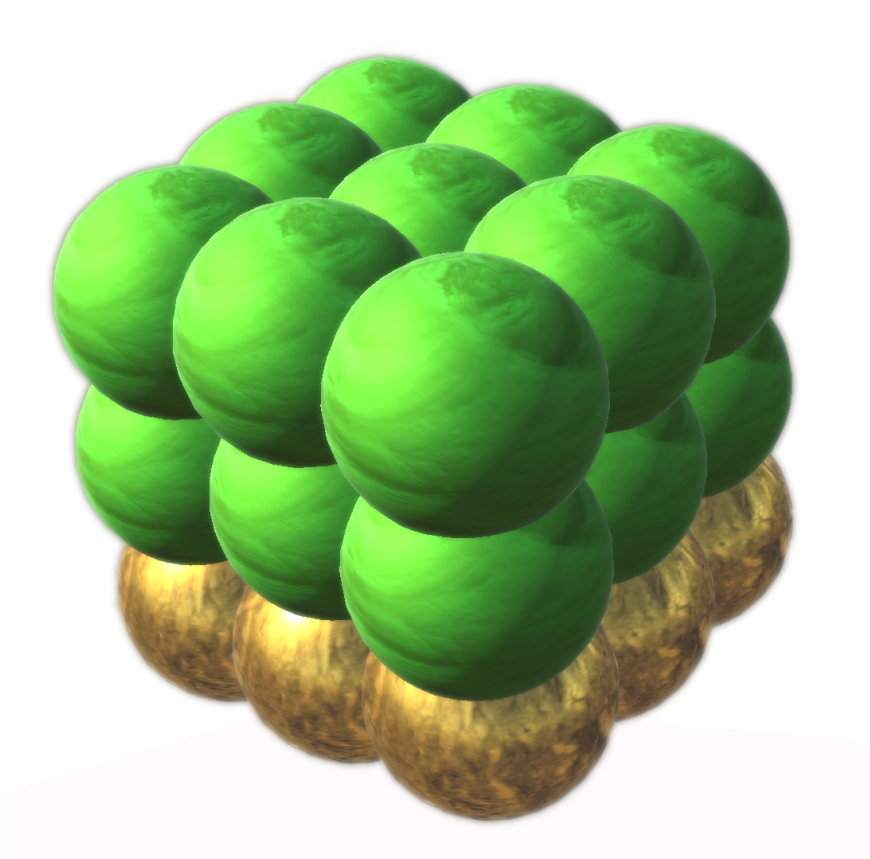




**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

---



## **Evolved ecosystems**

A simulation of an emerging complex adaptive system

Master's thesis in Complex Adaptive Systems

**JOAKIM JOHANSSON**





MASTER'S THESIS FRT 2017:17

# Evolved ecosystems

A simulation of an emerging complex adaptive system

JOAKIM JOHANSSON



Department of Energy and Environment  
*Division of Physical Resource Theory*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2017

Evolved ecosystems  
A simulation of an emerging complex adaptive system  
JOAKIM JOHANSSON

© JOAKIM JOHANSSON, 2017.

Supervisor: Claes Andersson, Department of Energy and Environment  
Examiner: Claes Andersson, Department of Energy and Environment

Master's Thesis FRT 2017:17  
Department of Energy and Environment  
Division of Physical Resource Theory  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Depiction of the organism used to initialise simulation. See section 3.1 for more information.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2017

Evolved ecosystems  
A simulation of an emerging complex adaptive system  
JOAKIM JOHANSSON  
Department of Energy and Environment  
Chalmers University of Technology

## Abstract

The study of artificial life has been ongoing since even before the term was coined by Christopher Langton in 1986 as an attempt to broaden the definition of life beyond its organic form. This master's thesis introduces *Grafliv*, a model of artificial life based on a particle simulation run on a general-purpose graphics card. This is done for the purpose of studying the construction of open-ended evolution, its principles and possibilities, in complex adaptive systems. The organisms in the model consists of cell particles determined through pattern-producing networks that evolve over time in a manner inspired by the CPPN-NEAT method proposed by Stanley in 2007.

Results from a 14-day simulation are presented and include a list of several distinctly adapted lineages originating from a single initial organism. Categorized trophic levels include plants, decomposers and predators, with examples from each level described and compared.

Selecting an appropriate modelling level was arguably one of the more difficult challenges in the project, thus different possible approaches are discussed, together with the problem of creating a suitable environment for the organisms.

Finally, it is concluded that, while life definitely is a hard-to-define subject, this work could be considered a step on the way towards an understanding of its nature, its characteristics and its limitations.

Keywords: Artificial life, CUDA, Complex Adaptive Systems, CPPN, NEAT, Ecosystem evolution, Niche construction, Ecosystem engineering



# Acknowledgements

First of all, I would like to thank my supervisor Claes Andersson for his inspiration, support and encouragement. Whenever I felt stuck or out of ideas during this year-long project a meeting with Claes always provided new energy. Many thanks also to the other members of the department of Physical resource theory, for their questions, recommendations and exciting lunch discussions. Moreover, I want to thank my opponent Victor Nilsson for his enquiries and for our valuable discussions. My deepest gratitude also to my fiancée Ellen Bohlin for both her love and support, as well as her knowledge of biology. Thank you for listening and helping not only when you were the expert, but also when you had no idea of what I was talking about. Thanks also to the rest of my family, for encouraging me in my choices and always being there when I needed you. This of course also includes the Bohlins. Finally, thanks to everyone in Stenungsunds Taidoklubb Taidokan, for being such great friends and for putting up with the physics and mathematics that often slip through into the dojo.

Joakim Johansson, Gothenburg, October 2017



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Ecological, societal and ethical aspects . . . . .	2
1.3 Aim . . . . .	3
1.4 Objectives . . . . .	3
1.5 Demarcations . . . . .	3
1.6 Method . . . . .	3
<b>2 Model</b>	<b>5</b>
2.1 Particle-based physics . . . . .	5
2.1.1 Interaction functions . . . . .	5
2.1.2 Particle types . . . . .	5
2.1.3 Cell types . . . . .	7
2.2 The arena . . . . .	8
2.2.1 Water and air . . . . .	8
2.2.2 Terrain . . . . .	8
2.2.3 Boundaries . . . . .	8
2.3 Definition of an organism . . . . .	9
2.3.1 Compositional Pattern Producing Networks . . . . .	11
2.3.2 Neuroevolution of Augmenting Topologies . . . . .	12
2.3.3 Genome . . . . .	13
2.3.4 Nervous system . . . . .	13
2.4 Organism coordinate system . . . . .	14
2.5 The energy cycle . . . . .	15
<b>3 Results</b>	<b>19</b>
3.1 Initial organism . . . . .	19
3.2 Lineages and trophic levels . . . . .	20
3.2.1 Plants . . . . .	20
3.2.1.1 Swimming upwards . . . . .	20
3.2.1.2 Larger organisms . . . . .	22
3.2.1.3 Pyramid-shaped organisms . . . . .	22
3.2.1.4 Higher ratio of fertile organisms . . . . .	25

3.2.1.5	Fewer eggs per organism . . . . .	25
3.2.1.6	Three plant organisms . . . . .	27
3.2.2	Decomposers . . . . .	28
3.2.2.1	Three decomposer organisms . . . . .	28
3.2.3	Predators . . . . .	30
3.2.3.1	Three predator organisms . . . . .	31
3.2.4	Inter-lineage relations . . . . .	33
<b>4</b>	<b>Discussion</b>	<b>37</b>
4.1	On choosing a suitable environment configuration . . . . .	37
4.2	Modelling level . . . . .	38
4.3	Future work . . . . .	39
<b>5</b>	<b>Conclusion</b>	<b>41</b>
	<b>Bibliography</b>	<b>43</b>
<b>A</b>	<b>Getting started with Grafiliv</b>	<b>I</b>



# List of Figures

2.1	Transformations between the different particle types. Since changing the total number of particles is computationally expensive, especially decreasing the particle count, a buffer is kept for creating new cell and energy particles. A cell usually turns into a detritus particle as it loses its energy, however it might in some cases turn directly into a buffer particle. Energy particles are created corresponding to the number of photosynthetic cells and will turn into buffer particles as their respective cells die. . . . .	6
2.2	Depiction of the arena terrain and water. These elements were introduced to enable organisms to settle in different environments and possibly diversify. The image is rendered at the initiation of the simulation described in Chapter 3, with the single initial organism barely visible in the centre of the arena. . . . .	9
2.3	Visualisation of all elements of the particle model. The purple object is the terrain, with the larger blue particles making up its vertices. The multicoloured particles above the terrain are energy, detritus and cell particles, with their colour representing their energy level. The opaque particles are the cells, while the transparent particles above are mostly energy particles, but also detritus. A blue energy particle belongs to an occluded photosynthetic cell while a red one indicates that it has not yet collided with anything before reaching the top. Detritus particles of varying remaining energy can be seen among the organisms beneath the water surface. The thin blue bounding box determines the boundary that the particles cannot pass. The large number of blue transparent particles beneath the terrain are buffer particles. . . . .	10
2.4	Example of a compositional pattern-producing network. Input is $z$ and $y$ coordinates. The actual simulation model also uses the $z$ -coordinate as well as $r$ , the radial distance to origo, but they are for the sake clarity not included here. For the same reason, all links have the weight of one. The pattern seen on the right is the result of evaluating the network seen on the left. The currently evaluated cell $(-2, 0)$ is marked with a blue border and is coloured green since the output node corresponding to the green colour has the highest output value. Numbers above the nodes are their input values while the numbers above are their output. . . . .	12

2.5	Identifiers for the six three-dimensional von Neumann neighbours to a cell. Subscripts denote distance in x, y, and z directions between the cell and each neighbour at birth. . . . .	14
2.6	Energy particles rising from an early population of photosynthetic organisms. Each energy particle belongs to a photosynthetic cell. If the energy particle reaches the top of the arena without colliding with any cells, the photosynthetic cell it belongs to will continue gaining energy every frame. If the particle does collide, the cell will not get any energy until its particle once again can complete the distance without collisions. In this illustration, energy particles that have collided are coloured blue, while the rest are red. The cells are opaque and coloured according to the amount of energy they contain, with warm colours indicating higher energy density. A video animating the process can be found at <a href="https://youtu.be/jsNgbxvU5Js">https://youtu.be/jsNgbxvU5Js</a> . . . .	16
3.1	Initial organism genome and visualisation. No hidden nodes exist in the network. Since the nervous system had no initial connections, the organism was inanimate and hence no nervous system network is displayed here. . . . .	20
3.2	Simulation arena at the end of the simulation, at t=1 160 000. While the organisms have spread all over the arena, the large majority is found on the surface level. A video of the full simulation can be viewed at <a href="https://youtu.be/yMY1dIjbFY8">https://youtu.be/yMY1dIjbFY8</a> . . . . .	21
3.3	The number of cells belonging to each respective cell type over time. Note that the y-axis is logarithmic and that the photosynthetic cells thus represents a large majority. At the final timestep t=1 160 000, the photosynthetic cells constitute 90.83% of the cells, while the digest and sting cells are at 6.45% and 0.23% respectively. Vascular, fat, sense and buoyancy cells, having no part in energy collection or reproduction are very uncommon. . . . .	21
3.4	Cell altitude density over time. The more cells that occupy a given altitude (y-coordinate) at a given timestep, the more light the point appears in the chart. The plot range is clipped at a cell density of 5000, but the lightest area reaches close to 40 000 at around t=600 000. Due to gravity, the initial organism, having spawned in the exact centre of the arena, quickly falls to the bottom, below the water surface. This is where the population starts out, at an altitude of about 30, which can be seen in the bottom-left corner of the chart. Quite quickly, however, the organisms evolve movement vectors upwards and the large majority end up at the altitude of 75 by the water surface, after about t=200 000. The fact that the organisms are spread out and focused on different altitudes shows signs of niche adaption. . . . .	22

3.5	Simulation statistics over time. As can be seen in the top-left chart, the number of particles stabilised after around $t=300\,000$ . However, the number of organisms (top-right chart) henceforth decreased steadily. This can be explained by the bottom chart, which clearly shows that the size of the organisms has, on average, increased over time. The jumps in the number of buffer particles are only the model ensuring that there is a sufficient amount of buffer particles available.	23
3.6	Example of morphology changing over time. The phenotype for every tenth ancestor of the organism with index 574174, which was one of the organisms living on the water surface at $t=1\,000\,000$ . It is located in the top-left corner, the oldest ancestor is in the bottom-right, as indicated by the generation labels. A trend of this ancestral line is that the organisms have increased in size, as well as turning more "pyramid-shaped". The radii of the top cells are, in the later organisms, significantly smaller than the bottom cells.	24
3.7	The number of fertile organisms divided by the total number of organisms over time.	26
3.8	Distribution of the number of children per organism, given all organisms throughout the whole simulation. A majority of the organisms never reproduced but multiple children are not uncommon among those that did, as can be seen in the histogram on the bottom. Note that the y-axis is logarithmic and that the number of children as such follows a power-law distribution.	26
3.9	The average number of egg cells per organism over time. At about the same time as the total number of cells in the simulation stabilised ( $t=400\,000$ ), each organism settled on having about one egg cell each.	27
3.10	Rendered view of decomposers living off eating the dead cells falling down from the surface-level plants. Note the occasional light-blue vascular cell in some of the decomposers. A full video of the whole simulation as seen from beneath the water surface can be found at <a href="https://youtu.be/evZK16QmrBk">https://youtu.be/evZK16QmrBk</a> .	29
3.11	Rendered frame of the predator lake at $t=1\,008\,020$ . Because of the topology of the terrain, the small lake at the corner of the arena was very beneficial for predators. Without having to move, the predators could use their black sting cells to take energy from the organisms sliding down the slope into the lake. A video of the same scene can be found at <a href="https://youtu.be/onh0dPyuJiE">https://youtu.be/onh0dPyuJiE</a> .	30
3.12	Rendered frame of the predator lake at $t=1\,008\,020$ . This is the same frame as in figure 3.11, but viewed from below, through the terrain. Here the predator organisms at the lake surface can be more easily seen. A video of the same scene can be found at <a href="https://youtu.be/Wm2TUDi2fPs">https://youtu.be/Wm2TUDi2fPs</a> .	31

3.13	Example of a predator lineage. The phenotype for every tenth ancestor of the organism with index 630807, which was one of the organisms living in the predator bay at the end of the simulation, at t=1 160 000. Note that the organism portraits are taken at certain exported frames and that, as such, some cannot be found while others have lost cells since their birth. The predators originate, as all organisms do, from the initial organism at generation 0. . . . .	32
3.14	Positions of the nine example organisms described in subsections 3.2.1.6, 3.2.2.1 and 3.2.3.1 . . . . .	33
3.15	Each of the nine example organisms described in subsections 3.2.1.6, 3.2.2.1 and 3.2.3.1 represent a lineage among either plants, decomposers or predators. This diagram represents them as the leaf nodes of their family tree, with their mutual most common ancestors included all the way to the initial organism. Note that the organism visualisations are taken from saved timesteps and that the organisms are at different stages of decay. The ancestor 624933 was not found on any saved timestep on record . . . . .	34
3.16	Genome (left) and nervous system (right) of the surface-dwelling plant with id 574174. The genome node #59 and the nervous system node #59 are both missing hidden nodes, where the nodes have been removed while the connections remain. Thus they have no activation function and no value. The reason for the connections remaining is an unknown bug. . . . .	35

# List of Tables

2.1	Example of a cell types energy configuration. <i>Energy in</i> is the portion of the energy offered by a neighbour that is absorbed, <i>Energy out</i> is the portion of the cell's surplus energy that is offered to a neighbour.	17
3.1	Statistics about the performed large timescale simulation, detailing its extent and magnitude . . . . .	19
3.2	Example of three different plant organisms, each using photosynthesis but occupying different parts of the arena and having different adaptions. . . . .	28
3.3	Example of three different decomposer organisms, each gaining energy from eating detritus particles . . . . .	29
3.4	Example of three different predator organisms, each gaining energy by stealing it from cells of other living organisms . . . . .	33



# 1

## Introduction

This chapter introduces the project, its goals and limitations, as well as the history and background of Artificial Life research.

### 1.1 Background

What is life? In the book *Life: the science of biology*, Sadava et al. (2014) list a set of characteristics shared by living organisms: They are composed of a certain set of chemical components, they consist of cells (mostly), they use energy to process molecules from their environment, they contain genetic information that changes as the populations evolve over time, et cetera. The authors admit that the list is not exhaustive and that it does not apply to all life at all times. A seed of a desert plant does not consume any energy and can go for years without water, still, it could be considered alive. Likewise, a virus has no cells, but many biologists would consider it a form of life.

Although it might feel as if the definition of life is easy and intuitive, we know that flowers may cease to be alive if we forget to water them, it is not so easy to settle on a definition. An approach toward a more clear definition of life would be to try creating life artificially. The term *Artificial Life* was first coined by Langton (1986) in his paper *Studying artificial life with cellular automata*, where he suggested a broadening of the definition of life beyond its organic form.

Also, Farmer & Belin (1990) argued for a wider definition of life, in a paper where they describe possibilities and dangers of artificial life and how it might broaden our understanding of the nature of living things.

Many different versions of artificial life have been investigated, either realised in software, hardware or as synthetic biology. These three kinds are usually called "soft", "hard" and "wet" artificial life respectively (Bedau 2003).

Many artificial life simulations have had their creatures optimise toward a specific goal, such as distance travelled or likewise, but a few have instead tried the more implicit fitness function of survival alone. As early as in 1970, John Horton Conway devised the now-famous cellular automaton called Game of Life (Gardner 1970). Although computers were used for his more time-consuming experiments, he carried out many of the smaller ones manually by hand. The cellular automaton was remarkably lifelike, especially considering the simplicity of its rules. However, it took until 2010 before a self-replicating creature could be created in the simulator, achieved by an enthusiast named Andrew Wade (Aron 2013), so, while somewhat lifelike, it is not the most optimal model for studying evolution.

In 1992, Thomas S. Ray developed the computer program *Tierra* (Ray 1992). *Tierra* used the computer memory and CPU time as analogues for energy and material resources in biology. Organisms, having a genome of machine code, reproduce by executing their code to copy themselves. Mutations occur at a given rate in the form of randomly flipped bits in the memory. *Tierra* showed many interesting results in the form of co-evolution, parasitism and symbiosis.

In 1994, Larry Yaeger introduced *Polyworld* (Yaeger 1994). *Polyworld* was a three-dimensional world populated by polygon creatures. By seeing a one-dimensional band of pixels, the creatures evolved the ability to, to some extent, hunt for food - in the form of either pellets or other creatures - and to avoid predators. *Species* (Schumacher 2011), created by James Schumacher is a similar but more modern example in the form of a simulation game, with plants also there being a non-evolving part of the ecosystem.

While both computing power and the knowledge of biology has increased over time, there are still a lot of questions remaining to be answered. Ray (1992) described his ambition to simulate an origin of diversity similar to that of the Cambrian explosion and his work with *Tierra* was a first step on the way. A central part of the study of the Cambrian explosion and origin of the Metazoan ecosystems is the process of niche-construction, where the evolution of one species create new niches for additional species (Erwin & Valentine 2013). This master thesis introduces the simulation program named *Grafiliv* and will be on the level of basic research in the principles of ecosystem evolution, niche construction and ecosystem engineering.

## 1.2 Ecological, societal and ethical aspects

Simulating an artificial ecosystem certainly has ecological aspects; insights gained by constructing and studying artificial ecosystems can be adapted in order to better the understanding of natural ecosystems and their dynamics. Although many differences will exist, at the very least a model will not - by definition - be the same as the system it tries to model, the effects of an event (e.g. a removal of a top predator species) in a simulation can on a principle basis hint to the possible effects of such an event in a natural ecosystem.

Considering societal aspects, open-ended evolution has bearing not only on biological evolution but also on innovation processes in general. By investigating how adaptive novelty can continually drive the emergence of more adaptive novelty elsewhere, transforming the environments to which it adapts, a better understanding can be reached in the dynamics of other complex systems as well; one such example being societal systems.

Finally, on a more philosophical note, how do you define life? Since artificial life can be considered a subset of life, what conditions does an artificial creature need to fulfil in order to be considered living? If you decide to consider a virtual creature to be living, will there at any point arise an ethical obligation to keep it alive? At the current time being, however, experimenting with an artificial ecosystem can arguably be considered more ethical than experimenting with its biological counterpart.



### 1.3 Aim

Understanding the nature of open-ended evolution is an old and open problem and if this project can shed some light on the issue, it could be of large scientific value. This project aims to investigate how life can be simulated using GPU (Graphics Processing Unit) computing.

### 1.4 Objectives

The goal is to construct open-ended evolution in a simulated ecosystem, using behavioural models and genetically driven morphology. Emphasis will not be placed on detailed similarity with biological evolution and innovation but on in-principle similarity with evolution and innovation in complex adaptive systems.

### 1.5 Demarcations

Although the computer used for running the simulations can be considered powerful, it still has limitations regarding the number of particles it is able to simulate in a certain amount of time. Simulating more particles, more generations and more interactions will result in longer simulation times. Finally, since particles are allocated on GPU memory, there will be a limitation on how easily the number of particles simulated could be changed dynamically.

### 1.6 Method

The simulation model was constructed in iterations, with further complexity added along the course of the project. In order to simulate large populations over long periods of time, the project was implemented using the Nvidia CUDA programming on a high-end GPU (nVidia Quadro M6000).

This work was facilitated using the library *Fluidix* (OneZero Software 2016). *Fluidix* makes it possible to write custom interaction function to be executed on either each particle or each pair of particles within a certain distance from one another. Such interactions are executed in an optimised manner by utilising the power of GPU computing.



# 2

## Model

The simulation is at the essence a particle-based model. Spherical particles are used to represent entities analogous to biological cells, both dead and living, along with energy particles. This chapter will describe the simulation model, including the physics implemented to handle the particle interactions as well as the evolutionary and organism-related structures created around them.

### 2.1 Particle-based physics

Each particle has properties such as position, velocity, force, radius, density and energy level. Particles also have a given particle type and, if their particle type is *Cell*, also a given cell type. These properties are accessible and modifiable in concurrent interaction functions and will be described in detail in this section.

#### 2.1.1 Interaction functions

Functions that need to be applied to a majority of particles, such as handling terrain collisions, boundary conditions, buoyancy, energy decay and position updates, are implemented as interaction functions and are executed in parallel on the GPU through the Fluidix library.

This is also true for particle-pair interactions: For all non-buffer particles within a predefined distance, a repulsive force is applied unless they are cell neighbours<sup>1</sup>. Depending on the particle types, and in the case of cells also their cell types, energy will be exchanged between the particles upon contact. For example, a digestive cell colliding with a detritus particle will gain some of its energy.

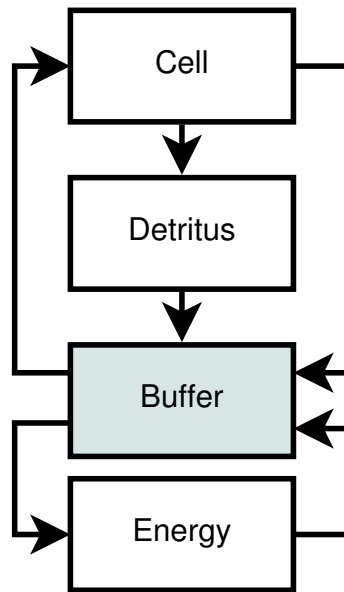
Finally, the model also make use of particle-to-surface interactions to handle collisions between particles and the terrain described in subsection 2.2.2.

#### 2.1.2 Particle types

There are four particle types implemented in the model: *Cells*, *Detritus*, *Buffer* and *Energy particles*. Particles change type depending on what happens to them in the simulation and transform according to Figure 2.1. The number of particles might be decreased if there exists an excess number of buffer particles at the end of the particle array, but the normal action for an unneeded particle is to turn into a buffer particle.

---

<sup>1</sup>Cell neighbours are defined in Figure 2.5



**Figure 2.1:** Transformations between the different particle types. Since changing the total number of particles is computationally expensive, especially decreasing the particle count, a buffer is kept for creating new cell and energy particles. A cell usually turns into a detritus particle as it loses its energy, however it might in some cases turn directly into a buffer particle. Energy particles are created corresponding to the number of photosynthetic cells and will turn into buffer particles as their respective cells die.

### Cell particle



The cell particle is analogous to the biological cell, although a significant simplification. Apart from the ordinary particle properties, a cell also has knowledge of which organism it belongs to (by an organism ID), which cells are its neighbours, and what type of cell it is. Neighbouring cells will, while in contact, exchange energy with each other, where the amount being exchanged depends on the cell types (see Table 2.1).

### Detritus particle



When a cell has less energy than the predefined amount *minCellEnergy*, the cell dies and turns into a detritus particle. All detritus particles are dead cells, but they might have an energy amount larger than *minCellEnergy* if they died of other causes than energy deprivation. Over time, the detritus will decay, losing energy, until none is left and the particle turns into a buffer particle and disappears.

### Buffer particle



A limitation of the Fluidix library is that changing the number of particles is a costly operation. Because of this, the model keeps a particle buffer in order to ensure that there always will be particles available. The ones not currently in use are set as buffer particles and positioned outside the simulation boundaries.

**Energy particle**

Photosynthetic cells receive energy while they have a clear path to the ceiling above them. To check if they do, energy particles are sent straight up from the photosynthetic cells to check for collisions.

**2.1.3 Cell types**

For cell particles, another property called *cell type* is also used. Cells each belong to one of eight discrete cell types, introduced to give evolution different adaptive options for harvesting and handling energy. Some cell types gain energy by interacting with other particles, while others store, transmit or simply spend energy.

**Photosynthetic cell**

Inspired by biological photosynthesis, the photosynthetic cell receives energy while it has a clear view to the top of the simulated volume (accomplished by the use of energy particles).

**Digestive cell**

The digestive cell instead gains energy by consuming detritus (dead cells), receiving their energy upon collision.

**Sting cell**

While the digestive cell consumes detritus, the sting cell "steals" energy from living cells of other organisms upon contact. Both digestive and sting cells are inspired by biological phagocytes.

**Vascular cell**

Whereas photosynthetic, digestive and sting cells all collect energy for the organisms, the fat, sensor and egg cells only consume (and/or store) energy. The vascular cell is implemented in order to transport energy between non-adjacent cells.

**Fat cell**

The fat cell works as energy storage (battery). It can store more energy than the other cell types (except eggs) and has a much higher energy inflow than outflow.

**Sensor cell**

In order for the nervous system (described in 2.3.4) to have any input from the surrounding world, the model includes a sensor cell. The sensor cell does not collect energy in any way, nor does it store it. However, it does observe the particles in its vicinity and sums their energy readings into a signal value. That signal is then used as one of the possible inputs to the nervous system.

### Buoyancy cell



Buoyancy cells have a significantly lower density than the other cell types, allowing them to float upwards.

### Egg cell



Last, but not least, is the egg cell. An egg cell, like the fat cell, stores energy. Differing from the fat cell, however, it does not have any energy outflow at all. Furthermore, the egg cell can store orders of magnitude more energy than the other cell types and it will continue to collect energy from its neighbouring cells until it has enough energy to form a new organism, in which case it immediately does.

## 2.2 The arena

This section will describe the environment created for the organisms to develop and adapt in. In order to create a non-homogeneous environment for the organisms, terrain and water were added to the model. Figure 2.2 and Figure 2.3 both shows the arena used in the final simulation described in Chapter 3, illustrating water and terrain as well as particles and boundaries respectively.

### 2.2.1 Water and air

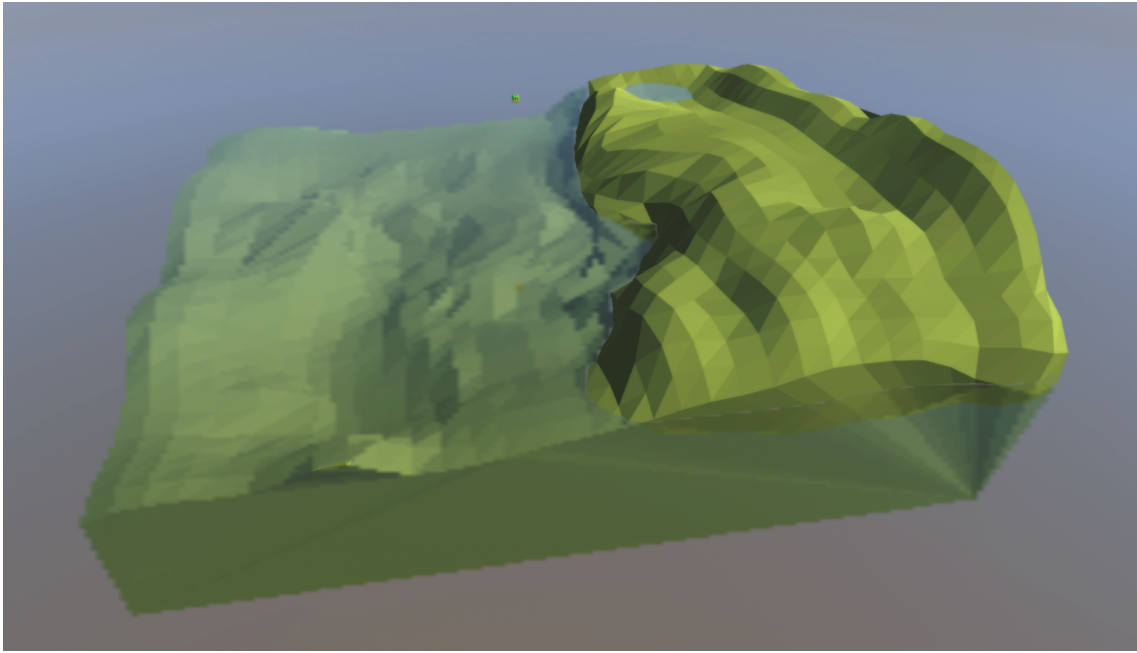
A basic implementation of water was introduced, where the density of the surrounding medium would be higher in the lower half of the simulation volume. The organisms, in normal configurations, have densities slightly higher than the water so that they can stay afloat with some movement effort. However, they need to spend significantly more energy to fly in the air. The water level is exemplified together with the terrain in figure 2.2, where the tiny dot of the initial organism is about to fall through the air and down to the bottom below the water surface.

### 2.2.2 Terrain

The terrain collision is implemented as a particle-to-surface interaction function in the Fluidix library. Each cell or detritus particle inside the 3d terrain volume, defined by linked vertex particles, will be moved back to the closest position on the terrain surface and will experience a ground repulsive force proportional to the distance it had penetrated. As can be seen in Figure 2.3, the particles remain on the outside of the volume defined by the larger blue terrain vertices.

### 2.2.3 Boundaries

The boundaries of the simulation volume, affecting cells and detritus particles, are implemented as linear soft-wall repulsion. Particles straying outside the simulation area will experience a repulsive force proportional to the trespassed distance. The boundaries can be distinguished as the thin blue edges in Figure 2.3, with the



**Figure 2.2:** Depiction of the arena terrain and water. These elements were introduced to enable organisms to settle in different environments and possibly diversify. The image is rendered at the initiation of the simulation described in Chapter 3, with the single initial organism barely visible in the centre of the arena.

particles remaining strictly on the inside (the light-green box instead encloses the terrain particle set).

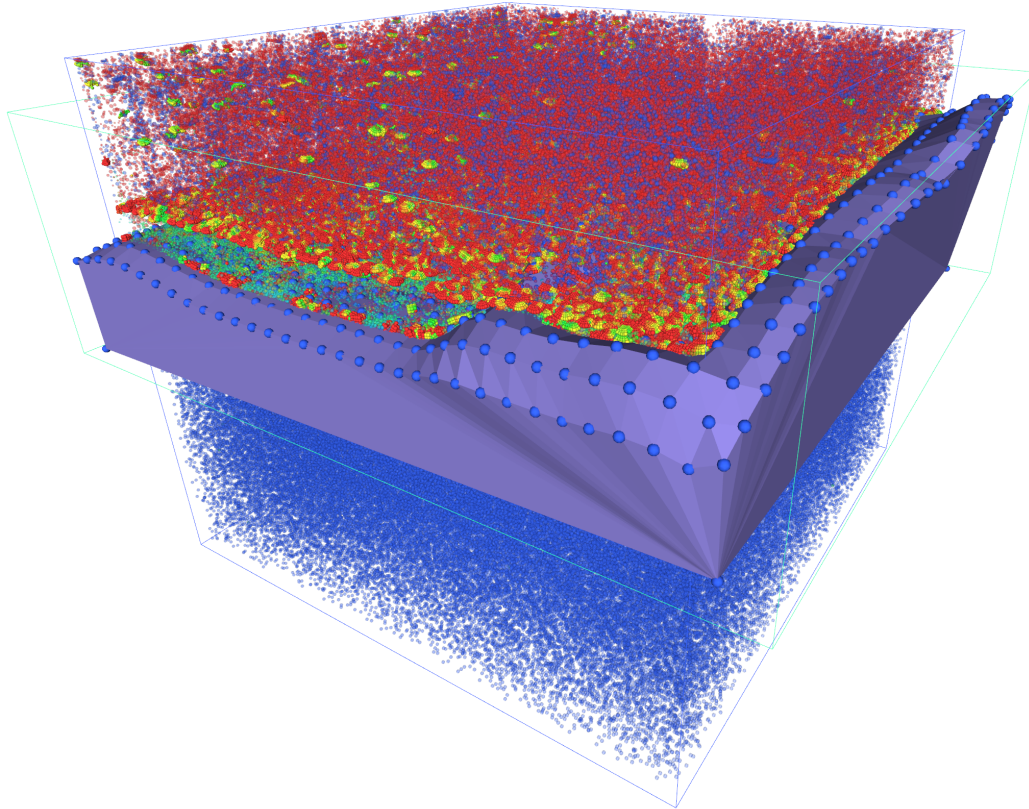
## 2.3 Definition of an organism

Similar to biological life, the information defining the organisms are encoded with a genotype-to-phenotype encoding. The choice of encoding is not trivial, as it will influence how mutations traverse the space of possible phenotypes. The approach chosen for this model is inspired by a method called CPPN-NEAT proposed by Stanley (2007). Organism phenotypes are defined using pattern-producing networks and are evolved over time through network topology mutations. This section will detail how the method was implemented for this project.

An organism is modelled to consist of three parts:

1. The genome, a genotype-to-phenotype mapping pattern-producing network detailed in subsection 2.3.3.
2. The nervous system, an artificial neural network to allow the organism to interact with its surroundings, detailed in subsection 2.3.4.
3. A list of the particles that constitute its cells. Each cell has a link to its von Neuman neighbours, as defined in figure 2.5, so that each pair of neighbour particles can exert a spring force holding the organism together.

Larger genome network cost more energy to reproduce, while larger nervous system networks cost more energy each timestep.




**Figure 2.3:** Visualisation of all elements of the particle model. The purple object is the terrain, with the larger blue particles making up its vertices. The multicoloured particles above the terrain are energy, detritus and cell particles, with their colour representing their energy level. The opaque particles are the cells, while the transparent particles above are mostly energy particles, but also detritus. A blue energy particle belongs to an occluded photosynthetic cell while a red one indicates that it has not yet collided with anything before reaching the top. Detritus particles of varying remaining energy can be seen among the organisms beneath the water surface. The thin blue bounding box determines the boundary that the particles cannot pass. The large number of blue transparent particles beneath the terrain are buffer particles.





### 2.3.1 Compositional Pattern Producing Networks

The encoding method CPPN, Compositional Pattern Producing Networks, is an abstraction of natural development proposed by Stanley (2007). It deviates from previous developmental encodings in that it requires neither local interactions nor temporal unfolding, but instead achieves similar properties by using a structure very similar to artificial neural networks.


Given coordinates as input, the network produces output used to set the properties of that specific coordinate. Each link in the network has a certain weight  $w$  and each node has a certain activation function  $f$ . The activation functions implemented are the following:

**Sine**   $f(x) = \sin(x)$

**Abs**   $f(x) = 1 - \text{abs}(x)$ , clamped within  $[-1, 1]$

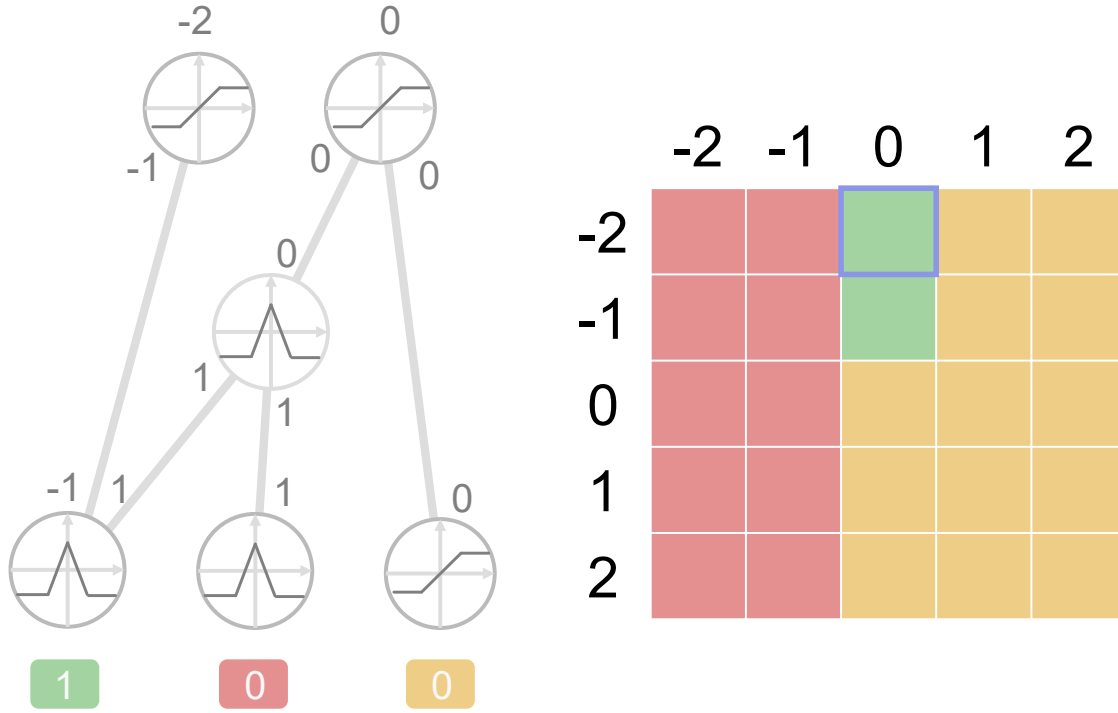
**Id**   $f(x) = x$ , clamped within  $[-1, 1]$

**Gaus**   $f(x) = e^{-\frac{x^2}{2}}$

**Mod**   $f(x) = x \bmod 1$

The *Sine* and *Mod* functions contribute with repetition while the *Abs* and *Gaus* provide symmetry.

Values are propagated through the network by calculating the next value  $v_{i,t+1}$  of each node as its activation function of the sum of the previous values times weights for all its connected nodes; that is  $v_{i,t+1} = \sum_{j \in n} f(v_{j,t} * w_{ij})$ , where  $n$  is the set of nodes connected to node  $i$ . A number of outputs are generated for any given location, resulting in a pattern. An example of this can be seen in Figure 2.4.



**Figure 2.4:** Example of a compositional pattern-producing network. Input is  $z$  and  $y$  coordinates. The actual simulation model also uses the  $z$ -coordinate as well as  $r$ , the radial distance to origo, but they are for the sake clarity not included here. For the same reason, all links have the weight of one. The pattern seen on the right is the result of evaluating the network seen on the left. The currently evaluated cell  $(-2, 0)$  is marked with a blue border and is coloured green since the output node corresponding to the green colour has the highest output value. Numbers above the nodes are their input values while the numbers above are their output.

### 2.3.2 Neuroevolution of Augmenting Topologies

NeuroEvolution of Augmenting Topologies (NEAT) is a method presented by Stanley & Miikkulainen (2002) where not only the edge weights of a network but also the network topology is evolved over time. This is done in order to avoid fixing the network structure beforehand.

In this model, mutations suggested by Stanley & Miikkulainen (2002) and Stanley (2007) are used, but crossover is disregarded since the organisms only reproduce asexually. While crossover could possibly improve the evolution of the organisms, its implementation was left outside the scope of this project. The following mutations were implemented:

**Mutate weights** With a certain mutation probability, each node is perturbed by a certain amount

**Add node** A hidden node is added at the site of a current connection, replacing the connection and creating new connections to the previously connected nodes

**Add connection** A new connection is added between two random nodes. However, there can be no connections going into input or bias nodes nor any out of output nodes.

**Remove connection** A random connection is removed. If this causes a hidden node to be without any connections, that node will be removed as well.

### 2.3.3 Genome

The genome consists of a pattern-producing network, as described in 2.3.1, as well as a discrete vector of organism diameters  $\mathbf{D}_{\text{org}} = \begin{pmatrix} d_x & d_y & d_z \end{pmatrix}$ . An organism consists of at least one cell, placed at the origin, while  $\mathbf{D}_{\text{org}}$  determines the number of cells outside of it in each direction. Thus, the size of the resulting rectangular cuboid of cells is  $\mathbf{S}_{\text{org}} = \begin{pmatrix} 2d_x + 1 & 2d_y + 1 & 2d_z + 1 \end{pmatrix}$ .

Given the cell index as input to the CPPN, the resulting output determines the cell type together with other optional cell properties, for example radius or density. Thus the network genotype defines the organism phenotype.

In the genome network implemented for the final long-term simulation (described in the results chapter 3), there are four input nodes plus one bias node with a constant value of one. Together they specify the initial distance and position of the current cell relative to the organism origin. Furthermore, there are nine output nodes: one for each cell type, plus one extra output node determining the cell radius. The node indices are assigned as follows:

**Input nodes**  $\begin{array}{|c|c|c|c|} \hline 0 & 1 & 2 & 3 \\ \hline x & y & z & r \\ \hline \end{array}$

**Output nodes**  $\begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ \hline \text{Photo} & \text{Digest} & \text{Sting} & \text{Vascular} & \text{Fat} & \text{Sense} & \text{Egg} & \text{Buoyancy} & \text{Cell radius} & & \\ \hline \end{array}$

**Bias nodes**  $\begin{array}{|c|} \hline 15 \\ \hline \text{Gives constant value of 1} \\ \hline \end{array}$

**Hidden nodes**  $\begin{array}{|c|} \hline 16 \text{ and onward} \\ \hline \end{array}$

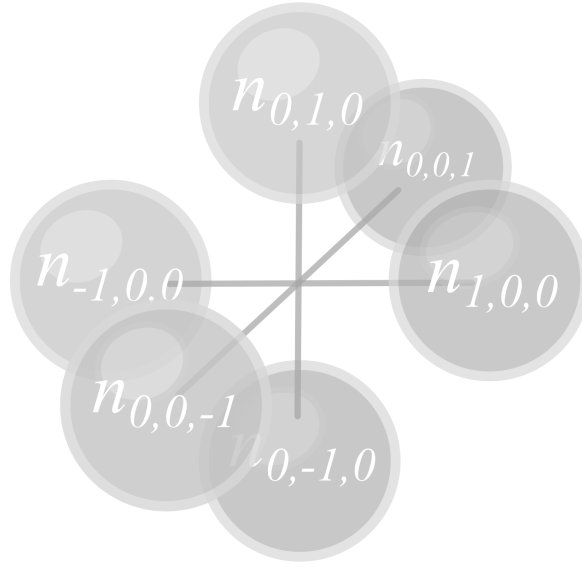
In order to determine the cell type, the output nodes 4 through 11 are inspected. The cell type corresponding to the output node with the largest value will be selected. Compare this with the example of the coloured lattice in Figure 2.4.

If two values are exactly equal, preference will be given to the lower node id. However, if no node produces a value over 0 the cell will not be created at all. Of the nodes 12, 13 and 14 only the node 12 fills a purpose. The remaining two were actually left as a mistake in the initial organism genome from earlier versions of the model, where the density and other properties of cells were also considered for genome output.

When an organism reproduces, its genome (and nervous system) will mutate as it is copied to the offspring. The CPPN will mutate according to the NEAT method described in 2.3.2, while  $\mathbf{D}_{\text{org}}$  might increase or decrease in either of its vector components.

### 2.3.4 Nervous system

The nervous system is an ANN (artificial neural network) very similar in structure to the CPPN in the genome and also evolved using the NEAT method (described in 2.3.2). The input to the nervous system consists of signals from the sensor cells as well as from a bias node sending out a constant value of 1. The output consists



**Figure 2.5:** Identifiers for the six three-dimensional von Neumann neighbours to a cell. Subscripts denote distance in x, y, and z directions between the cell and each neighbour at birth.

of a movement vector for the organism in its local coordinate system, converted to the global coordinate system as described in section 2.4.

## 2.4 Organism coordinate system

The most simple approach for generating movement from the output of the nervous system would be to directly output a force vector  $f = \begin{pmatrix} f_x & f_y & f_z \end{pmatrix}$  in the global coordinate system, where the three components of  $f$  are the three output values from the network and  $f$  is added to the current total force of the organism.

However, this approach creates an inherent knowledge of world direction, easily causing the organisms to move, for example: directly up, north or west. As such, the organism in this model instead moves according to their local coordinate system, an approach which will be explained in this section.

The organisms do not have any stored orientation since they consist of relatively independent cells. However, cells have links to their 3-dimensional von Neumann neighbours; front, back, left, right, up and down. These are denoted  $n_{x,y,z}$  as in Figure 2.5, where x, y and z are initial distances between the neighbouring cells.

For each neighbour  $n_{x,y,z}$ , we then calculate a vector  $v_{x,y,z}$  pointing in their direction as follows:

$$v_{x,y,z} = \begin{cases} pos(self) - pos(n_{x,y,z}), & \text{if } n_{x,y,z} \text{ exists} \\ (0, 0, 0), & \text{otherwise} \end{cases}$$

The function  $pos(n)$  gives the current position of the neighbour  $n$ . Cells at the edges of an organism does not have any neighbours in their "outward" directions, so a zero

vector is returned from  $v_{x,y,z}$  if the neighbour does not exist.

A property of the neighbourhood structure is that the top and bottom neighbour vectors, for example, should point in almost opposite directions. Thus, the six vectors  $v_{0,0,1}, v_{0,0,-1}, \dots, v_{-1,0,0}$  can be reduced to three by taking the sum of the linearly dependent pairs:

$$\begin{aligned} v_x &= v_{1,0,0} - v_{-1,0,0} \\ v_y &= v_{0,1,0} - v_{0,-1,0} \\ v_z &= v_{0,0,1} - v_{0,0,-1} \end{aligned}$$

These three vectors would work as base vectors defining the local coordinate system of the cell. However, to further account for missing neighbours, each is also combined with the cross product of the other two. Normalised and written as the columns of the transformation matrix  $\mathbb{M}$  we get:

$$\mathbb{M} = \left( \text{norm}(v_x + (v_y \times v_z)), \quad \text{norm}(v_y + (v_z \times v_x)), \quad \text{norm}(v_z + (v_x \times v_y)) \right)$$

Where  $\text{norm}(v) = \frac{v}{|v|}$ .

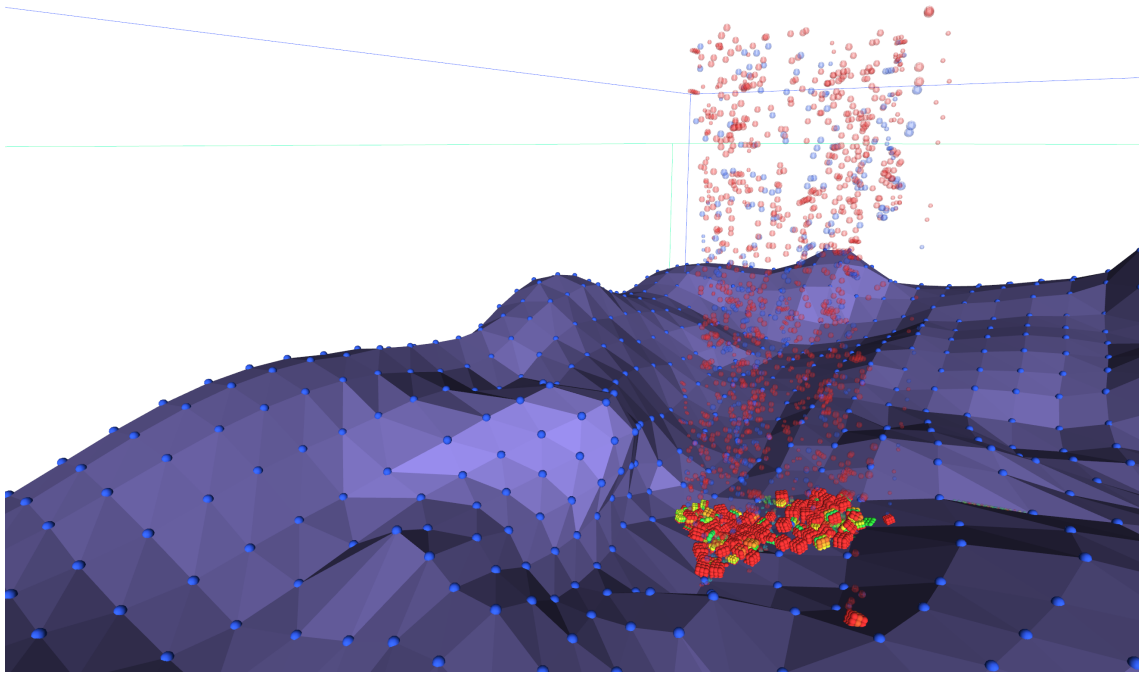
Thus from the movement force  $f_{local}$  given by the nervous system in the local coordinate system, we get the force to be applied to the cell in global coordinates as  $f_{global} = c_{move}(\mathbb{M} \cdot f_{local})$ , where  $c_{move}$  is a movement parameter.

## 2.5 The energy cycle

The final part of the model to be explained here is the energy cycle; how the energy is transferred through the system.

Each photosynthetic cell is paired with an energy particle. The energy particle is initialised just above the cell and travels straight upwards with a speed of one particle diameter per timestep. Each timestep the cell gains an amount of energy proportional to its cross-section area, provided that the energy particle reaches the top of the simulated volume without any collisions. If there is a collision, the cell will stop gaining energy until the view is once again clear. When the particle reaches the top of the volume it is returned to a position just above the cell. Figure 2.6 illustrates this with an example of photosynthetic cells and their respective energy particles.

Digestive cells and sting cells are the other two cell types that harvest energy to an organism, as described in subsection 2.1.3. From them and the photosynthetic cells, the energy is dispersed within the organism. Different cell types disperse energy in different amounts. For example, egg cells only absorb energy without ever giving anything back, while the energy-harvesting cells part with most of their energy. See Table 2.1 for an example of how the energy dispersion was configured in the final run described in the results.



**Figure 2.6:** Energy particles rising from an early population of photosynthetic organisms. Each energy particle belongs to a photosynthetic cell. If the energy particle reaches the top of the arena without colliding with any cells, the photosynthetic cell it belongs to will continue gaining energy every frame. If the particle does collide, the cell will not get any energy until its particle once again can complete the distance without collisions. In this illustration, energy particles that have collided are coloured blue, while the rest are red. The cells are opaque and coloured according to the amount of energy they contain, with warm colours indicating higher energy density. A video animating the process can be found at <https://youtu.be/jsNgbxvU5Js>

Cell type	Energy in	Energy out	Maximum energy
Photosynthetic	0.01	0.5	10
Digestive	0.01	0.5	10
Sting	0.01	0.5	10
Vascular	1	0.2	3
Fat	1	0.01	50
Sensor	1	0	5
Buoyancy	1	0	5
Egg	1	0	1000

**Table 2.1:** Example of a cell types energy configuration. *Energy in* is the portion of the energy offered by a neighbour that is absorbed, *Energy out* is the portion of the cell's surplus energy that is offered to a neighbour.





# 3

## Results

These results originate from a simulation run for 2 weeks, investigating the features emerging after long-term evolution in a large system. Many previous simulations had been performed with smaller world sizes, but while they achieved a larger number of total timesteps they did not contain the diversity and the trophic levels that resulted from this larger simulation. While it still remains a subject of further investigation, it at least seems as if a certain amount of organisms are needed on a trophic level before organisms on the next can be sustained.

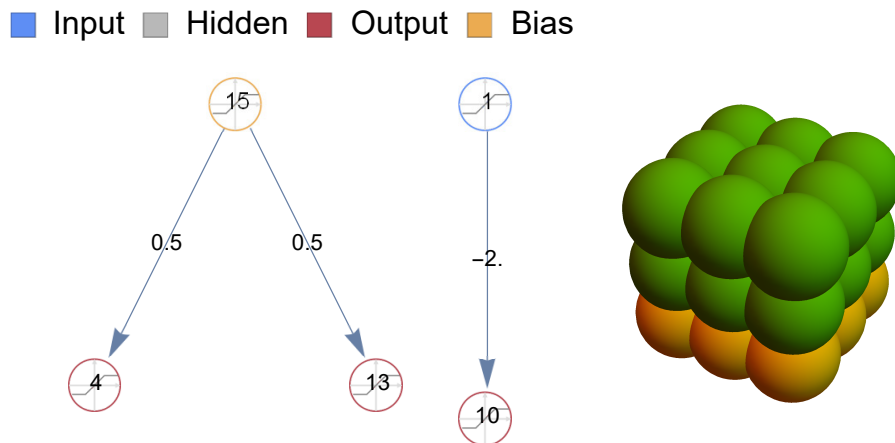
Parameter	Value
World size	$350 \times 150 \times 350$
Timestep length	0.01s
Timesteps simulated	1 160 000
Simulated time	3h, 46m, 20s
Simulation runtime	14 days, 2h, 46m

**Table 3.1:** Statistics about the performed large timescale simulation, detailing its extent and magnitude

### 3.1 Initial organism

The initial organism was designed such that it would have a very simple genome and a non-existent nervous system, but so that it would still be able to survive long enough to reproduce. As can be seen in Figure 3.1, the inputs to the genome network come from the input node 1,  $n_1$ , and the bias node 15,  $n_{15}$ . The interpretation of all node indices can be found in section 2.3.3.

While the bias node gives a constant signal of 1, the input  $n_1$  follows the value of the y-coordinate. The output node  $n_4$  corresponds to the photosynthetic cell type while  $n_{10}$  corresponds to egg cells and  $n_{13}$  does nothing. The connection to  $n_{13}$ , as well as the node itself, is a residue left by mistake from earlier versions of a model where more properties were determined by the genome.



**Figure 3.1:** Initial organism genome and visualisation. No hidden nodes exist in the network. Since the nervous system had no initial connections, the organism was inanimate and hence no nervous system network is displayed here.

## 3.2 Lineages and trophic levels

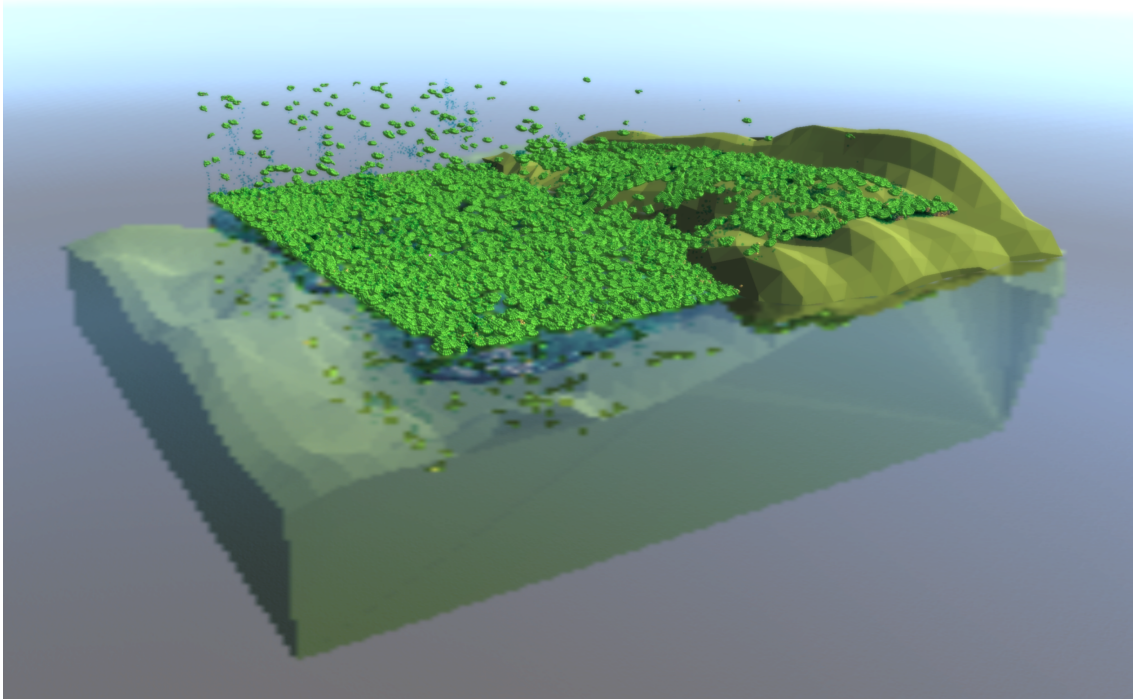
Since there is no sexual reproduction within a model the notion of species cannot really be applied, however, it is possible to follow and compare differently adapted ancestral lines. The clearly dominant type of organism belongs to the plant-like line exemplified in Figure 3.6, but the results also include organisms of higher trophic levels resembling the adaptations of decomposers and plant-eating animals (herein called predators). The order-of-magnitude differences between the amount of photo, digest and sting cells seen in figure 3.3 indicate how many plant organisms are needed to support the decomposers and predators.

### 3.2.1 Plants

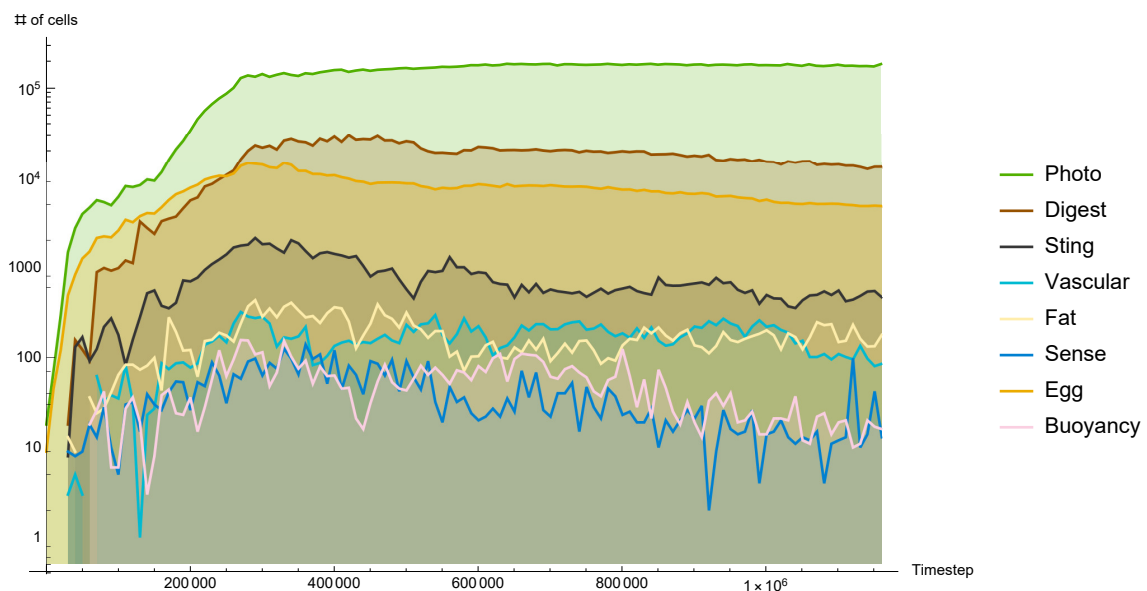
Plants are organisms that gain energy through their photosynthetic cells. Looking at figure 3.3 it is clear that a large majority of the cells (over 90 percent) are photosynthetic, something that is also supported by the green cover of vegetation seen in figure 3.2. Since the photosynthetic surface-dwellers are a large majority of the total population, the conclusions drawn here will mostly be taken from an analysis of the whole population average. This subsection will cover a number of the adaptations to be found in the photosynthetic organisms.

#### 3.2.1.1 Swimming upwards

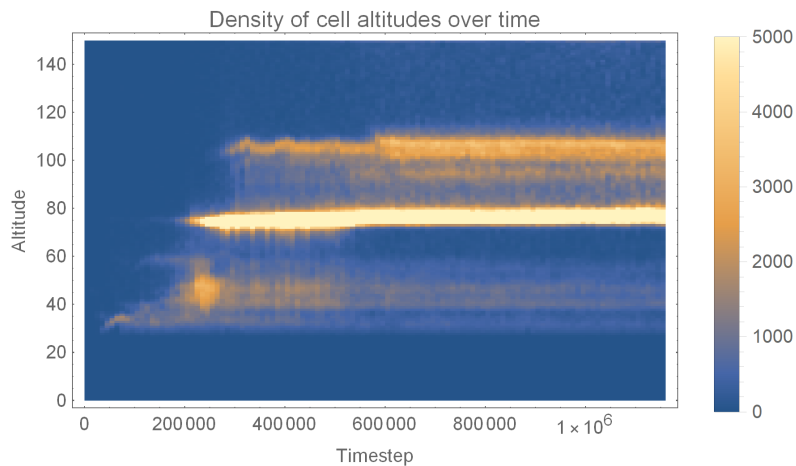
One of the first mutations that show signs of adaption is the upwards movement. There were mutations involving buoyancy cells that allowed very early organisms to float to the surface, but since those cells either replaced the photosynthetic cells or the egg cells those organisms were short-lived. After some sideways-moving organisms had started to spread across the water and, later on, some ventured a little higher, the organisms finally started inhabiting the surfaces at around  $t=130\,000$ . At  $t=200\,000$ , a population dense with organisms started out in the small lagoon at



**Figure 3.2:** Simulation arena at the end of the simulation, at  $t=1\,160\,000$ . While the organisms have spread all over the arena, the large majority is found on the surface level. A video of the full simulation can be viewed at <https://youtu.be/yMY1dIjbFY8>.



**Figure 3.3:** The number of cells belonging to each respective cell type over time. Note that the y-axis is logarithmic and that the photosynthetic cells thus represents a large majority. At the final timestep  $t=1\,160\,000$ , the photosynthetic cells constitute 90.83% of the cells, while the digest and sting cells are at 6.45% and 0.23% respectively. Vascular, fat, sense and buoyancy cells, having no part in energy collection or reproduction are very uncommon.



**Figure 3.4:** Cell altitude density over time. The more cells that occupy a given altitude (y-coordinate) at a given timestep, the more light the point appears in the chart. The plot range is clipped at a cell density of 5000, but the lightest area reaches close to 40 000 at around  $t=600\,000$ . Due to gravity, the initial organism, having spawned in the exact centre of the arena, quickly falls to the bottom, below the water surface. This is where the population starts out, at an altitude of about 30, which can be seen in the bottom-left corner of the chart. Quite quickly, however, the organisms evolve movement vectors upwards and the large majority end up at the altitude of 75 by the water surface, after about  $t=200\,000$ . The fact that the organisms are spread out and focused on different altitudes shows signs of niche adaption.

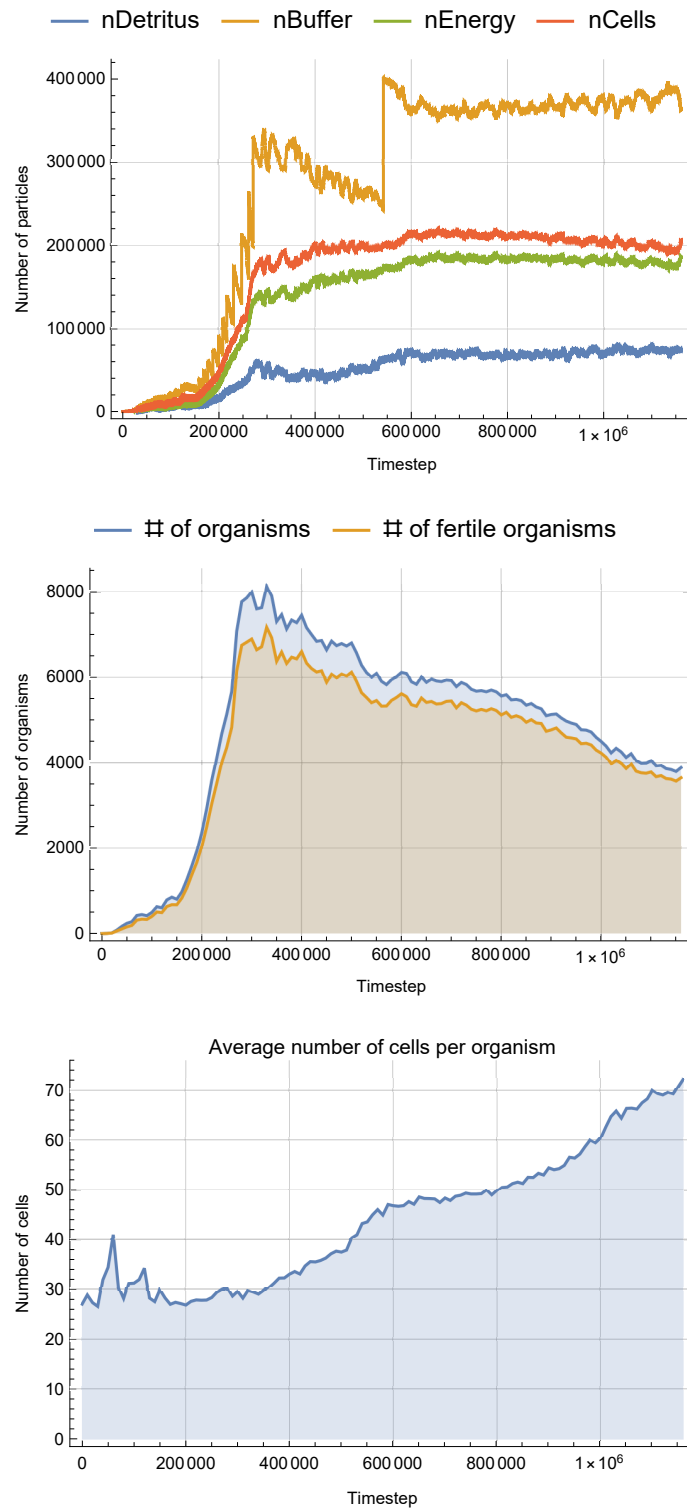
the corner of the map and soon the entire surface of the water was covered, as can be seen in Figure 3.4.

#### 3.2.1.2 Larger organisms

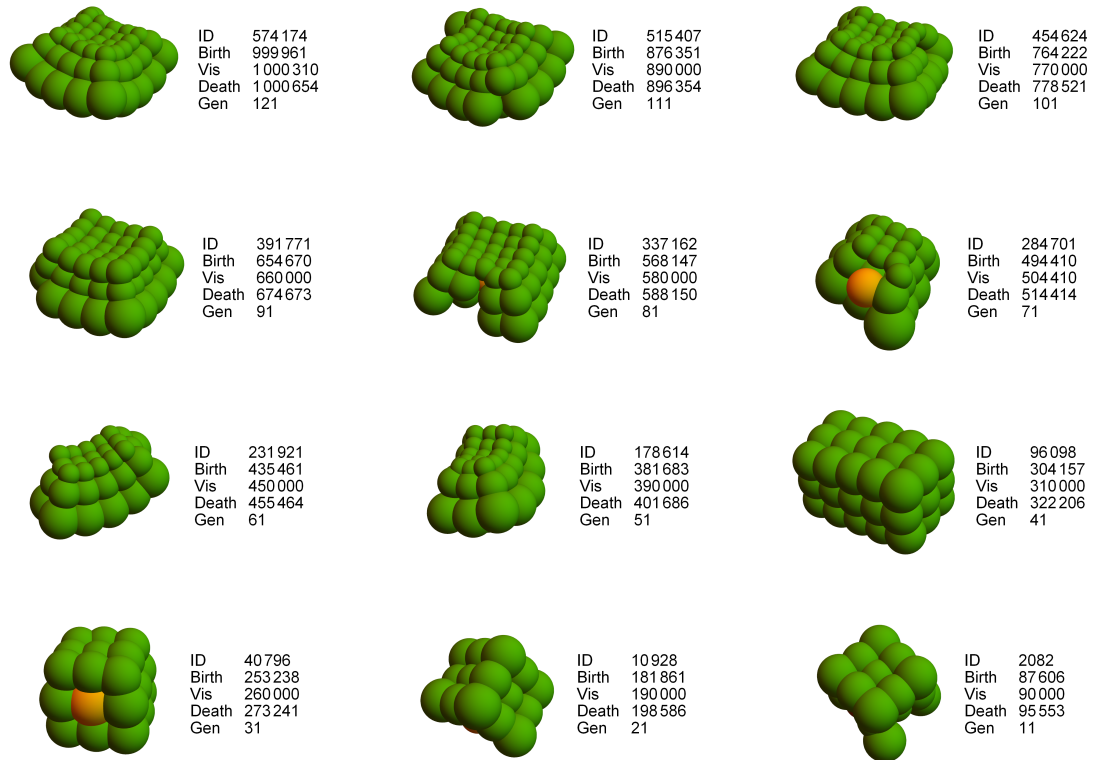
Another trend for the organisms is that they increase in size. Looking at the charts in Figure 3.5 it is clear that the population increased exponentially until about  $t=300\,000$  after which the number of cells remained approximately constant while the number of organisms steadily decreased. The bottom chart shows even clearer how the size of the organisms then started to increase. The time  $t=300\,000$  is also right after the water surface got crowded with photosynthetic organisms. A probable explanation for the increase in size is therefore that it is an adaption resulting from the competition for the water surface area; a larger more spread-out organism can harvest more energy and reproduce faster. This is supported by the observation of the surface organisms getting increasingly spread-out over the generations.

#### 3.2.1.3 Pyramid-shaped organisms

Apart from increasing in size, the water-surface organisms also became more "pyramid-shaped" over time, as seen in Figure 3.6. This enables the lower layers of the organism to extract photosynthetic energy as well, instead of relying on the energy from the top layer like their early ancestors.



**Figure 3.5:** Simulation statistics over time. As can be seen in the top-left chart, the number of particles stabilised after around  $t=300\,000$ . However, the number of organisms (top-right chart) henceforth decreased steadily. This can be explained by the bottom chart, which clearly shows that the size of the organisms has, on average, increased over time. The jumps in the number of buffer particles are only the model ensuring that there is a sufficient amount of buffer particles available.



**Figure 3.6:** Example of morphology changing over time. The phenotype for every tenth ancestor of the organism with index 574174, which was one of the organisms living on the water surface at  $t=1\,000\,000$ . It is located in the top-left corner, the oldest ancestor is in the bottom-right, as indicated by the generation labels. A trend of this ancestral line is that the organisms have increased in size, as well as turning more "pyramid-shaped". The radii of the top cells are, in the later organisms, significantly smaller than the bottom cells.

It is interesting to note that this seems to be a direct artefact from how the model was implemented, with the energy-particles checking for occluding particles above as described in 2.5. In previous implementations, the model was instead bombarded by photons from above that could bounce around until they run out of energy, but with the current implementation, the number of cells seen from directly above is even more important. A smaller cell radius decreases the energy gained per timestep, since it depends on the cross-section area of the cell, but also increases the probability of the energy particle reaching the top without any collisions.

Another possible reason for the plants to be pyramid-shaped might be that it helps them stay on the surface better. Cells have a constant density, so a larger cell will have a larger mass and thus the organisms will not risk tipping over. Having a pointier tip could also help them penetrate to the surface better after being born, as either they or their parent often are pushed below the surface at birth.

#### **3.2.1.4 Higher ratio of fertile organisms**

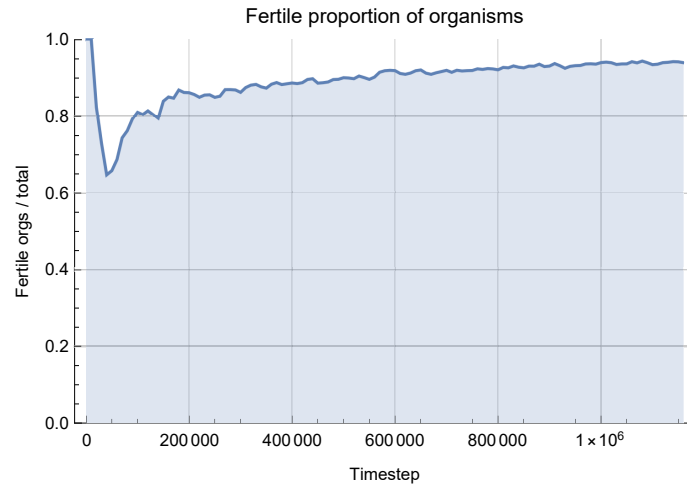
Observing the top-right chart of Figure 3.5 it looks as if the proportion of organisms that are fertile (meaning that they have at least one egg cell) are increasing relative to the total number of organisms over time. This is even more evident in Figure 3.7 where, after an initial drop from unity caused by the single initial organism and its immediate offspring, the proportion steadily approaches a level of almost 0.95.

A certain proportion of the organisms could be expected to not be fertile since there is always a certain non-zero probability for each offspring to be born without egg cells. Since non-fertile organisms per definition cannot produce offspring it is clear to see that their proportion should not increase. However, the fact that the fertile proportion is steadily increasing, instead of remaining constant, seem to show an emerging resilience in the genome against birthing non-fertile offspring.

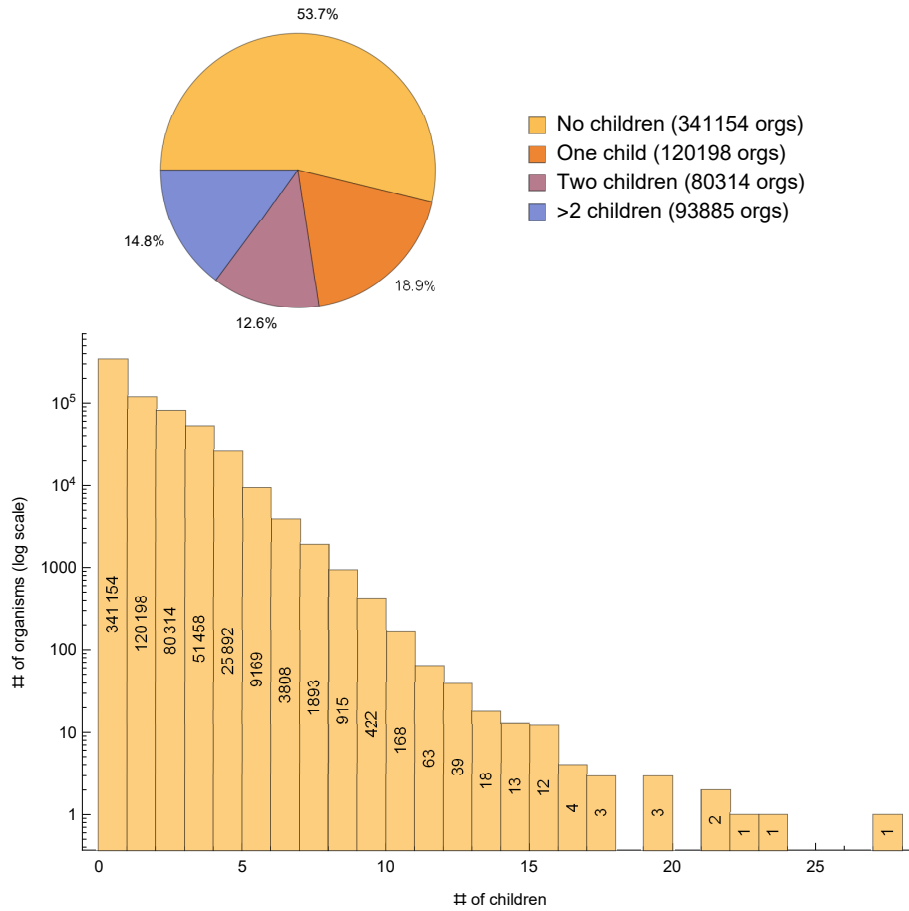
A majority of all the organisms that ever existed in the simulation never did reproduce, as can be seen in Figure 3.8. Some of those were not fertile while most simply died before they had gathered enough energy in their egg cells. Out of those that did reproduce, most had between one and two children, but close to 15 percent of the organisms had more than two children, with one having as many as 27.

#### **3.2.1.5 Fewer eggs per organism**

A clear adaption is the fact that the average number of egg cells per organism has dropped. The initial organism had a total of nine egg cells, resulting merely from the fact that it allowed the genome to stay simple while still allowing the organisms to reproduce. That many egg cells were however redundant and the population eventually evolved to settle on the lower value of two eggs per organism, as can be clearly seen in Figure 3.9.

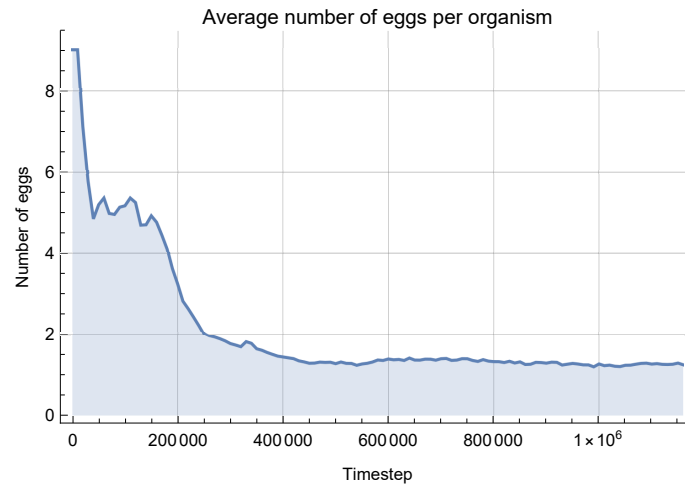


**Figure 3.7:** The number of fertile organisms divided by the total number of organisms over time.



**Figure 3.8:** Distribution of the number of children per organism, given all organisms throughout the whole simulation. A majority of the organisms never reproduced but multiple children are not uncommon among those that did, as can be seen in the histogram on the bottom. Note that the y-axis is logarithmic and that the number of children as such follows a power-law distribution.





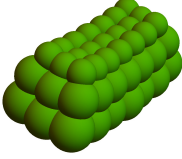
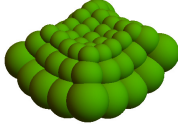
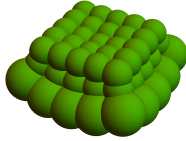
**Figure 3.9:** The average number of egg cells per organism over time. At about the same time as the total number of cells in the simulation stabilised ( $t=400\,000$ ), each organism settled on having about one egg cell each.

### 3.2.1.6 Three plant organisms

Table 3.2 shows a selection of three different plants and their properties. Noting the position of *plantA*, its altitude (y-coordinate) of just above 100 places it a bit above the water surface of 75. Studying the x and z coordinates it can be determined that it is located upon the terrain, upon dry land. Comparing this with *plantB*, which position places it directly at the water surface, it is clear that while both plants have adapted in size and shape for efficient photosynthesis, their separate environments have caused them to evolve in parallel.

As seen in Figure 3.15, the most recent common ancestor of *plantA* and *plantB* is at generation 61, while *plantB* and *plantC* share a more recent ancestor at generation 96. The closer relationship between *plantB* and *plantC* can also be seen in their more similar morphology. However, *plantC* has a much higher y-coordinate, placing it floating in the air far above the surface and very close to the top of the arena at 150. While this height altitude is beneficial in that it ensures a clear view for photosynthesis, it also costs more energy to keep afloat in the low-density air.

The third column of the table, *plantC*, shows a plant organism that resides in the air, far above the surface and close to the top of the arena.

	plantA	plantB	plantC
Organism ID	634078	574174	628131
Generation	148	121	136
Timestep of birth	1157422	999961	1140031
Timestep of death	(still alive)	1000654	1160034
Visualisation (at timestep)			
	1150000	1140000	1130000
Position $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$	$\begin{pmatrix} 172.198 \\ 101.756 \\ 74.0961 \end{pmatrix}$	$\begin{pmatrix} 108.528 \\ 74.7777 \\ 341.747 \end{pmatrix}$	$\begin{pmatrix} 39.015 \\ 149.276 \\ 254.073 \end{pmatrix}$

**Table 3.2:** Example of three different plant organisms, each using photosynthesis but occupying different parts of the arena and having different adaptations.

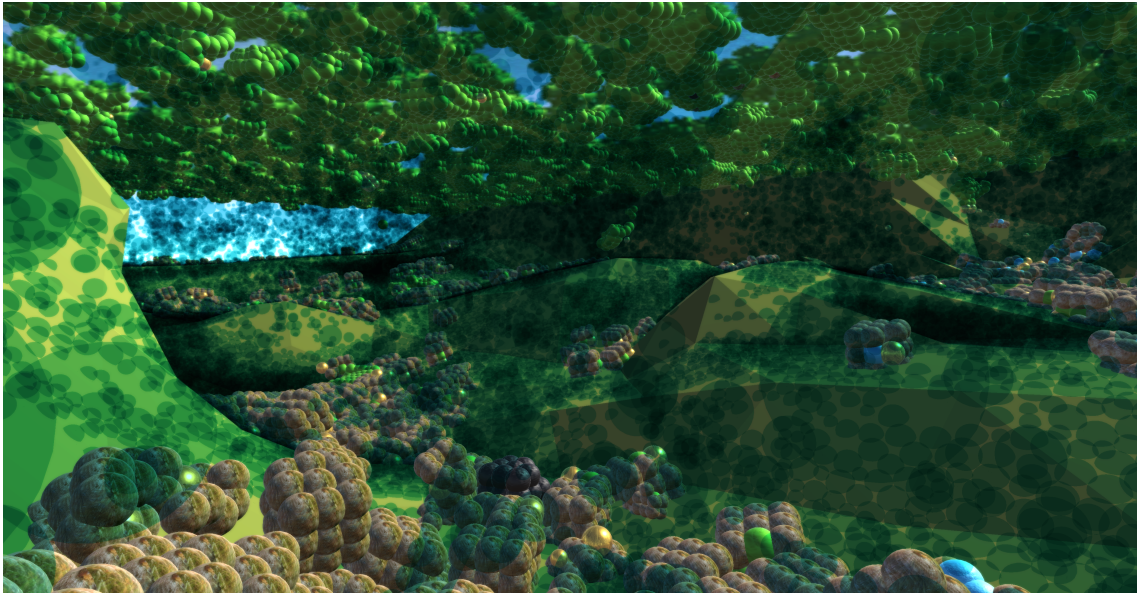
### 3.2.2 Decomposers

With the surface plants covering the sunlight, gaining energy through photosynthetic cells is practically impossible below the surface. Decomposers solve this issue by instead gaining energy through their digestive cells; eating detritus particles as they fall down from dying organisms above. Differing from the plants, the decomposers have maintained the dimensions of the initial organism. Being significantly fewer, there is not the same need to compete for space.

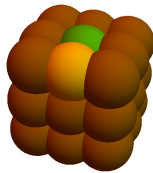
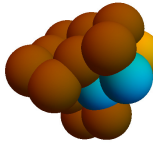
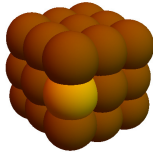
The decomposers are mostly made up of digestive cells and a single egg cell, although some decomposing lineages contain other cell types as well. Some, like *decompA* in Table 3.3, have a single (or a few) photosynthetic cells, perhaps useful when they get a clear view of the sky, but maybe just a relic from previous generations. There is also a lineage that has a single (or multiple) vascular cell adjacent to their egg cell. An example of this is *decompB* in Table 3.3. This might be an adaption to better transport the energy from digestive cells to the egg cell, like the vascular cell was intended.

#### 3.2.2.1 Three decomposer organisms

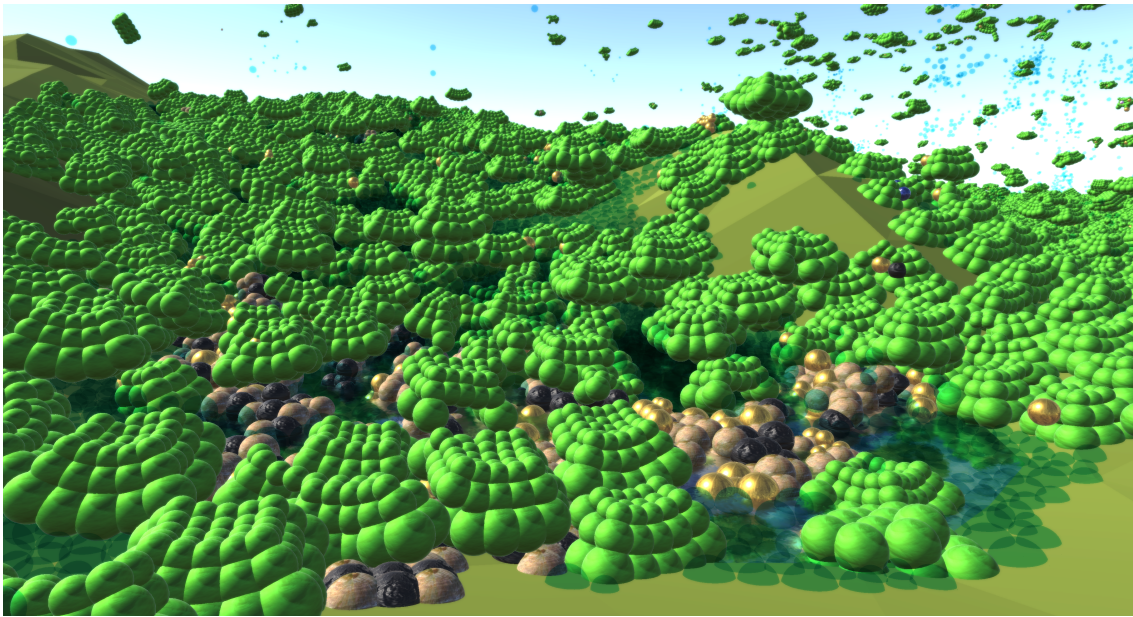
As can be seen in Table 3.3, three decomposer organisms have been selected as examples of their lineages. As discussed above, they contain mostly digestive cells, a single egg cell and, in the first two organisms, cells of other types. The positions of the decomposer organisms are closer together than those of the plant organisms (as can be seen in Figure 3.14), explained by the fact that decomposers occupy a single niche at the bottom below the water surface.



**Figure 3.10:** Rendered view of decomposers living off eating the dead cells falling down from the surface-level plants. Note the occasional light-blue vascular cell in some of the decomposers. A full video of the whole simulation as seen from beneath the water surface can be found at <https://youtu.be/evZK16QmrBk>.

	decompA	decompB	decompC
Organism ID	634597	634413	634596
Generation	220	234	206
Timestep of birth	1158806	1158364	1158806
Timestep of death	(still alive)	(still alive)	(still alive)
Visualisation (at timestep)	 1160000	 1160000	 1160000
Position $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$	$\begin{pmatrix} 102.983 \\ 45.5508 \\ 198.9 \end{pmatrix}$	$\begin{pmatrix} 107.391 \\ 52.922 \\ 143.383 \end{pmatrix}$	$\begin{pmatrix} 128.907 \\ 45.5792 \\ 184.384 \end{pmatrix}$

**Table 3.3:** Example of three different decomposer organisms, each gaining energy from eating detritus particles



**Figure 3.11:** Rendered frame of the predator lake at  $t=1\,008\,020$ . Because of the topology of the terrain, the small lake at the corner of the arena was very beneficial for predators. Without having to move, the predators could use their black sting cells to take energy from the organisms sliding down the slope into the lake. A video of the same scene can be found at <https://youtu.be/onh0dPyuJiE>.

#### 3.2.3 Predators

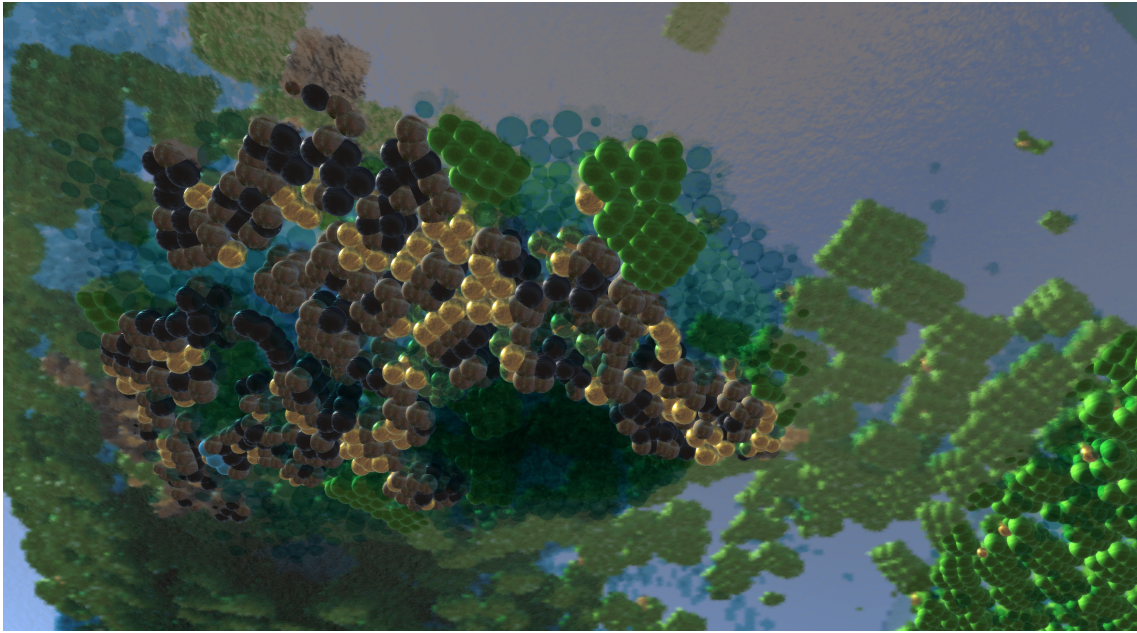
The adapted line called predators is characterised by their use of sting cells to steal energy from the living cells of other organisms. They are however similar to decomposers in that they also have digestive cells, utilised when their prey dies or when no living organisms can be found.

Some predators could be found among the decomposers at the bottom at around  $t=500\,000$ , but their ability to reproduce was not high enough to survive as a lineage. A better source of prey was however found in the small lake upon the terrain. As can be seen in figure 3.11, both plants and detritus slide down the slopes and into the lake filled with the lurking predators. A view of the predators from below as seen in figure 3.12 might give a better idea of their morphology.

Compared to plants and decomposers, which as a rule tend to have one egg cell per organism, the predators have a lot more. This adaption can be explained by the fact that predators gain their energy at non-regular intervals; sting cells only gain energy while in contact with living cells from other organisms and there are not always other organisms present. When they do gain energy, however, it could very well be more than what fits in a single egg cell and so multiple egg cells ensure that no energy goes to waste. This also explains the fact that Figure 3.9 approaches a value slightly above, but not equal to, one.

Figure 3.13 shows an example of a predator lineage, where the evolution from plants to predatory organisms can be seen.





**Figure 3.12:** Rendered frame of the predator lake at  $t=1008020$ . This is the same frame as in figure 3.11, but viewed from below, through the terrain. Here the predator organisms at the lake surface can be more easily seen. A video of the same scene can be found at <https://youtu.be/Wm2TUDi2fPs>.

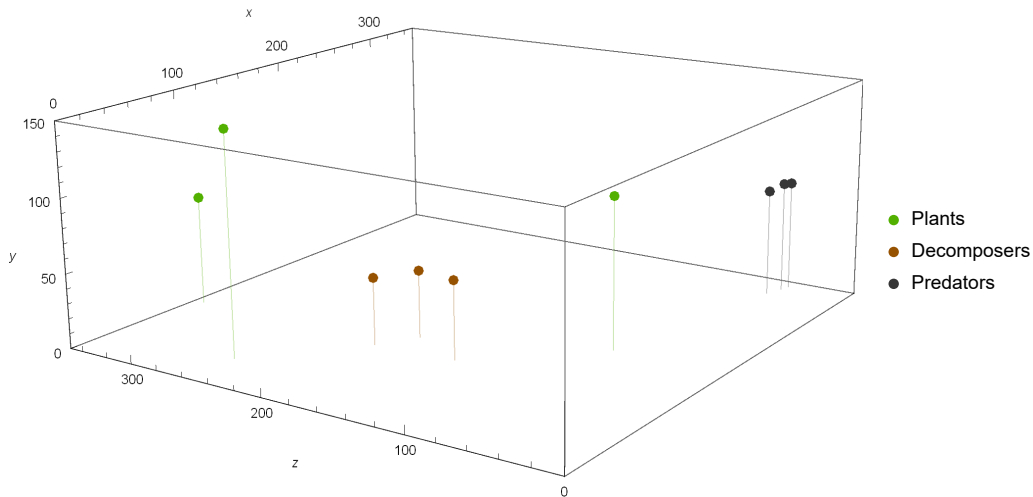
### 3.2.3.1 Three predator organisms

Compared to the three example organisms from the plant and decomposer lineages, the predator organisms are even more geographically constricted, as can be seen in Figure 3.14. Looking at Table 3.4, the predator organisms are also quite similar in their morphology; *predA* and *predC* are almost identical and while *predB* has a larger ratio of sting cells, its original form of  $3 \times 3 \times 3$  cells has changed since its conception. The similarity between the organisms is not strange given their relatively recent common ancestor just a few generations earlier, as seen in Figure 3.15.

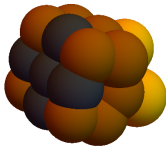
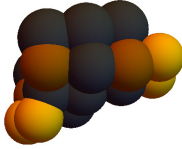
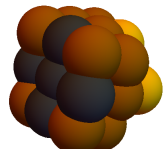
### 3. Results



**Figure 3.13:** Example of a predator lineage. The phenotype for every tenth ancestor of the organism with index 630807, which was one of the organisms living in the predator bay at the end of the simulation, at  $t=1\,160\,000$ . Note that the organism portraits are taken at certain exported frames and that, as such, some cannot be found while others have lost cells since their birth. The predators originate, as all organisms do, from the initial organism at generation 0.



**Figure 3.14:** Positions of the nine example organisms described in subsections 3.2.1.6, 3.2.2.1 and 3.2.3.1

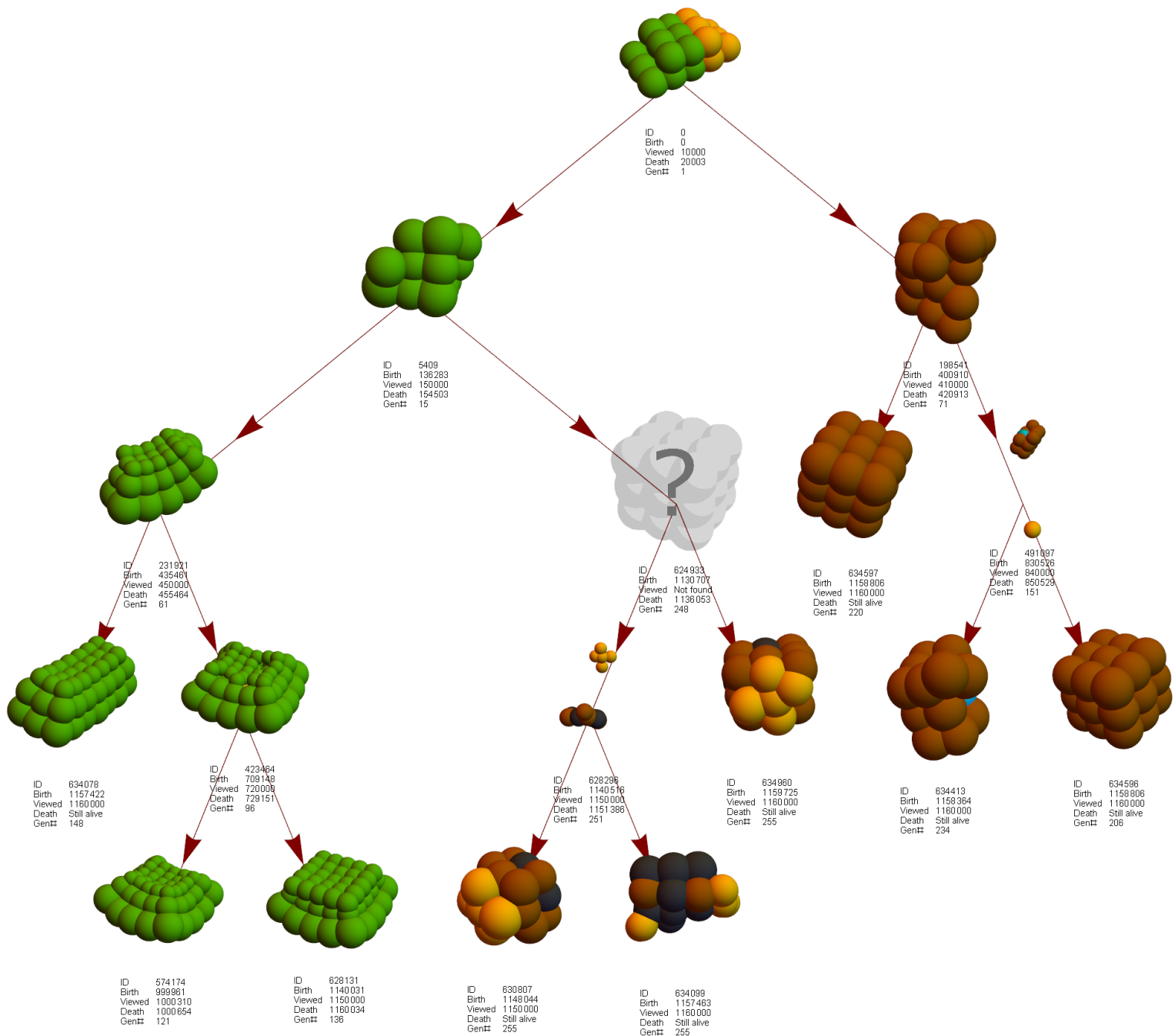
	predA	predB	predC
Organism ID	630807	634099	634960
Generation	255	255	255
Timestep of birth	1148044	1157463	1159725
Timestep of death	(still alive)	(still alive)	(still alive)
Visualisation (at timestep)			
Position $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$	$\begin{pmatrix} 318.756 \\ 73.1252 \\ 46.4154 \end{pmatrix}$	$\begin{pmatrix} 330.843 \\ 76.0001 \\ 42.1454 \end{pmatrix}$	$\begin{pmatrix} 338.39 \\ 74.9417 \\ 40.7016 \end{pmatrix}$

**Table 3.4:** Example of three different predator organisms, each gaining energy by stealing it from cells of other living organisms

### 3.2.4 Inter-lineage relations

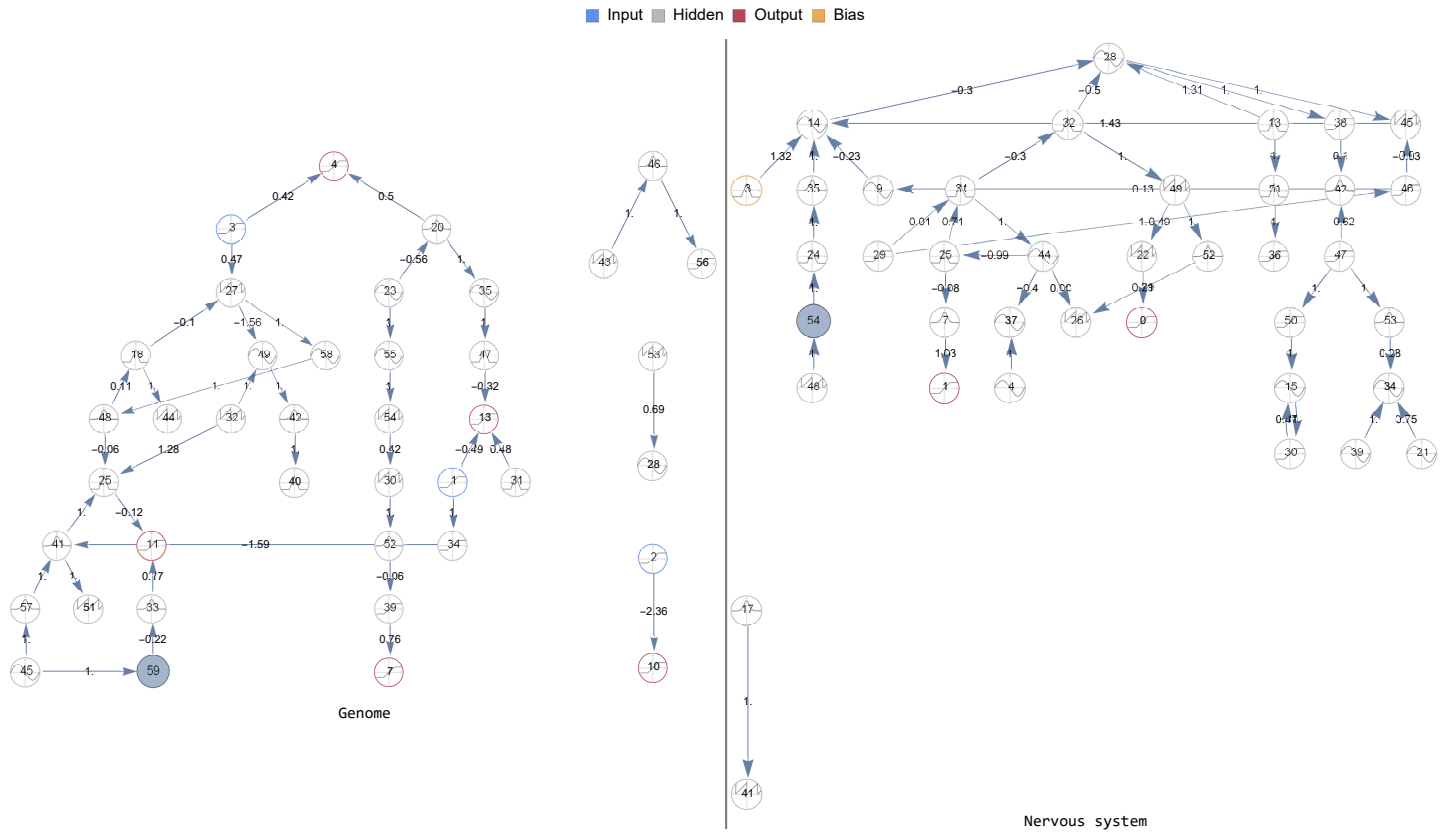
From each lineage type, plants, decomposers and predators, three example organisms were chosen and investigated for their common ancestors. The resulting family tree is illustrated in figure 3.15. It can be seen already from the phenotype that similar organisms tend to be more closely related. The plants and the predators actually share a more recent common ancestor compared to the decomposers, however since that plant is born very early, at generation 15, this should not hold much significance and can be mostly regarded as a result of the selected organisms.

### 3. Results



**Figure 3.15:** Each of the nine example organisms described in subsections 3.2.1.6, 3.2.2.1 and 3.2.3.1 represent a lineage among either plants, decomposers or predators. This diagram represents them as the leaf nodes of their family tree, with their mutual most common ancestors included all the way to the initial organism. Note that the organism visualisations are taken from saved timesteps and that the organisms are at different stages of decay. The ancestor 624933 was not found on any saved timestep on record





**Figure 3.16:** Genome (left) and nervous system (right) of the surface-dwelling plant with id 574174. The genome node #59 and the nervous system node #59 are both missing hidden nodes, where the nodes have been removed while the connections remain. Thus they have no activation function and no value. The reason for the connections remaining is an unknown bug.



# 4

## Discussion

The results were more numerous than anticipated and it is fascinating to be able to list so many different observed adaptations. Earlier simulations performed with the model still had adaptations within plant organisms where they moved upwards in order to gain photosynthetic energy and changed in size to gain more area. There were, however, no population of decomposers that managed to stay alive for any longer duration. Instead, they fluctuated out of existence due to their small numbers. Even more notably, sting cells were nothing but the result of unsuccessful mutations; the smaller systems had no lineage of predators.

The higher trophic levels in this final run are likely due to the increased scale of the model. With a larger top area, the number of photosynthetic cells that can gain energy also increases, which means more energy input to the system. It would seem as if there is a need for a certain number of plant organisms to sustain a population of decomposers or predators, something that can be likened with the concept of ecological pyramids within biology: The (biomass) quantity of plankton, for example, is much larger than the quantity of fish. These relations in this model are however not certain and should be investigated in future work.

### 4.1 On choosing a suitable environment configuration

The results acquired seem to indicate that the environment configuration chosen for the final result was quite good, but it is hard to know how the results would have compared for other configurations without a proper investigation of the parameter space. As for now, the parameters were chosen by a process of trial and error. As long as the environment is hospitable enough for the organisms to survive, the emerging population will evolve and adapt to thrive in that environment. Even a very hostile environment might have life emerging and thriving, given enough time to adapt. You could ensure this by having random initial organisms inserted in a simulation as others die out; eventually, some organism should be able to survive with the selected environment configuration. However, in order to avoid spending a large amount of initial time, a form of life and an environment somewhat initially compatible with each other were chosen, with the initial organism as described in Section 3.1.

Still, while a population usually survived with this approach, it took some time before signs of adaption and the more interesting results began to appear in this simulation. Because this delay of results (it often took more than a day to decide if

a given run was viable), together with the difficult search in the huge space of environmental parameters, a long time had to be spent adjusting parameters. This leads to a stopping-rule dilemma of whether to abort a simulation or to leave it running in hope for better results. If something seems to be wrong with the simulation, is it more strategic to abort it and try another, possibly better environment setup? Or is it better to continue to run and instead hope that the extra time already spent running will be worth more for the end result? Also, since larger-scale simulations seem to give better results, how large should the simulations be while evaluating other parameters?

### 4.2 Modelling level

An important thing to consider when modelling any system is which level to formulate the model on. A lower-level model would increase the open-endedness of the system, allowing for a larger design space. However, it would also require more calculations to simulate and with a modelling-level too low there will not be enough time for interesting results to appear.

Originally, the model was run solely on a cell-level; with cells replicating individually. But such a setup proved to be too low-level for any interesting results to appear within a reasonable timeframe. The organisms were still intended to be multicellular, but that would require an emergent change of individuality, something the model did not allow for. Instead, the egg cells were introduced and the organism reproduction was centralised to the organism, not handled by each cell.

A model including mutable cell types, where the cells themselves consists of different components, or where the cell types were continuous instead of discrete, could allow for new adaptations unconstrained by the pre-set cell types used here. However, it would make visualisation harder and could make the dynamics even trickier to balance.

Not all of the cell types were used. Vascular cells were used only to a small extent, possibly as an adaption, but their numbers were declining during the last 150 000 time steps as per figure 3.3. Fat cells were about as numerous but probably did not have any advantages over egg cells since reproduction was the primary need for energy. Perhaps both vascular and fat cells would still prove useful in larger and more complex phenotypes. Sensor cells were almost never used, this might be due to the way sensors were handled in the model, perhaps more sophisticated senses would work better, or the signal input itself might somehow be incorrectly handled; the nervous systems instead made use of the bias node as input.

The buoyancy cells were used as little as the sensors, while many plant organisms evolved the strategy of swimming upwards to gain more sunlight, none were successful in using buoyancy cells instead to save energy. This was likely because a buoyant cell tends to orient the organism so that it is at its top, and thus getting in the way of photosynthesis. Some phenotype configurations should be able to work with buoyancy cells if they were spaced symmetrically, however, it would seem as if the energy saved from not having to swim upward is not enough to overcome the cost of sacrificing a possible position for a photosynthetic cell.

### 4.3 Future work

To conclude the discussion part, there are some areas of future work that might allow us to explore and develop the potential of the model:

- Performing larger-scale simulations for longer durations. The main interest of future work would be to perform even larger and longer simulations. Because of the significant increase in diversity in the final simulation compared to the smaller previous ones, it would be of great interest to investigate how this trend continues forward. This is needed also for developing the model further as we need to see where the bottlenecks are.
- Improving the model visualisation. The visualisation would need to be improved in order to handle a larger amount of particles while still being interactive.
- Investigating the effects of different environments, such as with and without water, different terrain or day-and-night cycles.
- Developing a better nervous-system model. Since the nervous system was mainly used for moving upwards, some changes to the model might improve its usability. Furthermore, other input and output nodes could be included, while a higher cost of large networks could be applied to avoid unnecessarily large nervous systems.
- Introducing sexual reproduction. Since full NEAT implementations allow for crossover between genomes, organisms could be allowed to reproduce sexually, for example by colliding egg cells. This would allow for a definition of species as well as possibly decrease adaption time.
- Investigate other genetic encodings. A genetic model that is more congruous with biological genetics might be used for gaining insights into the evolution of adaptive organisation of gene regulatory networks (GRN).



# 5

## Conclusion

This thesis was introduced by asking the somewhat existential question of how to properly define life. It would then seem appropriate to conclude by asking whether the organisms presented in this work can be considered alive. I would argue that, at least under some reasonable minimal definitions, this *is* a form of life. The organisms evolved in this project certainly feel more lifelike than a dry desert plant seed and it would seem unwise to limit the definition of life to include only the form of life we have encountered here on Earth.

A more substantial conclusion considers the actual nature of open-ended evolution and emergent selection pressures. Biological ecosystems drive variability by dynamically creating niches, as introduced in section 1.1. The organisms presented in this work have been shown to drive variability among each other; inventing niches for new adaptive lines to exploit.

A both obvious and expected example of this is the plant organisms covering the water surface, thereby creating a niche for non-plant decomposer organisms living on the bottom (subsection 3.2.2). Without the surface plants, the fitness of organisms at the bottom would be determined by their ability to collect sunlight. However, with the surface plants covering the sunlight, the fitness of the bottom-living organisms is instead determined by their ability to harvest energy from the detritus particles falling down from the dying plants above.

A less expected example is the predator-lagoon niche, seen in figures 3.11 and 3.12. Created by a combination of the terrain topology and the land-living plants, the predators overcame mobility- and sensor-related deficiencies in the model to discover their niche of using sting cells. The original expectation was that predators, if they would emerge at all, would move around to hunt for prey. But with the large population of plant organisms sliding down into the "lagoon", there was no need for predators to move or to sense their surroundings; the food was already provided.

To be truly open-ended, evolution need to continually introduce novelty and complexity into a system. The Grafiliv simulation has not yet run for long enough to be able to determine if that is the case, but this work could still be seen as a contribution toward a better understanding of the nature of life and how it can be simulated using GPU computing. More importantly, it also provides insights into and examples of the emergence of endogenous niches and selection pressures.





# Bibliography

- Aron, J. (2013), ‘First replicating creature spawned in life simulator’, *New Scientist*. Retrieved **12**.
- Bedau, M. A. (2003), ‘Artificial life: organization, adaptation and complexity from the bottom up’, *Trends in cognitive sciences* **7**(11), 505–512.
- Erwin, D. & Valentine, J. (2013), *The Cambrian Explosion: The Construction of Animal Biodiversity*, W. H. Freeman.
- URL:** <https://books.google.se/books?id=vnqRQQAACAAJ>
- Farmer, J. D. & Belin, A. d. (1990), Artificial life: The coming evolution, Technical report, Los Alamos National Lab., NM (USA).
- Gardner, M. (1970), ‘Mathematical games: The fantastic combinations of john conway’s new solitaire game “life”’, *Scientific American* **223**(4), 120–123.
- Langton, C. G. (1986), ‘Studying artificial life with cellular automata’, *Physica D: Nonlinear Phenomena* **22**(1-3), 120–149.
- OneZero Software (2016), ‘What is fluidix?’, <http://www.fluidix.ca/>. Accessed: 2016-09-13.
- Ray, T. S. (1992), Evolution, ecology and optimization of digital organisms, Technical report, Technical Report 92-08-042, Santa Fe Institute, Santa Fe, NM.
- Sadava, D. E., Hillis, D. M., Heller, H. C. & Berenbaum, M. (2014), *Life: the science of biology*, Vol. 10, Macmillan.
- Schumacher, J. (2011), ‘What is species?’, <http://www.speciesgame.com>. Accessed: 2016-12-20.
- Stanley, K. O. (2007), ‘Compositional pattern producing networks: A novel abstraction of development’, *Genetic programming and evolvable machines* **8**(2), 131–162.
- Stanley, K. O. & Miikkulainen, R. (2002), ‘Evolving neural networks through augmenting topologies’, *Evolutionary computation* **10**(2), 99–127.
- Yaeger, L. (1994), Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or poly world: Life in a new context, *in* ‘Santa Fe Institute Studies in the Sciences of Complexity’, Vol. 17, Addison-Wesley Publishing Co., pp. 263–263.



# A

## Getting started with Grafiliv

Grafiliv (short for *grafikkortsliv*, graphics card life) is the program created for running the simulations. Source code and binaries can be found on GitHub: <https://github.com/Akodiat/grafiliv>.

In order to run a grafiliv simulation:

1. Obtain binaries by either:
  - (a) Compile the source code:
    - i. Download and install the Fluidix library at: <http://www.fluidix.ca/>
    - ii. Using the Fluidix app, compile `\fluidix\grafiliv\grafiliv.cu`
    - iii. If you want to compile the GrafilivViewer as well (instead of using the provided binary), download Unity3d <https://unity3d.com/> and open and compile the project at `\GrafilivViewer`
  - (b) Download binaries from the Git repository:
    - i. Download and extract the zip-file corresponding to the version you want to run from <https://github.com/Akodiat/grafiliv/tree/master/app>
2. Configure `\fluidix\grafiliv\config.txt` with the parameters you desire.
3. If compiled for terrain, make sure `terrain.stl` is present in the same directory as `grafiliv.exe`
4. Launch simulation by starting `grafiliv.exe`. If an earlier simulation was aborted, you have the option to restart it.
5. Use `GrafilivViewer.exe`, also located in the same directory as `grafiliv.exe`, to inspect the simulated organisms