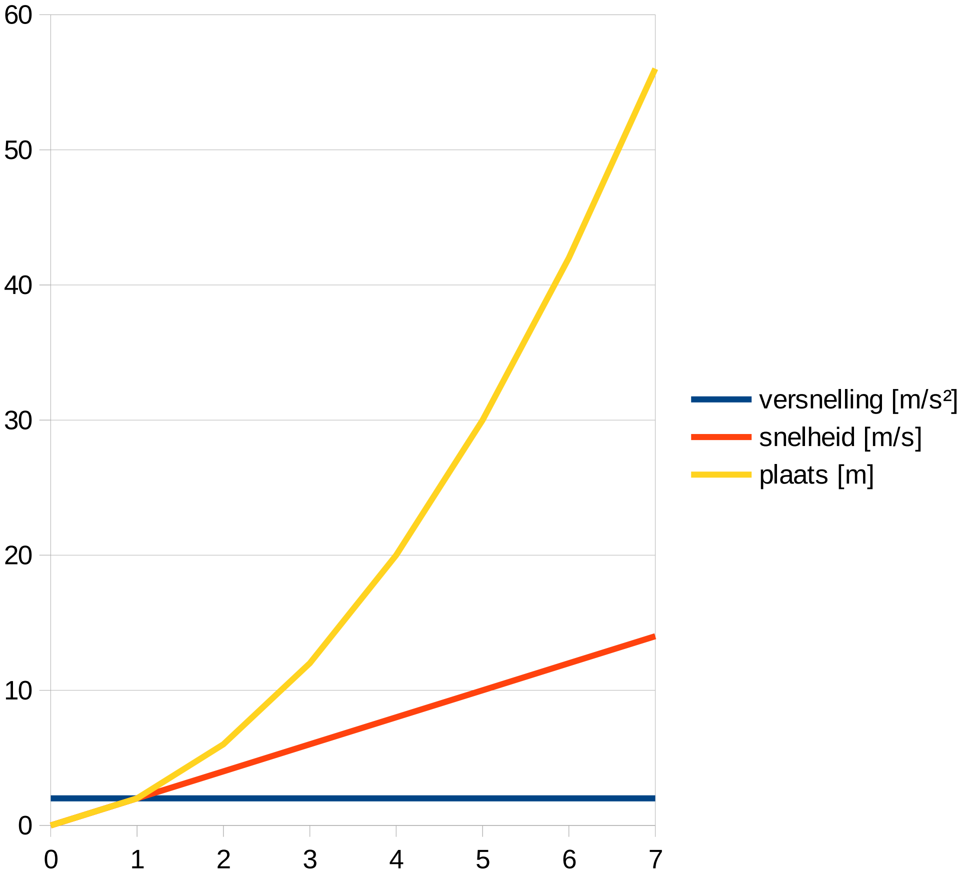
Snelheid auto neemt elke s toe met 2 m/s
Versnelling is 2 (m/s)/s = 2 m/s²
Bijv. elke seconde updaten, dan Δt = 1s



Integreren

tijd [s]	versnelling [m/s²]	snelheid [m/s]	positie [m]
0	2	0	0
0+Δt=1	2	0+2=2	0+2=2
1+Δt=2	2	2+2=4	2+4=6
2+Δt=3	2	4+2=6	6+6=12
3+Δt=4	2	6+2=8	12+8=20
4+Δt=5	2	8+2=10	20+10=30
5+Δt=6	2	10+2=12	40+12=42
6+Δt=7	2	12+2=14	42+14=56

Differentieren

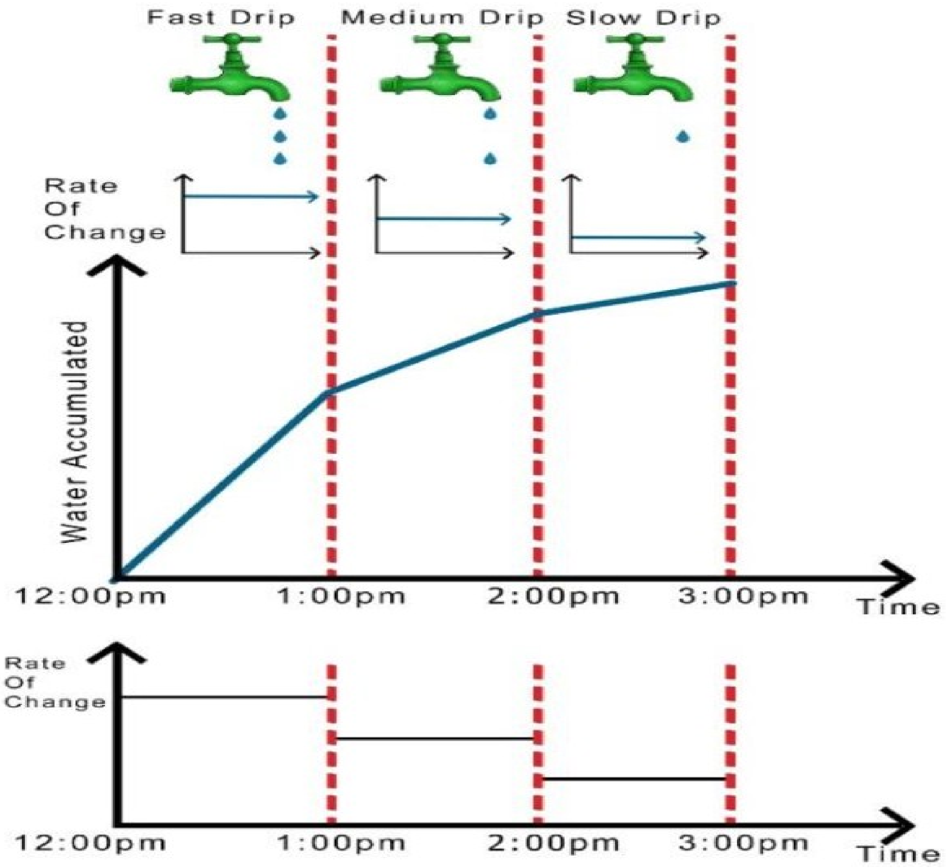
Hoe vaker je update, hoe beter ‘t klopt

Ideaal (‘limiet’): Δt → 0

Δt hoeft niet constant te zijn!

Kraan druppelt eerst snel, dan minder snel, dan nog minder snel

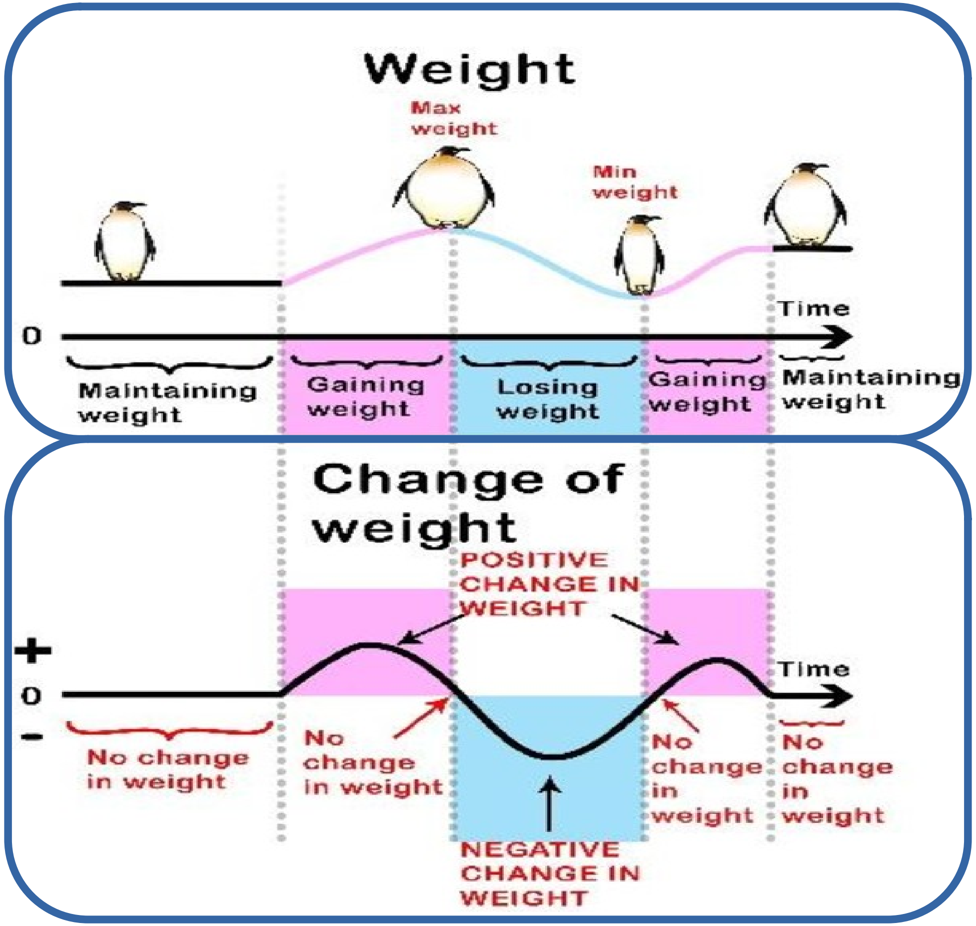
Integreren



Differentieren

Verandering kan zijn toename, maar ook afname…

Integreren



Differentieren

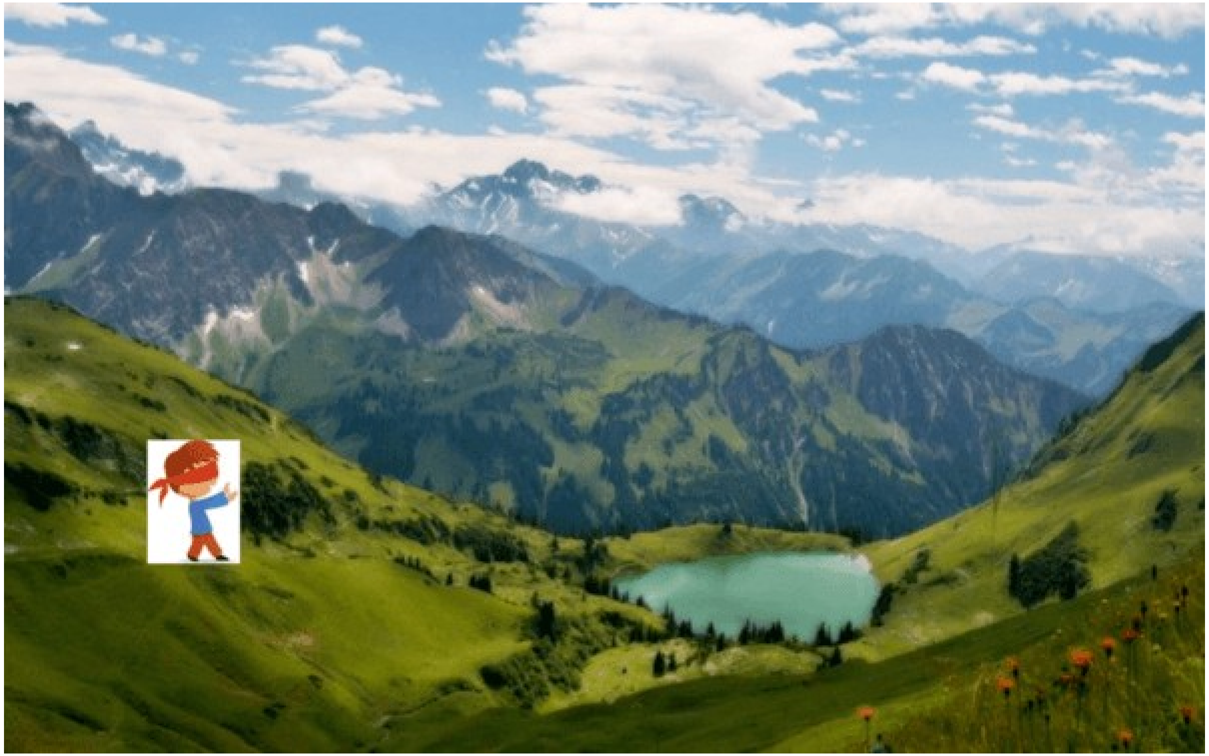
Verandering kan ook een richting hebben, dan heet het “gradient” en is het een vector (pijltje). Vaak werken we met <u>min</u> de gradient, die wijst precies de andere kant op...

De <u>richting</u> van <u>min</u> de gradient is de richting waarin je het snelst zou dalen (“steepest descent”).

De <u>grootte</u> van <u>min</u> de gradient is hoeveel meter je dan zou dalen per meter horizontale verplaatsing. Met andere woorden: Het is hoe snel de hoogte dan verandert, de afgeleide dus in de richting van het dal.

Deze skier skiet gelukkig niet volgens de steepest descent...

Gradient descent methode



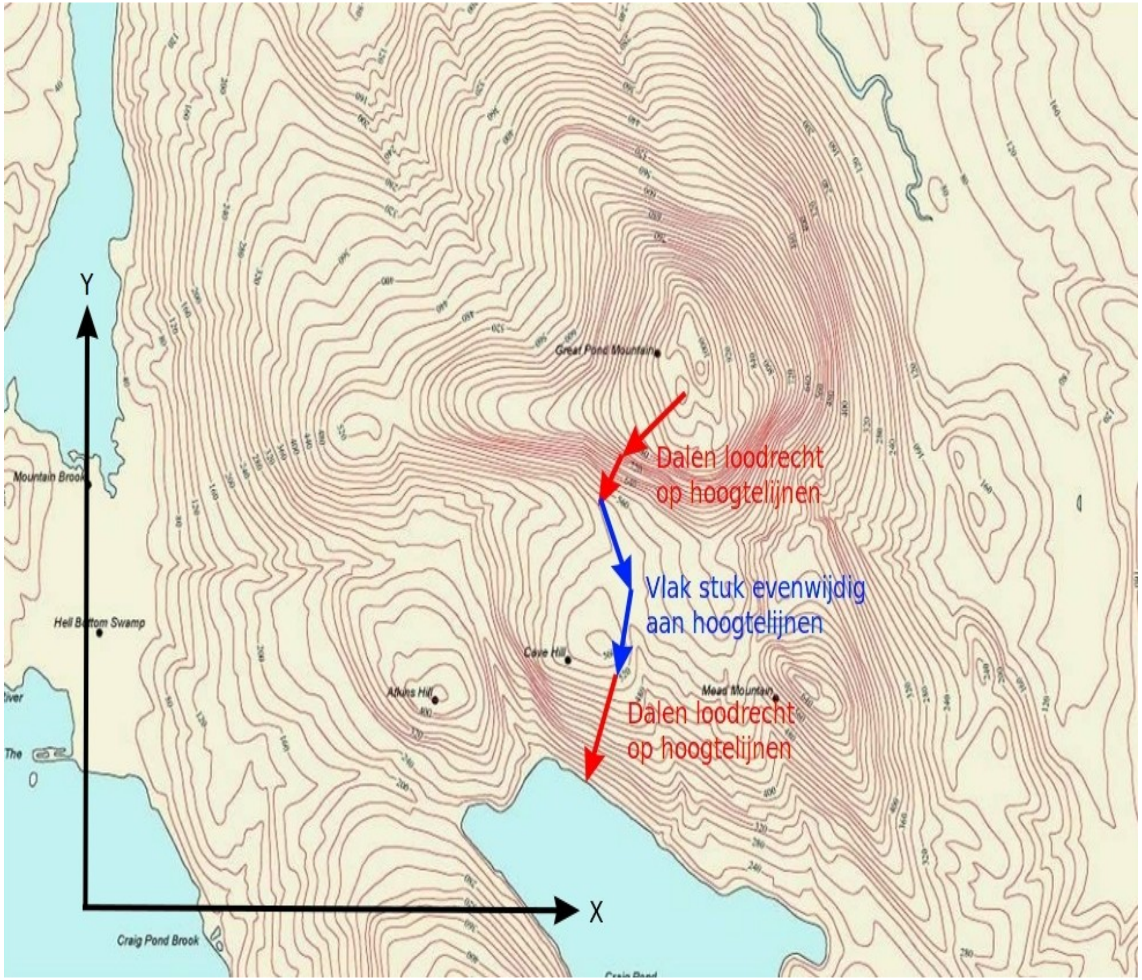
Hoe kom ik geblinddoekt zo snel mogelijk beneden?

Volg steeds de steilst dalende richting!

Maar wat als ik dan in een kuiltje strandt?

Verken af en toe de wijdere omgeving!

Van bovenaf gezien, <u>richting</u> van rode pijlen == <u>richting</u> van <u>min</u> de gradient



Als we afspreken dat:

z(x, y) de hoogte is bij een bepaalde x en y

dz/dx de toename van hoogte is per meter verplaatsing in de x richting ("differentieren naar x")

dz/dy de toename van hoogte is per meter verplaatsing in de y richting ("differentieren naar y")

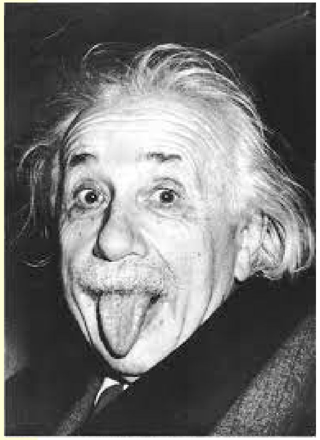
Dan blijkt dat de gradient-pijl een verplaatsing weergeeft van dz/dx in de x richting en dz/dy in de y richting. De richting van de rode pijlen is steeds de richting van de <u>min</u> de gradient-pijl ter plaatse.

<u>Dus als je dz/dx en dz/dy meet of weet, weet je waar je heen moet om te dalen!</u>

Q: Dit is heel leuk in 3D, maar als er nu eens meer dimensies zijn?

Bijv *f (a, b, c, d, e)* met *f* de "hoogte" (cost function) en *a, b, c, d* en *e* de "gewichten" in een neuraal netwerk?

A: Dan kan niemand, ook niet Einstein, zich er nog iets 3D-igs bij voorstellen!



Maar wiskundig valt aan te tonen dat je dan nog steeds de kosten kunt minimaliseren (het netwerk "trainen"), door *a*, *b*, *c*, *d* en *e* te veranderen in de richting van min de gradient, die dan geen 2 maar 5 componenten heeft, *df/da, df/db, df/dc df/dd* en *df/de*...

Wiskunde gaat dus verder dan ons 3D voorstellingsvermogen.
Niks bijzonders, de smaak van een appel is toch ook niet 3D…

Wiskunde is de manipulatie van betekenisloze symbolen volgens vaste regels!
(Zie video-les over getallen)

En zo'n wiskunde-spel blijkt soms overeen te komen met een deel van de realiteit dan niet ruimtelijk voor te stellen is.

Toeval? Of is wiskunde de realiteit *achter* de realiteit? (Google: Roger Penrose)