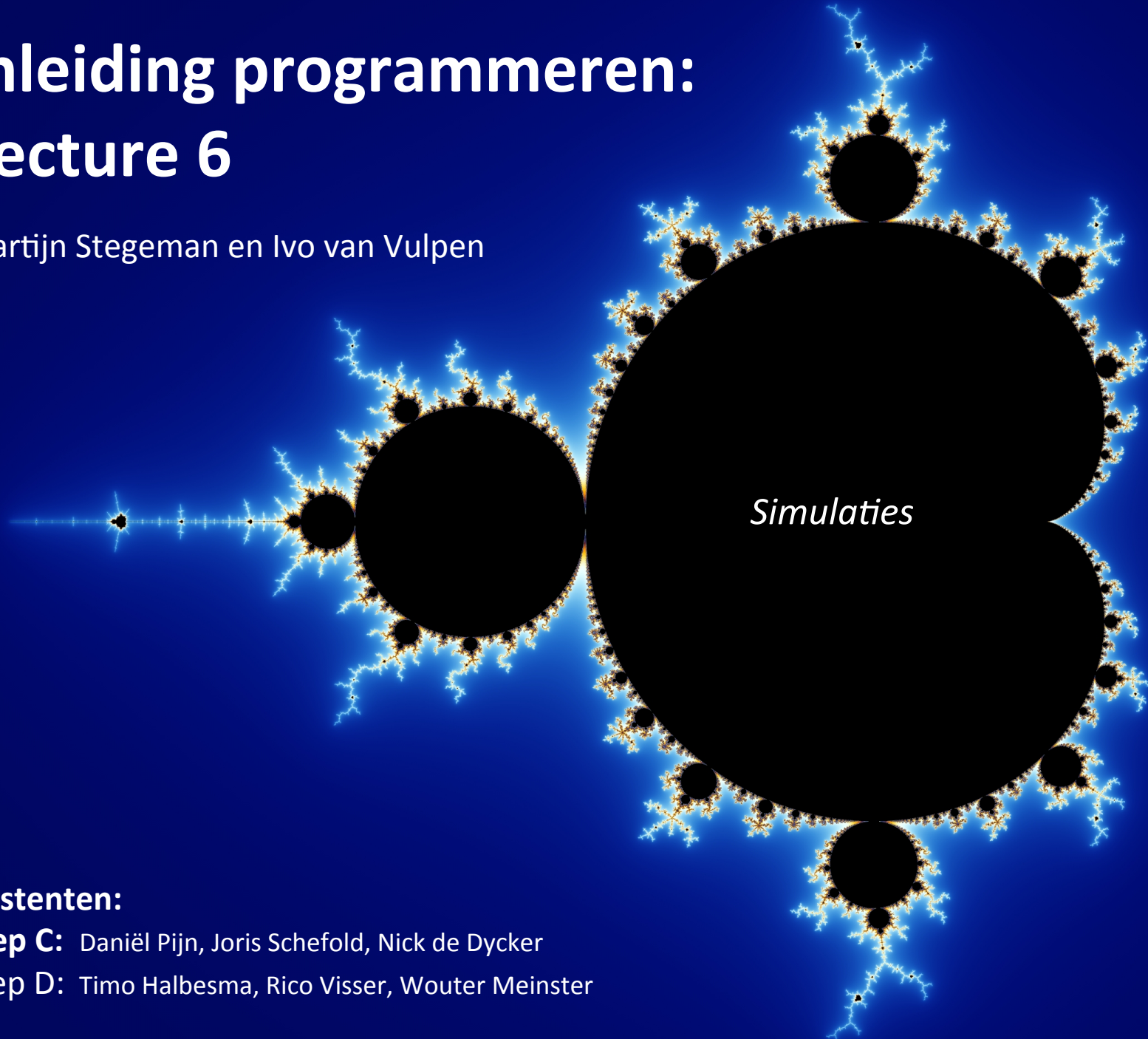


Inleiding programmeren: Lecture 6

Martijn Stegeman en Ivo van Vulpen



Assistenten:

groep C: Daniël Pijn, Joris Schefold, Nick de Dycker

groep D: Timo Halbesma, Rico Visser, Wouter Meinster



Week 2

- *Basis wiskunde*
- *Vergelijkingen*

Week 4

- *Numeriek integreren*
- *Fitten van data*

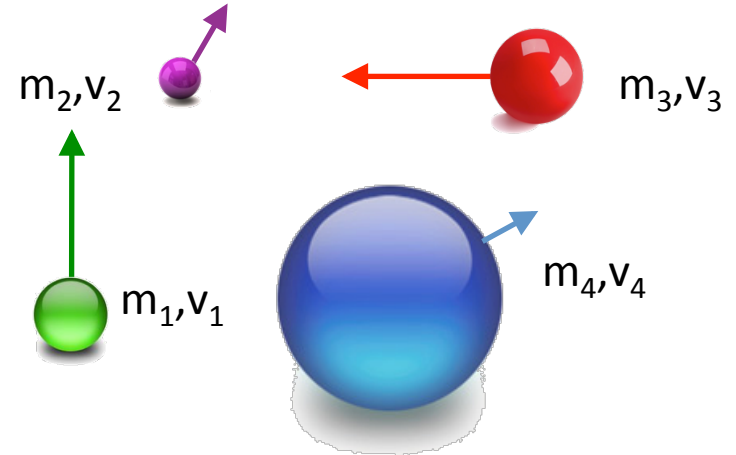
Week 6

- *Simulaties*

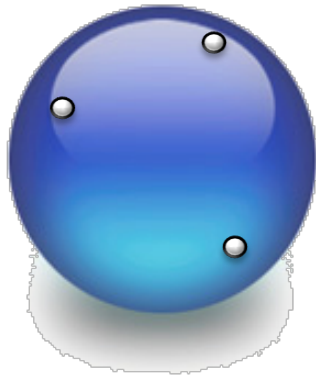
Week 7

- *Data-analyse*

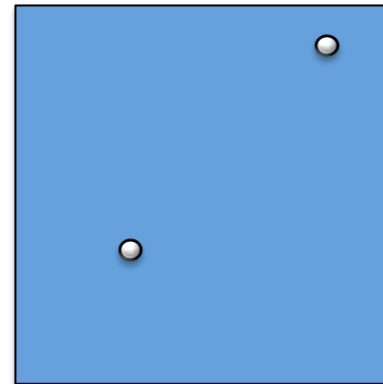
1) N-body probleem (gravitatie)



2) geometrie



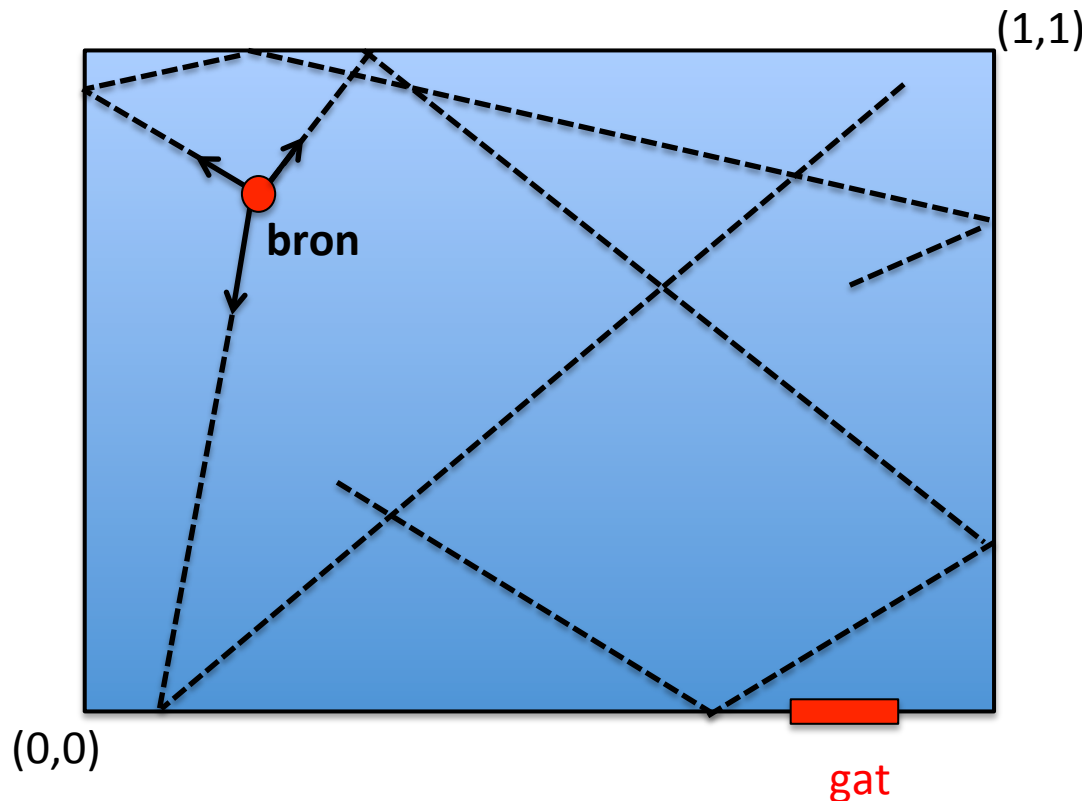
Punten op een bol equi-distant
3 / 10 ? Waar en afstanden ?



Gemiddelde afstand van 2 punten
op een bol of in een vierkant ?

Opgave deze week: deeltjes in een (2-dimensionale) doos

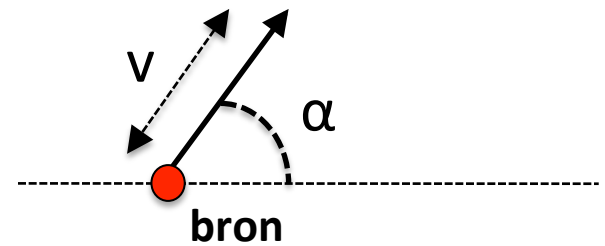
In een doos ($0 < x < 1$ en $0 < y < 1$) worden vanuit een bron $(x_{\text{bron}}, y_{\text{bron}}) = (0.25, 0.75)$ worden op $t=0$ een aantal deeltjes gegenereerd met random snelheid en richting



Deeltjes hebben random
snelheid en richting

snelheid (v_i): $0 < v_i < 0.10$

hoek (α): $0 < \alpha < 2\pi$



Tip: op $t=0$: v en $\alpha \rightarrow v_x$ en v_y

Doel: kijk hoe het systeem evolueert

neem kleine stapjes in de tijd en hou voor elk deeltje

de positie x , y en de snelheid v_x en v_y bij. Tuples als vorige week (of lists)

Basics

Aannames:

- de deeltjes botsen elastisch tegen de wanden
- de deeltjes hebben geen afmeting en botsen niet

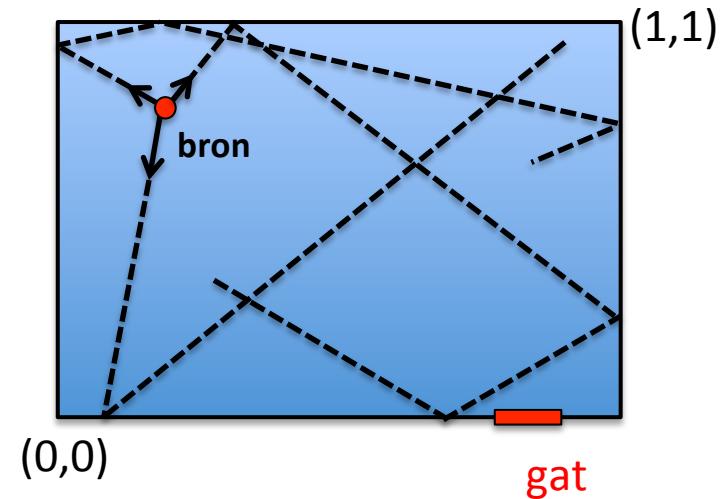
Opgave 1 (basics)

Tip 1: Neem kleine stapjes Δt in de tijd en bepaal voor elk deeltje de nieuwe x en y positie

$$x(t+1) = x(t) + v_x(t) * \Delta t$$

Tip 2: Behandel x en y afzonderlijk

Tip 3: begin met 1 deeltje en kijk of je zijn pad goed kan beschrijven. Teken het pad op het scherm ?



Opgave 1 (basics)

1/2) Plot aantal deeltjes aan de rechte kant van de doos ($x > 0.5$) als functie van de tijd en plot de gemiddelde afstand tussen de deeltjes

→ probeer altijdvooraf een inschatting te maken van het resultaat

Maak nu een gat in de doos ($y_{\text{gat}} = 0$ en $0.8 < x_{\text{gat}} < 0.9$)

3/4) Plot het aantal deeltjes in de doos als functie van de tijd. Hoe lang duurt het voor de helft van de deeltjes is verdwenen ?

→ probeer altijdvooraf een inschatting te maken van het resultaat

Hacker

Realisme:

- deeltjes hebben een afmeting en kunnen botsen
- animatie (film) van de evolutie van het systeem

Hacker uitbreidingen:

- 1) Gebruik animaties (`animation_template_circle.py`)
- 2) Geef deeltjes een afmeting
- 3) Laat deeltjes botsen (puntdeeltjes)
- 4) Laat deeltjes 'echt' botsen, waarbij afmeting (= massa) van deeltjes apart kunnen worden ingesteld en de botsing ook realistisch is.
- 5) Hacker hacker: Boltzman snelheidsverdeling

