{1}{3} \sqrt[3]{\frac{243}{2} - \frac{3\sqrt{5793}}{2}} - \frac{\sqrt[3]{\frac{1}{2}(81 + \sqrt{5793})}}{3^{2/3}}}$$

ved hjelp av Mathematica
svært vanskelig for studentene



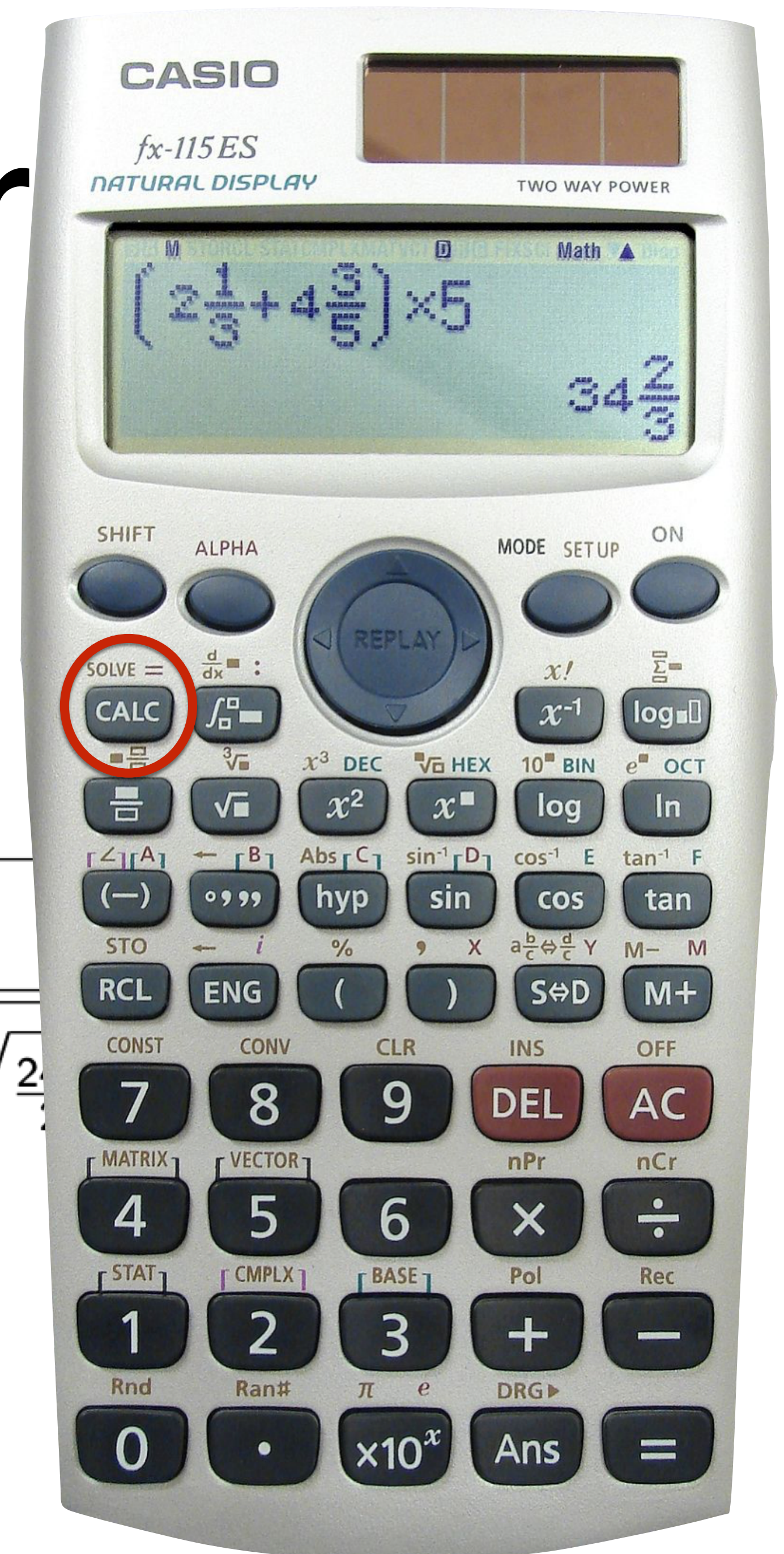
Løse ligninger

$$x^4 + 3x + 1 = 0$$

$$x = \frac{1}{2} \sqrt{\frac{1}{3} \sqrt[3]{\frac{243}{2} - \frac{3\sqrt{5793}}{2}} + \frac{\sqrt[3]{\frac{1}{2}(81 + \sqrt{5793})}}{3^{2/3}}} - \frac{1}{2} \sqrt{-\frac{1}{3} \sqrt[3]{\frac{243}{2} - \frac{3\sqrt{5793}}{2}} - \frac{\sqrt[3]{\frac{1}{2}(81 + \sqrt{5793})}}{3^{2/3}}}$$

$$\approx -0.337667$$

som regel mye nyttigere



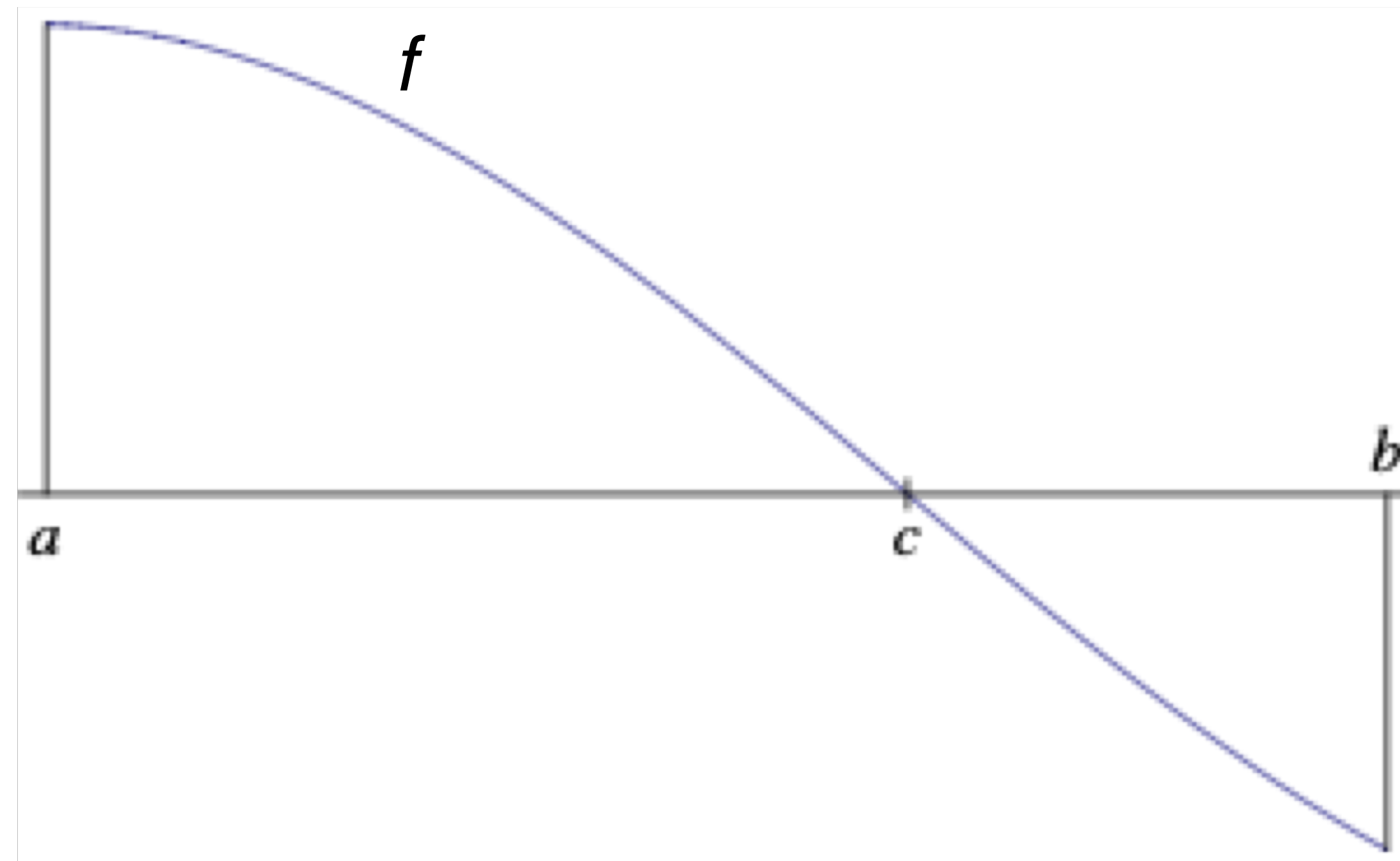
Løse ligninger

Når kan vi finne en formel for nullpunktet?

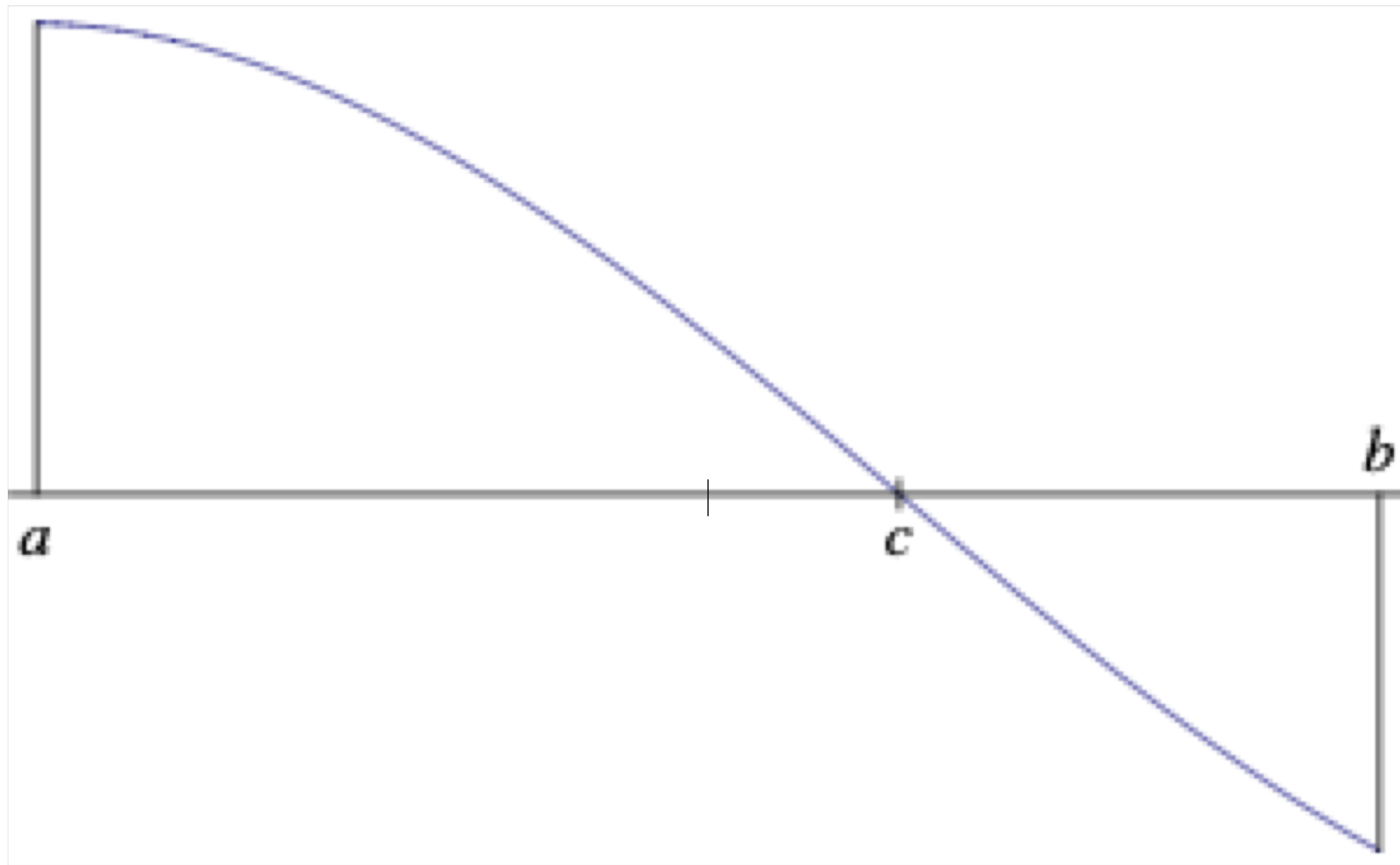
- Førstegradsligning, andregradsligning, noen eksponensielle og trigonometriske ligninger, tredje og fjerdegradsligninger
- Med andre ord noen veldig spesielle ligninger
- Ved hjelp av triks som ikke lar seg generalisere

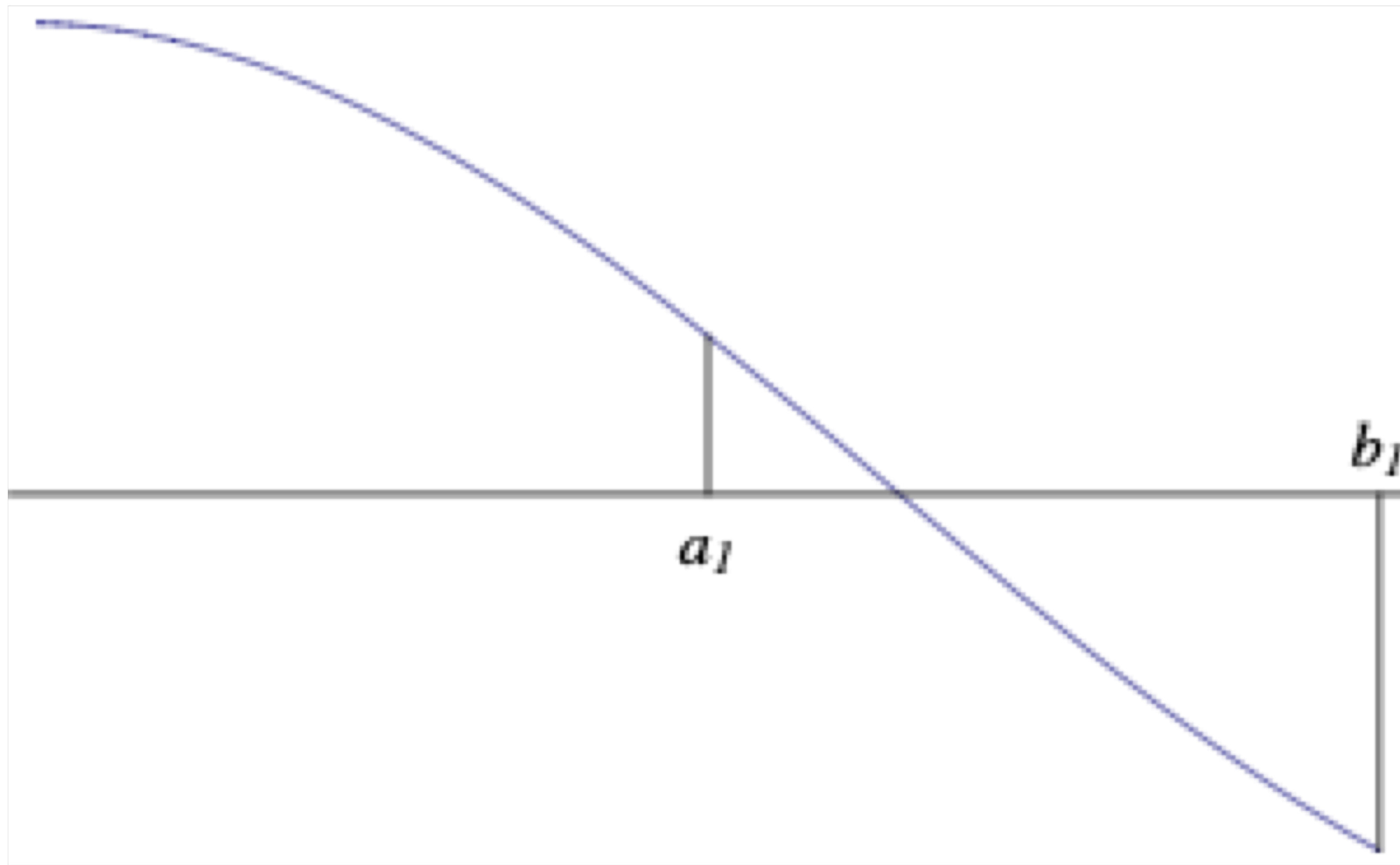
Hvordan finne nullpunkt for en generell funksjon f ?

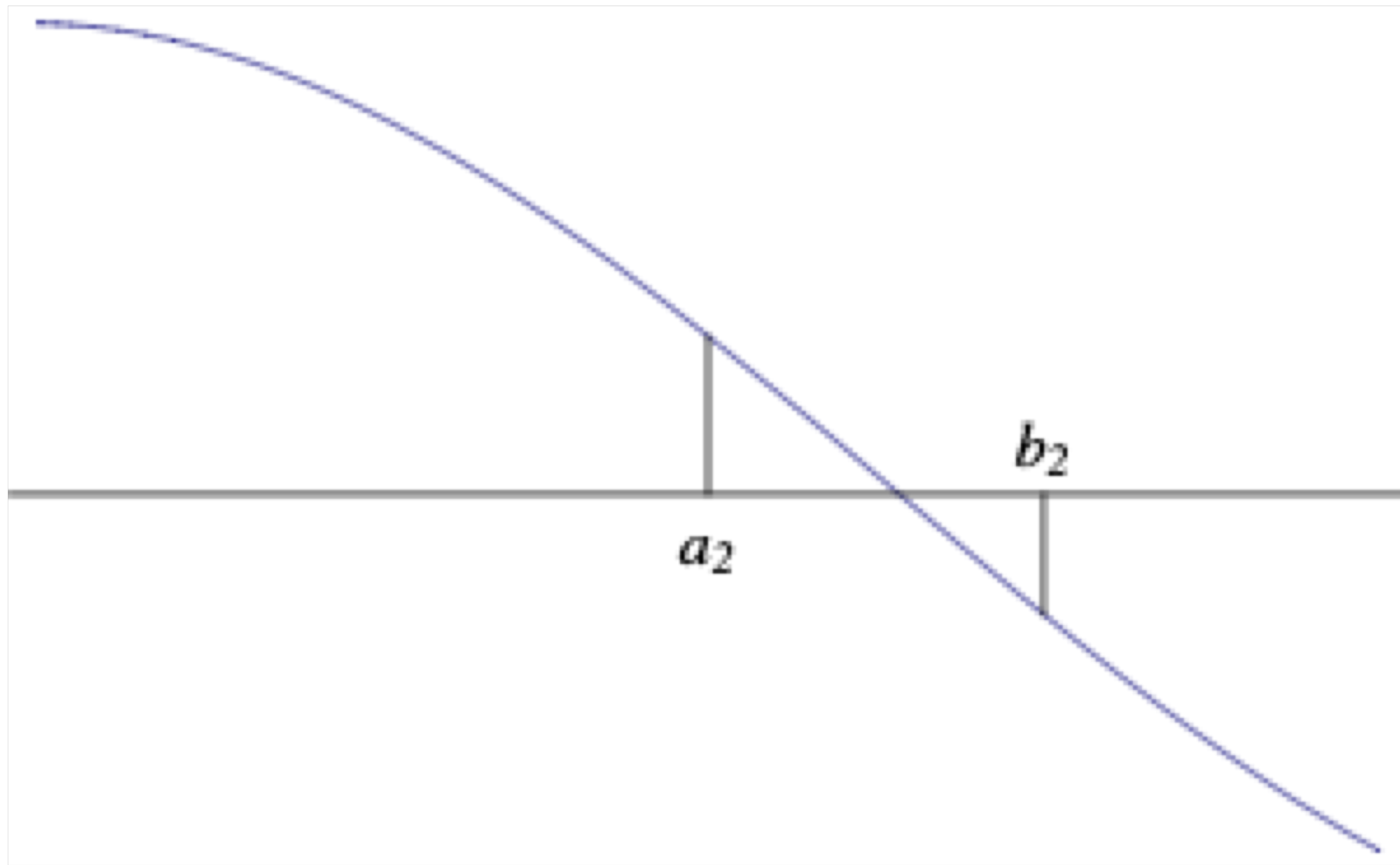
Skjæringssetningen. For en kontinuerlig funksjon f definert på $[a,b]$, med motsatt fortegn i a og b , fins det et tall $c \in (a,b)$ slik at $f(c) = 0$.

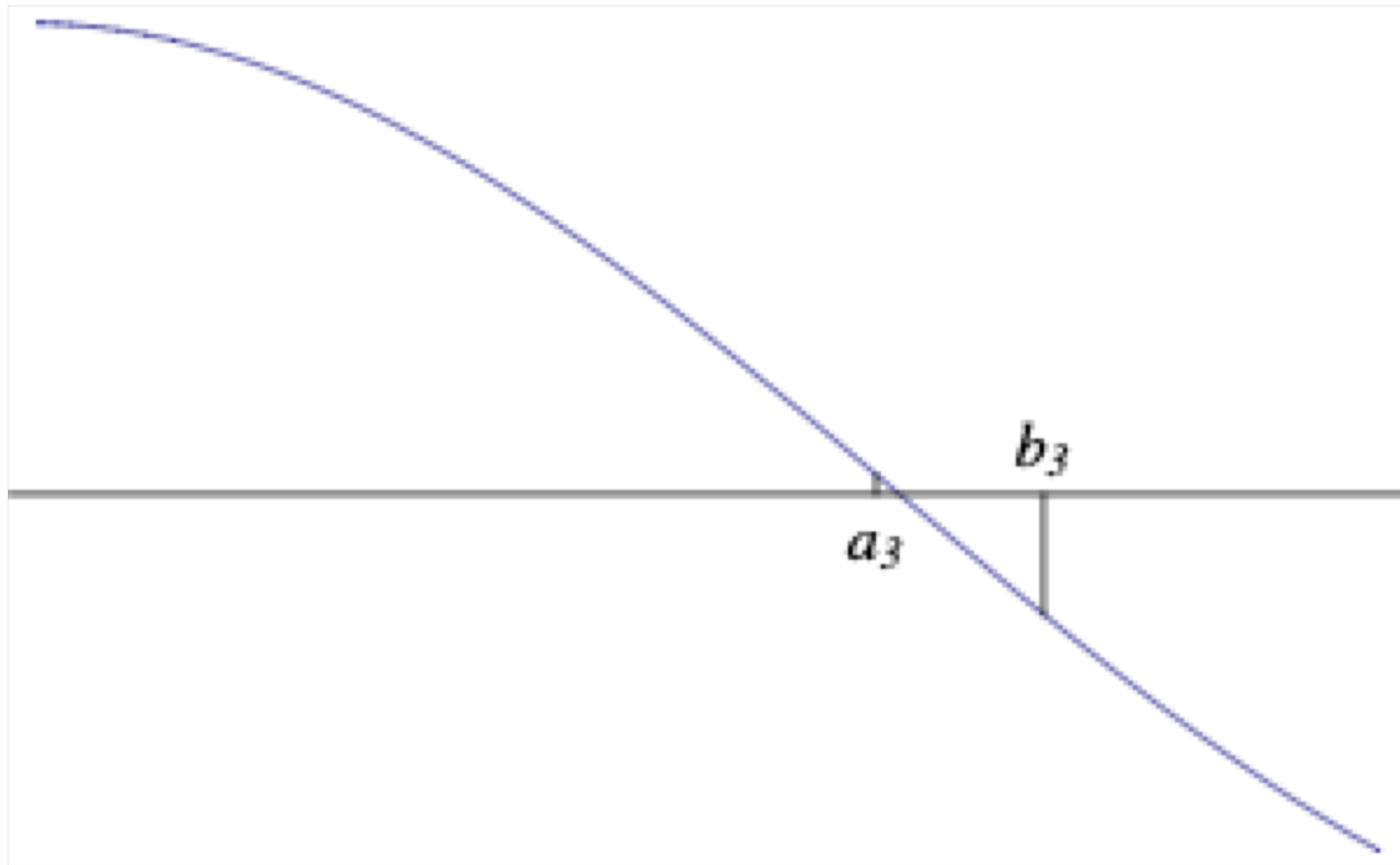


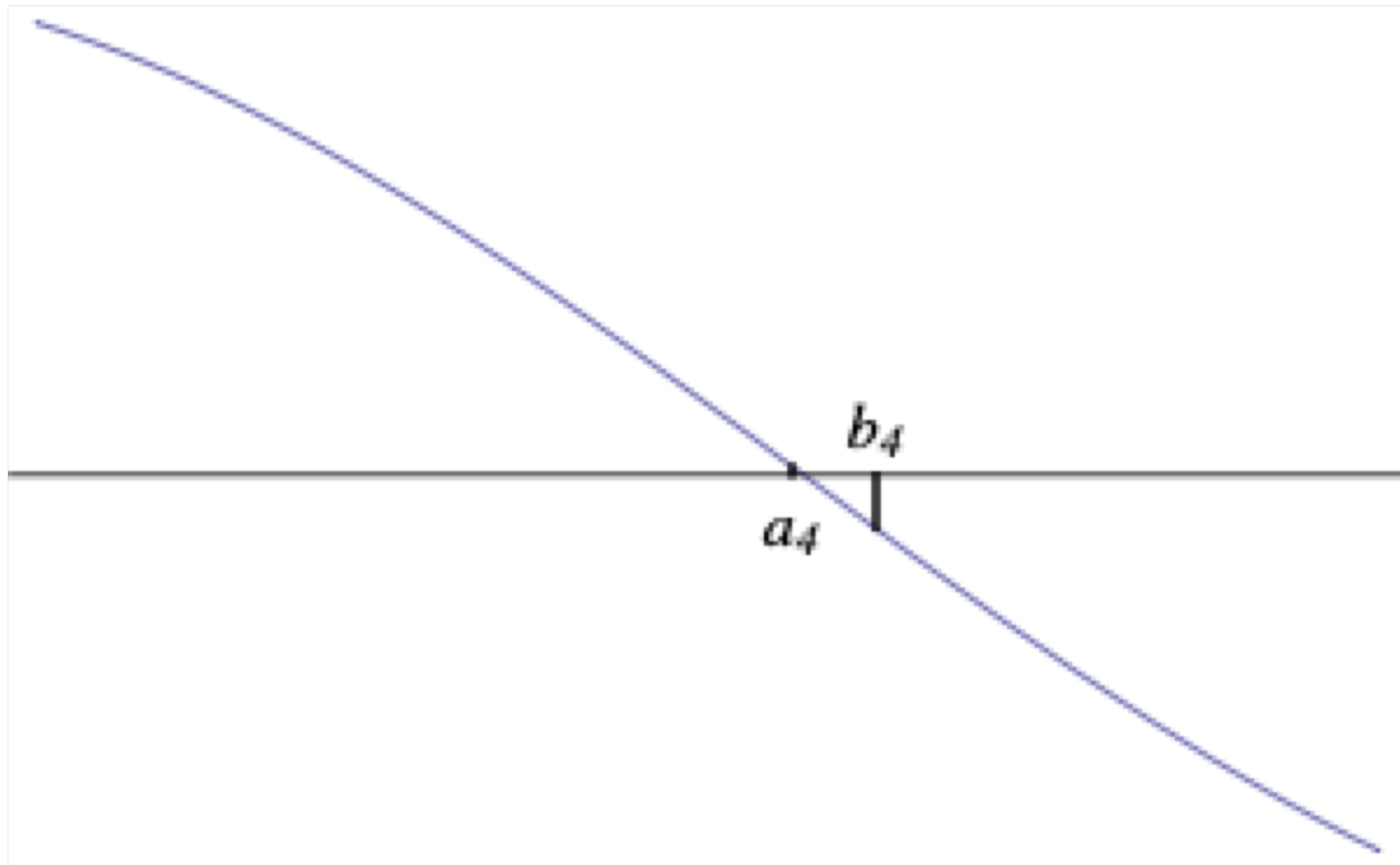
Må kunne regne ut funksjonsverdier

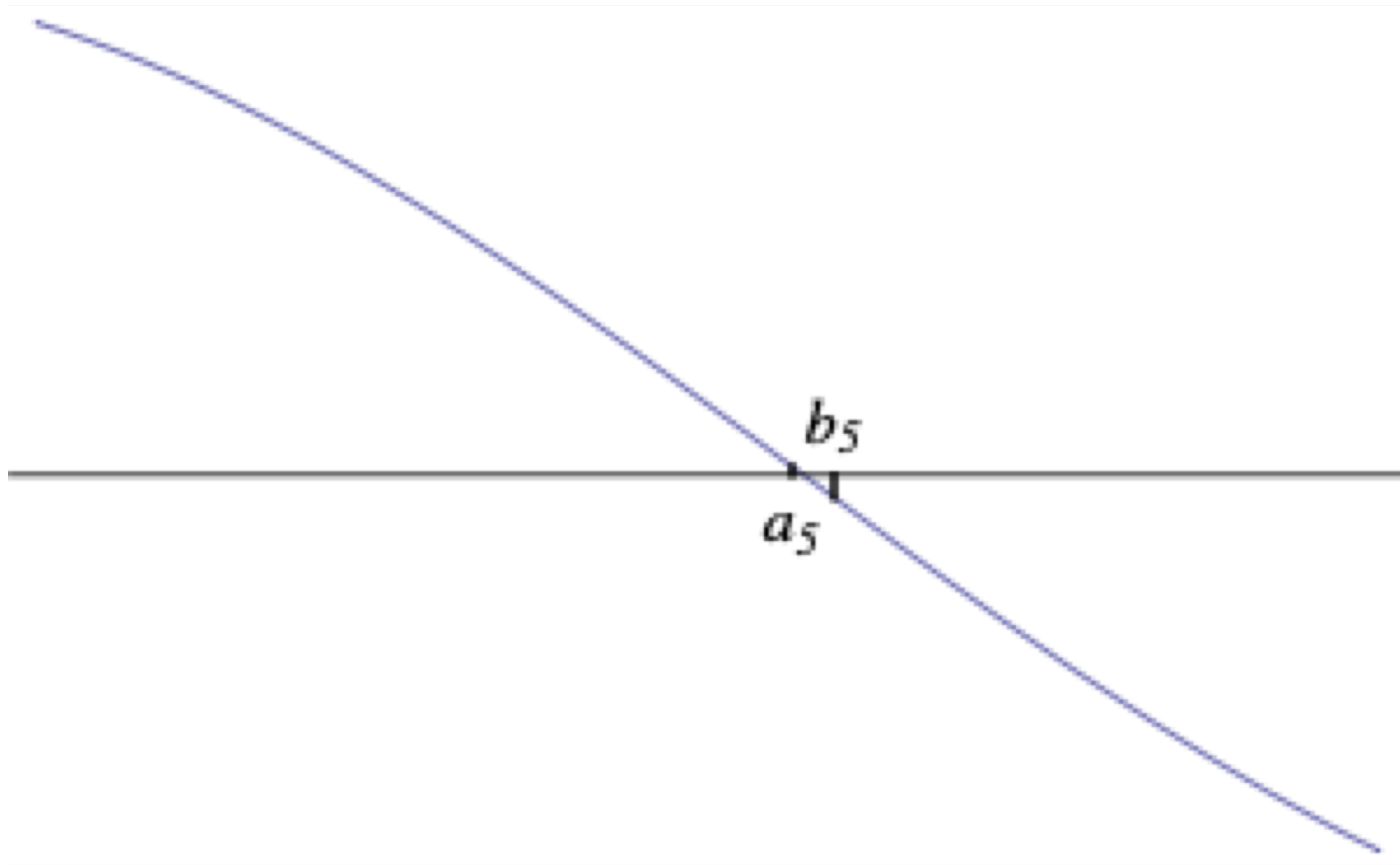












Halveringsmetoden – algoritme

$i = 0;$

$m = (a + b)/2;$

while $i \leq N$

if $f(m) == 0$

$a = b = m;$

if $f(a)*f(m) < 0$ **then** $b = m;$

else

$a = m;$

$i = i + 1;$

$m = (a + b)/2;$

Bare den aller siste delen er
«koding»

Utleddning av algoritmen er
matematikk

Pedagogiske muligheter

Bevissthet rundt begreper

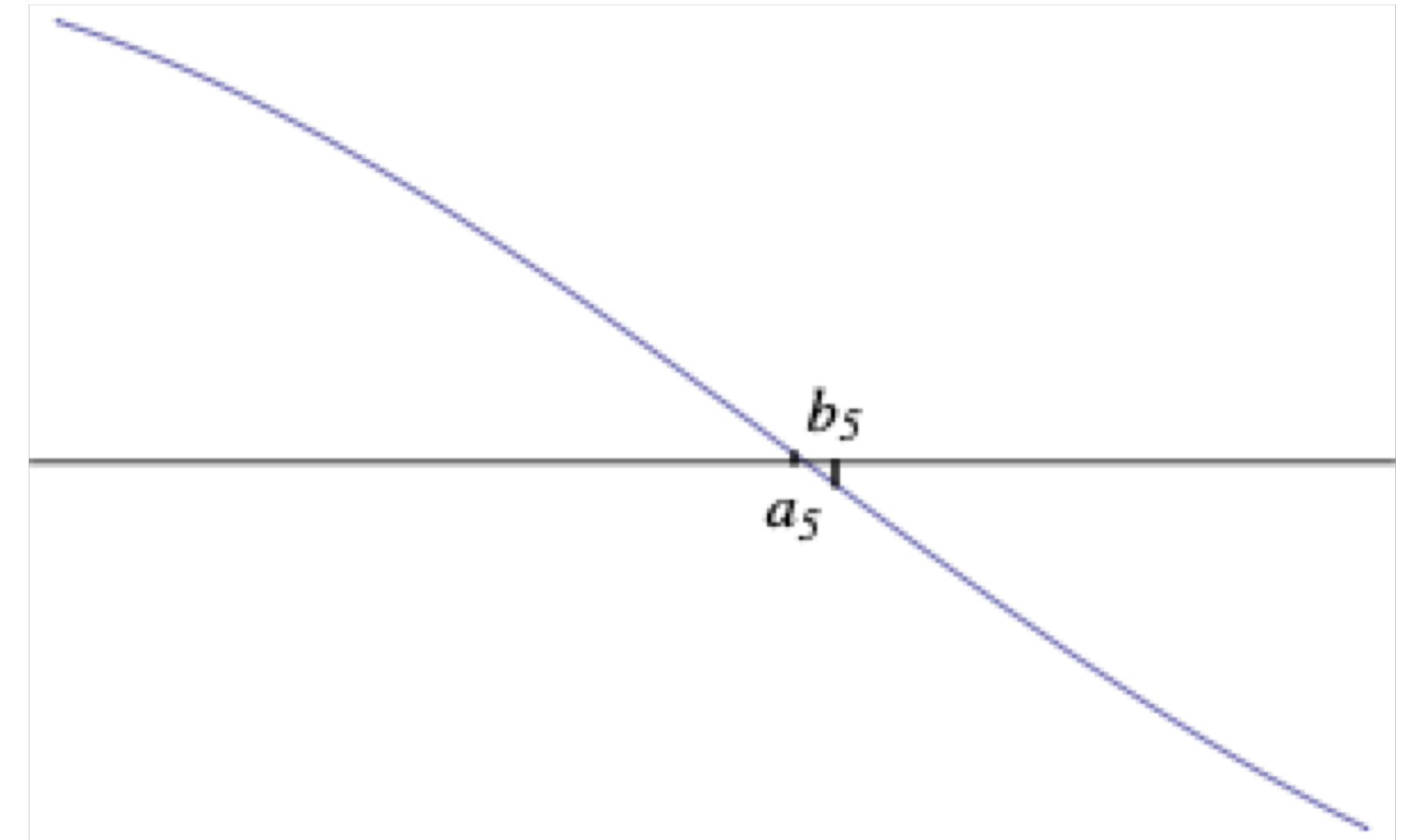
- Hva vil det si å løse en ligning?
- Eksakt formel — iterasjon

Kreativitet:

- Hva med å dele i tre?
- Kan vi utnytte størrelsen til f i endene?

Lekegrind for mer abstrakte begreper

- Kan vi si noe om feilen?
- «Divide and conquer» som en generell strategi



Andre muligheter

Vgs

- Numeriske metoder: Integrasjon, derivasjon, differensialligninger
- Modellering
- Grafikk / geometri

Ungdomsskole

- Aritmetikk, variable, likhetstegn, funksjoner
- Diskret modellering
- Geometri

Utfordringer

- 40000 lærere skal kunne programmere før 2020
- Lærerutdanningene: Det utdannes lærere nå som ikke kan programmere i matematikk
- Mangfold av elevtyper og lærere
- Programmeringsdidaktikk, i fag
- På sikt: Hvordan fornyes fagene?

Prosjekt med Oslo kommune

- 40 ungdomsskolelærere på kurs på UiO (1 + 4/2 dager)
- De har tilsvarende kurs med kolleger på egen skole
- Nettverksmodell, eneste måte å nå langt ut innen 2020

Hvordan får vi til et nasjonalt fellesløft for programmering i matematikk?