

# Programmeervaardigheden

## Opdracht 2: Cellulaire Automaten in MATLAB

*2B Bio-Ingenieur  
Academiejaar 2012-2013*

Bart Meyers  
bart.meyers@ua.ac.be

### Belangrijke informatie

- Maak deze opdracht per twee.
- Dien de opdracht in op Blackboard, onder “opdrachten”.
- De deadline is maandag 18 maart 2013 om 23u55 (volgens de klok van Blackboard!).
- Je krijgt feedback op 25 maart 2013.
- Belangrijke bestanden:
  - assignment2.pdf: de beschrijving van de opdracht.
  - gameoflife.m: je startpunt voor de implementatie van Conway’s Game of Life. Je zal dit bestand moeten aanpassen.
  - show.m: een MATLAB script met code die jouw cellulaire automaat visueel kan weergeven.
  - pulsar.m: een script waarin een “Pulsar” patroon als matrix wordt geïnitieerd.
  - puffer\_train.m: een script waarin een “Puffer Train” patroon als matrix wordt geïnitieerd.
- Benodigdheden: MATLAB.
- Materiaal: MATLAB slides hoofdstukken 1, 2, 3, 11, 12.

## Doelstelling

De bedoeling van deze opdracht is dat je twee voorbeelden van cellulaire in MATLAB implementeert, gebruik makend van de gegeven MATLAB bestanden. Deze opdracht test het gebruik van matrices, arrays, lussen en if-testen.

## Achtergrond

Een cellulaire automaat is een veelgebruikt simulatiemodel uit de theoretische biologie. Het kan gebruikt worden voor het simuleren, en dus analyseren en voorspellen van diffusie van gassen of vloeistoffen, de vlucht van vogels in een zwerm, ziekte-epidemieën in populaties, of stedenbouw, volgens een aantal eenvoudige regels.

Een cellulaire automaat wordt opgebouwd uit een seed en regels. De *seed* is een n- of meerdimensionaal raster van cellen dat een gediscretiseerde voorstelling is van een doorgaans continu universum voorstelt (doorgaans komen deze cellen overeen met coördinaten in dit universum). Elke cel kan een waarde hebben, bijvoorbeeld al dan aanwezig zijn van een besmette mens op een vierkante meter, of de dichtheid van een gas per vierkante centimeter.

Het gedrag van het universum (doorgaans in de tijd) wordt bepaald door de *regels*. Volgens deze regels wordt er per tijdstap voor elke cel een nieuwe waarde berekend, aan de hand van haar eigen waarde en die van haar nabije cellen. Een vogel zal zich bijvoorbeeld verplaatsen naar een positie met minder tegenwind, een gas zal makkelijker naar een plaats vloeien waar er minder massadichtheid is, en veel mensen zullen liever dichtbij stadskernen gaan wonen.

De regels en de seed moeten zo goed mogelijk overeenkomen met de werkelijkheid om een zinvolle simulatie te verkrijgen. De cellulaire automaat is dus een abstractie van een realistisch gedrag. Sommigen beweren zelfs dat het universum in wezen een cellulaire automaat is.

## Opdracht : Conway's Game of Life

Je zal in deze opdracht opnieuw John Conway's Game of Life<sup>1</sup> implementeren, maar deze keer voor MATLAB. John Conway bedacht in de jaren '70 de "Game of Life" als een wiskundig spel. Het gaat om een cellulaire automaat waarin het universum een tweedimensionale matrix is, en elke cel twee waarden kan hebben: 0 of 1 (oftewel dood of levend). Bij een nieuwe tijdstap wordt elke cel berekend aan de hand van haar eigen waarde en de acht omliggende cellen. De Game of Life bevat steeds de volgende regels:

- een levende cel (waarde 1) met minder dan twee levende omliggende cellen sterft door onderpopulatie;
- een levende cel met twee of drie levende omliggende cellen blijft leven;
- een levende cel met meer dan drie overlevende omliggende cellen sterft door overpopulatie;
- een dode cel met exact drie levende omliggende cellen wordt levend door reproductie;
- een dode cel met niet exact drie levende omliggende cellen blijft dood.

Naargelang de seed kunnen er interessante patronen optreden. In deze opdracht zijn de "Pulsar"- en "Puffer Train"-patronen bijgevoegd.

Een speciale behandeling is nodig voor de cellen aan de rand van het universum oftewel matrix, aangezien zij niet aan elke kant aangrenzende cellen hebben. Voor deze opdracht zal je de cellen aan de rand net zo moeten implementeren als normale cellen, volgens de regels.

---

<sup>1</sup>Lees hierover meer op Wikipedia: [http://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life).

Concreet luidt de opdracht als volgt:

1. Implementeer de Game of Life in `gameoflife.m`. Merk op dat de seed-matrix al gegeven is (door het laden van het `pulsar.m` script of het `puffer_train.m` script) die zal resulteren in een patroon. Je zal ervoor moeten zorgen dat je nieuwe universum cel per cel opgebouwd wordt volgens de vijf regels die hierboven beschreven zijn. Dit gebeurt in totaal voor **TIME** (zie begin `gameoflife.m`) tijdstappen. Zorg ervoor dat je de array `life` aanvult die het universum op elk tijdstip bevat, dus een array met **TIME** universums. Het is de bedoeling dat je voor elke tijdstap een aparte matrix aanmaakt (let dus op dat je je matrix van de vorige tijdstap niet rechtstreeks aanpast).
2. Zorg ervoor dat je oplossing ook werkt voor de cellen aan de rand van het universum.
3. Voer je oplossing uit voor meer tijdstappen (let er wel op dat het berekenen bij Puffer Train erg lang kan duren). Wat gebeurt er bij de Pulsar en bij de Puffer Train? Is het universum periodiek oftewel “stabiel”?
4. Schrijf een kort verslag (ongeveer 1000 woorden) waarin je je oplossing uitlegt, moeilijkheden en hun oplossingen aankaart en je resultaten analyseert. Leg het verschil uit met je implementatie in Python (opdracht 1) en vergelijk Python en MATLAB.