

CELLULAR COHERENCE

Investigations into spatial self-organization

Murat Erkurt

Supervised by Prof. Henrik Jeldtoft Jensen

A thesis presented for the degree of
Doctor of Philosophy in Mathematics of Imperial College London

Department of Mathematics

Imperial College London

United Kingdom

28 September 2020

Abstract

The goal of this Thesis is to investigate spatial self-organization. The cellular coherence is the mechanism of local actioning units ensembled over a network to generate organization. The cells are modeled as computational machines acting through their regulatory networks with different types being the attractors of such automata. The self-organized cell types across the tissue is the emergent *organism*.

The work consists of two building blocks: *Cellular dynamics* and *gene regulation*. In cellular dynamics, we cellularize continuous space-time dynamics on a general manifold into discrete dynamics on a mesh. *Singular attractor* is proposed as a novel dynamics that generates tissue-wide symmetry breaking which is key for generating *form* in an organism. Extending the model to dipolar attractor, it is shown that animal body plans can self-organize by self-generated positional information.

In gene regulation, we develop the *computron* as a generalized computational model of genetic regulatory networks where each gene is a node of the machine and the attractor configurations are the cell types. We provide a formal set-theoretic framework for embedding the gene regulatory network into a novel metric space, called the *diversity space*. A measure for complexity of regulation is proposed using entanglement complexity of basin partitioning. Combining cellular dynamics and gene regulation, a model for *morphogenesis* is presented.

In this Thesis, we invented a mechanism for developing fully fledged organism forms through self-organization, contributing a mathematical basis to biological theory of development. Whether Nature does it the same way remains to be seen. On the practical front, our work can be used for building self-organizing nano-machinery, and can find applications in soft robotics and biomimetic architecture.

Prologue

This is a long due project. In 1983, programming ZX Spectrum at high-school, I was trying to simulate attracting of particles. A small sign error in the code resulted in formation of string patterns. This was the first *in-silico* self-organization that I observed. It occurred to me that everything that was not nothing was a pattern in the Universe. Life must have emerged due to a fundamental law of pattern formation, a law that was yet to be found. Humans are blessed with intelligence to apprehend Nature. But, we are also cursed with not being able to comprehend it. Stuck in between, we yearn for a meaning, using science, art and religion. After all these years, I am back to where I started. And it feels like home.

Declaration

I herewith certify that all material in this dissertation which is not my own work has been properly acknowledged.

Murat Erkurt

Copyright

The copyright of this Thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives license. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the license terms of this work.

Contents

1	Introduction	20
1.1	Purpose	20
1.2	Key influencers	23
1.3	Related work	24
1.3.1	Mathematical and computational morphogenesis	24
1.3.2	Gene regulatory networks	30
1.3.3	Regulatory complexity	33
1.3.4	Artificial embryogeny	37
1.3.5	Morphogenetic engineering	37
1.4	Contributions and limitations	40
1.5	Outline	43
2	Inspiration	46
2.1	Patterning universe	46
2.2	Self-organizing nature	47
2.3	Emergent aesthetics	55
3	Background on patterning	59
3.1	Terminology	59
3.2	Reaction-diffusion	63
3.3	Advection	69
4	Cellular dynamics	72
4.1	Terminology	72
4.2	Discretization methods for PDEs	75
4.3	Discretization of time with FDM	78

4.4	Cellularization of space with FVM	79
4.5	Turing instability revisited	98
4.6	Attractors in cellular dynamics	103
4.6.1	Periodic attractor	103
4.6.2	Singular attractor	110
4.6.3	Dipolar attractor	119
4.7	Segmentation and body plans	121
4.8	Dynamics on 2D mesh graphs	124
4.9	Self-organized cellular automata	126
4.10	Discussion	132
5	Gene regulation	134
5.1	Terminology	134
5.2	Gene regulatory networks	137
5.3	Computron	139
5.4	Regulatory dynamics	142
5.4.1	Lyapunov characterization	144
5.4.2	Kolmogorov–Sinai entropy	145
5.4.3	Basin entropy	146
5.4.4	Entropy in discrete state-space	146
5.4.5	Dynamics of cellular automata	147
5.4.6	Back to basins	148
5.5	Diversity space	149
5.5.1	Purpose	149
5.5.2	Diversity measure	150
5.5.3	Entanglement	152
5.6	Dynamics of autonomous computrons	153
5.6.1	Embedding in metric space	153
5.6.2	Computron as a dynamical system	154
5.6.3	Chaotic computation	155
5.6.4	Complex computation	156
5.6.5	Case Study – Linear cyclic generating graph	158
5.7	Dynamics of driven computrons	159

5.7.1	Computron as a Markov Chain	159
5.7.2	Ergodic decomposition	162
5.7.3	Chaoticity – Complexity of probabilistic automata	164
5.7.4	Susceptibility	165
5.8	Discussion	166
5.9	Proof of Lemmas	167
6	Morphogenesis	175
6.1	Terminology	176
6.2	Background	178
6.3	Theory	181
6.3.1	Vector-Reaction-Diffusion-Drift	181
6.3.2	Biological implications of VRDD	185
6.3.3	GRN as a Finite State Automaton	187
6.4	Model	191
6.4.1	Topology module	192
6.4.2	Physical module	193
6.4.3	Chemical module	194
6.4.4	Regulatory module	195
6.5	System architecture	196
6.6	Results	198
6.6.1	Fundamental forms	198
6.6.2	Making a hydra	203
6.6.3	Mutating a hydra	206
6.6.4	Other organism forms	207
6.7	Discussion	208
7	Conclusion	213
7.1	Epilogue	213
7.2	Applications	216
7.3	Current limitations and future work	216
A	Codework	218

B Simulations	223
C Artwork	233

List of Figures

2.1	Bénard convection cells and mud cracks	48
2.2	Turing reaction-diffusion and Belousov-Zhabotinsky oscillator	50
2.3	Fish school and bird flock	51
2.4	Slime mold aggregation and morphogenesis	52
2.5	Sand pile pattern formation	53
2.6	Driven aggregation in snowflakes and radiolaria skeletons	54
2.7	Human embryo at eight weeks	55
2.8	Art forms of nature – Ernst Haeckel	57
3.1	Dynamical systems overview	60
3.2	Patterning in Gray-Scott system	67
3.3	Selected patterns generated by Gray-Scott system	68
3.4	Turing instability for Gray-Scott model in parameter space	69
3.5	Aggregation pattern generated by advection	70
4.1	Partitioning of 2D manifold into homogeneous voronoi cells	82
4.2	Generating mesh graph from Voronoi cell partitioning	83
4.3	Stitching algorithm building mesh around manifold	84
4.4	Eigenvalues and eigenvectors of linear Turing system	102
4.5	Time evolution of spatial organization in linear Turing system	104
4.6	Spatial organization of linear Turing system	105
4.7	Time evolution of pattern formation in Sharing-is-Caring game	107
4.8	Sharing-is-Caring Game: Spectral modifier $M(\omega)$	109
4.9	Spatial organization by periodic attractor in Sharing-is-Caring game	110
4.10	Illustration of Source-Field reaction and diffusion of Field	112
4.11	Illustration of drift of Source on gradient of Field	113

4.12	Single and double source diffusion	114
4.13	Piecewise Linear Field approximation	115
4.14	Time evolution of Actual Field vs Piecewise Linear Field	117
4.15	Singular peak formation	119
4.16	Dipolar attractor time evolution	121
4.17	Self-organized body plans	124
4.18	Parameter sets used for generating bodyplans	125
4.19	Embryogenesis simulation for fly and insect variants	126
4.20	Sharing-Caring-Game pattern formation on 2D regular grid	127
4.21	Self-generated tissue-wide anisotropy, $\alpha = 1$	127
4.22	Singular attractor on 2D regular grid	132
4.23	Self-organized cellular automata	133
5.1	Cell regulation by gene expression switching	137
5.2	Cell regulation by computron	139
5.3	Computron construction on a generating graph with gene units.	140
5.4	32-state machine on linear cyclic graph	141
5.5	Basin partitioning in quadratic Julia sets	146
5.6	Illustration of entanglement	153
5.7	Diversity-induced distance between configurations	154
5.8	Hierarchy of computron dynamics	155
5.9	Illustration of diversity-based complexity	158
5.10	Chaoticity λ vs complexity Ψ	159
5.11	Complexity synergy of driven cellular automata	165
5.12	Topological distortion to driven computron due to noise	166
6.1	Initial vs limit-cycle distribution	184
6.2	FSA model of a gene regulatory network	189
6.3	Mesh topology	192
6.4	Simulation model architecture	196
6.5	Symmetries in fundamental forms	199
6.6	Fundamental forms	202
6.7	Sequential form making with FSA	205

6.8	Hydra embryogenesis stages	206
6.9	Hydra mutants	208
6.10	Jellyfish	209
6.11	Sea anemone	210
6.12	Tendrils	211
A.1	Screenshot of DNAMaker showing DNA model of Jellyfish	222
B.1	Axis1only6 morphogenesis simulation and DNA states	224
B.2	Axis1only6.dna parameters	225
B.3	Urchin22 morphogenesis simulation and DNA states	226
B.4	Urchin22.dna parameters	227
B.5	Spikeynew3 morphogenesis simulation and DNA states	228
B.6	Spikeynew3.dna parameters	229
B.7	jelly23 morphogenesis simulation and DNA states	230
B.8	jelly23.dna parameters page 1 of 2	231
B.9	jelly23.dna parameters page 2 of 2	232

Notation

Convention

Large cap U is used for state variable on a continuous manifold. Small cap u is used for state variable on cellularized manifold, i.e. mesh. Bold versions $\mathbf{S}$ and $\mathbf{s}$ refer to state vectors on manifold and mesh, respectively. The following indexation convention is used to refer to state-space-time dimensions of the dynamics.

$${}_m S_i^k$$

m	state index of state vector
i	space index (n ode, <i>e</i> dge or f ace) of mesh
k	time index of dynamics

Morphogenetic Dynamics

Symbol	Explanation
Continuous manifold	
Ω	general shaped 2D manifold embedded in 3D
$\delta\Omega$	boundary of manifold Ω
$\mathbf{n}$	boundary normal vector of manifold
$\mathbf{x}$	spatial position on continuous manifold
$\Phi(\mathbf{x})$	scalar field Φ on manifold
$\mathbf{F}(\mathbf{x})$	vector field F on manifold
$\nabla \cdot \mathbf{F}$	divergence of vector field F on manifold
$\nabla \times \mathbf{F}$	curl of vector field F on manifold
$\nabla\Phi$	gradient of scalar field Φ on manifold
$\nabla^2\Phi$	Laplacian of scalar field Φ on manifold
$U(\mathbf{x}, t)$	state variable of continuous space-time dynamics
$\partial_t U(\mathbf{x}, t)$	time derivative of state variable at point $\mathbf{x}$, time t
$\mathbf{S}(\mathbf{x}, t)$	state variables of continuous space-time dynamics
$\partial_y \mathbf{S}(\mathbf{x}, t)$	space derivative along y-axis in Cartesian manifold
$\partial_{yy} \mathbf{S}(\mathbf{x}, t)$	second order space derivative along y-axis
$\varphi(\mathbf{S})$	function of state variables, for $\mathbf{S} \in \mathbb{R}^N$, $\varphi : \mathbb{R}^N \rightarrow \mathbb{R}^N$
Mesh graph	
$M(\mathbf{V}, \mathbf{F}, \mathbf{E})$	mesh graph with nodes, edges and faces
$\mathbf{V} = \{\mathbf{a}, \mathbf{b}, \dots\}$	nodes of the mesh
$\mathbf{E} = \{a, b, \dots\}$	edges of the mesh

Continued on next page

Table 1 – *Continued from previous page*

Symbol	Explanation
$\mathbf{F} = \{\mathbf{A}, \mathbf{B}, \dots\}$	faces of the mesh
$\mathbf{V}(a)$	nodes of edge a , two of them
$\mathbf{V}(\mathbf{A})$	nodes of face $\mathbf{A}$, three of them
$\mathbf{F}(a)$	faces of edge a
$\mathbf{E}(\mathbf{a})$	the set of edges emanating from node $\mathbf{a}$; i.e. $b \in \mathbf{E}(\mathbf{a}) \iff \mathbf{a} \in \mathbf{V}(b)$
$\mathbf{N}(\mathbf{a})$	the set of neighbor nodes of node $\mathbf{a}$, which is defined as having a common edge; i.e., $\mathbf{c} \in \mathbf{N}(\mathbf{a}) \iff \exists b : \{\mathbf{a}, \mathbf{c}\} = \mathbf{V}(b)$
$\underline{\mathbf{a}\mathbf{b}}$	edge that connects node $\mathbf{a}$ to $\mathbf{b}$
$\mathbf{p}_{\mathbf{a}} \in \Omega$	position in 3D of the anchor point of cell corresponding to node $\mathbf{a}$
$C_{\mathbf{a}}$	region on tissue that is cellularized to cell $\mathbf{a}$
$B_{\underline{\mathbf{a}\mathbf{b}}}$	border on tissue that is between cells $\mathbf{a}$ and $\mathbf{b}$
$\phi_{\mathbf{a}}$	value at node $\mathbf{a}$ of scalar field ϕ on mesh
$\mathbf{f}_a$	vector value at edge a of vector field $\mathbf{f}$ on mesh
$(\text{div } \mathbf{f})_{\mathbf{a}}$	divergence of vector field $\mathbf{f}$ on mesh at node $\mathbf{a}$
$(\text{curl } \mathbf{f})_{\mathbf{a}}$	curl of vector field $\mathbf{f}$ on mesh at node $\mathbf{a}$
$(\text{grad } \phi)_a$	gradient of scalar field ϕ on mesh at edge a
$(\text{lapl } \phi)_{\mathbf{a}}$	Laplacian of scalar field ϕ on mesh at node $\mathbf{a}$
$u_{\mathbf{a}}^k$	state variable of discrete dynamics at node $\mathbf{a}$, time k

Continued on next page

Table 1 – *Continued from previous page*

Symbol	Explanation
$\langle u^k \rangle_{\mathbf{N}(\mathbf{a})}$	spatial average (over the neighbor nodes) of state variable at node $\mathbf{a}$, time step k
$\delta u_{\mathbf{a}}^k$	time difference of state variable at node $\mathbf{a}$, time step k
$u_{\mathbf{a}} \rightarrow$	next value of state variable at node $\mathbf{a}$ upon time iteration
$\mathbf{s}_{\mathbf{a}}^k$	state vector at node $\mathbf{a}$, time step k
${}_m s_{\mathbf{a}}^k$	m -th state variable at node $\mathbf{a}$, time step k

Regulatory Dynamics

Symbol	Explanation
ω_i	state of <i>unit</i> i , binary valued, i.e $\omega_i \in \mathbb{Z}_2$. Each unit corresponds to a gene with binary states representing gene ON (expressing) or OFF
$M = [G, C, v_{\text{IN}}, v_{\text{OUT}}]$	<i>computron</i> : a computational machine defined by its generating graph G , connector set C and input and output nodes $v_{\text{IN}}, v_{\text{OUT}}$
$G(V, E)$	<i>generating graph</i> : each unit is assigned to a node $v_i \in V$. A unit v_j is said to be a neighbor of unit v_i if there exists the directional edge $e_{ji} \in E$
δ_i	<i>neighbors</i> of node v_i , represented as an ordered set. Each unit is assumed to be also its own neighbor, i.e. $v_i \in \delta_i, \forall i$. The cardinality of the neighbors set (the number of neighbors) is denoted as $ \delta_i $
Ω_i	configuration state of the neighbors of node v_i , i.e. for the neighbors given as $\delta_i = [v_a, v_b, \dots]$, the configuration state of neighbors is the ordered set $\Omega_i = [\omega_a, \omega_b, \dots]$
C	<i>connector set</i> : each unit is assigned a connector $c_i(\Omega_i) : \mathbb{Z}_2^{ \delta_i } \rightarrow \mathbb{Z}_2$, which a binary valued look-up table based on the state configuration of neighbors
v_{IN}	<i>input unit</i> which acts as interface for the external binary input σ , if the computron is driven with an external signal, $\omega_{\text{IN}} = \sigma$.
v_{OUT}	<i>output unit</i> which provides a binary output $\gamma = \omega_{\text{OUT}}$.
$s = [\omega_0, \omega_1, \dots, \omega_{N-1}]$	global state of computron as the configuration of local states on the generating graph.

Continued on next page

Table 2 – *Continued from previous page*

Symbol	Explanation
$d(x A)$	<i>individuality</i> of an element x given a set of constraining elements A .
$\vec{A} = [x_0, x_1, x_2 \dots]$	<i>tuple</i> , an ordered set.
$d(\vec{A} B)$	<i>diversity</i> of tuple $\vec{A}$ given a set of constraining elements, B .
$\ A B\ $	<i>diversity measure</i> of set A given constraining set B .
$d_w(x A)$	<i>individuality of weighted element</i> x given a constraining set A .
$\tilde{\cup}, \tilde{\subseteq}, \tilde{\Delta}$,	union, subset, symmetric difference of weighted sets.
$\varphi(A, B)$	<i>diversity induced distance</i> metric between sets.
$\mathbf{A} = \{A_0, A_1, A_2 \dots\}$	set of sets.
$\mathbf{P} = \{B_0, B_1 \dots\}$	a <i>partitioning</i> of Q (i.e. $\cup_i B_i = Q$, and $B_i \cap B_j = \emptyset, i \neq j$).
$\text{SPL}(v_i, v_j)$	<i>shortest-path-length</i> between two nodes v_i and v_j of a graph.
s^n	configuration state s of computron at time step n .
S	set of all configurations of a computron $s \in S$.
$s^{n+1} = T(s^n)$	T is the state <i>transition map</i> of the computron.
$\vec{A} = [s^0, s^1, \dots s^{p-1}]$	<i>attractor</i> (with period p) of computron dynamics, where $s^{i+1} = T(s^i)$, $s^0 = T(s^{p-1})$ and $s^n \neq s^m \Leftrightarrow n \neq m$.
$\mathcal{B}_\epsilon(s)$	<i>epsilon-ball</i> around state s , $\{q \in S : \varphi(s, q) < \epsilon\}$.
$\lambda(s)$	<i>Lyapunov value</i> of an initial configuration s .

Continued on next page

Table 2 – *Continued from previous page*

Symbol	Explanation
λ_M	Lyapunov value of a computron given as $\langle \lambda(s) \rangle_S$.
Ψ_M	<i>complexity</i> of a computron.
$\mathbf{P}$	<i>transition probability matrix</i> of a Markov chain.
π	<i>steady-state-distribution</i> vector of a Markov chain.
C_i	a <i>recurrent class</i> of a Markov chain.
$\mathcal{C}_M = \{C_1, C_2, \dots, C_K\}$	set of recurrent classes of a probabilistic computron M .
$\tilde{\Psi}_M$	complexity of a probabilistic computron.
$\Phi(M^A, M^B)$	<i>similarity</i> of two computrons.

Chapter 1

Introduction

1.1 Purpose

Nature is abundant with organisms having a wide variety of forms containing up to trillions of cells. Yet they all start from a single cell, the zygote, and the *embryonic development* process generates such intricate forms *replicably*. The zygote of a man becomes a man, and that of an ant yields an ant. The mechanisms of embryonic development are mostly unknown.

Coherence means *the quality of forming a unified whole*. This Thesis is about cellular coherence: How an ensemble of cells develop into an organism with predefined target form, be it a man or an ant. The *form* is the property of a unified whole generated by the cells' quality to act in unison. The overall argument of the Thesis is that biological form is an emergent property of locally interacting cells with combined dynamics of information exchange among cells (*cellular dynamics*) and information processing within cells (*gene regulatory dynamics*). The Thesis investigates spatial self-organization based on these two pillars of dynamics using mathematical notions from the theory of non-linear dynamical systems.

The first pillar explores inter-cellular dynamics that generates patterning, and more importantly, symmetry breaking across the tissue of locally interacting identical cells. Starting from continuous space-time reaction-diffusion systems showing instability against uniformity, originally proposed by Turing in 1952 [1], we cellularize the dynamics and analyze patterning in a discrete mesh where each node is a cell that exchanges information with local neighbors, showing the conditions that

result in patterning with characteristic wavelengths. We then propose our novel singular attractor model which fundamentally differs from Turing pattern formation as it generates singular peak formation across the whole tissue, breaking the topological symmetry. Using singular and dipolar attractors we show that tissue-wide positional information can be self-organized. We use this positional information to generate basic segmentation and body plans of an organism. Positional information is well known in developmental biology. An embryo is known to organize itself based on cues of positional information provided by concentration of certain chemicals called morphogens. How such positional information can be self-organized is mostly unknown. Our work shows a possible mechanism.

The second pillar investigates the dynamics of intra-cellular gene regulation. There is a long way from basic body plans to a full fledged form of an organism. We propose that intricate forms can be achieved using the symmetry breaking with dipolar attractor dynamics by orchestrating a hierarchy of such symmetry breakings at the right place and at the right time. Such orchestration can be done by the gene regulatory network within each cell. To understand the dynamics of gene regulation, we use a Boolean network, originally proposed by Kauffman [2], which we refer to as *computron* herein. We offer a novel set-theoretic framework to embed the computron into a metric space that captures similarity of configuration states, called the *diversity space*. This metric space allows us to analyze the dynamics of gene regulation, including complexity, chaoticity and susceptibility to noise. The attractors of the dynamics of computron are gene activation configuration patterns that define cell types.

Our goal in this Thesis is to come up with a minimal model for locally interacting cells to self-organize into target forms replicably. By *replicable*, we mean that given the gene regulatory network and related morphogenetic parameter set of a species scribed in its DNA, cells with identical DNA placed on a spheric blastula self-organize by way of differential growth and cell differentiation into a target form. We achieve this by combining the spatial dynamics of dipolar attractors that provide spatial symmetry breaking across the tissue, and gene regulatory dynamics within each cell that provide cell differentiation. The combined model is presented under Chapter 6, Morphogenesis.

The topic of spatial self-organization in embryogenesis is relevant and timely. Following the sequencing of DNA, vast amount of progress has been made in genetic engineering. Yet, the mechanisms for development of a full fledged organism from a single cell is still vastly unknown. The scope of the Thesis spans multiple disciplines, including dynamical systems, theory of computation, biology, engineering and computer graphics which can also cause shortcomings. The work is not part of any ongoing research programme of any lab or team. The question posed is simple and general. The approach to finding an answer is that of an engineering type, constructive rationalism. In order to understand how billions of cells in an embryo can self-organize into form, we simply try to make one. This is in stark contrast to empiricism of biology. Clearly, our resultant model of self-organization may not be how Nature is actually doing it. But, our findings are useful for developmental biology as the model raises a series of testable biological outcomes as we listed in Chapter 6. It is also useful for engineering applications. As described in Section 1.3, Related Work, *Morphogenetic Engineering* is a nascent field of engineering that seeks to generate architectures with desired functionality from building blocks that interact locally without central planning or coordination. Our model can have applications in morphogenetic engineering such as self-assembling nano-robots.

Our work resides in the prolific intersection of (i) theoretical biology (developmental), (ii) applied mathematics (complex systems and non-linear dynamics), (iii) theory of computation (finite state automata) and (iv) engineering (morphogenetic assembly). While cross-disciplinary research could run the risk of being lost in no-man's land and suffer from lack of well-defined audience base, modern scientific research increasingly uses cross-disciplines to generate novelty as mono-domain studies run the risk of being incremental and entrapped to locality within the realm of possibilities. The biologists could criticize our work as being far away from biological reality. The mathematicians can question technical rigor. The theoretical computer scientists can dismiss the work on gene regulation as already been part of a well-defined model of computation. Notwithstanding such specialists' challenges, the Thesis provides novel propositions and results across the board. There are many take aways for the biologists, provided in Chapter 6, Morphogenesis. Fellow mathematicians working on complexity would be interested in basin-centric complexity

definitions and theorems provided in Chapter 5, Gene Regulation. Computation theorists and non-linear dynamicists can enjoy the bridging that is provided under set-theoretic work we offer in Diversity Space. Given our engineering approach in tackling the central question of the Thesis, most resonant audience would be engineers with creative spirit, the *maker* types.

1.2 Key influencers

The key influencers in addressing the central theme of spatial self-organization in this Thesis are Hans Meinhardt and Stuart Kauffman.

Meinhardt (1938-2016) spent most of his research career on mechanisms of pattern formation in biology. His early work in 1972, “*A theory of biological pattern formation*” [3] was very similar to the model proposed by Alan Turing in 1952, “*A chemical basis for morphogenesis*” [1]: A reaction-diffusion system with a short-range activator with strong self-enhancing capabilities, coupled to an inhibitor of longer range that suppresses the expansion of the activator in surrounding areas.

In the ensuing 40 years, Meinhardt published several research papers, addressing the key structural challenges to self-organization in embryonic development including: “Formation of symmetrical and asymmetric structures during development of higher organisms” (1986) [4], “Tailoring and coupling of reaction-diffusion system to obtain reproducible complex pattern formation during development of higher organisms” (1989) [5], “Development of higher organisms – How to avoid error propagation and chaos” (1995) [6], “Models for organizer and notochord formation” (2000) [7], “Organizer and axes formation as a self-organizing process” (2001), and “Models for the generation and interpretation of gradients” (2009) [8].

Meinhardt focused on reaction-diffusion dynamics that could replicably generate biologically well-observed organizing regions and morphogen gradients. His theoretical models faced difficulties, such as (i) generation of secondary or periodic peaks as the tissue grows, (ii) exponential decay of gradients, (iii) difficulty in establishing orthogonal gradients for primary and secondary body axes and (iv) forming an exact number of segments. We sought to remedy these problematic areas by introducing our novel concepts: dipolar attractor model which provides tissue-wide

symmetry breaking, linear gradient formation that allow precise positioning, and robust orthogonal axis generation by vector-reaction-diffusion-drift dynamics.

Kauffman (born in 1939) is a theoretical biologist and complex systems researcher investigating emergence of life. His main line of argument throughout his career and several publications is that complexity of biological organisms might result from self-organization arising in far-from-equilibrium dynamical systems, as explained in his book *“Origins of Order”* (1993) [9]. He was the first to introduce boolean networks as minimal models of gene regulation. He proposed in 1971 article, *“Gene regulation networks: a theory for their global structure and behaviors”* [2] that cell types are attractors of dynamics in gene regulatory networks, and that cell differentiation is transition from one attractor to another. In our computron model for gene regulation, we use Kauffman’s boolean network concept, but embed such computational machine into a metric space that captures similarity of gene activation patterns which enables us to use concepts from dynamical systems theory to investigate the dynamics of gene regulatory networks.

1.3 Related work

1.3.1 Mathematical and computational morphogenesis

The development of an embryo from homogeneous ensemble of cells at blastula stage to differentiated cells organized in organs with definite form and function is an immensely intriguing and important area that had attracted various mathematical approaches.

Early days In 1917, D’Arcy Thompson [10] approached biological form as mathematical transformations at the level of organism. This was the first attempt to put a mathematical framework to embryogenesis. In 1952, Alan Turing [1] proposed a simple reaction-diffusion model that exhibits patterning when parameters are such that homogeneous diffusion becomes unstable. This was the start of chemical level mathematical modeling to explain biological development using partial differential equations. Whilst Turing’s work was suitable for generating patterns, it was not adequate for explaining morphogenesis.

Starting in early 70s, Wolpert [11] introduced his *positional information* model to explain development during early embryogenesis. His model assumed extrinsically defined organizing regions generating chemical gradients that served as cues for cells to understand their position within the organism. With *in vivo* evidence for such chemical gradients, Wolpert's model was satisfying to most biologists, but it could not answer how such organizers were self-organized to start with. Furthermore, Wolpert opposed Turing-like mathematical approach to self generated patterning via reaction-diffusion systems. Also starting in early 70s, Gierer and Meinhardt [3] introduced their theory of biological pattern formation, which had very similar formulations to that of Turing.

Shortcomings

The seminal work of Turing that proposed self-organization of tissue patterns through reaction-diffusion systems was not able to address generation of form in an organism. A pattern breaks local homogeneity on the tissue, yet it does not create tissue-wide symmetry breaking. If we look at the skin pattern of a zebra, we can clearly see the stripes, but the pattern does not have a head or tail. As Turing allegedly said "*Well, stripes are easy, but what about the horse part*".

The original theory of Wolpert for development of an embryo is based on cells getting positional information on local values of chemical gradients that extend from polar organizing regions of the early embryo. While this approach could explain self-organization of cells across the tissue to generate tissue-wide symmetry breaking and generation of form, it did not answer the fundamental question of how the organizing polar regions self-organized to start with.

Proposed solution

We propose a mathematical framework extending Turing's reaction-diffusion formalism where singular and dipolar attractor dynamics self-generate polar layout and positional axis in the early blastula, addressing the shortcomings of Turing and Wolpert's models and effectively offering a bridge between them.

Recent advances Over the last three decades, Turing-Meinhardt type models of embryonic development were extended to incorporate (i) gene regulatory networks making each cell a computing agent, rather than a stateless entity, and (ii) tissue growth by cell proliferation and motility making the tissue dynamic rather than static.

Particularly over the last decade, huge amount of progress was made in *in vivo* measurements of protein distributions during embryonic development in model organisms such *Drosophila*, *Xenopus leavis* (African frog), zebra fish and chicken limbs. With exponential increase in computing power, computational morphogenesis became a new research field where cell level simulations of organisms with realistic chemical and physical dynamics.

Extending reaction-diffusion Starting from the early 90s, the interplay between genetic regulation and development started to gain attention. Several models are proposed that built on genetic regulatory networks to explain cell differentiation which in turn generated organization across the tissue. Mjolsness, Shapiro *et al.* [12] introduced their connectionist model of development that sought correlations between experimental data on gene expressions and developmental processes. They expanded their connectivist framework to a simulation system called **Celerator** [13] which allowed effective representation of biochemical reactions within a cell as well as interaction among cells.

Expanding from patterning by reaction-diffusion equations into form generation by differential growth driven by morphogen concentrations was introduced by Dillon and Othmer in 1999 [14], in their mathematical model for limb development. The model treats tissue as a viscous, incompressible fluid whose volume increases by virtue of a distributed source that stems from cell division and growth. The growth is driven by morphogen concentrations across the tissue. The dynamics of growth and morphogen reaction-diffusion is coupled by partial differential equations.

In 2000, Salazar *et al.* [15] introduced a theoretical model for coupling inter-cellular morphogen dynamics (diffusional or cell-to-cell contact) with intracellular gene regulatory dynamics. Their model consisted of up to 30 genes for intracellular network of regulation and up to 30 cells linearly positioned on a tissue each communicating locally with selected protein levels generated by the intracellular gene

network. Their approach was to go through 10^6 varieties of gene network configurations and see which ones created a patterning at the tissue level. This way they established an *in silico* empirical correspondence between regulatory network graph type (genotype) and resulting patterning (phenotype).

Also in 2000, von Dassow *et al.* [16] published in Nature their development model for *Drosophila* segment generation. Their model is an interplay of intra-cellular genetic regulatory network and inter-cellular morphogen protein diffusions which are discretized as cell-to-cell couplings, simulated in a computational platform called **Ingeneue**.

Shortcomings

Mjolsness, Shapiro *et al.*'s **Celerator** simulation system was capable of representing biochemical reactions within and among cells, as well as diffusion of such biochemicals. But their model did not have a “*drift*” concept which is directional diffusion where direction is set by local gradients of an underlying morphogen field. Salazar's work on coupling reaction-diffusion dynamics to intra-cellular gene regulator dynamics was an important step towards regulating morphogen dynamics. However, their work did not propose an embryogeny that directly mapped the gene regulation to morphogenesis but instead sought patterning capabilities of all possible combinations of gene regulation with a limited number of genes.

Proposed solution

Our work extends Turing-Meinhardt type reaction-diffusion equations into vector reaction-diffusion-drift equations that are represented in a cellularized discrete tissue domain rather than as continuum equations. Our dipolar attractor model is novel and can generate polar organizing regions by self-organization. The vector format of the dynamics allows for orthogonal axis formations with linear gradients providing precise positional information for differential tissue growth. Our tissue is dynamic, allowing modeling of embryonic growth. Cellular differentiation is generated by finite state automata type model for intracellular gene regulatory network which is tightly coupled to the morphogen dynamics. Our proposed solution can self-organize positional information with orthogonal axis and linearly decaying morphogen gradi-

ents. Using such positional information, we are capable of precisely generating digit formations and segmentations. Further, our modeling of gene regulatory network as finite state automata allows us to generate self-organized sub-regions with sub polar systems, resulting in detailed forms self-organized in an orchestrated fashion.

Computational models of the last decade In 2007, Robertson *et al.* [17] developed a simulation model for tissue morphogenesis in *Xenopus leavis*, specifically the mesendoderm migration. The model uses *Xenopus* specific signaling pathway for intracellular gene regulatory network, and an agent-based model for migration of cells where each cell have direction sensitive interaction with surrounding cells.

In 2010, Hoehme and Drasdo [18] developed **Cellsys** as a modular software platform for efficient off-lattice simulation of growth and organization processes in multicellular systems in 2D and 3D using an agent-based model that approximates cells as isotropic, elastic and adhesive objects. Also in 2010, Boehm *et al.* [19] combined empirical data sets for shape change during the mouse hind-limb bud outgrowth with 3D finite element model of tissue growth. Their model suggested that elongated growth was not due to gradient of observed proliferation rates but due to directional cell orientations which was then confirmed by experimental data.

In 2011, Hester *et al.* [20] produced an integrated model for somitogenesis, the formation of body's primary segmental structure common to vertebrate development, using **CompuCell3D** simulation environment, including intracellular regulatory network, intercellular morphogen signaling and adhesion-driven cell sorting. In 2014, Glimm *et al.* [21] published a reaction-diffusion-adhesion model to explain observed patterning that serves as primordia for avian skeleton development. The model follows Turing-Meinhardt type approach. The analysis is tightly tuned to experimental data on early embryonic avian skeleton development. Their numerical analysis of reaction-diffusion-adhesion model showed that cell motility, which was not taken into account in traditional Turing-Meinhardt type reaction-diffusion formulations, was intrinsic to patterning process and that patterning was observed in parameter regions that would not have caused Turing-instability in a cell-static model.

In 2016, Villoutreix, Delile, Doursat *et al.* [22] published a research that integrated for the first time data from imaging technology that allows cell by cell viewing

of embryogenesis into a computational model. Most recently, in 2017, Delile *et al.* [23] published their comprehensive computational morphogenesis platform **MecaGen** which combines intracellular gene regulatory network dynamics, intercellular morphogen dynamics and tissue biomechanics at the cellular level. The simulation platform can be applied to a range of morphogenetic events in the embryonic development of different model organisms such as *Xenopus*, *Echinoida*, *Drosophila*, or other types of multicellular processes such as tumour formation.

Tanaka [24] provides a review of techniques used for tissue modeling, including continuum models, spheroid and vertex based cell discretizations, as well as simulation platforms including **CompuCell3D**, **CellSys**, **EPISIM**, **VirtualLeaf**, **Biocellion**, **Morpheus** and **LBIBCell**.

Shortcomings

The computational models of the last decade have the capabilities of modeling and simulating complicated regulatory and morphogenetic dynamics. Some of them are also capable of integrating biomechanics to the dynamics. These models have been extended into platforms that can be used by third-party researchers. Some of these platforms model only 2D tissue. The ones capable of 3D simulation require extensive processing power and simulation time due to the fact that cells are simulated as voxels filling the 3D tissue or reaction dynamics being defined and solved in continuous space.

Proposed solution

We developed our own simulation platform **EmbryoDeveloper** and GUI based genetic code design tool, **DNAMaker**. The **EmbryoDeveloper** is built on a dynamic mesh where each cell is node centric. Mesh is a 2D surface, embedded in 3D, that models the boundary of the organism. Modeling a 3D form by its surface boundary is a technique used in parametric architecture in the last decade, called *boundary representation* (BREP). As opposed to so-called solid-works of 3D models, BREP approach reduces computational load as what is inside the form is discarded for calculation purposes and focus is solely on the defining boundary surface. We borrowed this technique from modern architectural design and applied to our embryogenesis

simulation platform.

We model cell growth as expansion of the boundary mesh at each node in a direction normal to the surface. This minimal model approach is a deviation from realistic cell-by-cell models described above, yet it allows for very fast simulation times as cell dynamics within the tissue is discarded and only boundary growth is simulated. Appendix A, **Codework**, provides an overview of structure and routines used in **EmbryoDeveloper**. Appendix B, **Simulations**, provide parameter sets designed with **DNAMaker** and resultant forms. Appendix C, **Artwork**, shows renderings of life-like form generated by our of morphogenesis model, run on our simulation platform.

1.3.2 Gene regulatory networks

In their seminal paper of 1961, “*Genetic regulatory mechanisms in the synthesis of proteins*”, Jacob and Monod presented a model for regulation within a cell. In their model, the manufacturing of proteins is suppressed by the act of binding of a repressor protein to a specific site in the DNA responsible for encoding such protein. The repressor itself is encoded by a regulatory gene in DNA, generating a network of expression and suppression, called the *Gene Regulatory Network* (GRN). Jacob, Monod and Lwoff won the Nobel Prize in Physiology or Medicine in 1965 for their discoveries concerning genetic control of enzyme and virus synthesis. This marked the beginning of a new era where DNA was no longer just informational but had executional (or computational) role, regulating not only cell behavior but also cell differentiation.

Gene regulation has become the central mechanism for compiling the information provided in the DNA into executable processes that in turn define the development and behavior of an organism. Gene regulation as a network of expression control is a dynamical process with time scales ranging from cell metabolism in the order of seconds, the organism development in the order of days, to evolution in the order of thousands of years, as presented by Davidson in his book “*The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*” [25]. Vast amount of research is being undertaken on computational modeling of gene regulatory networks that approximates the observed biological reality as accounted by Bolouri in his book

“*Computational Modeling of Gene Regulatory Networks*” [26].

The theoretical biology community has proposed abstract models for gene regulation focusing on its dynamical and networked features. We refer to the review article published by De Jong in 2002 [27] for an excellent overview of mathematical formalisms used for gene regulatory networks, including Bayesian networks, Boolean networks, logical networks, ordinary and partial differential equations, qualitative differential equations, stochastic equations and rule-based formalisms.

In *Bayesian network* formalism for GRNs, each gene is represented as a node on a directional graph. Expression level of a gene is a random variable whose probability distribution is conditional on the the expression levels of genes impacting it (defined as a directional edge on the graph). The Bayesian network approach for modeling regulatory networks is useful when incomplete knowledge about the system is available, and provides a statistical approach to identifying causal gene expression relations. However, it has the disadvantage of leaving out dynamical aspects of gene regulation.

In *Boolean network* formalism for GRNs, genes are represented also as nodes on a directional graph with expression levels abstracted to ON/OFF boolean states. The state transitions are defined as logic functions of the boolean states of the impacting genes. The states of all genes update synchronously. This was the very original model proposed by Kauffman [2].

The *generalized logical network* formalism for GRNs is an extension of the Boolean approach where gene expression level can have more than two states and the transition between states can occur asynchronously. These generalizations allow logical networks to better represent the observed features of biological GRNs as demonstrated in the study of a number of genetic regulatory systems of limited scale such as λ phage infection in *E. coli*, dorso-ventra pattern formation and gap gene control in *Drosophila*, and morphogenesis in *Arabidopsis thaliana* [27].

As we move towards less abstraction, GRN dynamics within a cell are generally defined by *non-linear ordinary differential equations* (ODE) where the concentrations of expressed proteins are given as continuous time-dependent variables with regulatory interactions among them set by non-linear functions. In this formalism, gene regulation is modeled by rate equations expressing the rate of production of a

component of the system as a function of the concentrations of other components. These techniques have been applied to well studied organisms such as *Xenopus*.

In between Boolean network models and non-linear ODE models, we find *piece-wise linear differential equations* formalisms, where rate equations for gene expressions are given as linear combination of impacting gene expression levels, followed by a highly non-linear threshold function (sigmoid type). Similar formalism is also used in modeling switching dynamics of neural networks.

Other mathematical formalisms used in modeling genetic regulatory networks include *qualitative differential equations* where a discrete description is abstracted from a continuous model, and *stochastic differential equations* where the concentrations of expressed proteins are given as continuous time-dependent random variables with regulatory interactions defined through stochastic differential equations.

Shortcomings

Theoretical and experimental modeling of gene regulatory networks is a very active field. The Boolean network formalism for GRNs, originally proposed by Kauffman, is simple and provides a good framework for conceptualizing regulation function. Our goal is to analyze the dynamics of such regulation as a computational machine, a finite state automaton. The current literature does not offer a suitable mathematical foundation to analyze the dynamical characterization of finite state automata.

Proposed solution

In our work, we used Boolean network formalism for GRNs. With the equivalence theorem proven in Chapter 5, we abstracted the intra-cellular gene regulation to a finite state automaton with state transition given by fate functions whose inputs are chemical morphogen concentrations. By our novel embedding of the finite state automaton into a metric space that captures similarities of gene expression patterns, we pave the way to use concepts from non-linear dynamical systems theory to investigate behaviors of GRNs, including complexity, chaoticity and susceptibility to noise.

1.3.3 Regulatory complexity

Complexity of dynamical systems and that of computational machines are mostly disconnected areas of research. The dynamical systems typically live in metric state spaces and evolve in time by iterating the rules of dynamics. The system is said to be complex if the information required to describe the evolving paths is high, a notion formally defined as Kolmogorov-Sinai entropy. On the other side of the world where computational machines live, the aim is to map a set of inputs to a set of outputs.

A *cellular automata* (CA) is a dynamical system that consists of a grid network of units where each unit run the same evolution rule that tells it how to change its current state (which is drawn from a finite number of elements) into the next, given the neighbors' and its current state. The global state of CA is the configuration of local states of each units on the grid. The local evolution rule and state information exchange between neighbors synchronously update all local states in each time step, changing the configuration of CA to its next, hence generating the dynamics on discrete configuration state space. CA is called elementary if the grid is 1D, each unit takes binary state values, and the neighborhood is just the left-right nearest units.

We define *computron* as a computational machine composed of a finite number of units placed on nodes of a directional graph. The units have binary states which are updated according to local transition rules as a function of connected units' states. Computron is the generalization of a cellular automata where local rules are not necessarily uniform and graph not required to be a regular grid. Furthermore, a computron can have an external input, in which case it we call it *driven*, otherwise *autonomous*. The computron's global state is the configuration of its units' local states on its graph. It performs computation by mapping an initial configuration to an attractor, be it fixed or periodic, by iterative dynamics. Such mapping partitions its configuration space. An autonomous computron is equivalent to the Boolean network that was proposed by Kauffman as a minimal model for gene regulatory networks [2].

We attempt to bridge disparate worlds of dynamical systems and computation by embedding computrons in a suitably defined metric space that captures similarity of

its configurations. We develop notions of dynamical chaoticity using Lyapunov characterization of the machine's evolution and computational complexity by analyzing the basin partitioning of this proposed metric space.

In the early 1980s, Wolfram ignited research interest in cellular automata with a series of articles [28], [29], [30] where he classified the apparent spatio-temporal configuration complexity as: (1) tends to a spatially homogeneous state; (2) yields a sequence of simple stable or periodic structures; (3) exhibits chaotic aperiodic behavior; and (4) yields complicated localized structures, some propagating, i.e. *complex patterns*. Wolfram's classification made a distinction of chaotic vs. complex configuration evolution. This was important because entropy based definition used in dynamical systems theory did not differentiate chaotic behavior from complex. He conjectured that complex CA performed universal computation. Culik and Yu (1988) [31] proposed a framework for Wolfram's complexity classes. Key demarcation was whether an algorithm existed to decide for any two finite configurations if one ever evolves to the other. Since then there has been a wide range of classifications on elementary CA to capture the intuition that complex was something different than chaotic.

Li and Packard (1990) [32] looked into the structure of *rule space* and investigated bifurcations in the behavior of 1D CA by varying a classifier parameter λ defined as the ratio of non-zero entries in local rule table of CA. The rule-space seemed to be split into ordered (fixed or periodic patterns) and chaotic regions with a sharp phase boundary. However, in certain parts of the rule space, between the ordered and chaotic domains, the phase boundary was a thick wall that housed critical rules which showed complex patterns. Kurka (1997) [33] provided three formal ways of classification for 1D CA: (i) based on the complexity of languages generated by partitions of the state space; (ii) based on Lyapunov stability; and (iii) based on attractors of the configuration space. Kurka demonstrated correspondence between the classes of these three schemes and Wolfram's heuristic classification.

Chua *et al.* (2002) [34] proposed a complexity index to categorize the rules of elementary CA: Consider a cube graph where each vertex corresponds to one of 8 states of the input triplet of a unit. Let us color each such vertex with Red or Blue depending on what the output state ought to be based on the local rule table. Now,

imagine partitioning the cube by planes such that same colored vertices remain in the same partition. Chua's complexity index is the number of planes required to make such partition. He showed that Class IV elementary CA of Wolfram had index 2 or 3. To overcome certain mis-categorizations of Chua index, Ewert recently (2019) [35] proposed a novel method based on residual circuit complexity. It takes into account not only minimal form representation of the rule table, but also the reduction of input entropy as the system evolves which further reduces the needed minimal form. Applied to elementary CA, Ewert showed that the proposed method matched well to Wolfram's heuristic classification.

Marr and Hutt (2005) [36] investigated the pattern formation capacity of binary graph automata where each unit was executing a local law based on the average state value of the neighbors passed through a threshold function parametrized by value κ . They investigated both small-world and scale-free type graphs, varying the graph incrementally to see impact on pattern formation. They measured Shannon-entropy of time strings of a single unit averaged over all units. Since graphs were not regular grids, the authors did not define any spatial pattern concept, and investigated time patterns of single units only.

Tomassini (2006) [37] analyzed the performance of binary graph automata with homogeneous unit rules over small-world and scale-free graphs over two tasks. In density classification task, the units, starting from random configurations are expected to converge to all 1 if their initial 1 density were greater than 0.5 and vice-versa. In synchronization task, the units starting from random initial configurations are expected to synchronize into a global flip-flop cycle. Cattaneo *et al.* (2009) [38] introduced two types of non-homogeneous 1D CA: (i) local rules are variable only in a region outside which the rule is uniform, and (ii) the rule is non-uniform everywhere but the neighborhood has always same radius. The authors showed that most topological properties get demolished when CA becomes non-homogeneous.

Shortcomings

Very few attempted to go beyond classical cellular automata which has two built-in regularities: (i) units are arranged on a regular grid-like unit-space, and (ii) each unit executes the same local rule, i.e. homogeneous. These properties arguably give

the classical CA a physical-like feel with uniform laws and regular space, and make dynamical analysis tractable. It is however tempting to go beyond classical set up and consider generalized graph automata where units are placed at nodes of a general directional graph, and each executes a different local rule. Many real-life systems with interacting agents are typically organized into networks such as Watts-Strogatz type small-world graphs, or Barabasi-Albert type scale-free graphs with non-uniform local rules. Finite size discrete systems have been mostly dismissed in the entropy world since paths are of finite size and ultimately fixed or periodic. However, these systems as computational machines are very rich in the structure of their basins of attraction. In fact, mapping initial states to ultimate destinies is what computation means for these machines. As we shall formally present in Chapter 5, our definition of chaoticity and more importantly complexity of such finite state automata is all about quantifying the structure of basin partitioning, and as such fundamentally differs from previous work on complexity and entropy for automata.

Proposed solution

Our work differs from the literature as it deals with general finite state automata, both autonomous or driven, on arbitrary network topology and non-homogeneous, rather than autonomous CAs which is a special case. Our methodology is also different: Finite state automata are decomposed into boolean networks, called computrons, which are then embedded into a topological space using *diversity-induced-distance* as a metric introduced to define similarity of states and sets of states. Chaoticity is treated with Lyapunov separation in general metric space. A novel definition of complexity is proposed that is based on disentanglement of basins of attraction which captures *strangeness of the attracted* in state space as opposed to literature that deals with *strangeness of the attractor* in time domain. The theory is then extended into driven machines using Markov chains where recurrent classes with fuzzy-basins replace traditional basins of attraction. Our framework also provides a similarity measure between two machines operating on the same state space. Using this measure, we defined topological susceptibility and investigated impact of noise as a particular case.

1.3.4 Artificial embryogeny

Evolutionary computation is a search process in genotype space whereas selection is effected by fitness in phenotype space. The mapping of genotype to phenotype by developmental growth is called *embryogeny*.

In 1999, Bentley [39] introduced a conceptual framework for embryogeny, classified into three types as external, explicit and implicit. External embryogeny is designed, not evolved, and provides a parametric mapping between genotype and phenotype. In explicit embryogeny, genotype is program-like evolvable instructions that drive growth of organisms. In implicit embryogeny genotype is a set of chain of instructions that can be used iteratively and within sub-domains of a growing tissue. Bentley provided a very interesting framework to compare evolvability of embryogeny classes and conjectured that implicit embryogeny is maximally evolvable.

In 2003, Stanley and Miikkulainen [40] introduced artificial embryogeny as a sub-discipline of evolutionary computation in which phenotypes undergo developmental phase, and proposed a new taxonomy, dismissing the differentiation of embryogeny into grammatical (Lindemeyer-type phenotype developed by hierarchical pattern reuse) and chemical (Turing-type based on intercellular chemical dynamics).

Proposed future work

Our model of morphogenesis is an implicit embryogeny that is capable of growing biological forms from symmetric blastula driven by genotype operating as finite state automata inside each cell. We have not investigated the evolutionary potential of our morphogenesis model. Future research that can combine evolutionary aspects with developmental side could move our model into the domain of computational evolutionary development (*EvoDevo*).

1.3.5 Morphogenetic engineering

In 2009, Rene Doursat launched a new field called *morphogenetic engineering* and compiled research in his book “*Morphogenetic Engineering: Toward Programmable Complex Systems*” [41]

“Morphogenetic engineering explores the design, implementation, and

control (directly, by programming, and/or indirectly, by learning or evolution) of the agents of complex systems capable of giving rise autonomously and reproducibly to large heterogeneous architectures that will support a set of desired functionalities, without relying on any central planning or external drive.”

Rene Doursat

Doursat notes that complexity science traditionally approached self-organization by seeking to explain order and unstructured patterning with stateless agents interacting through laws of physics or chemistry, using the perspective of statistical physics. However, nature is abundant with self-organized architectures that are built by interacting agents that have *computational capability* that can be seen as stateful-machines running finite state automata. Doursat calls this *organized self-organization*. Such computational capabilities of stateful agents is understudied in complexity science, and conversely, the self-organization is overlooked in traditional engineering. The goal of morphogenetic engineering is to design a programmable and reproducible mapping between local computation rules acted on by the agents (the *genotype*) and the collective emergent forms and behaviors at the macroscopic level (called the *phenotype*).

Doursat identifies four categories of morphogenetic models based on dynamical process characteristics of the system: Constructing, coalescing, developing and generating. The developing category is where our work in this Thesis falls into: Understanding the mathematical framework for genotype-phenotype mapping in embryogenesis.

As a scientific and engineering field established only a decade ago, seeking to mimic the organized self-organization performed pervasively by nature, morphogenetic engineering has promising applications in artificial systems including self-assembling nano-robots, self-morphing micro drone swarms, and possibly self-constructing civil architectures.

In 2008, Bai *et al.* [42], published a system that self-organizes morphogenetic primitives (circular cells in 2D with orientation) into user selected forms by local interactions driven by rule sets that are not hand-designed, but rather selected through genetic programming. Cells move on chemical gradients they sense which

are based on the position and orientation of close-by cells. Each cell is identical in terms of rule set they execute and interactions are localized to neighborhoods of c. 4 cell size. The cells move around aggregating and aligning based on the rule set and the sensed chemical gradients. The rule sets are chosen by genetic programming, seeking closeness to a user defined target shape of the emergent shape collectively made by the cells. Bai *et al.* have shown that their system was capable of arranging cells into pre-defined shapes such as ellipse, star etc. However, the replicability was challenging.

In 2009, Doursat [43] published his 2D embryogeny model that generates form by cells that divide and aggregate based on self-organized positional cues provided by morphogen gradients. Cells types are selected based on morphogen levels which result in dynamic cell differentiation and generation of sub-tissue domains with hierarchical execution of morphogen dynamics. Using his model, Doursat showed body plan formation and segmentation.

In 2014, Doursat and Sanchez [44] published an article in Soft Robotics journal, explaining their *in silico* embryonic development platform, **MapDevo3D**. Their model consists of swarming cells with physical, chemical and genetic properties. The physical properties are modeled in terms adhesion and directional cell division with orientation normal to tissue membrane. The chemical part generates gradient information from so called source cells. The gradient diffusion is modeled as a cell-to-cell digital conveyance of information. Source cells are self-organized, not hand crafted. Sources repel each other and drift away. Sources are paired so that North-South and Front-Back axis are self generated. Positional information is determined at each cell based on gradients emanating from the sources. Cell differentiation occurs at genetic module based on positional cues. Such differentiations initiate organization on sub-tissue levels, generating form detail hierarchically.

Shortcomings

As an engineering solution, Doursat and Sanchez' work in **MapDevo3D** is impressive. The main challenge is replicability, detail in form and computational load. Furthermore, Doursat and Sanchez' engineering solution was decoupled from the reaction-diffusion mathematical formalism. Our proposed solution addresses the

engineering challenges and provides an analytical framework for cellular dynamics.

Proposed solution

Although independently created, our approach to modeling embryogenesis has many similarities to that of Doursat. Our chemical module also uses sources and gradient fields. However, it is based on a formal reaction-diffusion-drift dynamics in vector format. We provide analytical treatment of such dynamics in detail. We also provide a formalism on singular and dipolar attractors, followed by body plan generation. Our regulatory model is a finite state automaton formalism with switching cues taken from positional information, very similar to that of Doursat. However, we provide a computation theoretic analysis of what it means to have complex regulation in the manuscript. Starting from a Turing-Meinhardt type reaction-diffusion formalism and using mathematics of dynamical system theory for treating both the reaction-diffusion-drift and the regulation side of the model, we provide a formal framework which ends up, at the implementation level, convergent to the engineering works of Doursat. Our proposed solution of modeling 3D form as a boundary representation (BREP) of 2D surface helped us remedy the engineering challenges. The dynamics is run on the boundary surface only, making it more stable and fast converging with less computation required.

1.4 Contributions and limitations

This Thesis has the following original contributions to mathematical theories for biological development and gene regulation:

1. We introduced the singular attractor dynamics that generates global symmetry breaking across the tissue resulting in singular peak distribution. We provided an analytical treatment of the singular attractor dynamics, using notions from theory of heat transfer, and derived an upper bound for the expected convergence time to formation of tissue-wide singularity.
2. We proposed dipolar dynamics as an extension of singular dynamics that self-organizes an axis on spheric blastula and provides positional information across

the tissue. We have shown that using this self-organized positional information, it is possible to generate body plans and segmentation replicably which provides a novel possible mechanism to explain early embryonic development.

3. We transferred the continuous space-time dynamics into discrete dynamics on generalized mesh graph, and also discretized state variables which resulted in expressing the singular attractor dynamics as graph cellular automata. This gave us a simple digital algorithm for selecting a single node on a graph which is equivalent to the leader election algorithm in distributed computing.
4. We introduced computron as a computational machine which is equivalent to a finite state automaton but with the feature that flow of information and processing of information are segregated. The computron is then extended to have external input and also be probabilistic. As such computron is proposed as an extension of Boolean network formalism.
5. Diversity space is introduced as a novel metric space that captures the similarity distance of configuration states of graph automata. This set-theoretic concept allows us to bridge the theory of computation into theory of dynamical systems. This is important because we can now address the questions of chaoticity and complexity of computation from a dynamical systems point of view.
6. A new complexity concept is provided based on entanglement of basins of attractions generated by the computron. Basin centric complexity is a relatively new concept in dynamical systems theory as opposed to path centric complexity captured in entropy-based definitions.
7. We further extended the computron model to probabilistic automata and introduced an original concept of fuzzy-basin which enabled us to qualify chaoticity and complexity of probabilistic finite state automata, including their susceptibility to noise.
8. Vector-Reaction-Diffusion-Drift (VRDD) is proposed as a dynamics that builds on our singular attractor dynamics. Combining VRDD with gene regulation,

we generate a model for morphogenesis which is shown to be capable of generating organism forms replicably with testable propositions for biology.

We note the following limitations of our work. Future research paths are proposed in Chapter 7 to expand the scope and address some of the current limitations.

We sought to establish some form of homeomorphism in our embryogeny between the genotype, in terms of its regulatory complexity, and the resultant phenotype, in terms of its form. That is the reason we devoted a full chapter to characterize the regulatory network as a dynamical system. But, we have not managed to establish such characteristic mapping between genotype and phenotype representations defined by our embryogeny.

We design our artificial organisms' genotypes using our **DNAMaker** tool with an artisan approach to generate life-like forms. It may be possible to create an evolutionary development environment where novel forms are generated through evolution in genospace, rather than by intelligent design.

Our morphogenetic model starts from blastula stage (i.e. circa 10,000 identical cells on the surface of a ball) and develops the form thereon. We have not looked into the implications of starting self-organization at the single cell zygote state.

We did not investigate the regenerative properties of our proposed embryogeny. Certain species such as hydra have distinct regenerative capabilities. Whilst hydra typically reproduces sexually, it can also reproduce by budding with each bud giving a cloned offspring. It should be possible to analyze under what conditions budding would be feasible in our embryogeny.

Individual cell level defects in the regulatory network could result in abnormal growth such as tumour formation. We have not investigated the implications of individual cell level defects, i.e. altered GRNs, to organism development.

Our morphogenetic simulation model is membrane based. Growth of the organism is modeled by the expansion of the membrane defining the boundary of the organism in 3D. Whilst this provides for fast execution of simulation since we do not model cells inside the boundary, it is clearly a deviation from reality. Our model should be extensible to full 3D modeling of the organism which could generate forms defined with hard tissue rather than soft membrane.

The case studies provided on computron is limited to linear cyclic generating

graphs with only 5 genes. This is due to computational power limitations in complexity calculations. Graphs with more genes and alternative topologies, such as hierarchical ones, should be studied with calculations run on more powerful computational platforms.

The susceptibility analysis of gene regulatory networks is briefly touched upon where we investigate the distortion in computation performed by a GRN due to thermal noise. Impact of point mutations in GRN's generating graph (which gene impacts a given gene's expression) or connectors (how they impact the expression) should be investigated.

Clearly our proposed embryogeny is a conceptual model that shows how self-organization of cells into an embryonic form could be achieved. As such it can be very useful for morphogenetic engineering but it lacks footing in biology. In Chapter 6, we list some testable biological propositions implied by our proposed embryogeny.

1.5 Outline

In this thesis, we describe cellular coherence as the mechanism of local actioning units ensembled over a network to robustly and replicably generate organization. The units are called *cells*, and their spatial network the *tissue*. Cells are computational machines acting as finite state automata through their gene regulatory networks and cell types are attractors of the dynamics performed by such automata. Spatial organization of cell types across the tissue is the emergent *organism*.

Chapter 2 shares with the reader our source of inspiration for writing this Thesis. It is proposed that the Universe is endowed with a patterning instinct. Examples of self-organization in nature, pervasive across physical, chemical, biological and social scales, are provided. Emergent aesthetics of nature is noted with reference to works of Ernst Haeckel.

Chapter 3 gives background on the reaction-diffusion systems. Turing instability that make homogeneous solutions unstable and spontaneously generate stable spatial patterns is introduced with standard linear analysis applied to Gray-Scott model. Advection is then explained as an important dynamics generating aggregation by local fields which is the core of chemotaxis phenomenon.

Chapter 4 starts with transfer of continuous space-time dynamical systems into cellular systems. The two-dimensional manifold is partitioned into a mesh graph with each node representing a cell, and each edge defining the neighboring relationship. A framework for exporting vector calculus on the manifold into equivalent cellular operations on the mesh graph is provided using finite-volume-method. This framework allows us to apply pattern generating partial differential equations as discrete dynamics on the cellular space.. We analyze periodic pattern formation of Turing systems on cellularized tissue and derive analytic results for sufficient conditions to generate patterns as well as characteristic pattern wavelengths. We introduce our novel singular attractor dynamics and derive an upper bound on expected time to generate singularity on the tissue. We then expand the singular dynamics into dipolar dynamics and show that basic body plans can be self-organized with the self-generated positional information of this dynamics replicably. We lastly discretize state-space which moves the dynamics into a graph automata framework. The singular attractor dynamics on general graph automata is shown to be equivalent to a leader election algorithm of distributed computing.

Chapter 5 investigates gene regulation in a cell. Computron is introduced as a model of computation performed by gene regulatory networks. Cell regulatory complexity is addressed by providing a bridge between dynamical systems and computational machines. The chapter provides a background on entropy in dynamical systems in general and cellular automata in particular with an emphasis on complexity. We introduce our novel set theoretic concept of diversity space where configuration patterns of the computron are embedded. The diversity space which is shown to be a metric space, allows dynamical system theory to be used in analyzing computron, including chaoticity and complexity. The theory is then expanded to probabilistic machines using Markov chains which allows us to quantify susceptibility of cell regulation to noise.

Chapter 6 combines spatial pattern forming cellular systems and finite state automata as a model of cell-differentiation, demonstrating *in-silico* morphogenesis. The systems executes vector-reaction-diffusion-drift dynamics and each cell acts as a finite state automaton based on gene regulatory network. We develop a simulation model composed of physical, chemical and regulatory modules, and show its

capability to generate organic forms through self-organization.

Chapter 7 provides conclusion and suggestions for future research. Appendix A gives a list of objects used in our proprietary simulation environment, **EmbryoDeveloper** including the key routines. We also provide a brief overview of our GUI tool, **DNAMaker**. Appendix B gives examples of morphogenesis simulations together with the generating DNA including finite state diagrams and full parameter sets.

As a tribute to Ernst Haeckel, we provide in Appendix C our aesthetic projections of the self-organizing morphogenesis model which we exhibited in SSM museum in Istanbul, Turkey in October 2018 [45].

Chapter 2

Inspiration

“The Tao that can be told is not the eternal Tao. The name that can be named is not the eternal name. The nameless is the beginning of heaven and earth.”

Tao Te Ching – Laozi

2.1 Patterning universe

The Universe is endowed with a patterning instinct and so are we. Existence is not absolute but only with regards to and in relation with others as a nested hierarchy of patterns. Unfolding of the universe and life within it, is a continual self-organization of things into patterns. Humans have a patterning instinct that allow us to detect, classify, and conceptualize such patterns of the universe.

Our ability to extract patterns leads to our fundamental belief that there exists an extrinsic truth. Searching for such truth resulted in formation of religion and science. Plato contemplated the creator as a mathematician who formed the universe with rational principles which gave it beautiful regularity and order. This was the beginning of deification of reason in the Western world [46]. Descartes’ reductionism, introduced in his book *“Discourses”* (1637), forms a central pillar on Western philosophy of science which is today deeply entrenched into our perception of reality. Modern science is mostly built on reductionist dogma which stipulates that we can deduce the whole by understanding the pieces, such being the building blocks of reality that are extrinsically definable on a standalone basis.

Chinese philosophy, however, contemplates an alternative ground of existence:

“Throughout heaven and earth, there is li and there is qi. Li is the dao that organizes all forms above, and the roots from which all things are produced. Qi is the tools and raw material with which all things are made.”

Zhu Xi (1130-1200)

According to Chinese natural philosophy, the universe started as formless *qi* which continually self-organizes into patterns to generate reality. *Li* is the invisible organizing principle existing at all levels within the universe [47]. Leonardo da Vinci viewed the earth as an organism that was perpetually in a process of transformation with intrinsic reality defined through the dynamic patterns of nature. Alfred North Whitehead postulated in his book *“Process and Reality”* (1929) [48] that existence is made of processes rather than things, and that the dynamic unfolding is the primary account of reality. Things do not *exist*, but *occur*. As Norbert Wiener, the originator of cybernetics and one of the first to theorize that intelligence is a result of feedback processes stated [49]:

“We are but whirlpools in a river of ever-flowing water. We are not stuff that abides, but patterns that perpetuate themselves”

Norbert Wiener (1988)

2.2 Self-organizing nature

Self-organization is prevalent in nature across scales ranging from physical, biological to cultural. Sands of grains assemble into dunes, chemicals into spatial spirals. Cells organize into organs, humans into societies. Self-organization is a process in which a pattern of the system emerges solely from local interactions among its constituents. The organizing principle is intrinsic and distributed across the organized. The resulting pattern is a collective property with no extrinsic organizer nor a template. Process of self-organizing is the only intrinsic reality. It is like sailing in an uncharted ocean: Not the water or the boat, but the process of sailing is the sole

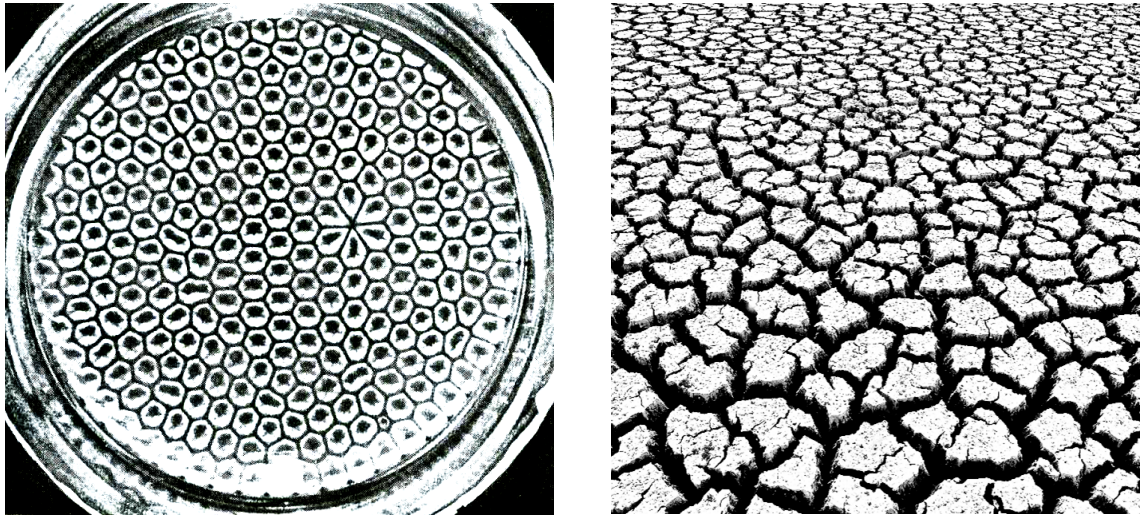


Figure 2.1: Bénard convection cells and mud cracks

reality. In a self-organizing system, the evolving pattern is created by the process and is not cognizable to its constituents nor knowable extrinsically. It occurs and unfolds in time.

Below, we provide cases of spatial self-organization in nature ranging from physical, chemical to biological systems. Per Bak's sand pile model and Bénard's convections are examples of physical pattern formation. Turing's reaction-diffusion and Prigogine's dissipative systems are examples of chemical pattern formation. The animals that interact based on simple rule sets can show purposeful emergent spatial organization, such as fish schools that can avoid predators and unicellular amoebae that can organize into stalk-spore formation to avoid starvation.

Bénard convection In 1901, Henri Bénard provided his dissertation on cellular thermal convection, which later became known as Bénard convection systems. When a thin layer of fluid is heated from below, the bottom expands becoming less dense and rises to top which is kept cool by contact to air. The cooling top part tends to sink, resulting in a convective fluid transport. Bénard's experiments were done with spermaceti oil, and as the temperature gradient between the bottom and the top of the liquid reached a critical level, fluid suddenly became a tessellated mosaic of polygons with a characteristic size in the order of 10mm. The partitioning of the surface into such polygons is similar to formation of mud cracks in dried up ponds, a Voronoi partitioning around local centers which emerge through spatial positive feedback on local fluctuations, as depicted in Figure 2.1.

Turing reaction-diffusion systems In 1952, Alan Turing postulated in his seminal paper on pattern formation “*The chemical basis of morphogenesis*” [1], that a simple two-reactant system can self-organize into stable spatial patterns of chemical concentrations. The reactants catalyze and inhibit each other respectively for which a single stable concentration state should prevail. Yet if the reactants diffuse in space at significantly different rates, then so-called Turing instability occurs which breaks the uniform spatial configuration upon slightest perturbation, resulting in intricate patterns of high and low chemical concentrations. Such reaction-diffusion systems generate patterns that are very similar to those observed in nature, including skin patterns of animals, as shown in Figure 2.2.

Prigogine dissipative structures Self-organization in chemistry arises in dissipative systems, a term introduced by Ilya Prigogine, who won the Nobel Prize in Chemistry in 1977 for his work on thermodynamic systems far from equilibrium [50]. Prigogine discovered that driven systems that take in and let out energy (and matter) could self-organize into spatio-temporal structures. He drew connections between dissipative structures and Rayleigh-Bénard instability and Turing instability. In Prigogine’s dissipative chemical systems, the energy-releasing reaction follows multiple pathways which are selected by the system based on local concentration of reaction intermediaries. Such dissipative systems can sustain far-from-equilibrium state with spatio-temporal patterns. Typical real-life example of far-from-equilibrium chemical system that show self-organization is the the Belousov-Zhabotinsky (BZ) reaction, depicted in Figure 2.2.

The above examples of physical and chemical systems generate spatial patterns. Form is different than pattern in the sense that patterns have local broken symmetry whereas form has global broken symmetry, a concept we elaborate on in Chapter 6. The examples that follow are from biological systems, ranging from basic clustering where a center is generated out of none, to dynamical forms of school that enable predator evasion with a community of 10,000 fish, to collective stalk-spore form generated by 100,000 starving amoebae, and finally human embryogenesis ultimately resulting in a functional adult with 70 trillion cells.

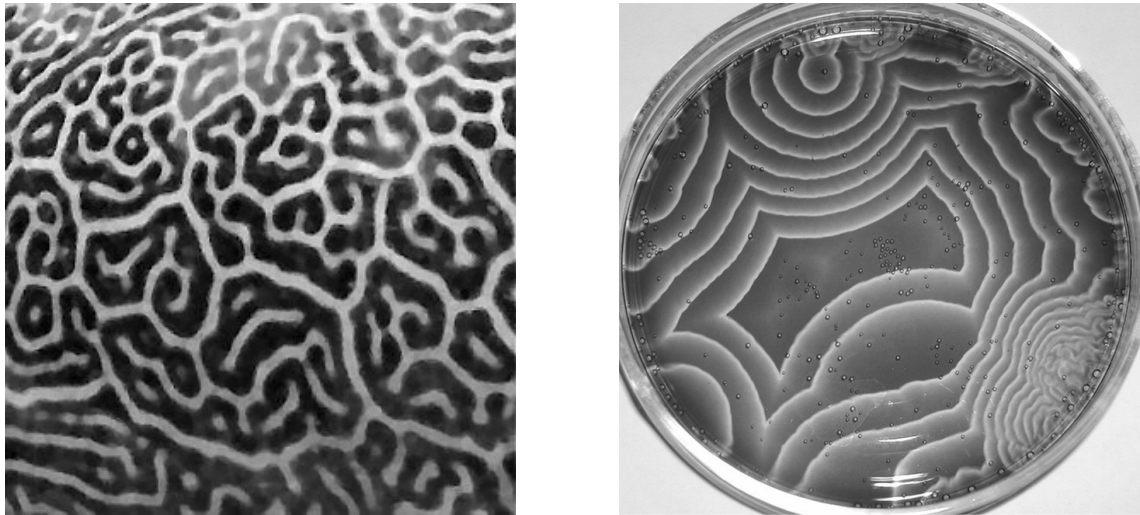


Figure 2.2: Turing reaction-diffusion and Belousov-Zhabotinsky oscillator

Clustering of beetles An important collective behavior in animals is spatial aggregation, or clustering. It is a trivial process provided that constituents can communicate their position in distance to others. This is easy for animals that can ‘see’ each other through the optical electro-magnetic field that fills the space and carries the information. In the case of many insect species, however, the spatial field is generated through chemicals deposited by the animals which is observed in a wide variety of species including honeycomb worm, Serpulid, Cirripedes, Jack pine sawfly, cockroach, beetles and caterpillar [51]. A simple model that achieves wide-range clustering is as follows: 1. Each larva emits pheromone continuously. 2. Pheromone diffuses spatially, forming a concentration gradient. 3. Larvae move upwards in gradient concentration if the concentration is above a threshold else randomly. This mechanism of motion of an individual based on the gradient of a substance concentration in the field is called chemotaxis.

Fish schooling A range of animals including fish, birds, wildebeest, lobsters and flies engage in large collective motion, called schooling, flocking or herding. These behaviors could provide survival advantage such as predator avoidance or energy optimization by improving hydrodynamic swimming efficiency [52]. A self-organization mechanism where each fish applies a few behavioral rules in response to local information from neighboring fish can effectively model school behavior of the ensemble [53]. A simple individual behavioral rule that generates such collective organization is as follows: 1. Go where your neighbors are. 2. Go where your neighbors are



Figure 2.3: Fish school and bird flock

going. 3. Do not get too close. Figure 2.3 shows the reaction of a fish school to an approaching potential predator. The school acts as a coordinated body with much enhanced sensory capability. Whereas each fish can only detect the presence of the predator at a very close distance, the ensemble can detect and communicate rapidly throughout the members of the predator utilizing the simple rule set given above. Figure 2.3 also shows elaborate motion by a bird flock. Whilst certain collective behavior can have survival advantage for the community, some just can be an emergent property with no clear evolutionary drive.

Unicellular morphogenesis by slime mold Can simple mechanisms of pattern formation such as clustering and collective motion be expanded into morphogenesis where a functional form, not merely a pattern, is generated collectively by incognizant masses? The slime mold is a marvelous example [54]. *Dictyostelium discoideum* is a unicellular organism, an amoeba, that can mobilize itself based on local chemical cues. In the presence of food (bacterium), each amoeba acts ambivalent of others, like gas molecules randomly moving around, satisfying individual dietary needs, growing and dividing about every three hours. However, when food supply falls to critically low levels, starving amoebae show collective behavior. First, they aggregate into clusters with emanating branches. Thereafter, these clusters

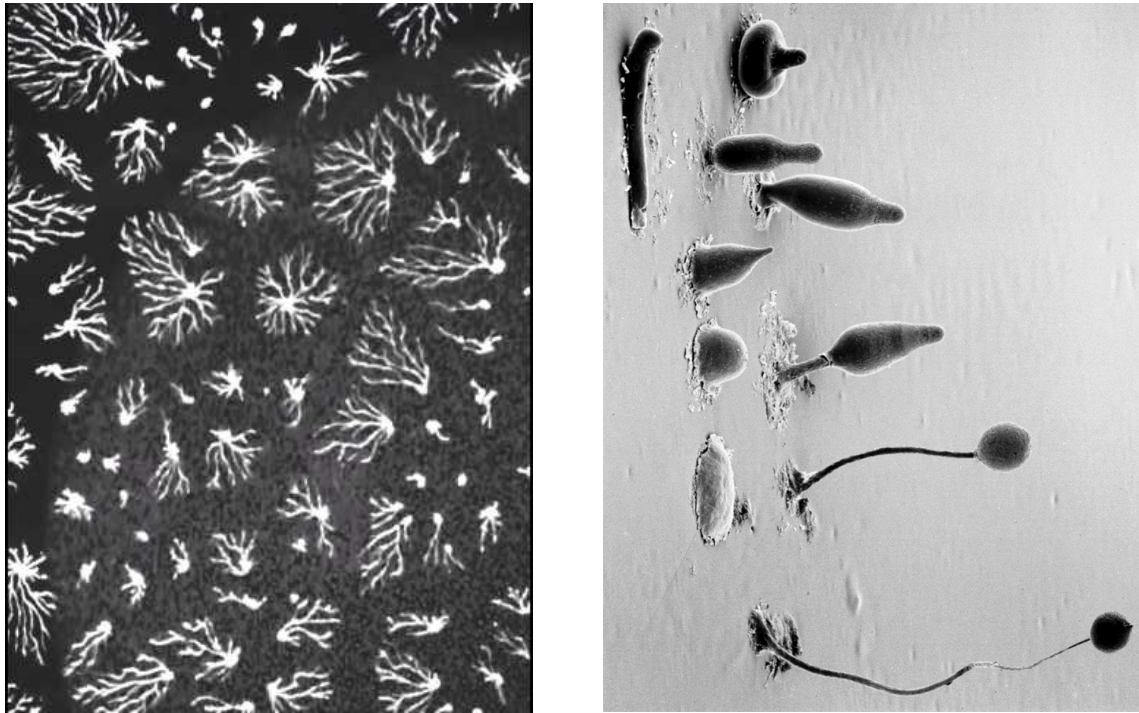


Figure 2.4: Slime mold aggregation and morphogenesis

organize into form of stalk and spores with 10,000-100,000 amoeba collectively generating an organism, as depicted in Figure 2.4. The stalk grows and at some point spores break out completing the cycle of collective behavior. The mechanism of this self-organization is unknown.

In prior examples the community set was closed. The constituents interacted locally and organized globally, but did not increase in number. The following examples of self-organization in nature are based on development where number of constituents increase resulting in system growth.

Per Bak sand pile In 1987, Per Bak, Chao Tang and Kurt Wiesenfeld introduced a simple model of sand pile to demonstrate self-organization of a system into critical state where the system showed self-similar patterns in space and time [55]. The abstract model is beautiful in its simplicity. Sand grains are dropped from a high place on to a pile, say at a single central point. The pile grows and steepens up. At some point, depending on friction of sand grains, the steepness reaches a threshold where grains topple down, creating an avalanche. The spatial organization of the pile is an intricate self-similar pattern, and the avalanche sizes in time would also exhibit a self-similarity. This is an open system which is pumped with matter

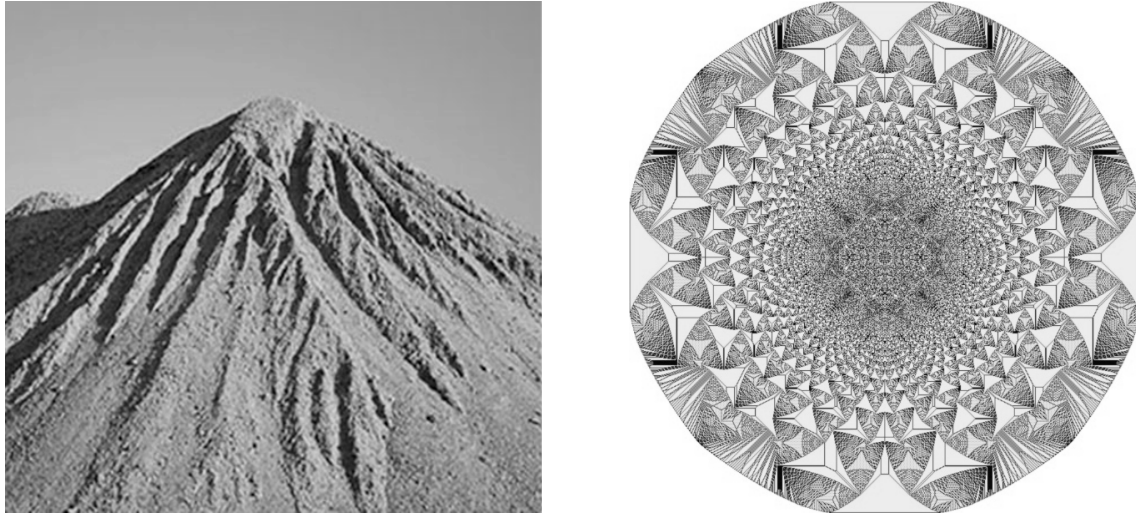


Figure 2.5: Sand pile pattern formation

(grain) and energy (dropping of grain from high up). Such dissipative systems tend to self-organize into complex spatio-temporal patterns as shown in Figure 2.5.

Snowflakes Snowflakes have fascinated artists and scientists as beautiful due to inherent simplicity of symmetries coupled with complexity that give the crystals life-like forms. Many positivists including Kepler, Descartes and Hooke sought scientific explanation to formation of snowflakes. Over the last decades, there have been volumes of theoretical work on the formation of snowflakes. Most recently, John Barrett from Imperial College London et.al [56] published their numerical simulations of snowflake growth in two and three spatial dimension that succeeded in generating various real-life morphologies of snow crystals. The development of a single snowflake starts with a minuscule dust particle nucleating the growth which then continues by aggregation due to condensation of vapor on the crystal surface. The aggregation dynamics depends on humidity and temperature as parameters set by the environment. The hexagonal formation is due to hexagonal close packing of water crystals due to angle between hydrogen and oxygen atoms in the water molecule. What causes the enormous variety of crystals is the slightest time variation of humidity and temperature of the environment as the crystal falls from the sky and keeps growing in the meanwhile. A snowflake's morphology is the signature of humidity and temperature time series on its developmental path.

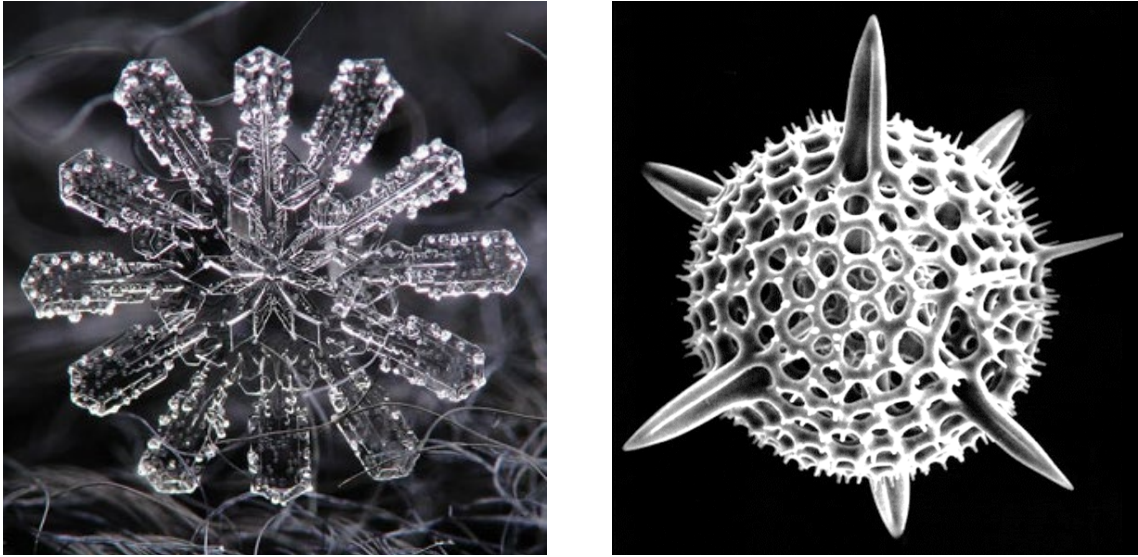


Figure 2.6: Driven aggregation in snowflakes and radiolaria skeletons

“The radiolarian skeletons... remind us of such things as snow crystals, rather than a collection of animals, constructed in accordance with functional needs.”

D’Arcy Thompson, On Growth and Form [10]

Radiolaria The radiolaria are unicellular organisms of diameter 0.1-0.2mm that have silica-based skeletons. They have been in existence throughout the oceans since the Cambrian period for over 500 million years. Radiolaria absorb silica from the marine environment and construct skeletons with intricate geometric patterns. Silica aggregation seems to be driven by external environmental parameters. The mechanism of skeletal self-organization of radiolaria is not well-understood, yet is reminiscent of snow flake formation as depicted in Figure 2.6.

Human embryogenesis Developmental biology is the study of the process by which animals and plants grow and develop. Embryogenesis is the phase of prenatal development involved in establishment of the body and organs of the individual starting from a fertilized egg. Embryogenesis, particularly the human one, has been well studied in biology. While what happens during embryogenesis is known to its minute details, theories on how it happens in terms of self-organizing mechanisms is still at its infancy and mainly centered about positional information theory introduced by Lewis Wolpert in his reference book *“Principles of Development”* [57].



Figure 2.7: Human embryo at eight weeks

With no explicit blueprint and no central coordination, how would cells acting as agents, dividing, signaling and differentiating, self-organize from the fertilized egg into a fully developed animal solely by mechanics of self-organization?

2.3 Emergent aesthetics

Ernst Haeckel has been an inspirational figure for this Thesis due to his joy and awe of observing forms of nature with an aesthetic and intellectual curiosity. In his monograph *“Das System Der Medusen”* (1879) on forms of jellyfish, he starts with:

“When I visited the ocean for the first time twenty-five years ago and, in August 1854, was introduced to the inexhaustible, wonderful world of marine life by my unforgettable master Johannes Muller, nothing exerted such a powerful force of attraction on me amongst the myriad animal forms, of which I had not seen living specimens until then, as the medusae. Never will I forget the delight with which I, as a twenty-year-old student, first observed Tiara and Irene, Chrysoara and Cyanea, and attempted to render with my paintbrush their splendid forms and colors.”

Ernst Haeckel, *Das System Der Medusen*

Haeckel was a contentious biologist, a strong advocate of Darwin at the period when Darwin was defending his theory, and an extraordinary artist dedicated to

reflecting the beauty of forms in nature with scientific rigor and creativity. As a scientist Haeckel introduced “*Biogenetic Law*” which stated that development of form of an individual in its lifetime (ontogeny) recapitulates the evolutionary accumulation of form in the lifetime of the specie (phylogeny). More than a century ahead of today’s theories on evolutionary development, Haeckel was postulating on developmental form being an expression of accumulated formatives on the evolutionary path. He is also the founder of “*Biological Theory of Cognition*” which was further developed by Konrad Lorenz and Karl Popper, where he countered Immanuel Kant’s a priori knowledge and stated that what seemed a priori knowledge for an individual (i.e. cognitive capability endowed to an individual before or without any interaction with the world) was a posteriori for its specie. Such cognitive capability being result of phylogenetically developed brain with again accumulated formatives on the evolutionary path [58].

In the same period as Darwin’s “*Origin of Species*” was published, Haeckel was working on a comprehensive classification of forms of living organisms with complexity reflecting phylogenic paths which to him showed the mechanics of evolution. Haeckel believed that knowledge of nature was not to be limited to experimentation or theoretical abstractions, but to be revealed through natural aesthetics. According to Haeckel, nature is self-organizing with an aesthetic urge. Haeckel was particularly intrigued by the symmetries of radiolarians on which he published a monograph based on items collected in the deep-sea English Challenger expedition. The diverse and complex forms of radiolarians is not necessarily due to survival benefits as these forms co-exist in similar environments. According to Haeckel, the diversity of forms is due to nature’s drive to explore aesthetic patterning niches. Haeckel’s illustrations are not depictions but investigations of forms in nature. Looking at this exhilarating diversity of form, even in a single-celled plankton floating in the oceans, one asks: What laws of nature is at work here? And why are they so beautiful? Haeckel maintained that nature worked with an intrinsic creativity, an artistic drive, as illustrated in his artwork from *Art Forms in Nature* (1904) [59] in Figure 2.8.

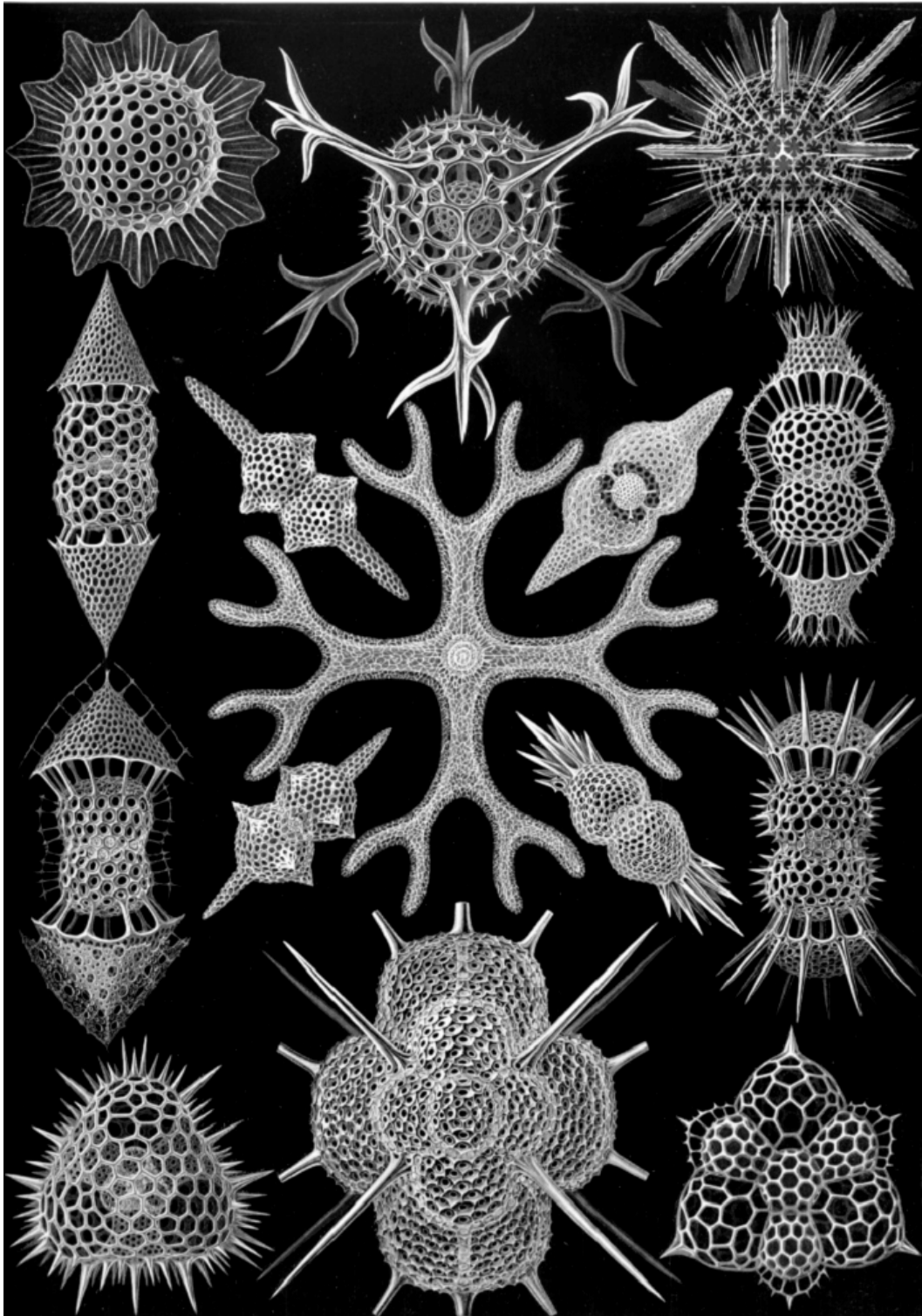
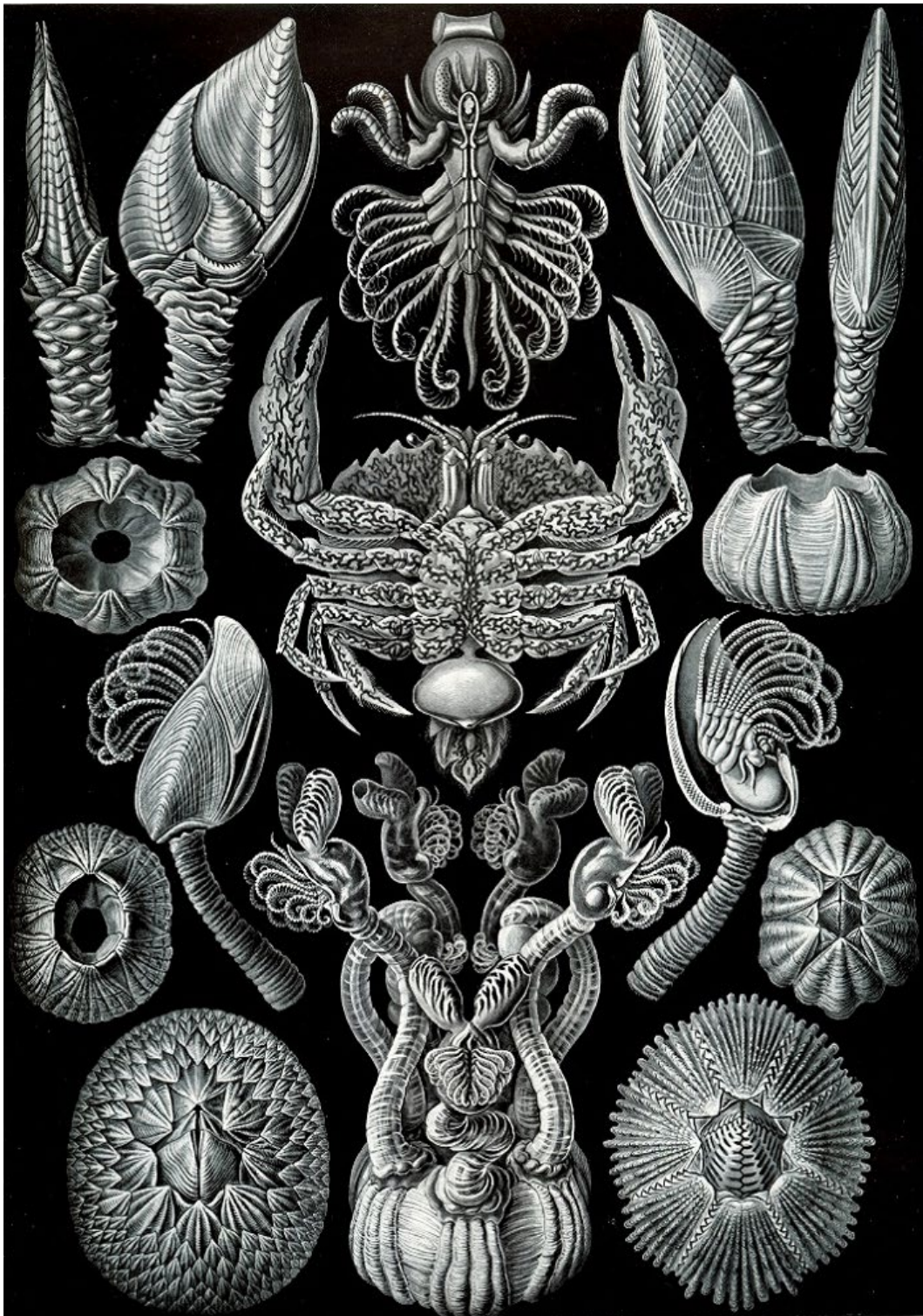


Figure 2.8: Art forms of nature – Ernst Haeckel



Art forms of nature – Ernst Haeckel

Chapter 3

Background on patterning

In the next chapters, we will investigate spatial self-organization in continuous space-time, in discrete space-time and in cellular automata which are expressed as increasing coarsening of state-space-time continuum starting from partial differential equations and ending in computational machines, yet all describing the common heuristic of spatial self-organization. Therefore mathematical domains that we will traverse are broad. Figure 3.1 provides a schematic on how these domains relate to each other. In this chapter, we will provide background on continuous state-space-time dynamical systems that exhibit patterning. None of the material provided in this chapter is novel or new. It is meant to provide a background, not a literature survey. A detailed review of the topic is available at “*Patterns and Waves*” (1991) [60].

3.1 Terminology

Point dynamical systems If dynamics is not defined in space but solely in time, we call it a *point dynamical system*. In these systems, time-evolution of state variables defined through ordinary differential equations where the incremental state change $\frac{d\mathbf{S}}{dt}$ is stated as a function φ of the current state values and possibly time.

$$d_t\mathbf{S} = \varphi(\mathbf{S}, t) \tag{3.1}$$

If φ is time independent i.e $\varphi(\mathbf{S}, t) = \varphi(\mathbf{S})$, then the system is said to be *time invariant*. If φ is linear, time evolution (i.e *solution*) is analytically tractable and

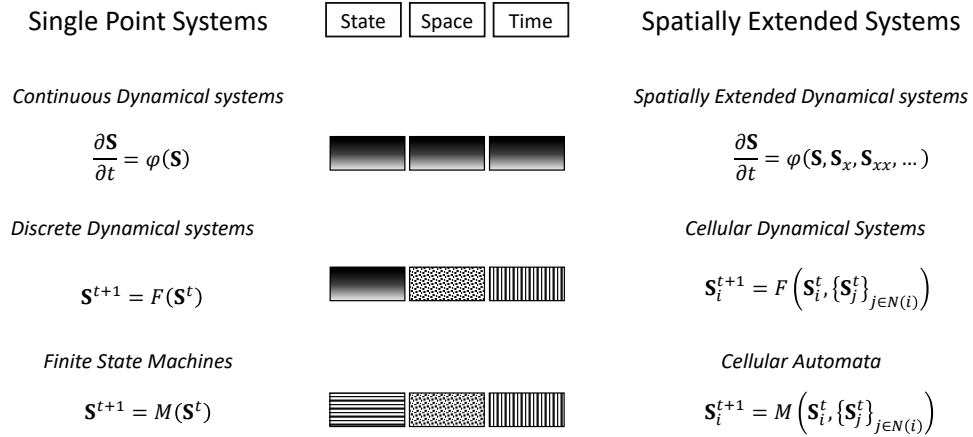


Figure 3.1: Dynamical systems overview

Point dynamical systems evolve in time dimension only, whereas spatially extended dynamical systems evolve in space-time. Depending on whether space-time or state variables are expressed in continuous or discrete terms, the mathematical domain ranges from partial differential equations to discrete dynamical systems to computation theory.

typically gets expressed as a linear combination of known function sets such as Fourier series. If φ is non-linear, analytical solution becomes difficult, if not impossible. We then start looking into long-term behavior of the system rather than exact expression of its time evolution.

Starting from an initial state, the dynamics evolves in time towards a fixed point, limit-cycle, or a chaotic orbit. These are called *attractors* of the dynamical system. Depending on which initial state the system starts from, it may end up in different attractors. Such set of states that converge to a given attractor is called its *basin* of attraction.

Then one asks how this long-term behavior of the system changes as we tweak some of its defining parameters. Most the time a small tweak results in no fundamental change to behavior, but then at some critical values, the system's long-term behavior shows a radical change. This is called a *bifurcation*.

Spatially extended systems Extending the dynamics in space where evolution is both in time and space becomes unwieldy very quickly. In continuous space-time, the dynamics is described in partial differential equations, most interesting and realistic ones being non-linear. The incremental change in time of the system state

at a given location in space $\frac{\partial \mathbf{S}(\mathbf{x})}{\partial t}$ is stated as a function φ of the current values of the state variables $\mathbf{S}(t)$ as well as its local variation in space, typically expressed as first and second order partial space derivatives (i.e. $\partial_x, \partial_{xx}$ if space is one dimensional, or in the case of a a general manifold, as gradient $\nabla \mathbf{S}$, and Laplacian $\nabla^2 \mathbf{S}$).

$$\partial_t \mathbf{S} = \varphi(\mathbf{S}, \partial_x \mathbf{S}, \partial_{xx} \mathbf{S}, t) \quad (3.2)$$

A subset of these systems, called parabolic partial differential equations, particularly the linear forms of it, lend itself to some analysis, and can be effectively used as models for many real physical, chemical, biological and economical systems. Reaction-diffusion and advection systems, as we will introduce below, have been widely studied for pattern formation and chemotaxis properties, respectively.

Diffusion The simplest spatially extended system showing collective patterning (albeit a homogeneous one) is diffusion phenomenon:

$$\begin{aligned} \partial_t U &= \partial_{xx} U \\ \partial_t U &= \nabla^2 U \end{aligned} \quad (3.3)$$

where $U = U(\mathbf{x}, t)$ is a state variable representing the density or concentration of some entity in space $\mathbf{x} \in \Omega \subseteq \mathbb{R}^3$ and $t \in \mathbb{R}^+$. With the initial distribution specified $U(\mathbf{x}, 0) = U_0(\mathbf{x}), \forall x \in \Omega$, and boundary condition possibly imposed by prescribing $u(\partial\Omega, t)$ on the boundary of Ω space domain, the behavior of diffusion equation can be stated in terms of known functions, such as Fourier transform. We are capable of analytically finding the solution because of the linearity of the diffusion equation. As we shall see reaction-diffusion equations in general lack this property.

Reaction-Diffusion Let us introduce the reaction term, f , on to the diffusion equation.

$$\partial_t U = \nabla^2 U + f(U) \quad (3.4)$$

with $f(u) = aU + bU^2 + cU^3$. Due to non-linear form of reaction f , it is not possible to find closed form or as series expansions. However, it is possible to study its behavior and represent the form of solutions.

Generalized Reaction-Diffusion-Drift The reaction function f of Equation 3.4 is space invariant (i.e. isotropic) and time independent. We can generalize it to be anisotropic and time dependent, and also expand it to be a function of the gradient of state variable. This generalized reaction-diffusion, given in Equation 3.5 can generate morphogenesis as we will show in the following chapters:

$$\partial_t U = \nabla^2 U + f(U, \nabla U, \mathbf{x}, t) \quad (3.5)$$

Balance Laws The reaction-diffusion equations are used for modeling dynamical systems in physics, chemistry, biology, and economics where spatio-temporal behavior of particle populations consisting of molecules, cells, or individuals, are represented with time evolving density functions, describing the spatial distribution of such populations. With this characterization, the balance laws state the following:

The change in number of particles in a region is the net creation due to reaction within the region plus net inflow of particles into it.

Let $U(\mathbf{x}, t)$ be the particle density function and $Q(\mathbf{x}, t, ..)$ be the net creation rate of particles at location $\mathbf{x}$ at time t . Let $\mathbf{J}(\mathbf{x}, t, ..)$ be the flux density, i.e. for any unit vector $\mathbf{n} \in \Omega$, the dot product $\mathbf{J} \cdot \mathbf{n}$ is the net rate at which particles cross a unit area in a plane perpendicular to $\mathbf{n}$. Then the balance laws have the form:

$$\partial_t U = -\nabla \cdot \mathbf{J} + Q \quad (3.6)$$

For a given model, be it chemical, biological etc, we can define the spatio-temporal dynamics by specifying local net creation function Q and local net flux function $\mathbf{J}$. For example, if the flux is proportional to the gradient of concentration (i.e. particles flow from high to low concentration), stated as Fick's law, i.e $\mathbf{J} = -D\nabla u$, with no local creation or destruction of particles, we then get $\partial_t u = -\nabla \cdot \mathbf{J} = D\nabla^2 U$ which is the simple diffusion dynamics, where $D > 0$ is the diffusion constant.

3.2 Reaction-diffusion

Certain reaction-diffusion systems exhibit stationary spatial inhomogeneous solutions in the long-term, called patterns. These are typically generated by coupled systems of more than one type of particles, such as catalytic chemical systems. Pattern formation can be analyzed, albeit locally, by bifurcation of trivial uniform equilibrium into stable patterned equilibria. The concept of spontaneous pattern generation by reaction-diffusion systems was introduced as a possible mechanism of morphogenesis (e.g. skin pattern formation in animals) by Alan Turing in 1952 [1].

Consider the following system of two coupled chemicals, represented as a state vector $\mathbf{S}(\mathbf{x}, t) = [U(\mathbf{x}, t), V(\mathbf{x}, t)]$:

$$\begin{aligned}\partial_t \mathbf{S} &= D \nabla^2 \mathbf{S} + \varphi(\mathbf{S}) & \mathbf{x} \in \Omega, \, t > 0 \\ \nabla U \cdot \mathbf{n} &= 0 & \mathbf{x} \in \partial\Omega, \, t > 0 \\ \nabla V \cdot \mathbf{n} &= 0 & \mathbf{x} \in \partial\Omega, \, t > 0\end{aligned}\tag{3.7}$$

where Ω is the spatial domain of reaction-diffusion system, what we call the tissue. $\partial\Omega$ is the boundary of the tissue, and $\mathbf{n}$ is the spatial vector that points perpendicular to the boundary, i.e. boundary normal vector. $\nabla U \cdot \mathbf{n} = 0$ means, the chemical has no flux leaving the tissue. This is the second-type boundary condition known as Neumann boundary condition in partial differential equations. since each of the two chemicals have their own diffusion constants, D_U, D_V , we notate D as a positive definite diagonal matrix where as $D = \text{diag}(D_U, D_V)$ and $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is the reaction function, specified by two individual functions $\varphi(\mathbf{S}) = [f(U, V), g(U, V)]$.

Now assume that $\mathbf{S}^*$ is spatially homogeneous equilibrium with $\varphi(\mathbf{S}^*) = 0$ and ask if inhomogeneous equilibria may bifurcate from $\mathbf{S}^*$ as the diffusivities D_U and D_V of the chemicals are varied. Clearly, when $\mathbf{S}^*$ is a stable solution to the ordinary differential equation $\partial_t \mathbf{S} = \varphi(\mathbf{S})$, then it is also a spatially homogeneous solution to Equation 3.7. But is it a stable solution or not? We will show that, so-called *Turing instability* may arise, depending on the reaction function, for certain values of diffusivities D_U, D_V , resulting in patterning. To catch this bifurcation point where the spatial attractor of the system moves from uniformity to patterned, we perform a linear analysis.

Linear instability analysis First we ignore diffusion and look into a simple coupled reaction that has no spatial dynamics. We start with the assumption that reaction system is stable for some reactant concentration levels (U^*, V^*) and approximate the reaction equation with linear version around such stability point by taking derivatives (in state, not space) of the reaction function, expressed as the Jacobian matrix, $\mathbf{J} = \begin{bmatrix} f_U & f_V \\ g_U & g_V \end{bmatrix}$, evaluated at the stability point $\mathbf{S}^* = (U^*, V^*)$. For small deviations $\delta\mathbf{S}$ of system state around the stability point, the linear PDE of the evolution for such small deviation is given as:

$$\partial_t \delta\mathbf{S} = \mathbf{J}_{|U^*, V^*} \delta\mathbf{S} \quad (3.8)$$

The linearized system (without diffusion) given in Equation 3.8 is stable at its zero point if its eigenvalues of the Jacobian are both negative, which means their sum must be negative and multiplication positive. This is the Trace and Determinant of the Jacobian matrix, respectively:

$$\text{Trace } \mathbf{J}_{|U^*, V^*} = f_U + g_V < 0 \quad \text{Condition 1} \quad (3.9a)$$

$$\text{Det } \mathbf{J}_{|U^*, V^*} = f_U g_V - g_U f_V > 0 \quad \text{Condition 2} \quad (3.9b)$$

Now we introduce diffusion to the system at this reaction equilibrium point:

$$\partial_t \delta\mathbf{S} = \mathbf{J}_{|U^*, V^*} \delta\mathbf{S} + \mathbf{D} \nabla^2 \delta\mathbf{S} \quad (3.10)$$

where $\mathbf{D} = \begin{bmatrix} D_U & 0 \\ 0 & D_V \end{bmatrix}$, is the diagonal matrix of diffusion coefficients. The spatial solutions to the linearized system (with diffusion) given in Equation 3.10 can be represented as linear combination of spatial harmonics the form $\delta\mathbf{S} = e^{j\omega x} q(t)$ which leads to $\nabla^2 \delta\mathbf{S} = -\omega^2 \delta\mathbf{S}$. We can rewrite the linearized reaction-diffusion system, at and around the stability point U^*, V^* as:

$$\begin{aligned} \partial_t \delta\mathbf{S} &= \mathbf{J}_{|U^*, V^*} \delta\mathbf{S} - \mathbf{D} \omega^2 \delta\mathbf{S} \\ &= \tilde{\mathbf{J}}_{|U^*, V^*} \delta\mathbf{S} \end{aligned} \quad (3.11)$$

As we can see the introduction of diffusion modified the Jacobian of the linear

approximate system from $\mathbf{J}$ to $\tilde{\mathbf{J}} = \mathbf{J} - \mathbf{D}\omega^2$. For Turing instability to onset, we need the Trace of the modified Jacobian to become positive, or its Determinant to turn negative. This would cause spatial ripples to grow exponentially, deeming homogeneous state unstable.

$$\text{Trace } \tilde{\mathbf{J}} = \text{Trace } \mathbf{J} - (D_U + D_V)\omega^2 \quad (3.12)$$

Since diffusion constants are positive, and because $\text{Trace } \mathbf{J} < 0$ to start with, $\text{Trace } \tilde{\mathbf{J}} < 0$. So we look into Determinant of $\tilde{\mathbf{J}}$ becoming negative as a possible condition Turing instability:

$$\text{Det } \tilde{\mathbf{J}} = q(\omega^2) = \text{Det } \mathbf{J} - \omega^2(D_U f_U + D_V g_V) + \omega^4 D_U D_V < 0 \quad (3.13)$$

This is a quadratic equation in ω^2 . We note that for $\omega = 0$ (i.e. homogeneous spatial function) we had $q(0) = \text{Det } \mathbf{J} > 0$. Therefore for $q(\omega^2)$ to become negative for some ω , we need the roots of $q()$ to be real, which leads to:

$$\frac{f_U}{D_U} + \frac{g_V}{D_V} > 2\sqrt{\frac{1}{D_U D_V} \text{Det } \mathbf{J}} \quad \text{Condition 3} \quad (3.14)$$

We observe from Equation 3.14 that diffusion effectively *suppresses intensity of the reaction* for each reactant by its diffusion constant ($f_U \rightarrow \frac{f_U}{D_U}$, and $f_V \rightarrow \frac{f_V}{D_V}$). We know from Condition 1 (stability of non-diffusive system) that $f_U + g_V < 0$, so for Condition 3 to hold we observe that either f_U or g_V had to be positive. And further, whichever is negative, its diffusivity has to be larger. This simple linear analysis gives a fundamental insight to pattern formation dynamics of Turing-like systems with two reactants:

Heuristics of Turing patterning The reaction should be expansive in one reactant $f_U > 0$ and contractive in the other $f_V < 0$. The diffusion should be slow in the expansive reactant and fast in the contractive one.

Case Study: Gray-Scott Model The Gray-Scott model represents a chemical reaction of two reactants $\mathcal{U}$, and $\mathcal{V}$, with $\mathcal{P}$ being an inert by-product:



The concentrations are shown with state variables u and v . The Gray-Scott model is governed by the following reaction-diffusion equations:

$$\mathbf{f} = \mathbf{f}(U, V) = \begin{bmatrix} -UV^2 + F(1 - U) \\ +UV^2 - (F + k)V \end{bmatrix} \quad (3.16)$$

- UV^2 is the reaction term: One $\mathcal{U}$ molecule needs to combine with two $\mathcal{V}$ molecules to convert it from $\mathcal{U}$ to $\mathcal{V}$. Given that U, V are density functions, the intensity of this reaction scales with UV^2 .
- F is the external *feed* rate of chemical $\mathcal{U}$ with saturation at level 1 captured by $1 - U$ term.
- $(F + k)$ is decay rate of chemical $\mathcal{V}$ with k representing the *kill* rate above and beyond the *feed* rate.

Gray-Scott model is a simple reaction-diffusion system that exhibits a wide variety of patterns depending on parameter set which has been widely studied starting with Pearson and Lee's work in 1993 [61], [62]. Figure 3.2 shows generation of different patterns on a 2D substrate with varying parameters (x-axis: F , y-axis: k) with further selected pattern details provided in Figure 3.3. In order for patterning to occur, the diffusion speed of $\mathcal{U}$, D_U has to be much faster than the diffusion speed of $\mathcal{V}$, D_V . This allows $\mathcal{U}$ to *disseminate local information* as a fast messenger, whereas slow diffusion of $\mathcal{V}$ sets the characteristic size of the patterns.

Deolman *et al.* [63] provided a detailed analysis of pattern formation of Gray-Scott reaction-diffusion system in one-dimensional substrate. In addition to a wide range of periodic steady-state spatial patterns, they have shown the existence of single-pulse and multi-pulse stationary waves.

As a case study, we investigate the onset of Turing instability for Gray-Scott system using linear analysis. The non-diffusive system has a trivial zero at $(U, V) =$

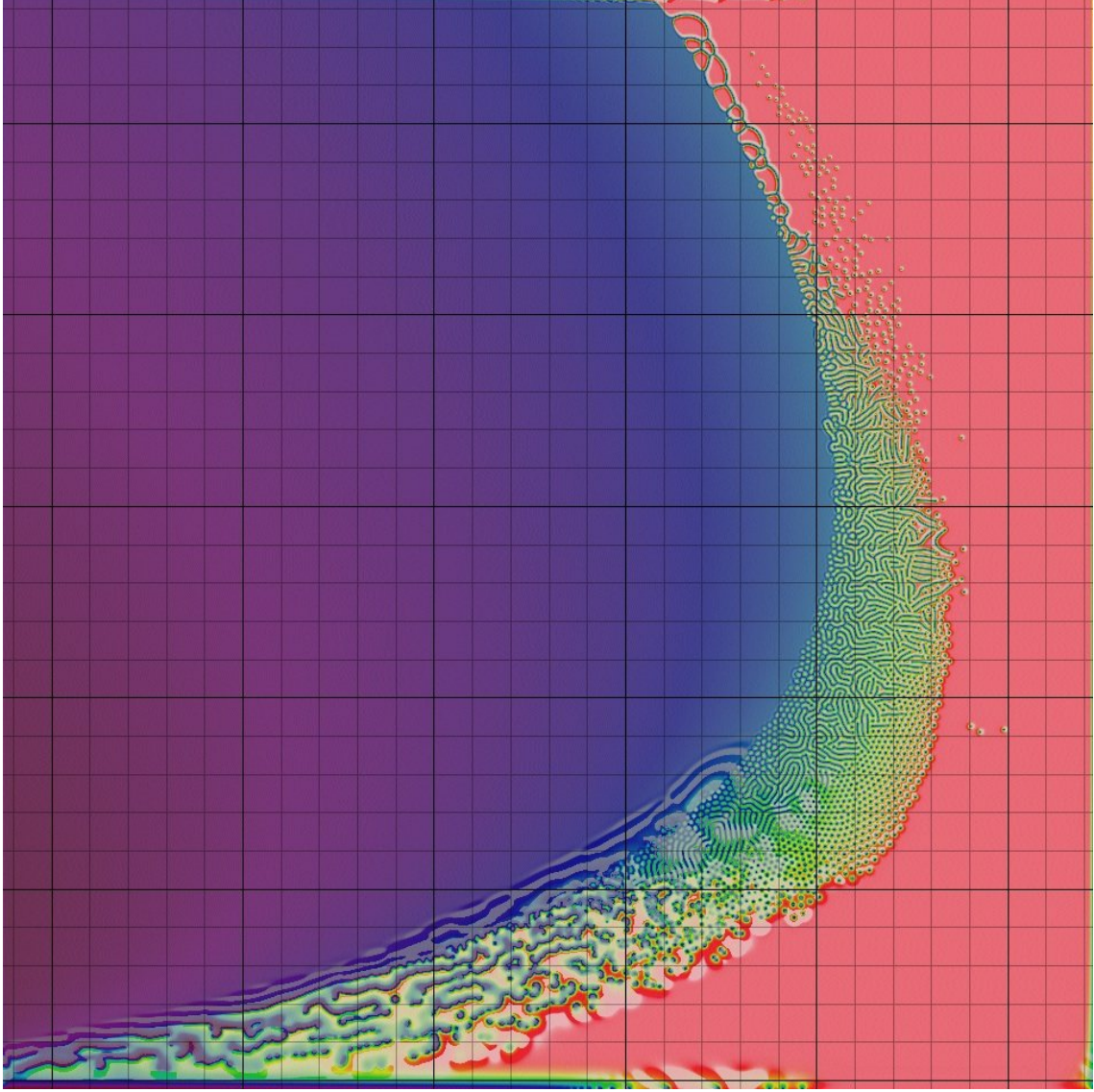


Figure 3.2: Patterning in Gray-Scott system

x-axis : $k : (0.02, 0.07)$, y-axis $F : (0.00, 0.12)$, relative diffusivity $D_U/D_V = 2$. Source : www.mrob.com

$(1, 0)$, it also has the zeros at:

$$(U^*, V^*) = \left(\frac{1-a}{2}, \frac{F}{F+k} \frac{1+a}{2} \right) \quad (3.17a)$$

$$(U^{**}, V^{**}) = \left(\frac{1+a}{2}, \frac{F}{F+k} \frac{1-a}{2} \right) \quad (3.17b)$$

provided that $\frac{(F+k)^2}{F} < \frac{1}{4}$; where $a = \sqrt{1 - 4\frac{(F+k)^2}{F}}$. The Jacobian of the non-diffusive system is:

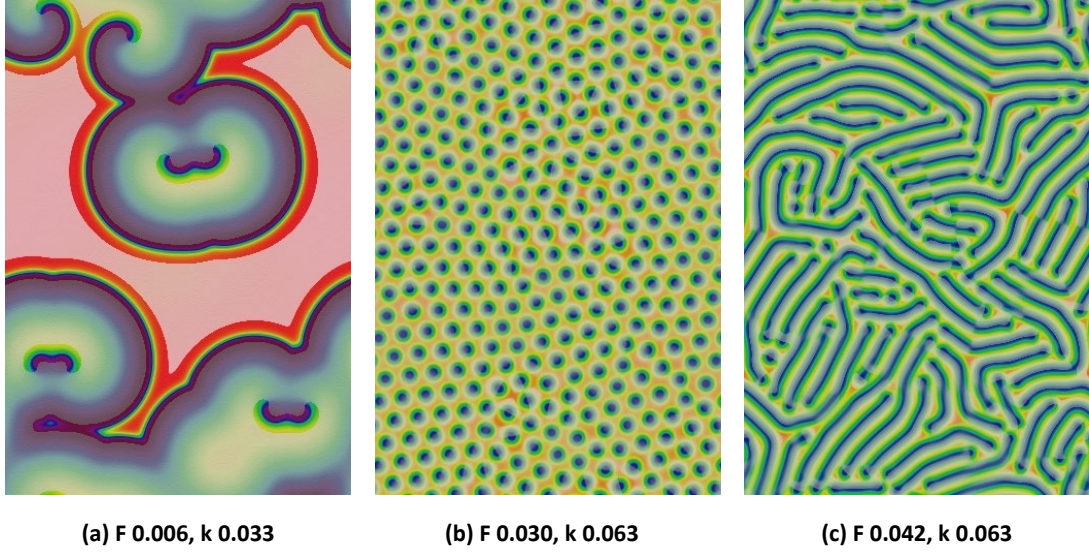


Figure 3.3: Selected patterns generated by Gray-Scott system

(a) shows oscillatory spatiotemporal patterns, reminiscent of Belousov-Zhabotinsky chemical reaction. (b) shows spatially periodic dots, reminiscent of Bénard convection cells and (c) shows Turing-like pattern formation. Source : www.mrob.com

$$\mathbf{J} = \begin{bmatrix} -V^2 - f & -2UV \\ V^2 & 2UV - (F + k) \end{bmatrix} \quad (3.18)$$

At the trivial zero of $(U, V) = (1, 0)$, the system is stable without diffusion. But since both f_U and g_V are negative at the trivial point, Condition 3 of Equation 3.14 can never hold, i.e. diffusion instability can not be induced for the trivial point. We look into the other two zeros given at Equation 3.17. We calculate for a range of system parameters $F, k, D_U/D_V$ where Turing instability conditions are fulfilled, the results are shown in Figure 3.4. The linear stability analysis matches closely the patterning regions observed in simulations given in Figure 3.2.

In Chapter 4, we will introduce a novel simple model that captures the heuristics of Turing patterning which is an interplay between expanding in one reactant and contracting in the other at widely different diffusion rates and analytically calculate typical pattern size formed by the system.

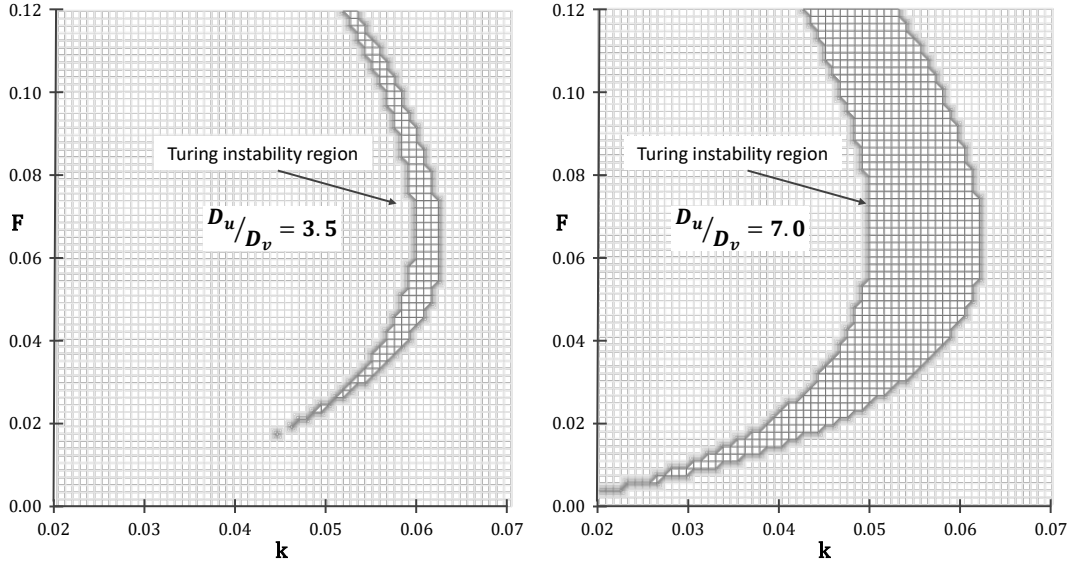


Figure 3.4: Turing instability for Gray-Scott model in parameter space

As relative diffusivity is increased Turing instability condition is satisfied for a wider range of parameters: Left chart $D_U/D_V = 3.5$, right chart $D_U/D_V = 7.0$

3.3 Advection

The pattern formation in the previous section was governed by reaction-diffusion dynamics where diffusion was given as divergence of the gradient of the state variable, i.e. $\nabla^2 U \equiv \nabla \cdot \nabla U$, and further ∇^2 was incorporated linearly into the model equations. We now expand the concept by allowing diffusion to be driven of an arbitrary gradient field, and we also allow non-linear dispersion. This opens up a whole new set of possible behaviors to be modeled, including advection (drift). Up-gradient drift for example can explain aggregation of particles in a single location which is observed in many biological phenomena such as clustering of slime molds mentioned in Chapter 2. Consider the following simple, yet somewhat misleading, example. Let say a population of insects with density function $U(\mathbf{x}, t)$ *swarm* according to the following rule:

$$\partial_t U = \epsilon \nabla^2 U - \partial_x(UW) \quad x \in \mathbb{R}, t \geq 0 \quad (3.19a)$$

$$W = \int_x^\infty U(y) dy - \int_{-\infty}^x U(y) dy \quad (3.19b)$$

The velocity field W cause individuals to drift towards the center of mass of the population. It is a bit misleading because velocity field is not a local function but a kernel across full space, but gives the idea of aggregation through drift.

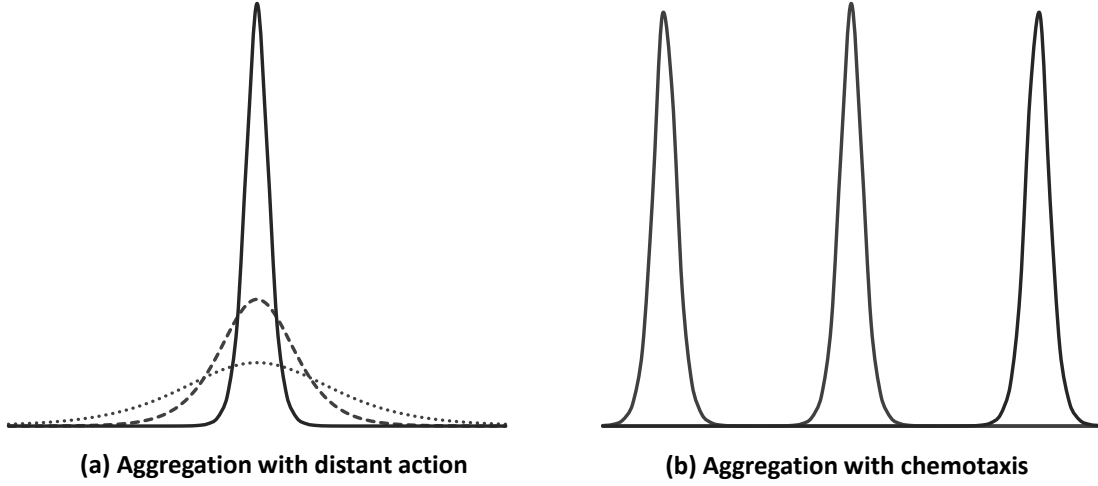


Figure 3.5: Aggregation pattern generated by advection

Gradient field formed by mass attraction at distance (a) vs chemotaxis (b)

The following chemotaxis model was introduced by Keller and Segel in 1971 [64]. It represents the aggregation dynamics of motile cells (with concentration U) that swim up on a chemo-attractant potential C . The chemo-attractant is itself generated by the cells, resulting in the following coupled partial differential equations:

$$\partial_t U = M \nabla^2 U - a \nabla \cdot (U \nabla C) \quad (3.20a)$$

$$\partial_t C = D \nabla^2 C + bU/(U + h) - \mu C \quad (3.20b)$$

$a \nabla \cdot (U \nabla C)$ is the *drift* term; U moves up the gradient field ∇C . M and D are the diffusion rates of cells and the chemo-attractant. $bU/(U + h)$ is *secretion* term, with $b, h > 0$ defining a saturation curve. μC is the decay rate of the chemo-attractant.

As we have seen in linear stability analysis of Gray-Scott model, in order for patterning to onset, chemo-attractant needs to diffuse much faster than the cell diffusion. In a way, the chemo-attractant is a messenger of the concentration of cells and such messenger has to travel much faster than the cells themselves so that it will quickly form the potential on which cells then start migrating up-gradient.

In Chapter 4, we will introduce our novel Reaction-Diffusion-Drift model that generates singular peak across the full space and develop an upper bound on expected time to such singular peak formation for a given space size, starting from a random initial system state.

Chapter 4

Cellular dynamics

In the previous chapter, we provided a brief background on patterning in continuous space-time expressed in partial differential equations. In this chapter, we discretize time, cellularize space and describe the dynamics as an iterative system. We will introduce three novel models: the spatial periodic attractor, singular attractor and dipolar attractor. We will demonstrate that self-organization of an embryo into various animal body plans are achievable with dipolar attractor which self generates positional information.

4.1 Terminology

Partial differential equations are used for describing the evolution of continuous state variables (such as heat) on a continuous space (such as a metal rod) over continuously flowing time. These were the type of equations we used in the previous chapter. In this chapter, we will use *discrete space* (such as a school building made of classrooms) and *discrete time* updates (such as students changing classrooms only at ring of the bell). We will keep the state variable continuous until the very last section at which we will limit the state variable to take on values from a *finite state* list (such as the number of students in each classroom). The reason we use discrete systems in our work is three fold: Analytical treatment is easier; computer simulations are feasible; and after all, biological tissue is inherently spatially discrete as it is made of cells. We use the term *cellularization* in referring to discretizing space which resonates better with biological realm.

To transfer the dynamics expressed in continuous world to a discrete one, there are well developed techniques in the vastly studied field of *numerical analysis*. With the advances in computational power, today almost all engineering calculations are done using numerical methods such as static calculations for a skyscraper or aerodynamic calculations of a jet. The three methods used in discretizing continuous systems are *finite difference method*, *finite element method* and *finite volume method*, which are described in the coming up section.

We use the terminology *manifold* to refer to a 2D curved surface embedded in 3D. We call the cellularized manifold a *mesh graph*. The mesh graph consists of *nodes* (also called vertices in literature) and *edges* connecting the nodes. Each node represents the location of a cell's center, and each edge connecting two nodes represents the existence of a boundary between two cells with the corresponding two nodes.

Multivariable calculus on a manifold is concerned with differentiation and integration of fields defined on the manifold. *Scalar field* on a manifold assigns a scalar value to each point on the manifold. Vector field on a manifold assigns a vector that lies on the manifold, to each point on the manifold. Key operators that measure rate of change of the fields as we move on the manifold are the following: *Gradient* operator acting on a scalar field gives the maximal change of that scalar field, as a vector pointing in that maximal change direction. *Divergence* operator acting on a vector field gives the net amount of vector emanating at any point. *Curl* operator acting on a vector field gives the total amount of twisting that the vector field has at any point. *Laplacian* operator acting on a scalar field measures the divergence of its gradient, or differently said the total amount of curvature of the scalar field at any given point. When we transfer our pattern forming dynamical systems from continuous manifold to a cellular world, we also need to transfer the operators of calculus. The framework of such transfer is provided in this chapter.

State variables are observables which we define the dynamics of. For example, heat at any given point on a rod would be a state variable. Typically the systems we investigate have more than one state variable, for example, the concentration of red and green dyes on a piece of fabric. In the case of two state variables, one can visualize the state of the system, at a given point on the fabric, using a 2-

dimensional depiction where one axis shows the concentration of red dye and the other axis the concentration of green dye. This 2-dimensional space where each red-green combination color is represented as a dot is called the *state space*, not to be confused with the spatial space. If we have reaction between the two colors (say red annihilates green, but green is self generated in the meanwhile), then this reaction would cause the red-green concentrations to change over time, i.e. the dot on the 2D state space would move around, its path defined by the reaction equation. If we plot little arrows at each dot to specify where that dot would move to, given the reaction equation, we would get a visual depiction of the reaction dynamics, called the *phase portrait*.

If we were to take into account the states of all points (or nodes, or cells) of the system as an ensemble, we get the *system state* or *configuration state*. Let us consider two non-reacting dyes of red and green color and assume that the fabric they are deployed on is diffusive, say water on a petri dish. Now clearly, no matter where you drop your initial dyes, they would diffuse over and result in a homogeneous light red and light green throughout the dish. The *initial system state* (or *initial configuration state*) is where the dyes are deployed at the beginning. The final system state is called the *attractor*. The attractor in this example is a homogeneous color across the tissue, not oscillating in time.

The core of pattern generation is to define systems with dynamics such that the attractors are prescribed patterns or shapes. To be able to prescribe a shape drawn on a tissue as the canvas, we need to be able to pin point some reference poles, and then be able to make each location know its distance to such reference poles. Therefore, a dynamical system where the attractor is a singular pole is fundamental for shape generation. In this chapter, we will introduce our novel *singular attractor* dynamics that generates a singular peak concentration starting from almost any initial system condition.

In our work we use the terminology *morphogen concentrations* to refer to state variables like the concentration of dyes mentioned above. The biological relevance of morphogens will be addressed in Chapter 6. In the singular attractor model, we use two morphogenes that are coupled through a reaction equation, called the *source* morphogen and the *field* morphogen. The field terminology used herein should not

be confused with vector and scalar fields of calculus.

Our analysis of singular attractor dynamics, and its extension to dipolar attractor, will be performed on a *linear cyclic tissue* which is a ring made of cells connected one to the other. Segmentation and body plan are terminologies from biology that we use in shape generation. *Segmentation* is differentiation of cells at particular pin points to generate segments in an organ (for example five fingers are the segments). Vertebrates have two main *body axis* (head-foot and front-back, or more technically dorso-ventral and anterior-posterior axis). *Body plan* is the overall location of where main limbs reside with regards to main body axis. In this chapter, we will demonstrate dynamics where the attractors are body plans with prescribed number of segments such as insects with three leg pairs or arachnics with four leg pairs, self-organized at precise locations.

When we discretize state space to a list of predefined states, we move away from biology to the realm of computation. There, our cellularized dynamics becomes *cellular automata* and our singular attractor dynamics becomes equivalent to *leader election* algorithms of distributed computing.

4.2 Discretization methods for PDEs

In the previous chapter, pattern forming reaction-diffusion systems were described using partial differential equations (PDE) in space $\Omega \subset \mathbb{R}^3$. In general, time-space solutions to these non-linear PDEs can not be found analytically. Hence, the problem in hand quickly becomes that of engineering domain where numeric approximations is sought taking into account the nature of the problem and computational resources. Engineers face the same problem of numerically approximating non-linear PDEs in vast array of traditional fields such as structural analysis, heat transfer, fluid dynamics, mass transport, and electromagnetics. Given the engineering applications ranging from aerospace, construction, telecommunications to weather forecast, great progress has been made in numerical analysis of non-linear PDEs. There are three methods used today for numerically approximating non-linear PDEs. *Finite element method*, *finite difference method* and *finite volume method*. As the word *finite* common to three methods imply, the goal is to convert the continuous space-time

description of the natural phenomena expressed as PDE into a discrete domain of finite items. In this Thesis, finite volume method is used for reasons explained below.

Finite difference method (FDM) is the most direct approach to discretizing partial differential equations on a regular grid where differential equations are replaced with difference equations. Given the simplistic nature it is suitable for large-scale massively parallel simulations on supercomputers with applications in meteorological and seismological simulations, but not suitable for simulations over shapes that require not regular meshes.

Finite element method (FEM) is a computational method that subdivides the domain Ω into small but finite-sized simple shapes, typically polygons. The collection of all these simple shapes constitutes the finite element mesh. The variables of PDE are approximated with linear combinations of so-called *coordinate functions*, typically polynomials. The global model of the problem is generated by stitching the elements to depict a given domain. [65]. Introduced in early 1960s, finite element method is widely used in engineering applications ranging from structural analysis to electromagnetics. It has a strong mathematical theory. One reason for the method's success is that it is possible to increase the number of the elements to increase the approximation locally, selectively. This technique is often critical, for example, in the case of accurate stress analysis at the corners of a mechanical piece. However, the mathematics behind the finite element method is quite advanced. In comparison, the implementations of finite difference and finite volume methods are more straightforward.

Finite volume method (FVM) is similar to the finite element method in that the domain is first divided into small but finite sized elements of geometrically simple shapes. Apart from this, the finite-volume method is very different from the finite element method, beginning with the concept of elements, which are instead referred to as cells. The finite volume method is based on the fact that many physical laws are conservation laws — what goes into one cell on one side needs to leave the same cell on another side, or increase the stock in the cell. Following this idea, we end up with a formulation that consists of flux conservation equations defined in an

averaged sense over the cells. Historically, this method has been very successful in solving fluid flow problems. Efficient technology for computational fluid dynamics with the finite element method has become increasingly popular over the last 10 to 15 years.

The finite volume method's strength is that it only needs to do flux evaluation for the cell boundaries. This also holds for nonlinear problems, which makes it extra powerful for robust handling of nonlinear conservation laws. The local accuracy of the finite volume method, such as close to a corner of interest, can be increased by refining the mesh around that corner, similar to the finite element method. However, the functions that approximate the solution when using the finite volume method cannot be easily made of higher order. This is a disadvantage of the finite volume method compared to the finite element method.

Finite volume element method (FVE) More recently [66], Cai *et al.* introduced a combination of finite volume and finite element methods, called finite volume element method which uses volume integral formulation of the problem with a finite partitioning set of volumes to discretize the equations, as standard in FVM, and then restricts the admissible functions to a finite element space to discretize the solution, following FEM approach.

We use finite volume method in our work because (i) it is more suitable to handle non-linearity in PDEs, (ii) our formulations of pattern forming morphogen dynamics are inherently conservation laws, (iii) our goal of simulating morphogenesis requires us to handle domains of arbitrary shapes (such as the surface of a jellyfish), and (iv) the development of the embryonic form requires the domain to expand and morph, therefore the domain needs to be dynamic. The cellularization approach of finite volume method suits very well to the cellular nature of the organism. In fact formulating pattern formation using continuous space partial differential equations is an abstraction. The actual phenomenon is done by cell-to-cell communication and cell differentiation, hence, the true nature of the problem at hand is cellular.

4.3 Discretization of time with FDM

Ordinary differential equations that describe evolution of dynamical systems can be discretized in time into difference equations. Such discretization schemes, called *finite difference methods* (FDM), were originally introduced by Euler in the later half of eighteenth century, and are widely used for solving non-linear differential equations numerically.

An ordinary non-linear differential equation that defines time evolution of a point system starting from an initial condition $U(0)$ is given as $\partial_t U = \varphi(U, t)$. The continuous-time dynamics can be approximated with discrete time intervals of Δt separation using difference equations. Let $u^0 = U(0)$ and $t^0 = 0$. Euler's forward method uses the iterative difference equation:

$$u^{k+1} = u^k + \varphi(u^k, t^k) \Delta t \quad (4.1a)$$

$$t^{k+1} = t^k + \Delta t \quad (4.1b)$$

The time discretization of dynamical system seeks to approximate the continuous time evolution by discrete time step:

$$U(t^k) \approx u^k \quad (4.2)$$

Error bound If φ is Lipschitz continuous, then the approximation error is bounded, and can be reduced linearly by increasing the granularity of time discretization [67]. A function $f(x)$ is called Lipschitz continuous in a domain $D \subset \mathbb{R}$ if for any $x_1, x_2 \in D$, there exists a finite constant L such that $|f(x_1) - f(x_2)| \leq L|x_1 - x_2|$. Whilst the forward Euler method is simple and explicit in calculation, its approximation error reduces only linearly with Δt and error rate can grow exponentially with time. This can be improved by higher-order approximation of f . There is also the Euler's backward method: $u^{k+1} = u_k + f(u^{k+1}, t^{k+1}) \Delta t$. Clearly, the forward method is explicit in calculation whereas the backward method is implicit and requires solving a non-linear equation specified by f , however, the growth in error rate is typically less in backward method.

4.4 Cellularization of space with FVM

Finite volume methods for convection-diffusion equations was first introduced in the early sixties by Tichonov and Samarskii [68]. The *finite volume method* (FVM) is a discretization method which is well suited for the numerical simulation of various types (elliptic, parabolic or hyperbolic, for instance) of conservation laws; it has been extensively used in several engineering fields, such as fluid mechanics, heat and mass transfer or petroleum engineering. It may be used on arbitrary geometries, using structured or unstructured meshes. An additional feature is the local conservativeness of the numerical fluxes, that is the numerical flux is conserved from one discretization cell to its neighbour. This last feature makes the finite volume method quite attractive when modeling problems for which the flux is of importance, such as in fluid mechanics, semi-conductor device simulation, heat and mass transfer.

Example: Linear Transport We introduce the concept by a simple example of linear transport phenomenon. Let $U(\mathbf{x}, t)$ be the state variable corresponding to concentration of physical entity being transported, and $\mathbf{F}(\mathbf{x}, t) \in C^1(\mathbb{R}^2, \mathbb{R}^2)$ the explicitly given velocity field that defines the transport, with $\mathbf{x} \in \mathbb{R}^2, t \in \mathbb{R}_+$

$$\begin{aligned} \partial_t U &= -\nabla \cdot (U\mathbf{F}) \\ U(\mathbf{x}, 0) &= U_0(\mathbf{x}), \quad \text{initial condition} \end{aligned} \tag{4.3}$$

Let $\mathcal{T}$ be a *partitioning* of space $\mathbb{R}^2$ consisting of polygonal bounded convex subsets of the space, which do not overlap and together cover the space. Let $K \in \mathcal{T}$ be a region in such partitioning, called a *cell*. Integrating the first equation in 4.3, over cell K yields the following *balance equation* over K :

$$\int_K \partial_t U(\mathbf{x}, t) d\mathbf{x} = - \int_{\delta K} U(\mathbf{x}, t) \mathbf{F}(\mathbf{x}, t) \cdot \mathbf{n}_K(\mathbf{x}) d\mathbf{l} \tag{4.4}$$

where δK denotes the boundary of cell K , $\mathbf{n}_K$ denotes the normal vector to δK , outward of K . Let Δt be a constant time discretization step and let $t^i = i\Delta t$ for $i \in \mathbb{N}$. Writing Equation 4.4 with Euler's forward method for time discretization, using the index notation $U^i(\mathbf{x}) := U(\mathbf{x}, t^i)$, and $\mathbf{F}^i(\mathbf{x}) := \mathbf{F}(\mathbf{x}, t^i)$ we get:

$$\frac{1}{\Delta t} \int_K (U^{i+1}(\mathbf{x}) - U^i(\mathbf{x})) d\mathbf{x} = - \int_{\delta K} U^i(\mathbf{x}) \mathbf{F}^i(\mathbf{x}) \cdot \mathbf{n}_K(\mathbf{x}) d\mathbf{l} \quad (4.5)$$

Let us split the border of cell region K with the neighboring cells and convert the border integral on right hand side of the Equation 4.4, into a summation over each border with the neighboring cells as follows:

$$\int_{\delta K} U^i(\mathbf{x}) \mathbf{F}^i(\mathbf{x}) \cdot \mathbf{n}_K(\mathbf{x}) d\mathbf{l} = \sum_{L \in N(K)} \int_{B_{KL}} U^i(\mathbf{x}) \mathbf{F}^i(\mathbf{x}) \cdot \mathbf{n}_{KL}(\mathbf{x}) d\mathbf{l} \quad (4.6)$$

where $N(K)$ denotes set of neighboring cells of cell K . B_{KL} denotes the border between cell K and L , with $\mathbf{n}_{KL}$ the normal vector on the border pointing from cell K to L .

So far we have not done any discretization or coarsening in space. Now let us assign a single u_K^i value for the cell region K , instead of its continuous version $U^i(\mathbf{x})$ for all $\mathbf{x} \in K$. And also define the total velocity field pointing out from the border of cell K towards cell L as:

$$f_{KL}^i := \int_{B_{KL}} \mathbf{F}^i(\mathbf{x}) \cdot \mathbf{n}_{KL}(\mathbf{x}) d\mathbf{l} \quad (4.7)$$

With u^i being now constant across the cell in the discretized version, we get the cellularized version of transport dynamics using finite volume method:

$$\frac{|K|}{\Delta t} (u_K^{i+1} - u_K^i) = - \sum_{L \in N(K)} g_{KL}^i \quad (4.8)$$

where g_{KL}^i is define as following, using the so called *upwind* convention:

$$g_{KL}^i := \begin{cases} f_{KL}^i u_K^i & \text{if } f_{KL}^i \geq 0 \\ f_{KL}^i u_L^i & \text{else} \end{cases} \quad (4.9)$$

The above example shows the basis principles behind the finite volume method. Simply said, the space is partitioned to cells; state variable is coarsened to single value across the cell; dynamics is moved from partial differential equation in continuous space to discrete equation set by integrating the dynamics equations across cell region and its boundary using the *Gauss' Theorem*, given that the PDE was

expressed as a balance equation. This can be applied to parabolic PDEs and hence is applicable to reaction-diffusion-drift systems that we will be studying.

Stability of approximations and error bound A key question in discretization methods is the stability of the discretized (approximate) solution to the dynamics, and bound in error of coarsening the original continuous system to the discretized version. By stability, we mean a structural Lyapunov stability, i.e. all potential solutions that start out nearby an equilibrium solution asymptotically converge to it. To define the coarsening error, we need to compare the state values for each cell in the discrete version to a corresponding value of the cell region of the continuous system. Here there are two approaches: (i) to use the average of the state value across the cell region as the benchmark to compare against, or (ii) use the value at the center of the region. Whilst, first is tempting, error bound proofs in literature is based on the second approach.

There is a myriad of papers published on stability and error bounds on finite volume and finite element methods over the last thirty years. Some selected ones that are close to our system where discretization is on Voronoi meshes and dynamics is reaction-diffusion-convection (drift) type are given below.

In 1997, Morton *et al.* showed that for convection-dominated diffusion systems on a plane with quadrilateral uniform mesh partitioning, the finite volume method has error bounded by h^p where h is the cell size of the mesh and $p \in [1, 2]$ is the convergence rate of approximation to continuous solution as mesh size gets smaller.

In 2001, Vanselow showed that convection-diffusion system discretized with FVM on a Voronoi partitioned mesh, the error is linearly bounded by the maximum Voronoi cell separation defined as the linear distance between centers of the Voronoi boxes [69].

More recently in 2017, Deuring *et al.* [70] applied an hybrid cellularization approach to convection-diffusion-reaction system where time varying solution is discretized by FDM in time (implicit Euler method), diffusion is cellularized in space by FEM and the convection is cellularized by FVM. They showed that the scheme is unconditionally stable, except if the boundary condition is inhomogeneous and the convective velocity is tangential at some points of the boundary.

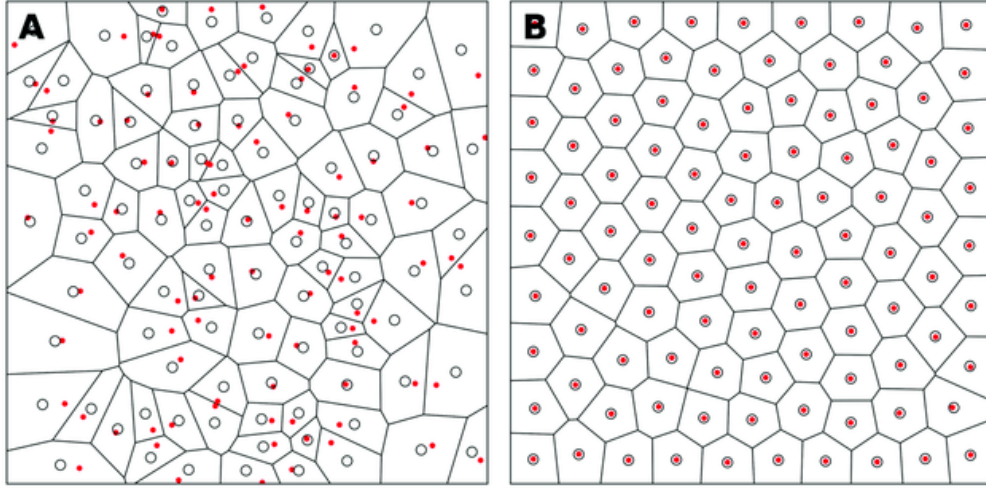


Figure 4.1: Partitioning of 2D manifold into homogeneous voronoi cells

(a) Voronoi partitioning of randomly placed anchor points. (b) Voronoi partitioning with homogeneously placed anchor points.

From manifold to mesh graph In our work, we investigate pattern formation on 2D general manifolds, called the *tissue*. We use the term *cellularization* for discretizing 2D manifold into a *mesh graph*. Our approach in moving from continuous space-time dynamics on manifold to discrete space-time dynamics on mesh graph is as follows: Time discretization is based on explicit (forward) Euler method which is finite difference method in time domain. Space discretization is based on finite volume method, where the tissue is partitioned into regions of cells. Dynamics is discretized using within cell values and cell boundary flows invoking the *Gauss' Theorem* and *Stoke's Theorem* of vector calculus.

The *nodes* of mesh graph represent the cells. The borders between neighboring cells are depicted as *edges* on the mesh graph between the corresponding nodes. Since we work on general 2D manifolds, an orthogonal coordinate system is not necessarily available. Therefore, discretization of the space into a regular grid has to be replaced with a general mesh.

We use homogeneously distributed Voronoi cells for partitioning the manifold as shown in Figure 4.1. Once the manifold Ω is partitioned into homogeneous Voronoi cells, we generate corresponding mesh graph $M(V, E)$ as depicted in Figure 4.2. Each cell corresponds to a node on the mesh graph. Neighboring cells on the partition are connected with an edge on the mesh graph. The nodes are labeled

with notation $V = \{a, b, \dots\}$, and edges are labeled with notation $E = \{c, d, \dots\}$ with $c = \underline{ab}$ if edge connects node a to b . Each node a is attributed with the position $\mathbf{p}_a \in \Omega$ of the anchor point that generated the corresponding cell. Tissue is partitioned into cell regions with the Voronoi partitioning logic which simply says that any region belongs to the anchor point that it is closest to. More formally, $C_a = \{\mathbf{x} \in \Omega; \|\mathbf{x} - \mathbf{p}_a\| \leq \|\mathbf{x} - \mathbf{p}_m\|, \forall m \in V \neq a\}$. The area of each cell region is an attribute assigned to the corresponding node on the mesh, i.e. $c_a = |C_a|$. The border between two neighbor cells a and b on the manifold is defined and denoted as $B_{ab} = C_a \cap C_b$. The length of each border is an attribute assigned to the edge connecting nodes corresponding to cells on the tissue with that border.

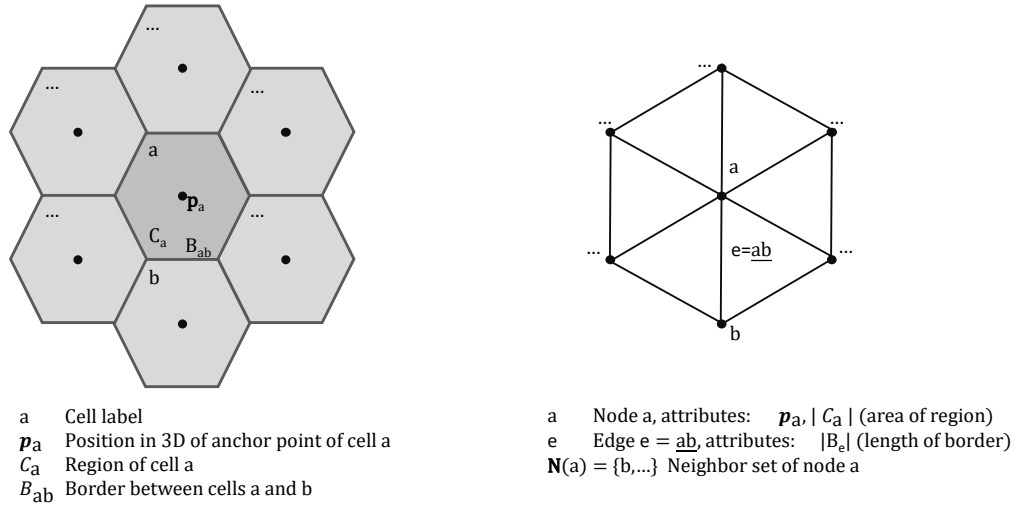


Figure 4.2: Generating mesh graph from Voronoi cell partitioning

Left: Voronoi cell partitioning with anchor points $\mathbf{p}_a, \mathbf{p}_b, \dots$, cell region $C_a = \{\mathbf{x} \in \Omega; \|\mathbf{x} - \mathbf{p}_a\| \leq \|\mathbf{x} - \mathbf{p}_m\|, \forall m \in V \neq a\}$, and cell borders $B_{ab} = C_a \cap C_b$. Right: Mesh graph with each node corresponding to a cell of the Voronoi partition and each edge corresponding to a border between neighboring cells with attributes, position in 3D of anchor point, cell area and border length.

Even if the manifold is analytically defined, randomly picking N points on it is not trivial to implement in general. For this reason, we also developed a *stitching algorithm*, which builds a mesh on an analytically defined surface. Starting from a seed point and a given target edge length, the algorithm stitches triangular faces on the analytical surface and closes it up if the manifold were a closed surface such as sphere or torus. Figure 4.3 shows stitching up of a sphere and torus using our algorithm.

Having partitioned the tissue into cells and established the corresponding mesh

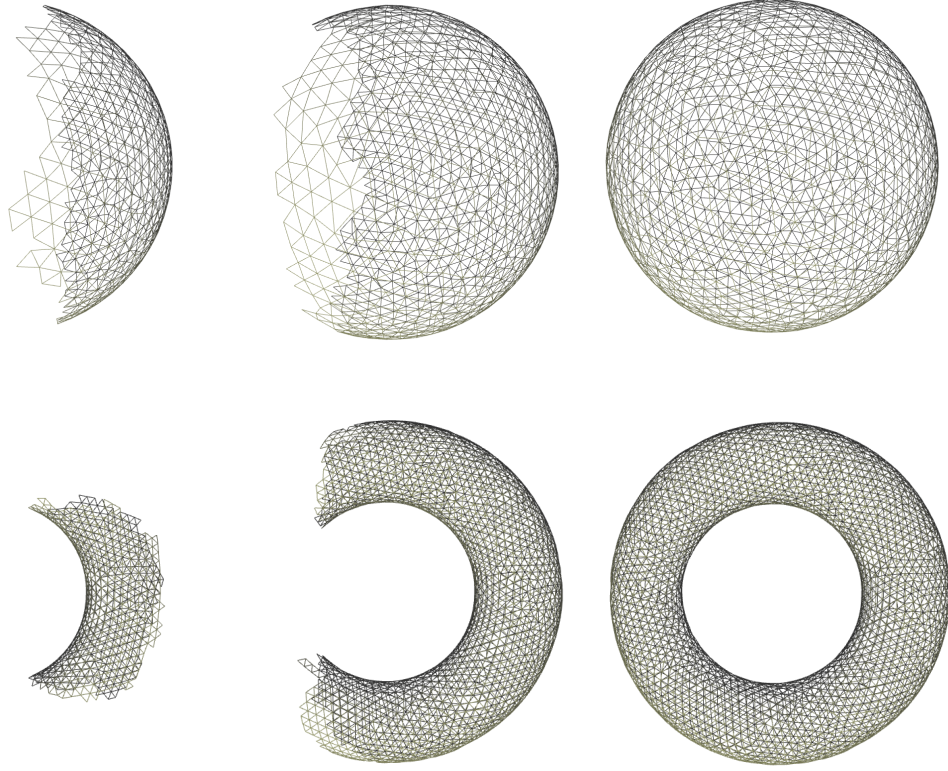


Figure 4.3: Stitching algorithm building mesh around manifold

(a) Sphere (b) Torus

graph, we are now in a position to define equivalent differential operators on the mesh, derived using *Fundamental Theorem of Calculus*, *Gauss's Theorem* and *Stoke's Theorem*. Table 4.4 sets forth the results.

A **Scalar field** on a continuous 2D manifold Ω is a function that assigns a real value to each point $\mathbf{x} \in \Omega$, notated as $\Phi(\mathbf{x})$. The corresponding concept on the mesh M is a function that assigns a real value to each cell of the cellularized manifold. We note that a node of the mesh graph corresponds to a cell, and the edge connecting the nodes on the mesh corresponds to the border between two cells. To find the scalar field value of a node $\mathbf{a}$, we integrate the scalar field of the manifold of the corresponding cell, normalized by the cell area. Since we use irregular sized cells, normalization by cell size is needed, otherwise, larger cell would have large field value, even though the corresponding scalar field of the manifold may be constant. Therefore, the scalar field gets cellularized as: $\Phi(\mathbf{x}) \rightarrow \phi_{\mathbf{a}} := \frac{1}{c_{\mathbf{a}}} \oint_{C_{\mathbf{a}}} \Phi(s) ds$, where $C_{\mathbf{a}}$ the region of cell indexed $\mathbf{a}$ on the manifold, and $c_{\mathbf{a}}$ its area.

A **Vector field** on a continuous 2D manifold Ω is a function that assigns a 2D vector to each point $\mathbf{x} \in \Omega$, notated as $\mathbf{F}(\mathbf{x})$. Note that the vector lies on the manifold, meaning it has no component normal to the surface of the manifold. We define the vector field on each edge of the mesh graph as follows: The vector field of the manifold is integrated *along the border* $\underline{\mathbf{ab}}$ between cell $\mathbf{a}$ and cell $\mathbf{b}$. This border is a curve on the manifold, hence the integration is a curve integral. To perform the curve integrals, we need to define unit parallel vector $\mathbf{n}^{\parallel}$ and unit perpendicular vector $\mathbf{n}^{\perp}$ at each point the border curve. Here we need a convention on the orientation of these unit vectors. In the case of a closed 2D manifold embedded in 3D, such as a spheroid or torus, the manifold creates a membrane that splits the 3D space into two volumes, inside and outside. Our convention for picking the surface normal vector of the closed manifold at any point on the manifold is *from inside towards outside*. With the surface normal direction set, we then use *positive orientation* convention where $\mathbf{n}^{\parallel}$, $\mathbf{n}^{\perp}$ and the surface normal correspond to local x-axis, y-axis and z-axis, respectively. This convention still leaves us with two orientation choices, as one can flip both $\mathbf{n}^{\parallel}$ and $\mathbf{n}^{\perp}$ 180 degrees on the surface and still satisfy the positive-orientation. In the case of open 2D manifolds, we arbitrarily pick an orientation for the surface normal at a given point by selecting one of the two orientation possibilities, and then consistently transfer the surface normal vector along the continuous manifold to any other point.

Integrating the tangential component of the vector field along the border gives us the tangent value $f_{\underline{\mathbf{ab}}}^{\parallel} := \int_{B_{\underline{\mathbf{ab}}}} \mathbf{F} \cdot \mathbf{n}^{\parallel} dl$, and similarly integrating the perpendicular component of the vector field along the border gives us the perpendicular value $f_{\underline{\mathbf{ab}}}^{\perp} := \int_{B_{\underline{\mathbf{ab}}}} \mathbf{F} \cdot \mathbf{n}^{\perp} dl$. This results in each edge being assigned a 2D vector $\mathbf{f}_{\underline{\mathbf{ab}}} = \left(f_{\underline{\mathbf{ab}}}^{\perp}, f_{\underline{\mathbf{ab}}}^{\parallel} \right)$. Simply stated, the vector field lying on the continuous manifold is cellularized where flow of the field *across* the cell boundary, and flow of the field *along* the cell boundary, are assigned to the boundary (i.e. the edge on the mesh) as two values. The use of this field discretization will become evident soon, as one can now invoke the Gauss' Theorem and Stoke's Theorem to calculate the cellularized version of divergence and curl of the vector field on the manifold by simply summing the perpendicular and parallel components, respectively, of the cellularized version of the field along the closed boundary of the cell.

Divergence operator acting on a vector field defined over the manifold Ω creates a scalar field which measures the net amount of vector field emanating at any given point on the manifold. The equivalent operator on the mesh can be derived using the cellularization of scalar and vector fields as defined above, and by using the *Gauss' Theorem* which states that the total divergence of a vector field on a region, calculated as the integral of the divergence over the region, equals the line integral of the vector field leaving the region. Intuitively it is a coarse graining argument. If one partitions the tissue into cells, then total field emanating from a cell is the amount *leaving the boundary* of the cell. This coarse graining intuition underlying Gauss' Theorem leads to definition of divergence of a vector field defined on the mesh at a given node being equal to the sum of perpendicular components of the vector field assigned to the edges departing from such node. More formally, $\oint_{C_a} \nabla \cdot \mathbf{F} \, ds = \sum_{b \in \mathbf{E}(a)} \int_{B_b} \mathbf{F} \cdot \mathbf{n}^\perp \, dl = \sum_{b \in \mathbf{E}(a)} f_b^\perp$, where $\mathbf{E}(a)$ denotes the edges emanating from node a and B_b denotes the border between two cells with corresponding nodes connected by edge b .

Curl operator acting on a vector field defined over the 2D manifold Ω creates a scalar field which measures the net amount of vector field rotating at any given point on the manifold; depending on whether rotation is clockwise or counterclockwise with respect to normal of the 2D manifold, curl gets its sign (or direction with regards to such normal vector). The equivalent operator on the mesh can be derived using the cellularization of scalar and vector fields as defined above, and by using the *Stokes' Theorem* which states that the total curl of a vector field on a region, calculated as the integral of the curl over the region, equals the line integral of the vector field *along the boundary* of the cell. This is also a coarse graining argument which leads to definition of curl of a vector field defined on the mesh at a given node being equal to the sum of parallel components of the vector field assigned to the edges departing from such node. More formally, $\oint_{C_a} \nabla \times \mathbf{F} \, ds = \sum_{b \in \mathbf{E}(a)} \int_{B_b} \mathbf{F} \cdot \mathbf{n}^\parallel \, dl = \sum_{b \in \mathbf{E}(a)} f_b^\parallel$, where $\mathbf{E}(a)$ denotes the edges emanating from node a and B_b denotes the border between two cells with corresponding nodes connected by edge b .

Gradient operator acting on a scalar field defined over the 2D manifold Ω creates a vector field that lies on the manifold with direction and magnitude of the steepest change of the scalar field at any given point on the manifold. We define the

analogue concept of directional change on the mesh by calculating the difference of the scalar field at a given node and its neighbor, and assigning such difference to the edge between the nodes as the perpendicular component of a vector field defined on such edge. We note that the parallel component of the gradient field on the edge is zero since gradient field by definition has no curl.

We can readily describe the **Laplacian operator** which is the convexity of a scalar field defined over the 2D manifold that is calculated as the divergence of gradient of the scalar field. Formally, $\nabla^2\Phi(\mathbf{x}) = \nabla \cdot \nabla\Phi(\mathbf{x})$. Using the corresponding divergence and gradient definitions on the mesh given above, the Laplacian operator on a mesh becomes $(\text{lapl } \phi)_a = (\text{div } (\text{grad } \phi))_a = \frac{1}{c_a} \sum_{b \in N(a)} (\phi_b - \phi_a)$. Or simply said, the Laplacian of the scalar field at a node is the deviation of that node's field value from the average of its neighbors. This is a useful and interesting insight. As we will see the diffusion dynamics on a mesh simply becomes an incremental change of a node's value towards the average of its neighbors' values.

Lastly, we describe the **Drift operator** which maps a gradient field (drift vector) and a scalar field onto a scalar field. Drift operator acting on a 2D manifold is given as: $\nabla_{\mathbf{F}} \cdot \Phi(\mathbf{x}) := \nabla \cdot \Phi(\mathbf{x})\mathbf{F}(\mathbf{x})$. Applying the *product rule*, we get $\nabla_{\mathbf{F}} \cdot \Phi(\mathbf{x}) = \nabla\Phi(\mathbf{x}) \cdot \mathbf{F}(\mathbf{x}) + \Phi(\mathbf{x})\nabla \cdot \mathbf{F}(\mathbf{x})$. Using the cellularized analogues of the gradient, divergence and dot product on the mesh, we calculate the cellularized version of the drift as: $(\nabla_{\mathbf{f}}\phi)_a = \sum_{b \in N(a)} \phi_b f_{ab}^\perp$.

Table 4.1: Vector calculus on Manifold vs Mesh

	Manifold Ω	Mesh M
Scalar Field	Scalar field on continuous manifold. Ω is the 2D manifold. $\Phi : \Omega \rightarrow \mathbb{R}$	Discretized scalar field over the cell region, assigned to corresponding node of the mesh. V is the set of nodes of the mesh. $\phi : V \rightarrow \mathbb{R}$

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
	$\Phi(\mathbf{x})$	$\phi_{\mathbf{a}} := \frac{1}{c_{\mathbf{a}}} \oint_{C_{\mathbf{a}}} \Phi(s) \, ds$ $C_{\mathbf{a}}$ and $c_{\mathbf{a}}$ are the region of cell $\mathbf{a}$ on the manifold and its area, respectively
Vector Field	Vector field on continuous manifold. $\mathbf{F} : \Omega \rightarrow \mathbb{R}^2$	Discretized vector field along a cell border assigned to corresponding edge of the mesh. E is the set of edges of the mesh. $\mathbf{f} : E \rightarrow \mathbb{R}^2$

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
	$\mathbf{F}(\mathbf{x})$	$\mathbf{f}_{\underline{\mathbf{ab}}} = \left(f_{\underline{\mathbf{ab}}}^{\perp}, f_{\underline{\mathbf{ab}}}^{\parallel} \right)$ $f_{\underline{\mathbf{ab}}}^{\perp} := \int_{B_{\underline{\mathbf{ab}}}} \mathbf{F} \cdot \mathbf{n}^{\perp} dl$ $f_{\underline{\mathbf{ab}}}^{\parallel} := \int_{B_{\underline{\mathbf{ab}}}} \mathbf{F} \cdot \mathbf{n}^{\parallel} dl$ $B_{\underline{\mathbf{ab}}}$ is the border between cells $\mathbf{a}$ and $\mathbf{b}$ on the manifold. $\mathbf{n}^{\perp}$ and $\mathbf{n}^{\parallel}$ represent the positively oriented normal and tangent vectors along the border $B_{\underline{\mathbf{ab}}}$ on the manifold Ω with <i>positive orientation</i> in respect to the manifold normal $\hat{\mathbf{n}}$.
Dot Product	Dot product of two vectors lying on continuous manifold gives a scalar value. $\mathbf{F}(\mathbf{x}) \cdot \mathbf{G}(\mathbf{x})$	Cellularized analogue of dot product of two vectors defined on the edges of a given node $\mathbf{a}$ of the mesh: $(\mathbf{f} \cdot \mathbf{g})_{\mathbf{a}}$ $= \sum_{b \in \mathbf{E}(\mathbf{a})} f_b^{\perp} g_b^{\perp} + f_b^{\parallel} g_b^{\parallel}$ where $\mathbf{E}(\mathbf{a})$ denotes the edges emanating from node $\mathbf{a}$

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
Cross Product	Cross product of two vectors lying on continuous manifold gives a scalar value (and a direction normal to surface manifold which we drop from notation). $\mathbf{F}(\mathbf{x}) \times \mathbf{G}(\mathbf{x})$	Cellularized analogue of cross product of two vectors defined on the edges of a given node $\mathbf{a}$ of the mesh: $(\mathbf{f} \times \mathbf{g})_{\mathbf{a}} = \sum_{b \in \mathbf{E}(\mathbf{a})} f_b^{\perp} g_b^{\perp} + f_b^{\parallel} g_b^{\parallel}$ where $\mathbf{E}(\mathbf{a})$ denotes the edges emanating from node $\mathbf{a}$
Divergence operator	Divergence operator acting on a vector field on the manifold gives a scalar field. It measures the amount of vector field <i>emanating</i> at each point of the manifold $\nabla \cdot : \mathbf{F}(\Omega) \rightarrow \Phi(\Omega)$	Cellularized analogue of divergence operator acting on the vector field assigned to edges of the mesh calculates the total amount vector field emanating from the node surrounded by its edges. $\text{div} : \mathbf{f}(E) \rightarrow \phi(V)$

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
	$\nabla \cdot \mathbf{F}(\mathbf{x})$	$(\operatorname{div} \mathbf{f})_{\mathbf{a}} :=$ $\frac{1}{c_{\mathbf{a}}} \oint_{C_{\mathbf{a}}} \nabla \cdot \mathbf{F} \, ds$ $(\operatorname{div} \mathbf{f})_{\mathbf{a}} = \frac{1}{c_{\mathbf{a}}} \sum_{b \in \mathbf{E}(\mathbf{a})} f_b^{\perp}$ which follows from <i>Gauss' Theorem</i>. $\mathbf{E}(\mathbf{a})$ denotes the edges emanating from node $\mathbf{a}$
Curl operator	Curl operator acting on a vector field on the manifold gives a scalar field (and a direction normal to manifold surface which we drop from notation). It measures the amount of vector field <i>rotation</i> at each point of the manifold. $\nabla \times : \mathbf{F}(\Omega) \rightarrow \Psi(\Omega)$	Cellularized analogue of curl operator acting on the vector field assigned to edges of the mesh calculates the total amount vector field rotating around the node surrounded by its edges. $\operatorname{curl} : \mathbf{f}(E) \rightarrow \psi(V)$

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
	$\nabla \times \mathbf{F}(\mathbf{x})$	$(\text{curl } \mathbf{f})_{\mathbf{a}} :=$ $\frac{1}{c_{\mathbf{a}}} \oint_{C_{\mathbf{a}}} \nabla \times \mathbf{F} \, ds$ $(\text{curl } \mathbf{f})_{\mathbf{a}} = \frac{1}{c_{\mathbf{a}}} \sum_{b \in \mathbf{E}(\mathbf{a})} f_b^{\parallel}$ which follows from <i>Stoke's Theorem</i>. $\mathbf{E}(\mathbf{a})$ denotes the edges emanating from node $\mathbf{a}$
Gradient Operator	Gradient operator acting on a scalar field of continuous manifold gives the direction and magnitude of maximal change of the field $\nabla : \Phi(\Omega) \rightarrow \mathbf{F}(\Omega)$ $\nabla \Phi(\mathbf{x})$	Gradient field on the border of two cells is approximated as by the change of discretized scalar field from one cell to the next. $\text{grad} : \phi(V) \rightarrow \mathbf{f}(E)$ $(\text{grad } \phi)_{\underline{\mathbf{ab}}} = \mathbf{f}_{\underline{\mathbf{ab}}} = (f_{\underline{\mathbf{ab}}}^{\perp}, f_{\underline{\mathbf{ab}}}^{\parallel})$ $f_{\underline{\mathbf{ab}}}^{\perp} = \phi_{\mathbf{b}} - \phi_{\mathbf{a}}$ $f_{\underline{\mathbf{ab}}}^{\parallel} = 0$ Note that gradient fields do not have any curl, hence $f_{\underline{\mathbf{ab}}}^{\parallel} = 0$.

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
Laplacian operator	Laplacian operator acting on a scalar field of continuous manifold gives the convexity of the scalar field, which is the divergence of its gradient. $\nabla^2 : \Phi(\Omega) \rightarrow \Psi(\Omega)$ $\nabla^2 \Phi(\mathbf{x}) = \nabla \cdot \nabla \Phi(\mathbf{x})$	Applying the divergence and gradient analogues of a scalar field on a mesh, we get the Laplacian of the scalar field at each node as that node's deviation from its neighbors' average. $\text{lapl} : \phi(V) \rightarrow \psi(V)$ $(\text{lapl } \phi)_{\mathbf{a}} = (\text{div } (\text{grad } \phi))_{\mathbf{a}}$ $= \frac{1}{c_{\mathbf{a}}} \sum_{\mathbf{b} \in \mathbf{N}(\mathbf{a})} (\text{grad } \phi)_{\mathbf{b}}$ $= \frac{1}{c_{\mathbf{a}}} \sum_{\mathbf{b} \in \mathbf{N}(\mathbf{a})} (\phi_{\mathbf{b}} - \phi_{\mathbf{a}})$
Drift operator	Drift operator, specified by an underlying gradient vector field (the <i>drift field</i>), acts on a scalar field and calculates the <i>biased convexity</i> of the scalar field where such biasing is provided by the underlying drift field. $\nabla_{\mathbf{F}} : \Phi(\Omega) \rightarrow \Psi(\Omega)$	As shown below, applying the product rule, we get the simple intuition that on the mesh, the drift operator becomes the gradient field weighted average of a node's neighbors. $\nabla_{\mathbf{f}} : \phi(V) \rightarrow \psi(V)$

Continued on next page

Table 4.1 – *Continued from previous page*

	Manifold Ω	Mesh M
	$\nabla_{\mathbf{F}} \cdot \Phi(\mathbf{x}) := \nabla \cdot \Phi(\mathbf{x})\mathbf{F}(\mathbf{x})$ $= \nabla \Phi(\mathbf{x}) \cdot \mathbf{F}(\mathbf{x}) + \Phi(\mathbf{x})\nabla \cdot \mathbf{F}(\mathbf{x})$ follows from <i>Product Rule</i>. Note that if the underlying drift field is set equal to the normalized gradient field of Φ, (i.e. if $\mathbf{F} = \frac{\nabla \Phi}{\Phi}$) the the Drift operator reduces to Laplacian operator (i.e. <i>biased convexity</i> measurement becomes regular <i>convexity</i>).	$\nabla_{\mathbf{f}} \phi = \text{div}(\phi \mathbf{f})$ $= \text{grad } \phi \cdot \mathbf{f} + \phi \text{div } \mathbf{f}$ $(\nabla_{\mathbf{f}} \phi)_{\mathbf{a}} = \sum_{\mathbf{b} \in N(\mathbf{a})} (\phi_{\mathbf{b}} - \phi_{\mathbf{a}}) f_{\underline{\mathbf{a}\mathbf{b}}}^{\perp}$ $+ \phi_{\mathbf{a}} \sum_{\mathbf{b} \in N(\mathbf{a})} f_{\underline{\mathbf{a}\mathbf{b}}}^{\perp}$ $(\nabla_{\mathbf{f}} \phi)_{\mathbf{a}} = \sum_{\mathbf{b} \in N(\mathbf{a})} \phi_{\mathbf{b}} f_{\underline{\mathbf{a}\mathbf{b}}}^{\perp}$

With a cellularized tissue represented as a mesh, and equipped with discrete calculus (e.g. scalar and vector fields, divergence, curl, gradient, laplacian and drift operators on the mesh), we can now investigate pattern forming dynamics on the mesh. The cellularized system is more amenable to formal analysis and provides an effective linkage to biology where tissue is naturally formed out of cells. Table 4.4 provides the correspondence between continuous pattern forming dynamics on the manifold and discrete dynamics on the mesh.

State of a spatially-extended dynamical system on manifold Ω is a scalar field defined over the manifold that changes over time, denoted as $U(\mathbf{x}, t), \mathbf{x} \in \Omega$. The corresponding analogue of the state of discrete-time cellular-space dynamics defined on a mesh is a scalar field defined at each node of the mesh that changes over time,

denoted as $u_{\mathbf{a}}^k$. **Time derivative** of the state, $\partial_t U(\mathbf{x}, t)$ is given as time difference of discrete-time state values on the mesh at each node, denoted as $\delta u_{\mathbf{a}}^k = u_{\mathbf{a}}^{k+1} - u_{\mathbf{a}}^k$.

Diffusion dynamics on manifold was given as $\partial_t U = D \nabla^2 U$ where D is the diffusion constant. This meant points where state value U had higher convexity, increased their state value over time, hence reducing such convexity. This is the intuitive mechanism behind diffusion dynamics. We readily find the analogue of it in the cellular dynamics over mesh M . As shown in table below, the diffusion dynamics on the mesh is given as (using iterative notation): $u_{\mathbf{a}} \rightarrow (1 - \lambda)u_{\mathbf{a}} + \lambda \langle u \rangle_{\mathbf{N}(\mathbf{a})}$, where $\langle u \rangle_{\mathbf{N}(\mathbf{a})}$ means averaging u state value over neighbors $\mathbf{N}(\mathbf{a})$ of node $\mathbf{a}$. Here, $\lambda \in [0, 1]$ corresponds to diffusion constant in cellular dynamics. We note that for $\lambda = 0$, there is no diffusion, and the local state value of a cell is ambivalent to the state values of the neighboring cells; for $\lambda = 1$, local state value of a cell becomes equal to the average state value of its neighbors within one iteration.

Reaction-diffusion dynamics with two state variables (chemical reactants), $\mathbf{S} = [U, V]$ over a continuous manifold was given as $\partial_t \mathbf{S} = D \nabla^2 \mathbf{S} + \varphi(\mathbf{S})$, where φ is the state vector reaction function defined with two functions $\varphi = [F, G]$. The cellular reaction-diffusion dynamics on the mesh is given as (using iterative notation): $\mathbf{s}_{\mathbf{a}} \rightarrow (\mathbf{I} - \Lambda)\mathbf{s}_{\mathbf{a}} + \Lambda \langle \mathbf{s} \rangle_{\mathbf{N}(\mathbf{a})} + \varphi(\mathbf{s})$, where $\mathbf{s}_{\mathbf{a}} = [u, v]_{\mathbf{a}}$ are the two state variables at node $\mathbf{a}$. $\Lambda = \text{Diag}(\lambda, \beta)$ is the diagonal diffusion matrix with λ and β corresponding to diffusion constants for state variables u and v , respectively. $\varphi = [f, g]$ is the state vector reaction function defined with two functions f and g .

Table 4.2: Discrete Cellular Dynamics on Mesh

	Continuous Dynamics	Cellular Dynamics
State	State of spatially-extended dynamical system is a time-dependent scalar field on the space manifold.	State of discrete-time cellular-space dynamical system are scalar node values that change over discrete time.

Continued on next page

Table 4.2 – *Continued from previous page*

	Continuous Dynamics Ω	Cellular Dynamics M
	$U(\mathbf{x}, t)$	$u_{\mathbf{a}}^k$ where k is time index for state of cell $\mathbf{a}$.
Time Derivative	$\partial_t U(\mathbf{x}, t)$	Time derivative of state becomes time difference in discrete-time dynamical system $\delta u_{\mathbf{a}}^k = u_{\mathbf{a}}^{k+1} - u_{\mathbf{a}}^k$

Continued on next page

Table 4.2 – *Continued from previous page*

	Continuous Dynamics Ω	Cellular Dynamics M
Diffusion	$\partial_t U = D \nabla^2 U$	$\delta u_{\mathbf{a}}^k = D (\text{lapl } u)_{\mathbf{a}}$ $= \frac{\lambda}{ \mathbf{N}(\mathbf{a}) } \sum_{\mathbf{b} \in \mathbf{N}(\mathbf{a})} (u_{\mathbf{b}}^k - u_{\mathbf{a}}^k)$ using iterative notation for the time dynamics where $u_i^k \rightarrow u_{\mathbf{a}}^{k+1} = u_{\mathbf{a}}^k + \delta u_{\mathbf{a}}^k$, and dropping the time indexation as evident in iterative notation, we get: $u_{\mathbf{a}} \rightarrow (1-\lambda)u_{\mathbf{a}} + \lambda \frac{1}{ \mathbf{N}(\mathbf{a}) } \sum_{\mathbf{b} \in \mathbf{N}(\mathbf{a})} u_{\mathbf{b}}$ $\rightarrow (1-\lambda)u_{\mathbf{a}} + \lambda \langle u \rangle_{\mathbf{N}(\mathbf{a})}$ where $\langle u \rangle_{\mathbf{N}(\mathbf{a})}$ denotes averaging of u state values over the neighbors of cell $\mathbf{a}$. The result simply says that diffusion is the change of cell's state variable towards the average value of its neighbors; i.e. the heuristics of diffusion equation is '<i>go where your neighbors are in state space</i>'.

Continued on next page

Table 4.2 – Continued from previous page

	Continuous Dynamics Ω	Cellular Dynamics M
Reaction-Diffusion	$\partial_t U = D_U \nabla^2 U + F(U, V)$ $\partial_t V = D_V \nabla^2 V + G(U, V)$ where U and V are two reacting state variables with reaction functions F and G, respectively that are functions of local values of two states, and D_U and D_V the respective diffusion rates.	$u_{\mathbf{a}} \rightarrow$ $\rightarrow (1-\lambda)u_{\mathbf{a}} + \lambda \langle u \rangle_{\mathbf{N}(\mathbf{a})} + f(u_{\mathbf{a}}, v_{\mathbf{a}})$ $v_{\mathbf{a}} \rightarrow$ $\rightarrow (1-\beta)v_{\mathbf{a}} + \beta \langle v \rangle_{\mathbf{N}(\mathbf{a})} + g(u_{\mathbf{a}}, v_{\mathbf{a}})$ which states that reaction-diffusion is intracellular local averaging of state values at diffusion rates of λ and β, driven by intercellular reaction.

4.5 Turing instability revisited

Turing instability in continuous space-time was discussed in Section 3.2. We now analyze it in discrete space-time where the mesh is a cyclic linear graph. Let $u_{\mathbf{a}}, v_{\mathbf{a}}$ denote the two reactants (state variables) at cell $\mathbf{a}$. The state equations, linearized around stability point, are given as follows:

$$\delta u_{\mathbf{a}} = D_u (\langle u \rangle_{\mathbf{N}(\mathbf{a})} - u_{\mathbf{a}}) + f(u_{\mathbf{a}}, v_{\mathbf{a}}) \quad (4.10a)$$

$$\delta v_{\mathbf{a}} = D_v (\langle v \rangle_{\mathbf{N}(\mathbf{a})} - v_{\mathbf{a}}) + g(u_{\mathbf{a}}, v_{\mathbf{a}}) \quad (4.10b)$$

$$f(u_{\mathbf{a}}, v_{\mathbf{a}}) = au_{\mathbf{a}} + bv_{\mathbf{a}} \quad (4.10c)$$

$$g(u_{\mathbf{a}}, v_{\mathbf{a}}) = cu_{\mathbf{a}} + dv_{\mathbf{a}} \quad (4.10d)$$

Let $(u_{\mathbf{a}}, v_{\mathbf{a}}) = \omega_{\mathbf{a}} \hat{\mathbf{e}}_{\mathbf{a}}$ be the position of cell $\mathbf{a}$ on the state space in vector notation. Assume that at steady state all cells are aligned on state space, i.e. $\hat{\mathbf{e}}_{\mathbf{a}} = \hat{\mathbf{e}}, \forall \mathbf{a}$. We

will come back to justify the alignment assumption later. The prototype for Turing instability leading to spontaneous pattern formation is a linear reaction system that has a stable attractor with following properties:

$$\mathbf{F} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} D\mu & 0 \\ 0 & \mu \end{bmatrix}$$

$$\text{Tr} = a + d < 0 \tag{4.11a}$$

$$\text{Det} = ad - bc > 0 \tag{4.11b}$$

$$ad < 0 \tag{4.11c}$$

Whereas, the first two conditions above are the standard stability condition of the attractor, the third condition is a special requirement to generate Turing instability upon diffusive coupling. The third condition says that a and d should have opposite signs, i.e. the reaction system should be contracting (-) in one reactant but expanding (+) on the other reactant around the stability point. Whilst this sounds like the attractor is unstable, because of the *curl* generated by b and c of the reaction parameters with $|bc| > |ad|$, the attractor is indeed stable (albeit in the absence of diffusion).

We observe from simulations that in the steady-state of pattern forming Turing dynamics, cells attain state values $(u_{\mathbf{a}}, v_{\mathbf{a}})$ that are *aligned on a line* when plotted on the state space diagram. This can be seen Figure 4.6 where the top sub-plots show the steady-state of each cell in state space where x-axis is the u values and y-axis is the v values. Each cell is plotted with a blue dot on this state space, with the flow lines on state space given by the reaction dynamics shown as green curves. This is the phase portrait of a single point dynamics of two variables. If each cell was running its own reaction dynamics with no diffusion among them, then each cell would flow on the green flow lines and converge to the attractor of reaction dynamics which is a stable attractor in the middle of the phase plots. But, the interesting phenomenon of patterning arises because each cell is effectively connected to its spatial neighbors in state space due to diffusion part of the dynamics. We have noted that diffusion in cellular dynamics is a process of *go towards your neighbors'*

average. One way to internalize this concept is what we call *rubber bands* analogy. Imagine that each cell, in state space, is tied to its spatial neighbors with elastic rubber bands. These bands pull the cells, in state space, towards each other due to diffusion. If there were no reaction dynamics, the cells would simply lump into a single position which would be the averaged out state values. But due to reaction dynamics which pushes the cells to move on the green lines of Figure 4.6, the result is, for certain parameter values that we will investigate below, an alignment of cells on the state space.

Now assuming that this alignment of cells' state values has occurred in the steady-state in state space, we can represent the state (u_a, v_a) of any cell as $(u_a, v_a) = \omega_a \hat{\mathbf{e}} \forall a$, with $\hat{\mathbf{e}}$ being the unit vector in state space showing the direction of this alignment. Now each cell's state is defined by a single parameter ω_a . We now ask the question as how does one cell's ω value compare to the average of its spatial neighbors. We define the percent difference of a cell's state value ω_a to the average of its neighbors $\langle \omega \rangle_{\mathbf{N}(a)}$ as its *local differentiation*, denoted by ϵ_a .

$$\epsilon_a = \frac{\omega_a - \langle \omega \rangle_{\mathbf{N}(a)}}{\omega_a} \quad (4.12)$$

This is an important parameter which defines the ordering of cells in the state space, which as we shall show, determines the periodicity of spatial patterning. The dynamics with diffusion, along the state-space alignment unit vector $\hat{\mathbf{e}}$ becomes:

$$\begin{pmatrix} \delta u_a \\ \delta v_a \end{pmatrix} = \mathbf{F} \omega_a \hat{\mathbf{e}} + \mathbf{D} (\langle \omega \rangle_{\mathbf{N}(a)} - \omega_a) \hat{\mathbf{e}} \quad (4.13a)$$

$$= (\mathbf{F} - \epsilon_a \mathbf{D}) (\omega_a \hat{\mathbf{e}}) \quad (4.13b)$$

$$= \begin{bmatrix} a - \epsilon_a D\mu & b \\ c & d - \epsilon_a \mu \end{bmatrix} (\omega_a \hat{\mathbf{e}}) \quad (4.13c)$$

$$= \tilde{\mathbf{F}}_a (\omega_a \hat{\mathbf{e}}) \quad (4.13d)$$

In order for the alignment of cells in state space to be persistent, the differential change in state variables need to be along the alignment direction, i.e.:

$$\lambda_{\mathbf{a}}^+(\omega_{\mathbf{a}}\hat{\mathbf{e}}) = \tilde{\mathbf{F}}_{\mathbf{a}}(\omega_{\mathbf{a}}\hat{\mathbf{e}}) \quad (4.14)$$

This means alignment is along the eigenvector of $\tilde{\mathbf{F}}_{\mathbf{a}}$ corresponding to the positive eigenvalue $\lambda_{\mathbf{a}}^+$. Let $\tilde{\text{Tr}}_{\mathbf{a}}$ and $\tilde{\text{Det}}_{\mathbf{a}}$ denote the trace and determinant of $\tilde{\mathbf{F}}_{\mathbf{a}}$.

$$\tilde{\text{Tr}}_{\mathbf{a}} = a - \epsilon_{\mathbf{a}}D_{\mu} + d - \epsilon_{\mathbf{a}}\mu = \text{Tr} - \epsilon_{\mathbf{a}}\mu(D+1) \quad (4.15a)$$

$$\tilde{\text{Det}}_{\mathbf{a}} = (a - \epsilon_{\mathbf{a}}D_{\mu})(d - \epsilon_{\mathbf{a}}\mu) - bc \quad (4.15b)$$

$$= \text{Det} - \epsilon_{\mathbf{a}}\mu(Dd + a) + \epsilon_{\mathbf{a}}^2\mu^2D \quad (4.15c)$$

The eigenvalue of $\tilde{\mathbf{F}}_{\mathbf{a}}$ is required to be real with $\lambda_{\mathbf{a}}^- < 0 < \lambda_{\mathbf{a}}^+$.

$$\lambda_{\mathbf{a}}^{\pm} = \frac{\tilde{\text{Tr}}_{\mathbf{a}}}{2} \pm \sqrt{\left(\frac{\tilde{\text{Tr}}_{\mathbf{a}}}{2}\right)^2 - \tilde{\text{Det}}_{\mathbf{a}}} \quad (4.16)$$

Since $\tilde{T}_{\mathbf{a}} < 0$, we need $\tilde{\text{Det}}_{\mathbf{a}} < 0$ to satisfy the requirement that both eigenvalues are real and $\lambda_{\mathbf{a}}^- < 0 < \lambda_{\mathbf{a}}^+$. Therefore, Turing instability condition simply becomes:

Turing instability condition

$$\tilde{\text{Det}}_{\mathbf{a}} < 0$$

$$\text{Det} - \epsilon_{\mathbf{a}}\mu(Dd + a) + \epsilon_{\mathbf{a}}^2\mu^2D < 0 \quad (4.17)$$

Note that for a given set of parameters of linear cellular reaction-diffusion system, $\mathbf{F}, \mu, D$, there could be a band of ϵ values that satisfy the Turing instability condition given in Equation 4.17 with corresponding eigenvalues and eigenvectors, $\lambda(\epsilon_{\mathbf{a}})$ and $\hat{\mathbf{e}}_{\mathbf{a}}$. It can be shown that $\lambda(\epsilon)_{\mathbf{a}}$ is concave in the region that satisfies the Turing instability condition, with a single maximum λ^* and the corresponding eigenvector $\hat{\mathbf{e}}^*$ attained at local differentiation value of ϵ^* . We call this the *coherence mode*. The coherence mode will exponentially override all other modes as time progresses. Figure 4.4 shows the $\lambda(\epsilon)_{\mathbf{a}}$ for varying values of D , and the eigenvectors scaled with corresponding eigenvalues.

We now investigate the periodicity of patterns generated by the coherence mode. The underlying spatial graph of cellular tissue is assumed to be linear cyclic, hence

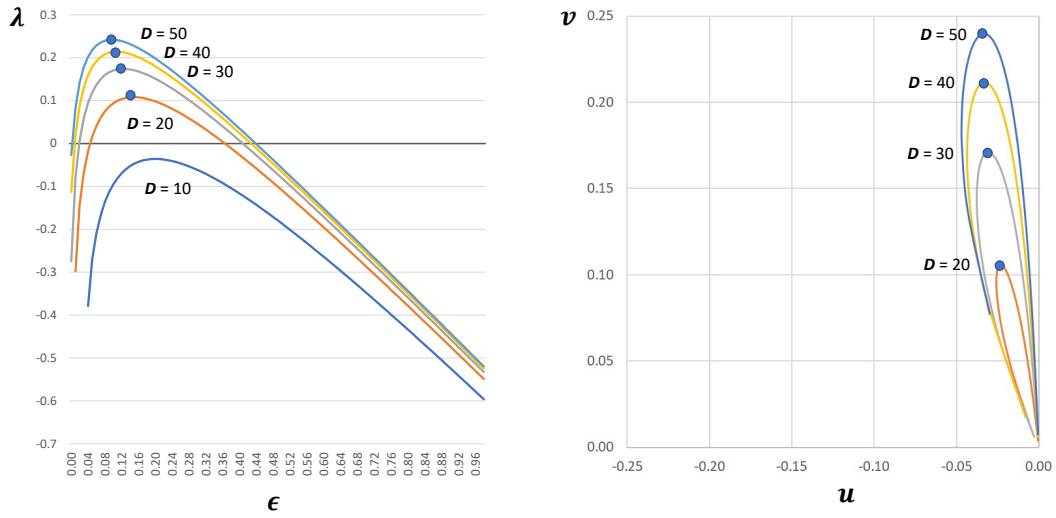


Figure 4.4: Eigenvalues and eigenvectors of linear Turing system

each cell is connected to its two neighbors. Let us label the cells with consecutive integers. Linear cyclic tissue means: $\mathbf{N}(i) = \{i - 1, i + 1\}$. Let $\omega_i \hat{\mathbf{e}}^*$ denote the position of cell i in phase space, with $\hat{\mathbf{e}}^*$ the eigenvector of the coherent mode. Equation 4.12 applied to linear tissue yields the following difference equation among cells:

$$\omega_{i+1} - 2(1 - \epsilon^*)\omega_i + \omega_{i-1} = 0 \quad (4.18)$$

Taking z-transform of the difference equation, we get:

$$z^2 - 2(1 - \epsilon^*)z + 1 = 0 \quad (4.19a)$$

$$z = (1 - \epsilon^*) \pm j\sqrt{1 - (1 - \epsilon^*)^2} \quad (4.19b)$$

with the roots on the z-transform ending up on the unit complex circle, the resulting sequence $\{w_n\}$ has the form $e^{j2\pi n/K}$ where K is the wavelength of the spatial periodic pattern on the tissue:

$$K = \frac{2\pi}{\arcsin(1 - \epsilon^*)} \quad (4.20)$$

Figure 4.5 show the evolution of a linear Turing system on a 1D cyclic tissue on both linear plots and polar plots, for sake of clarity. Sub-charts on top are linear plots where x-axis is the line-up of the nodes indexed by consecutive integers and

y-axis being the value of state variable v , whereas the sub-charts on the bottom are radial plot where consecutive nodes are lined up on the radial angle of the plot and the radial distance is set by the value of state variable v .

Figure 4.6 shows, for various parameter sets, the phase portrait of the dynamics in state space u, v with each cell plotted on the state space based on their long-term state values. It is noted that cells self-organize in state space to a straight line. This is because, the dynamics is contracting in one state variable and expanding in the other. The direction of alignment of cells on the phase portrait corresponds to the eigenvectors of the coherent mode that we derived analytically and depicted in Figure 4.4(b). We investigated the wavelength of each pattern (or number of lobes, given the tissue size is constant in all cases) against the analytical formula derived in Equation 4.20. The simulation results and analytical calculation, as provided in Figure 4.6, have precise match.

4.6 Attractors in cellular dynamics

4.6.1 Periodic attractor

Turing instability induced pattern formation is interesting because it shows that diffusion, which itself is a homogenizing phenomena, can lead to differentiation under certain bifurcation conditions as presented above. The differentiation is actually caused by the effective anti-diffusion (repulsion in state space) along one reactant. On the other hand homogenization (attraction in state space), which generates thickness for the patterns formed, is caused by the diffusion of the other reactant. This *yin-yang* of pushing and pulling in the state space is the key for pattern formation in the physical space.

In this section, we introduce a simpler cellular dynamics where the interplay between anti-diffusion vs diffusion creates robust periodic patterns. The model is inspired by the following game:

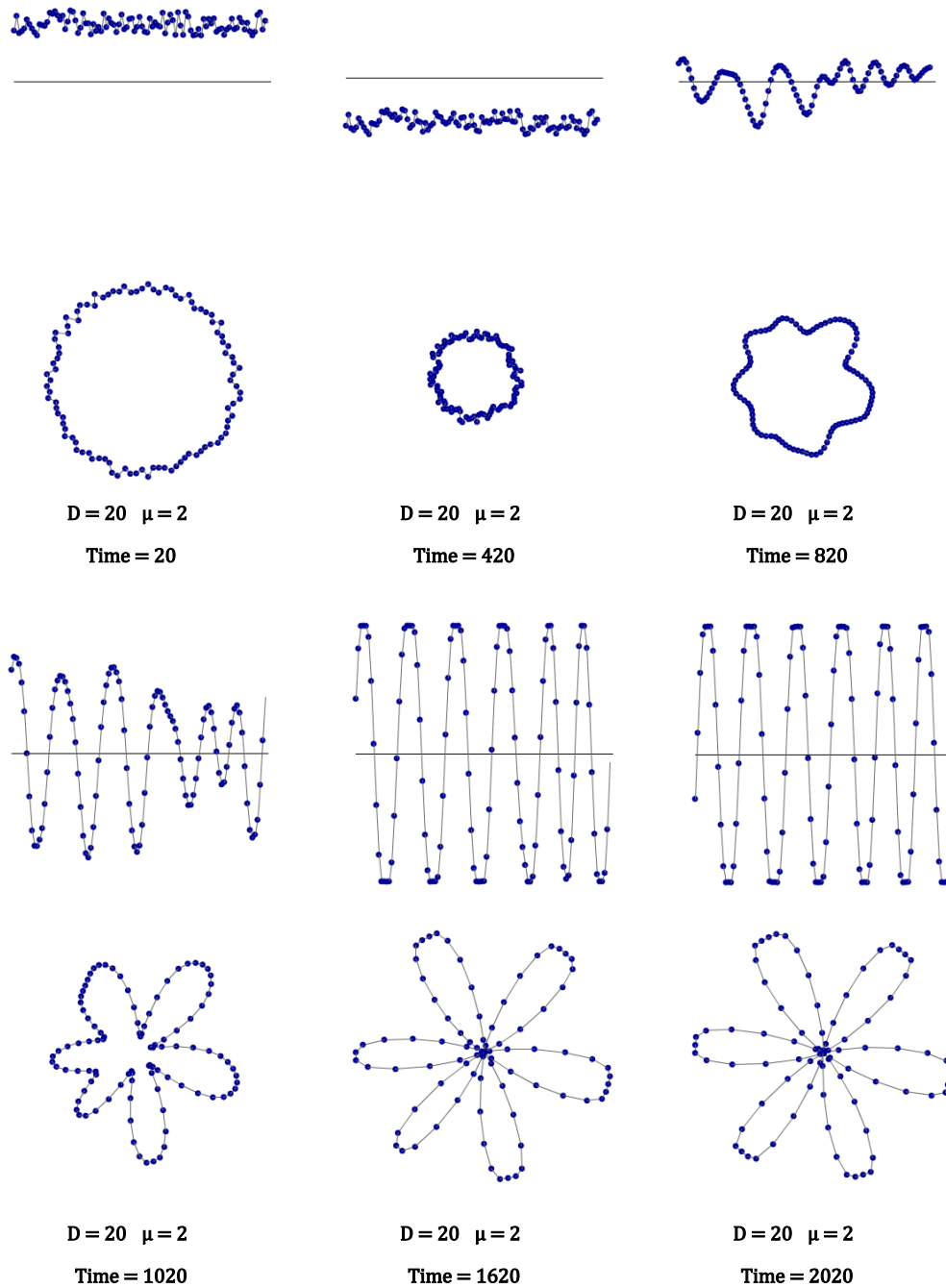


Figure 4.5: Time evolution of spatial organization in linear Turing system

Cells are organized on a 1D cyclic linear tissue. The charts depict the evolution of state value v across the tissue. Linear plots have x-axis as cell positions, y-axis as cells' v value. Polar plots depict the same data with rotational angle as cell positions, radial distance as cell's v value, capturing the cyclical organization of the 1D tissue (i.e. last cell is connected back to the first cell on the tissue). Relative diffusivity constant D is set at 20, common diffusivity constant μ is set at 2. Time evolution shows converging to a periodic attractor in c. 2000 time steps.

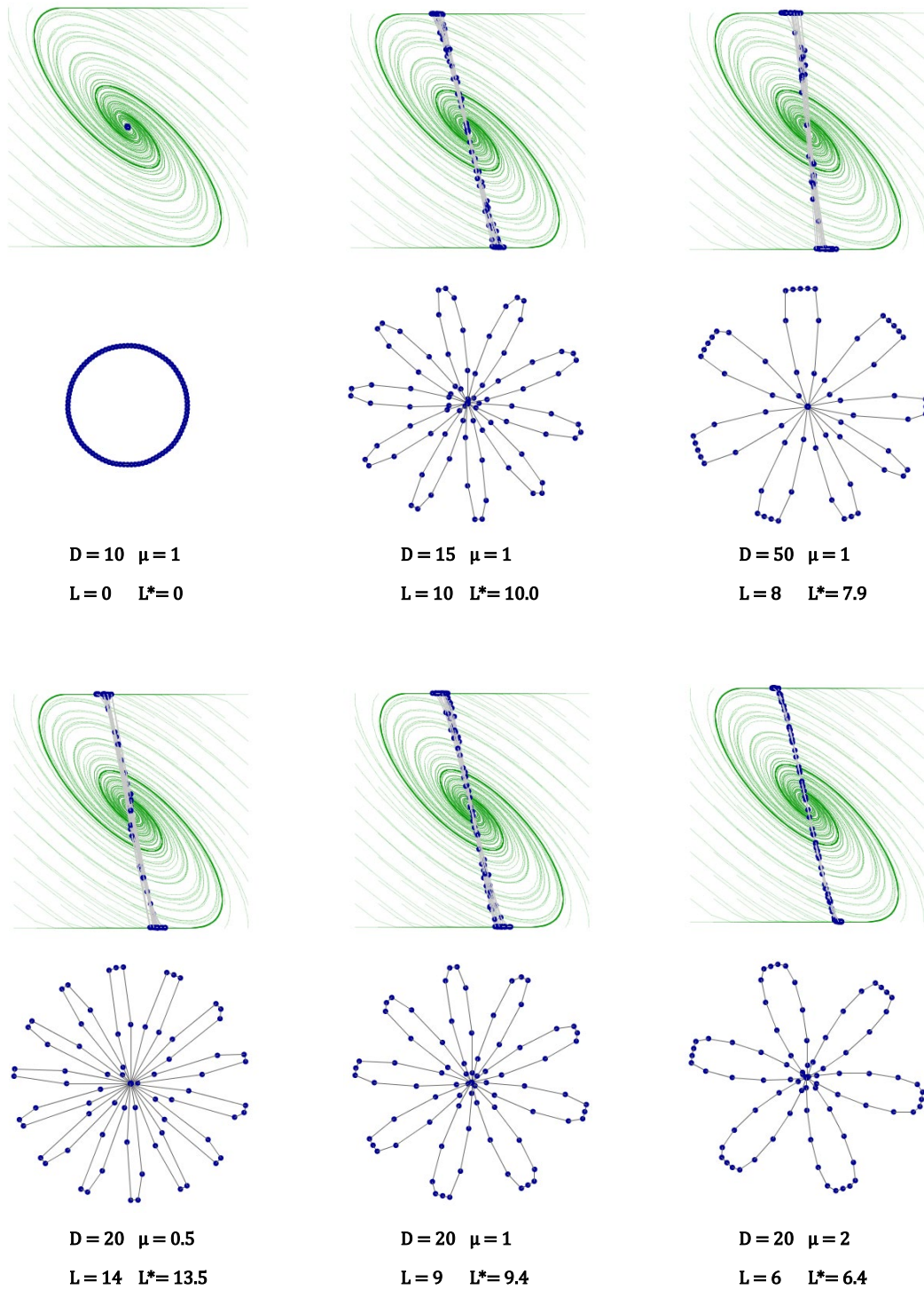


Figure 4.6: Spatial organization of linear Turing system

Organization in state space and spatial space. Relative diffusivity constant D is varied from 10-50. Common diffusivity constant μ is varied from 0.5-2. Number of lobes L in simulations is compared to our analytical calculations L^* .

Sharing-is-Caring Game

N identical agents are placed on a circle, each having two neighbors. They are given an initial amount of wealth, which is drawn randomly from $w_i \in (-1, 1)$, positive meaning asset, negative debt. At each year, players compare their wealth to the average of their neighbors. If they are richer, they gain an additional wealth equal to such difference. If they are poorer, they lose an amount equal to such difference. This is the rich-gets-richer, poor-gets-poorer set up. However, once such gain (loss) is realized, then each player shares a portion, $\beta \in (0, 1)$ of the gain with its neighbors, and receives from them their shared portions. This sharing back and forth is repeated P times. After that players put their final post-sharing gain (loss) into their bank accounts. This finishes the current year. The game continues year after year.

This rich-get-richer system coupled with sharing among neighbors results in strictly periodic *demarcation of society* with very rich and very poor bands of neighborhoods. Figure 4.7 shows the evolution of *sharing-is-caring* system, described above, on a 1D cyclic tissue in polar plots where the consecutive nodes are lined up on the radial angle of the plot and the radial distance is set by the value of state variable s . To visually demonstrate convergence to spatially periodic attractor over time, we plotted polar plots of each the full history in each diagram, by plotting radial plots one on top of the other at every 50 time step. Therefore, $Time = 100$ charts shows two polar plots (one for $T = 50$ and another for $T = 100$). Similarly for example $Time = 1000$ charts shows 20 polar plots (starting from $T = 50$, and then plotting on polar plot every 50 time increments). $Time = 7000$ charts shows the converged periodic pattern, starting from, faintly plotted random initial condition and evolving towards the attractor, with 140 polar plots drawn on top of each other showing the evolution of the pattern. Somewhat of an unusual charting method, Figure 4.7 captures the time evolution and convergence of the state variable S to a stable periodic spatial pattern.

We now analyze the spatial periodicity of this dynamics. Let s_i^t denote the wealth of agent i at beginning of year t , and $w_i^{t,0}$ the gain(loss) realized during the year, before sharing, and $w_i^{t,k}$ after k rounds of sharing. Then the Sharing-is-Caring Game is described by the following dynamics:

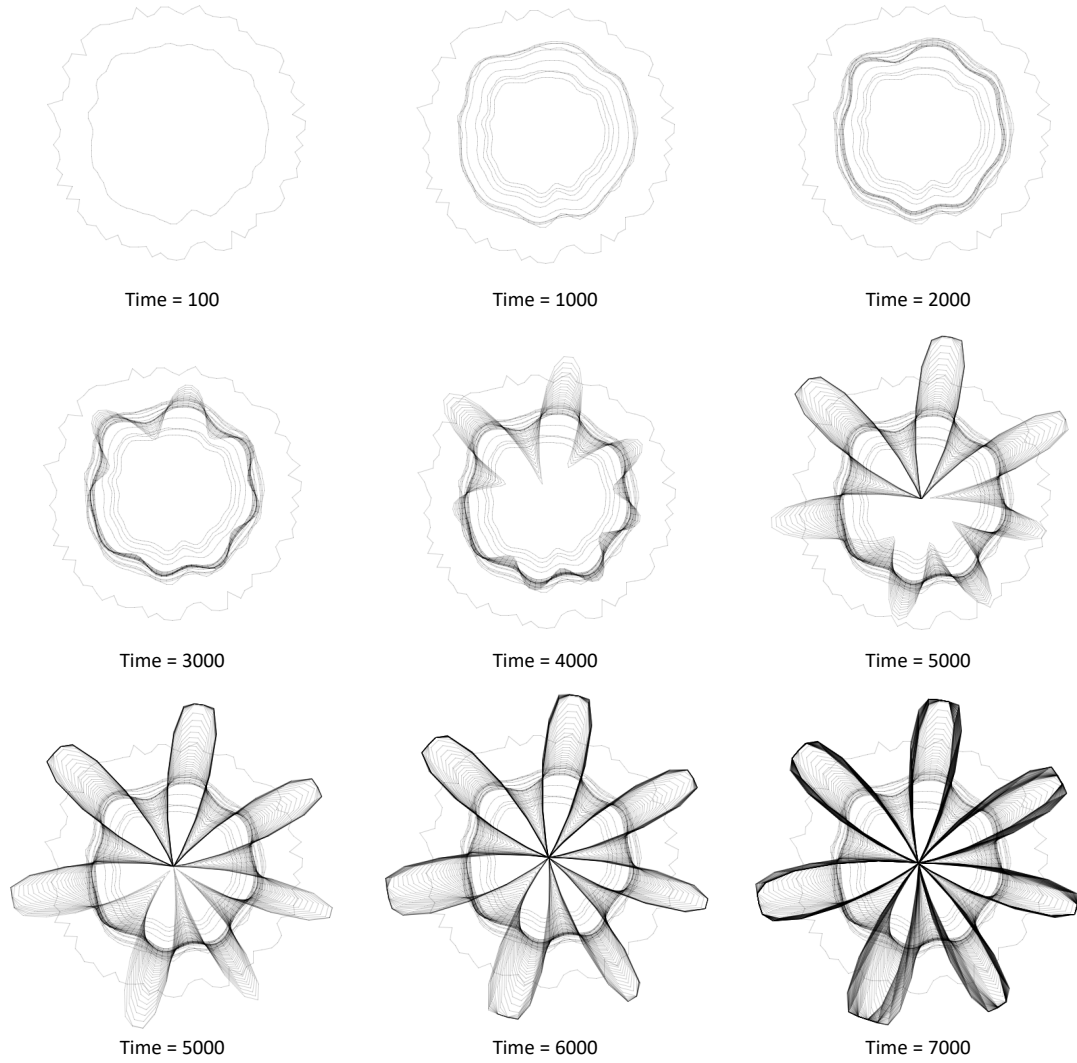


Figure 4.7: Time evolution of pattern formation in Sharing-is-Caring game

Sharing constant β is set at 0.5. Sharing repeat parameter P is set at 20. Tissue size is 100 cells. Evolution time taken up to 7000 steps. Each subplot shows the polar diagram of each cell's state value s as radial distance. History is depicted by plotting polar charts on top of each other for every 50 time increment.

$$w_i^{t,0} = -\frac{1}{2} s_{i-1}^t + 1 s_i^t - \frac{1}{2} s_{i+1}^t \quad (4.21a)$$

$$w_i^{t,k} = +\frac{\beta}{2} w_{i-1}^{t,k-1} + (1 - \beta) w_i^{t,k-1} + \frac{\beta}{2} w_{i+1}^{t,k-1} \quad (4.21b)$$

$$s_i^{t+1} = s_i^t + w_i^{t,P} \quad (4.21c)$$

Let us define two kernels to simplify notation:

$$\text{Differentiator} \quad d = \begin{bmatrix} -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}$$

$$\text{Homogenizer} \quad h = \begin{bmatrix} \frac{\beta}{2} & 1 - \beta & \frac{\beta}{2} \end{bmatrix}$$

Let $\mathbf{q} = \{s_i\}$ denote the space sequence, the dynamics can be stated as:

$$\mathbf{q} \rightarrow \mathbf{q} + \mathbf{q} * d * h^P \quad (4.22)$$

where $*$ denotes *convolution* operator, and $h^P := \overbrace{h * \dots * h}^{\text{P times}}$. We can represent the dynamics in spectral domain rather than space domain by taking Discrete-Time Fourier Transform of Equation 4.22. With convolution becoming multiplication upon Fourier transform, we get:

$$Q(\omega) \rightarrow Q(\omega) [1 + D(\omega)H^P(\omega)] \quad (4.23)$$

Upon each annual iteration of the Sharing-is-Caring Game, the spectral profile of the wealth distribution among the lined up agents become modified by multiplying by $M(\omega) := 1 + D(\omega)H^P(\omega)$. We now investigate the spectral profile of the modifier $M(\omega)$:

$$D(\omega) = \sum_n d[n]e^{-j\omega n} \quad (4.24a)$$

$$= -\frac{1}{2}e^{+j\omega} + 1 - \frac{1}{2}e^{-j\omega} \quad (4.24b)$$

$$= 1 - \cos(\omega) \quad (4.24c)$$

and similarly,

$$H(\omega) = \sum_n h[n]e^{-j\omega n} \quad (4.25a)$$

$$= \frac{\beta}{2}e^{+j\omega} + (1 - \beta) + \frac{\beta}{2}e^{-j\omega} \quad (4.25b)$$

$$= (1 - \beta) + \beta \cos(\omega) \quad (4.25c)$$

The spectral profile of the modifier $M(\omega)$ becomes:

$$M(\omega) = 1 + (1 - \cos(\omega)) [(1 - \beta) + \beta \cos(\omega)]^P \quad (4.26)$$

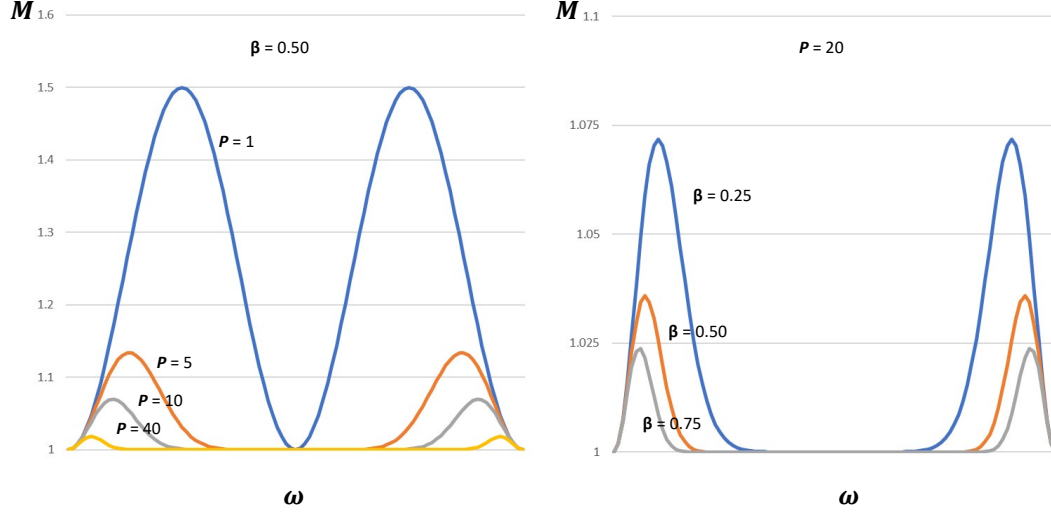


Figure 4.8: Sharing-is-Caring Game: Spectral modifier $M(\omega)$

Sharing percentage β and repeat value P are varied.

Figure 4.8 shows $M(\omega)$ for varying β (sharing percentage) and P (repeat value). The peaks of the modifier are resonance frequencies that magnify the spectral characteristics of wealth distribution $S(\omega)$ upon annual iteration, year-after-year. Resonant frequencies ω^* are identified by $\partial_w M(\omega)|_{\omega^*} = 0$, which gives:

$$\omega^* = \arccos\left(1 - \frac{1}{\beta(1+P)}\right) \quad (4.27)$$

Therefore the spatial periodicity of the rich-poor neighborhood pattern generated by the Sharing-is-Caring Game becomes:

$$K = \frac{2\pi}{\arccos\left(1 - \frac{1}{\beta(1+P)}\right)} \quad (4.28)$$

Figure 4.9 show simulation of pattern formation under Sharing-is-Caring Game dynamics with 100 agents. The number of lobes is compared to the analytical results given by Equation 4.28.

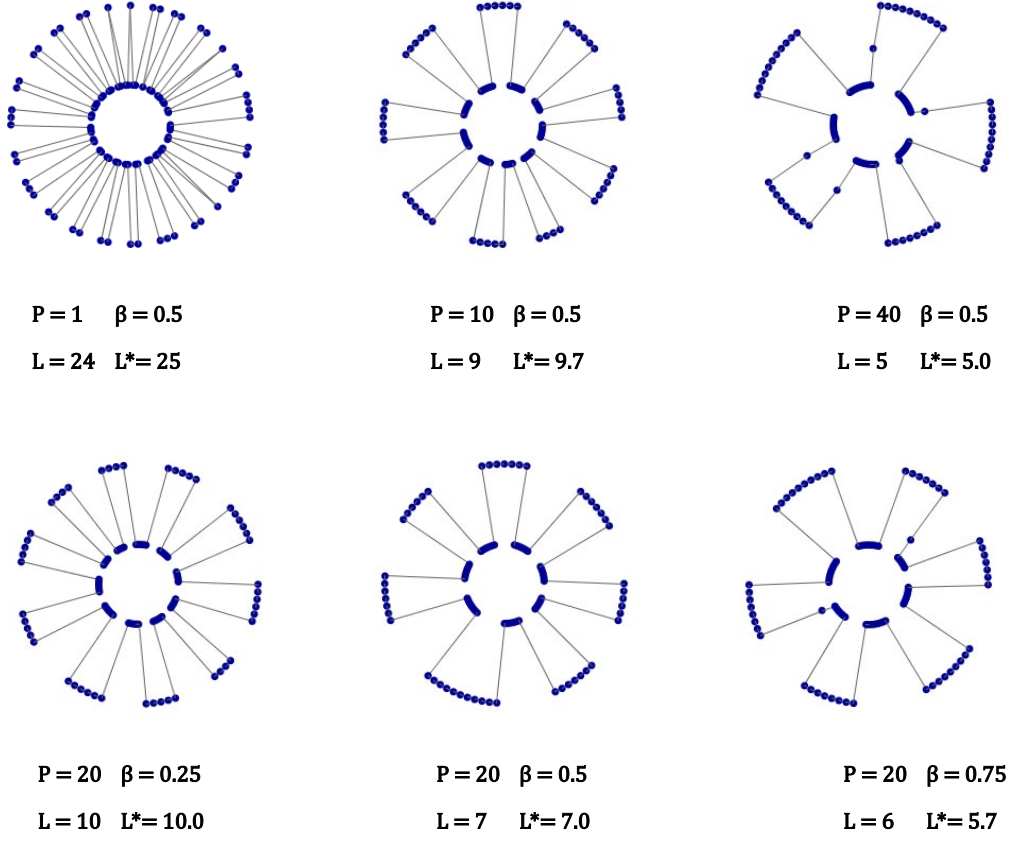


Figure 4.9: Spatial organization by periodic attractor in Sharing-is-Caring game

Sharing constant $\beta \in [0, 1]$ and sharing repeat parameter P are varied. Tissue size is set at 100 cells. Number of lobes L in simulations is compared to our analytical calculations L^* .

4.6.2 Singular attractor

In the previous sections, we investigated periodic pattern formation on a linear tissue. Here we introduce a novel reaction-diffusion-drift dynamics where the attractor is not a periodic pattern but a singularity that breaks tissue-wide symmetry. The concept is very important in form generation during embryogenesis where polarization of blastula in early embryo generates axis of development. This is discussed in Chapter 6. Our model is inspired by the following game:

Lego Tower Game

N identical agents are placed on a circle, each having two neighbors. They are each given a bag with random number of Lego bricks (ranging from 0 to 100 bricks). Each agent can exchange bricks with her neighbors and communicate only locally. Can we define a dynamics such that in finite time, all Legos are collected at one of the agents creating a tall tower. We introduce the Source-Field coupling model where local Lego tower at each agent is defined as a Source. Sources generate a tissue-wide Field by the reaction-diffusion dynamics detailed below. Then each agent exchange the bricks based on up-gradient of the Field. This is a novel chemotaxis model where advection (i.e. up-gradient drift) generates aggregation, but the Field formation dynamics ensures that all Lego bricks are aggregated at a single agent within finite time.

Reaction-Diffusion-Drift We formally define dynamics of singular attractor as follows: Let $S(\mathbf{x}, t)$ and $F(\mathbf{x}, t)$ denote the Source and Field concentrations in tissue $\mathbf{x} \in \Omega$ at time $t \in \mathbb{R}^+$

$$\partial_t F = \mathcal{R}(S - F) + D \nabla^2 F \quad (4.29a)$$

$$\partial_t S = -w \nabla \cdot (S \nabla F) \quad (4.29b)$$

where $\mathcal{R}(z) = \max(z, 0)$ is the *ramp* function, D the diffusion rate of the Field F , and w the drift rate of Source S . The heuristic is simple: Source pumps up the Field (through the ramp function); Field diffuses, effectively propagating the local Source level information throughout the tissue. Source then drifts up-gradient of the Field (i.e. advection).

Figure 4.10 illustrates the dynamics of Field given by Equation 4.29a. The reaction term is $\mathcal{R}(S - F)$, and diffusion term $D \nabla^2 F$. Reaction term alone would result in Field increasing in time in regions where it is less than Source, eventually reaching Source level. This is what we mean by Source pumps up Field. With diffusion of Field kicking in, the Field peaks would spread across the regions. The secondary Source peaks would eventually be surpassed by the diffusion of the Field. Once surpassed, due to the ramp function being zero in such regions, reaction term driven by local Source field would have no contribution to the dynamics of the Field. Therefore, the primary maximal Source peak would dominate the Field profile, and Field would effectively propagate that Source peak information across the tissue.

Figure 4.11 illustrates the dynamics of Source given by Equation 4.29b. For example, if we assume Field was stationary with a single peak, then in areas where Field gradient is pointing towards right and where Source is increasing, the divergence of $S\nabla F$ would be positive, hence $\partial_t S = -w\nabla \cdot (S\nabla F)$ would be negative and vice-versa. This would result in S profile drifting towards right. Converse would occur on regions where gradient of Field was pointing left. Therefore, Source would aggregate towards the point where gradient of Field turns, i.e. the Field peak. This is the advection mechanism.

Combination of Field dynamics running reaction-diffusion given above, and Source dynamics drifting up in Field gradient would eventually result in aggregation of all Source at a single point, which we call the singularity event. What follows below is a lengthy analytical derivation of an upper bound on expected time for a singular peak to form across a linear cyclic tissue.

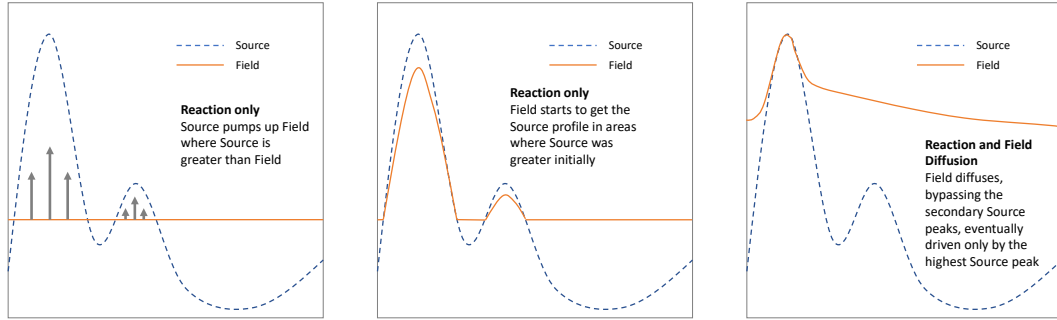


Figure 4.10: Illustration of Source-Field reaction and diffusion of Field

Assume *Source* profile is with two peaks, and *Field* is initialized to zero. Left chart shows *Field* being pumped towards *Source* as given by Field reaction term in Equation 4.29a, ramp function $\mathcal{R}(S - F)$. Without *Field* diffusion, we would get middle chart; in regions where S was higher than F initially, F would be pumped all the way to S level. Right chart shows when *Field* also diffuses. It will eventually surpass secondary S peak and would end up being driven solely by the primary, maximal S peak. In this dynamics, F would propagate peak Source information across the tissue.

Field formation Assume the system is initialized such that Field is zero across the tissue and Source is only present at specific locations, zero elsewhere. We seek to understand the Field formation between two Source points. First, we look at the formation of the Field with a single non-drifting Source at location $x = 0$, i.e. $S(0, t) = A$ and $S(x, t) = 0$ elsewhere, at all times. The dynamics simplifies to:

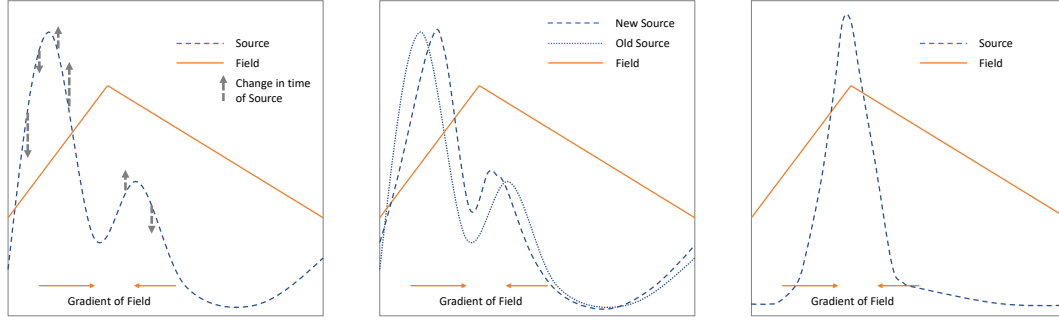


Figure 4.11: Illustration of drift of Source on gradient of Field

Assume *Field* profile is with a single peak and stationary. In areas where gradient of *Field* is towards right, if *Source* has an increasing profile, then divergence of $S\nabla F$ is positive. With $-w$ being negative, the *Source* would decrease; and *vice-versa* if *Source* has a decreasing profile then divergence of $S\nabla F$ is negative, with $-w$ being negative, the *Source* would increase. These are depicted in left chart. This results in *Source* profile drifting up gradient of *Field* as depicted in middle chart. Eventually, *Source* aggregates around the local *Field* peak, as depicted in right chart.

$$\partial_t F = D\nabla^2 F \quad (4.30a)$$

$$F(0, t) = A, \forall t \quad (4.30b)$$

This is actually the heat transfer equation on a linear rod where one end is kept at constant temperature (i.e. attached to a heat bath). The solution of it can be found by integrating the Gaussian distribution which itself is the solution of diffusion with a Source at a single location at time zero only.

$$F(x, t) = A \operatorname{erfc} \left(\frac{1}{2\sqrt{D}} \frac{x}{\sqrt{t}} \right) \quad x \geq 0, t \geq 0 \quad (4.31)$$

where $\operatorname{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty e^{-x^2} dx$, is the complementary error function. Next we seek to tackle the main formative case where Field is driven by two distinct Sources, separated by distance d , (*Double Source Diffusion*):

$$\partial_t F = D\nabla^2 F \quad (4.32a)$$

$$F(0, t) = A, F(L, t) = B, \forall t \quad (4.32b)$$

Equation 4.32 is the heat transfer on a linear rod with constant temperature heat baths at two ends of the rod, L being the length of the rod. It can be analytically solved by Fourier Transform, and the solution can be represented by a series summation [71]. Figure 4.12 shows formation of Field, pumped by a single Source with the closed form solution given in Equation 4.31 and pumped by two Sources with no closed form solution, other than the Fourier series expansion.

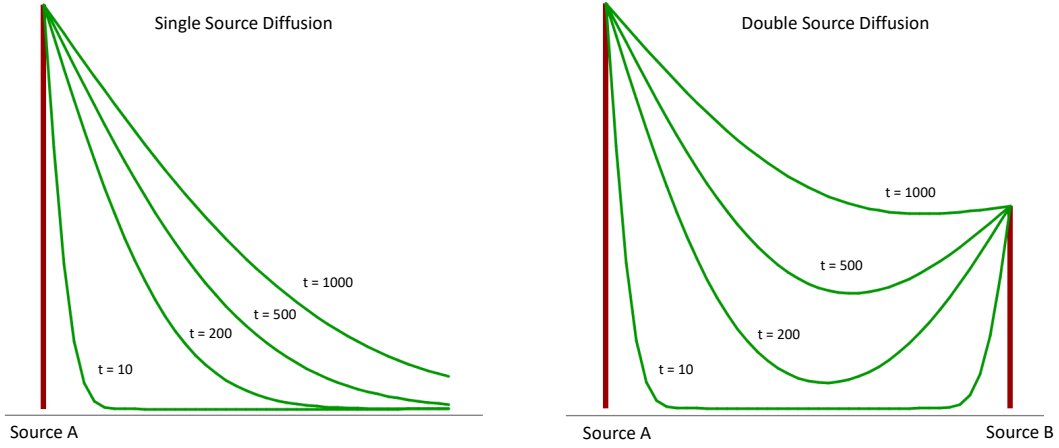


Figure 4.12: Single and double source diffusion

(a) Time evolution of Field driven by a single constant Source. (b) Time evolution of Field driven by two constant Sources.

We are seeking to find the time required for the Field established by the double source diffusion dynamics to reach a profile with positive gradient in the direction of B to A, upon which Source B will start drifting towards Source A due to Equation 4.29(b). Since, we do not have a closed form solution to Double Source Diffusion dynamics, we introduce a novel approximation technique.

Piecewise Linear Field We conceptualize the Field formed between two constant sources to consist of two piecewise linear segments as depicted in Figure 4.13. The dynamics has two phases: In Phase I, there are two anchor points $\mathbf{p}_A$ and $\mathbf{p}_B$, that move along the x-axis so that the area under PWL field is equal to the area under actual field; we call this *Single Source PWL Dynamics*. In Phase II, the two anchor points emanating from Source A and B, meet in the middle, creating a single anchor point $\mathbf{p}$ which moves in y-direction and along x-axis based on the steepness of the piecewise linear segments $\overline{A\mathbf{p}}$ vs $\overline{B\mathbf{p}}$; we call this *Double Source PWL Dynamics*.

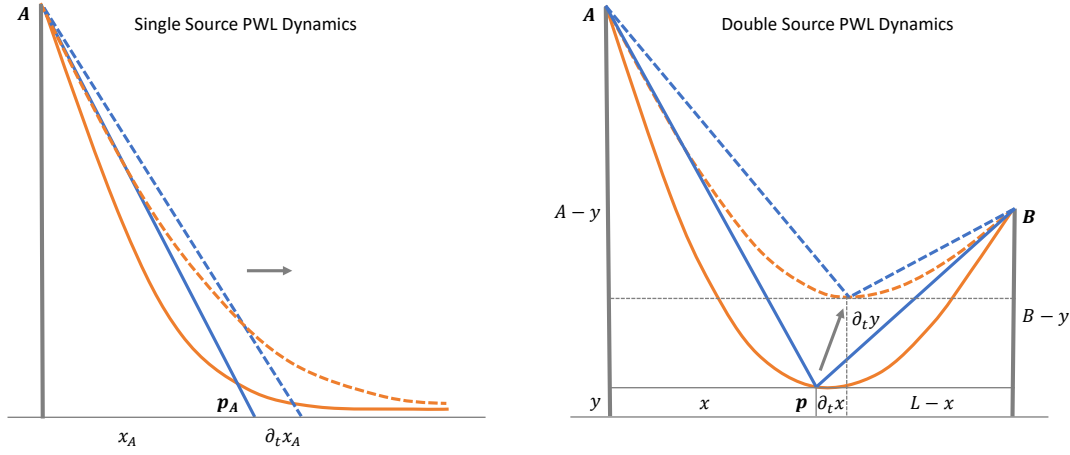


Figure 4.13: Piecewise Linear Field approximation

First we tune Single Source PWL Dynamics to Single Source Diffusion by equating the areas below the PWL field to that of actual field generated by Single Source Diffusion.

$$\frac{A x_A(t)}{2} = \int_0^{\text{inf}} A \operatorname{erfc}\left(\frac{1}{2\sqrt{D}} \frac{z}{\sqrt{t}}\right) dz \quad (4.33a)$$

$$\frac{B (L - x_B(t))}{2} = \int_{-\text{inf}}^L B \operatorname{erfc}\left(\frac{1}{2\sqrt{D}} \frac{L - z}{\sqrt{t}}\right) dz \quad (4.33b)$$

The integrals in Equation 4.33 a) and b) clearly have same value (easy to see by change of variable $y = L - z$), and can be readily solved with integration by parts, which results in closed form solution for Phase I as:

$$x_A = L - x_B = 4\sqrt{\frac{D}{\pi}} \sqrt{t} \quad (4.34)$$

which leads to the differential equation for Single Source PWL dynamics and the time required for Phase I to complete, i.e. $\mathbf{p}_A$ to collide with $\mathbf{p}_B$ as:

$$\partial_t x = \frac{8D}{\pi} \frac{1}{x} \quad (4.35a)$$

$$\tau^I = \frac{L^2}{64} \frac{\pi}{D} \quad (4.35b)$$

Next we derive the differential equation for Double Source PWL dynamics with the principle that (i) velocity of $\mathbf{p}$ at any given position equals to pressure generated by $\overline{\mathbf{A}\mathbf{p}}$ and $\overline{\mathbf{B}\mathbf{p}}$ line segments which are equal to their steepness, and (ii) $\mathbf{p}$ drifts

on x-axis to reach its equilibrium at any time where steepness of $\overline{\mathbf{A}\mathbf{p}}$ equals that of $\overline{\mathbf{B}\mathbf{p}}$, which leads to the following set of coupled differential equations for x and y coordinates of $\mathbf{p}$:

$$\frac{A-B}{2}\partial_t x + \frac{L}{2}\partial_t y = \frac{4D}{\pi} \left(\frac{A-y}{x} + \frac{B-y}{L-x} \right) \quad (4.36)$$

$$\partial_t x = \frac{4D}{\pi} \frac{2}{A-B} \left(\frac{A-y}{x} - \frac{B-y}{L-x} \right) \quad (4.37)$$

upon equilibrium condition for x reached, y dynamics reduces to:

$$\partial_t \beta = -\frac{16D}{\pi L^2} \frac{(\Delta + 2\beta)^3}{\Delta^2 + (\Delta + 2\beta)^2} = f(\beta) \quad (4.38)$$

where $\beta := B - y$ and $\Delta := A - B$. Equation 4.38 gives the decay dynamics of $\mathbf{p}$ along the y -axis towards $\mathbf{B}$. By observing the shape of $f(\beta)$ function, we can linearize it to find a close lower-bound on decay rate:

$$\partial_t \beta \gtrsim -\frac{16D}{\pi L^2} \left(2\beta + \frac{\Delta}{2} \right) \quad (4.39)$$

We are now in a position to give a closed form solution to slower-bound y -dynamics:

$$\beta(t) = \left(B + \frac{\Delta}{4} \right) e^{-\frac{32D}{\pi L^2} t} - \frac{\Delta}{4} \quad (4.40)$$

We are interested in the upper-bound on time required to end Phase II of PWL dynamics, i.e. when $\mathbf{p}$ reaches $\mathbf{B}$.

$$\tau^{II} = \frac{\pi L^2}{32D} \ln \left(1 + \frac{4B}{\Delta} \right) \quad (4.41)$$

When Phase II ends, then the drift part of Reaction-Diffusion-Drift Equation 4.29 kicks in with Source B drifting up-gradient on Field towards Source A. We simplify the drift dynamics, without losing the spirit of the model, as constant unit velocity. Therefore, time required to complete Phase III (drift) dynamics of the model, upon which Source B merges into Source A as:

$$\tau^{III} = L \quad (4.42)$$

We now have a closed form tight upper-bound on *time-to-merger* for two sources of magnitudes A and B , separated by distance L :

$$\begin{aligned}\tau &= \tau^I + \tau^{II} + \tau^{III} \\ &= \frac{\pi L^2}{64D} \left(1 + 2 \ln \left(\frac{A + 3B}{A - B} \right) \right) + L\end{aligned}\quad (4.43)$$

Figure 4.14 (a) shows the actual Field evolution vs Piecewise Linear Field approximation, while (b) shows comparison of time-to-merger generated by simulation vs analytically derived closed form tight upper bound based on PWL model, for varying distances between two sources L , ratio of their magnitudes B/A and diffusion rate D .

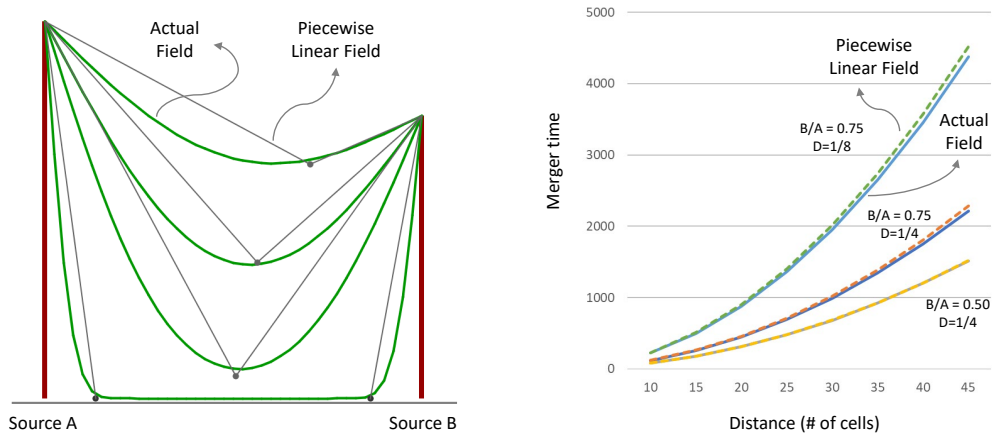


Figure 4.14: Time evolution of Actual Field vs Piecewise Linear Field

(a) Field samples from time evolution (b) Merger time comparison.

Expected time to singularity Having derived a tight upper-bound on merger time between two sources, we can now find an upper-bound on expected time for a randomly distributed source profile on a tissue to aggregate into a single position through our Reaction-Diffusion-Drift dynamics, so called a *singularity event*.

Let $\mathcal{L}$ be a random variable with uniform probability distribution in $[0, K]$; and $\mathcal{A}, \mathcal{B}$ two independent random variables with uniform probability distributions in $[p, q]$. First we find the expected half life of the sources, i.e. expected time-to-merger of randomly picked two sources A and B , separated by L :

$$T_{p,q,K} = E(\tau) = E \left(\frac{\pi}{64D} \mathcal{L}^2 \left(1 + 2 \ln \frac{\mathcal{A} + 3\mathcal{B}}{|\mathcal{A} - \mathcal{B}|} \right) + \mathcal{L} \right) \quad (4.44)$$

using $\int_a^b \ln z \, dz = (z \ln z - z)|_a^b$ and $\int_a^b z \ln z \, dz = (\frac{1}{2}z^2 \ln z - \frac{1}{4}z^2)|_a^b$, we calculate:

$$T_{p,q,K} = \frac{\pi}{64D} \frac{K^2}{3} (1 + 2 (F_{p,q} - F_{q,p} - G_{p,q})) + K \quad (4.45)$$

where $F_{a,b}$ and $G_{a,b}$ are given as:

$$F_{a,b} = \frac{1}{6} 16a^2 \left(\ln(4a) - \frac{1}{2} \right) - \frac{1}{6} (a + 3b)^2 \left(\ln(a + 3b) - \frac{1}{2} \right) \quad (4.46a)$$

$$G_{a,b} = -(a - b)^2 \quad (4.46b)$$

To calculate the expected time-to-singularity $\mathring{\mathbf{T}}$, we consider an approximation case where number of sources are halved by each iteration. Expected time-to-singularity is given as the sum of half-life for each iteration. Starting from a random initial distribution for Source magnitudes drawn from a uniform initial distribution in $[0, 1]$ at each cell location, at each iteration, the distribution of distance between Sources double on average. We approximate Source distribution at a given iteration drawing from uniform probability in $[0.5 - \sigma, 0.5 + \sigma]$. At each iteration the variance of Source magnitudes reduce by at most $\sqrt{2}$, because of addition of magnitudes of two Sources that merge. At each half-life iteration, the distance between subsequent Sources and their magnitude distribution gets updated as $K \rightarrow 2K$ and $\sigma \rightarrow \sigma/\sqrt{2}$, respectively. Then, an upper-bound on expected time-to-singularity $\mathbf{T}$ is given as:

$$\begin{aligned} \mathring{\mathbf{T}} < \mathbf{T} &= \sum_{m=1}^M T_{p_m, q_m, K_m} \\ p_m &= 0.5 - \sigma/\sqrt{2^m}, \quad q_m = 0.5 + \sigma/\sqrt{2^m} \\ K_m &= K * 2^m, \quad K_M < N(\text{tissue length}) \end{aligned} \quad (4.47)$$

We investigated the analytically derived upper-bound on expected time-to-singularity in Reaction-Diffusion-Drift system against observed time in a series of computer simulations. Diffusion constant D is set at 0.25, field values are initialized to zero, and source values are initialized randomly, drawn from uniform probability distribution in $[0, 1]$. 250 simulation run is performed for each tissue size (i.e. of cells), ranging

from 25 to 100 cells. The analytically derived upper bound is shown as dashed blue line in Figure 4.15. The orange line shows the average of actual time to formation of singularity over 250 simulation for each tissue.

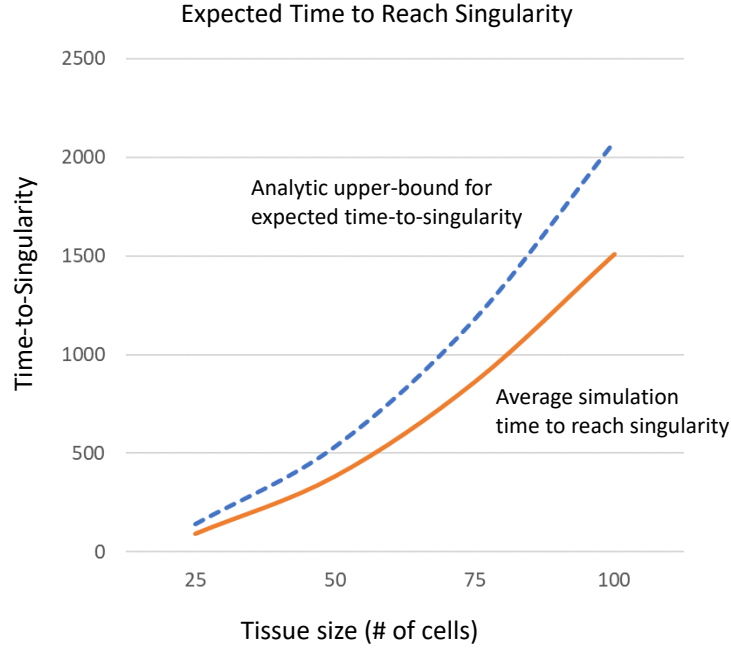


Figure 4.15: Singular peak formation

Analytic upper bound for expected time-to-singularity versus average time to reach singularity of 250 simulations on each tissue size. Diffusion rate $D = 0.25$, initial source for each cell is independently drawn from uniform probability distribution in $[0, 1]$

4.6.3 Dipolar attractor

In the previous section, we introduced singular attractor dynamics based on Source-Field coupling given in Equation 4.29. The dynamics results in a global singular attractor where all Source is combined at a single cell, starting from almost all initial configurations. An analytical upper bound on expected time to converge to singularity is provided. In this section, we are introducing the *Dipolar Attractor* model which is based on two types of Sources, we call *Green* and *Red*. Each Source type runs a singular attractor dynamics on a standalone basis, which result in a singular Red peak and a singular Green peak. These two dynamics are coupled through a newly introduced *dipole-field* H . The idea is that the dipole-field extends from Green peak (positive) to Red peak (negative). We show that dipole-field forms

a linear profile between the peaks. We then impose a drift on Red source on the gradient of dipole-field, which results in Red peak drifting away from Green peak (effectively a repulsion). The steady state of this dynamics is two peaks formed at exactly polar opposite sides of the tissue.

Let ${}_rS$ and ${}_gS$ denote the Red and Green Source types with corresponding Fields ${}_rF$, ${}_gF$, and H the dipole-field. The dynamics of dipole attractor model is given by:

$$\partial_t {}_gF = \mathcal{R}({}_gS - {}_gF) + D\nabla^2 {}_gF \quad \text{Green Field formation} \quad (4.48a)$$

$$\partial_t {}_gS = -w\nabla \cdot ({}_gS\nabla {}_gF) \quad \text{Green Source aggregation} \quad (4.48b)$$

$$\partial_t {}_rF = \mathcal{R}({}_rS - {}_rF) + D\nabla^2 {}_rF \quad \text{Red Field formation} \quad (4.48c)$$

$$\partial_t {}_rS = -w\nabla \cdot ({}_rS\nabla {}_rF) \quad \text{Red Source aggregation} \quad (4.48d)$$

$$-w_H\nabla \cdot ({}_rS\nabla H) \quad \text{Red Source drift away from Green} \quad (4.48e)$$

with dipole-field H formed by the following dynamics. Let $\Delta S = {}_gS - {}_rS$.

$$\partial_t H = D_H\nabla^2 H + \begin{cases} +\mathcal{R}(\Delta S - H) & \Delta S \geq 0 \\ -\mathcal{R}(H - \Delta S) & \Delta S < 0 \end{cases} \quad (4.49)$$

The dynamics of dipolar attractor model is therefore specified by the constants of dynamics, where D , D_H are the diffusion speeds of field and dipole-field; and w , w_H are the drift speeds of source on field and dipole-field, respectively.

Figure 4.16 shows the simulation of dipole attractor model on a linear cyclic cellular tissue of 200 cells with random initial Red and Green Source configuration. Green and Red sources execute singular peak formation as per the singular attractor model. This is the fast dynamics. The dipole-field formation is a slower dynamics and field H is shown in the diagram as line graph. The drift of Red source away from Green source on the net gradient of the dipole-field is the slowest dynamics and results in Red source peak drifting away from Green peak to the furthest away position on the linear cyclic tissue. At the steady state, the system has Red and Green poles that are positioned on polar opposite sides with a linear dipole-field formation between those two poles.

It is important to note that the steady state attractor of the system has a linear

dipole-field between the two poles. This field provides a very effective mechanism for each cell to know its distance from the poles. This gives each cell a *positional information* on the self-organized polar coordinates.

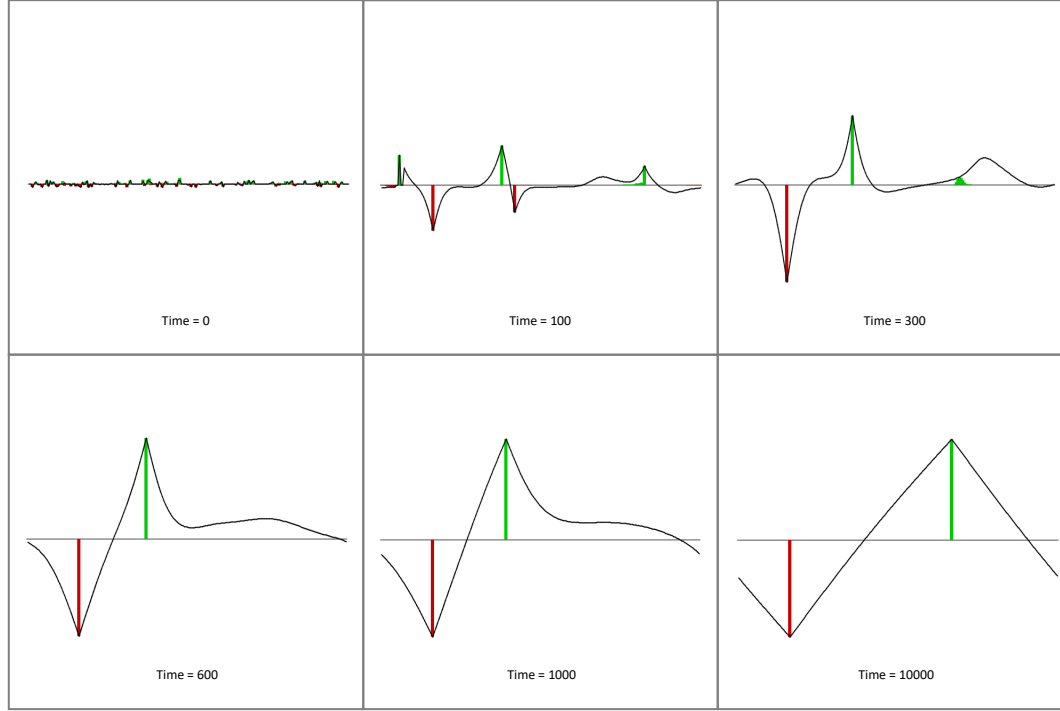


Figure 4.16: Dipolar attractor time evolution

The horizontal axis is the linear tissue. Red and Green bars show respective Source levels at any cell along the tissue. The black line is the dipole-field that extends from Red Source to Green Source. At the steady-state, the system gets attracted to a system state where two poles are formed at opposite sides of the linear circular tissue, and the dipole-field stretches linearly between these peaks, providing positional information to each cell in reference to the poles.

4.7 Segmentation and body plans

In the previous section, we introduced the dipole attractor model where steady state attractor of the system is two poles (Red and Green) at opposite sides of the tissue with a linear dipole-field established between them which effectively provides positional information. In this section, we use this positional information to generate precise segmentation of the tissue that leads to self-organized body plans.

Let us introduce a dynamics for *broadcasting* of the peak Source level across the tissue by a fast diffusing messenger S_g^* and S_r^* , for Green and Red sources, respectively:

$$\partial_t {}_gS^* = \mathcal{R}({}_gS - {}_gS^*) + D^*\nabla^2 {}_gS^* \quad (4.50a)$$

$$\partial_t {}_rS^* = \mathcal{R}({}_rS - {}_rS^*) + D^*\nabla^2 {}_rS^* \quad (4.50b)$$

As a broadcasting mechanism, the diffusion is fast, and soon after the formation of the Red and Green source peaks, all cells get communicated these peak levels, i.e. $\forall x \in \Omega$, ${}_gS^*(x) = {}_gS^{\text{peak}}$ and ${}_rS^*(x) = {}_rS^{\text{peak}}$. Now, we can locally calculate the *distance-to-pole* metrics ${}_gd$ and ${}_rd$ of each cell, using the fact that dipole-field settles at $H \in [-{}_rS^{\text{peak}}, +{}_gS^{\text{peak}}]$ as per Equation 4.49:

$${}_gd(x) = \frac{{}_gS^* - H(x)}{{}_gS^* + {}_rS^*} \in [0, 1] \quad (4.51a)$$

$${}_rd(x) = \frac{{}_rS^* + H(x)}{{}_gS^* + {}_rS^*} \in [0, 1] \quad (4.51b)$$

$${}_gd(x) + {}_rd(x) = 1 \quad (4.51c)$$

With self-organized polar peaks and a locally distributed distance-to-pole metric at each cell, the cells can differentiate according to their local positional information, generating segmentations and body plans. Figure 4.17 show variants of body plans with two segments (fly wings), three segments (insect legs) and four segments (arachnid legs) generated on a linear circular tissue, starting from random initial Source distributions, running the dipole attractor dynamics. For physical depiction of body plan formation a simplistic toy model is used where the cyclic tissue is embedded on a plane as a circle, with each cell's initial distance to center being same. This can be thought as an abstraction of the blastula stage of an early embryo. As the dynamics ensues, the physical position of each cell evolves as a radial distance from the center $r(x), x \in \Omega$. This radial distance is set as a function of the local distance-to-pole metric.

We used *notch* functions, parametrized by position, height and width of a given

organ:

$$\psi_{\text{head}} : \begin{pmatrix} \alpha_{\text{head}} : \text{position of head} \\ \beta_{\text{head}} : \text{height of head} \\ \gamma_{\text{head}} : \text{width of head} \end{pmatrix} \quad (4.52)$$

The notch function $\psi_{\text{head}}(x)$ is centered at position α , has a gain of β at the central position and decays linearly to zero within a width of γ , given as below:

$$\psi_{\text{head}}(x) = \begin{cases} \beta(1 - \frac{|x-\alpha|}{\gamma}) & |x - \alpha| \leq \gamma \\ 0 & |x - \alpha| > \gamma \end{cases} \quad (4.53)$$

Simple body plan of a self-organized toy organism is specified by a set of notch filters that define differential growth based on local positional information; e.g. one for head, tail and each segment pair with the full set defined as *body plan parameter set*:

$$\Psi = \begin{pmatrix} \alpha_{\text{head}} & \alpha_{\text{tail}} & \alpha_{\text{wing1}} \\ \beta_{\text{head}} & \beta_{\text{tail}} & \beta_{\text{wing1}} & \dots \\ \gamma_{\text{head}} & \gamma_{\text{tail}} & \gamma_{\text{wing1}} \end{pmatrix} \quad (4.54)$$

The full body plan is generated as the sum of all organ defining notch filters:

$$r(x) = \sum_{\text{organs}} \psi_{\text{organ}}(x) \quad (4.55)$$

Starting from a random initial distribution of Source across the tissue, first head and tail poles are formed, followed by establishment of dipole-field stretching between these two poles. Poles drift away on the gradient of the dipole-field and position themselves at the opposite sides of the tissue. Then each cell grows radially based on the body plan vector and distance from the poles. In *dragon-fly* body plan example, the tissue between poles is segmented into two which represent the positions of the wings. In *ant* and *arachnid* (e.g. spider) body plans, the tissue is segmented into three and four segments, respectively, which represent the positions of the legs. Note that due to circular symmetry of the tissue, broken by a single axis formed by Red-Green pole pair, the body plans necessarily have left-right symmetry. This symmetry can be broken by introducing a vector form of dipole system. This

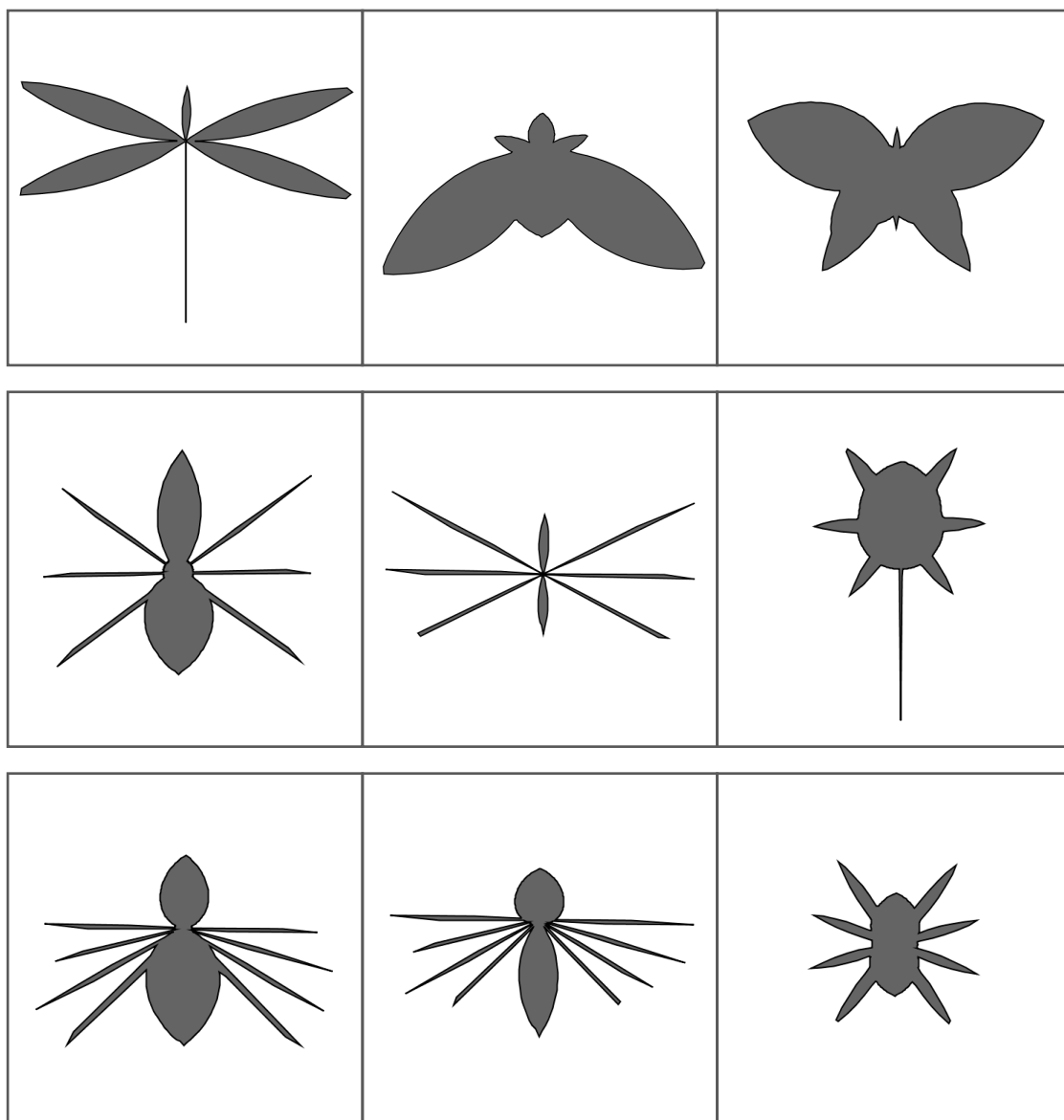


Figure 4.17: Self-organized body plans

(a) 2-segment variants: Fly wings, (b) 3-segment variants: Insect legs, (c) 4-segment variants: Arachnid legs. Parameter sets used in generating these bodyplans are provided in Figure 4.18.

will be introduced in Chapter 6 where we provide our Vector-Reaction-Diffusion-Drift model for self-organized embryogenesis.

4.8 Dynamics on 2D mesh graphs

In the previous sections, we investigated cellular dynamics on a 1D cyclic graph including linear Turing systems, periodic attractor, singular attractor, and dipolar attractor. Below we show simulation results for dynamics on regular 2D cyclic grid.

Fly 1				Fly 2				Fly 3			
	position	width	growth		position	width	growth		position	width	growth
Overall			0.070	Overall			0.070	Overall			0.150
Head	1.000	0.100	0.300	Head	1.000	0.300	0.200	Head	1.000	0.050	0.100
Tail	0.000	0.010	1.000	Tail	0.000	0.150	0.050	Tail	0.000	0.100	0.200
Wing1	0.600	0.100	1.000	Wing2	0.600	0.100	0.100	Wing1	0.700	0.250	0.400
Wing2	0.400	0.100	1.000	Wing1	0.200	0.300	0.500	Wing2	0.300	0.150	0.150
Insect 1				Insect 2				Insect 3			
	position	width	growth		position	width	growth		position	width	growth
Overall			0.070	Overall			0.000	Overall			0.150
Head	1.000	0.200	0.500	Head	1.000	0.050	0.400	Head	0.000	0.350	0.070
Tail	0.000	0.400	0.400	Tail	0.000	0.050	0.400	Tail	1.000	0.010	0.900
Leg1	0.300	0.017	0.700	Leg1	0.300	0.017	0.900	Leg1	0.300	0.040	0.200
Leg2	0.500	0.017	0.700	Leg2	0.500	0.017	0.900	Leg2	0.500	0.040	0.200
Leg3	0.700	0.017	0.700	Leg3	0.700	0.017	0.900	Leg3	0.700	0.040	0.200
Arachnid 1				Arachnid 2				Arachnid 3			
	position	width	growth		position	width	growth		position	width	growth
Overall			0.040	Overall			0.020	Overall			0.100
Head	0.000	0.400	0.300	Head	0.000	0.500	0.100	Head	0.000	0.350	0.050
Tail	1.000	0.400	0.500	Tail	1.000	0.200	0.400	Tail	1.000	0.350	0.050
Leg1	0.500	0.018	0.700	Leg1	0.500	0.018	0.700	Leg1	0.200	0.040	0.200
Leg2	0.583	0.018	0.700	Leg2	0.583	0.018	0.700	Leg2	0.400	0.040	0.200
Leg3	0.667	0.018	0.700	Leg3	0.667	0.018	0.700	Leg3	0.600	0.040	0.200
Leg4	0.750	0.018	0.700	Leg4	0.750	0.018	0.700	Leg4	0.800	0.040	0.200

Figure 4.18: Parameter sets used for generating bodyplans

As depicted in Figure 4.17 using Equations 4.54, 4.55

In Chapter 6, we will be running dynamics on general 2D mesh graphs.

Figure 4.20 shows a simulation result of 10000 cells arranged on 2D cyclic mesh (i.e. regular torus) for Sharing-is-Caring game which self-organize into Turing stripes. The governing dynamics equations are given as:

$$w_a^{t,0} = \sum_{b \in N_a} (s_a^t - s_b^t) |s_a^t - s_b^t|^\alpha \quad (4.56a)$$

$$w_a^{t,k} = (1 - \beta) w_a^{t,k-1} + \beta \langle w^{t,k-1} \rangle_{N(a)} \quad k \in [1, K] \quad (4.56b)$$

$$s_a^{t+1} = s_a^t + w_a^{t,K} \quad (4.56c)$$

Equations 4.56 are same as Equation 4.22 of Sharing-is-Caring dynamics that was described earlier, except for the multiplier term $|s_a^t - s_b^t|^\alpha$ introduced. For $\alpha = 0$, we get the pattern formation of sharing-is-caring dynamics. For this case, Equation 4.56a is linear with regards to a cell's state difference from its neighbors. The resulting pattern as depicted in Figure 4.20a has no tissue-wide elongations The

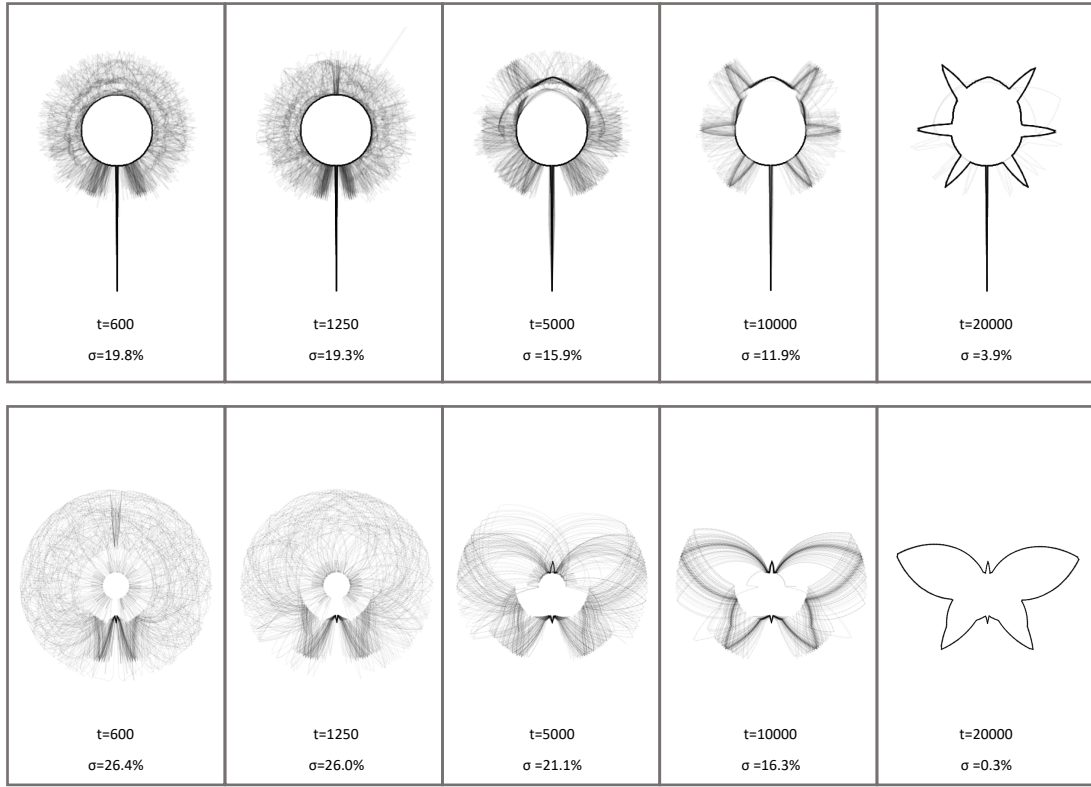


Figure 4.19: Embryogenesis simulation for fly and insect variants

100 independent simulations of insect embryogenesis are overlaid showing strong convergence to a definitive body plan with diminishing variance. The parameter set for generating the bodyplans are given in Figure 4.18: *Insect 3* and *Fly 3*.

symmetry of the tissue is not broken; there is no special direction.

However, when we increase α from zero to one, non-linearity is introduced to Equation 4.56a with regards to a cell's state difference from its neighbors. The resulting dynamics, interestingly, generates tissue-wide broken symmetry with elongations in directions that emerge randomly from the very initial fluctuations of state distributions of the initial state. This interesting phenomena of self-organized tissue-wide anisotropy from locally isotropic reaction equations is depicted in examples in Figures 4.21.

4.9 Self-organized cellular automata

In the previous sections, we developed our singular attractor and dipolar attractor dynamics. Starting from continuous space-time reaction-diffusion-drift equations, we discretized both space and time establishing cellular dynamics. In this domain,

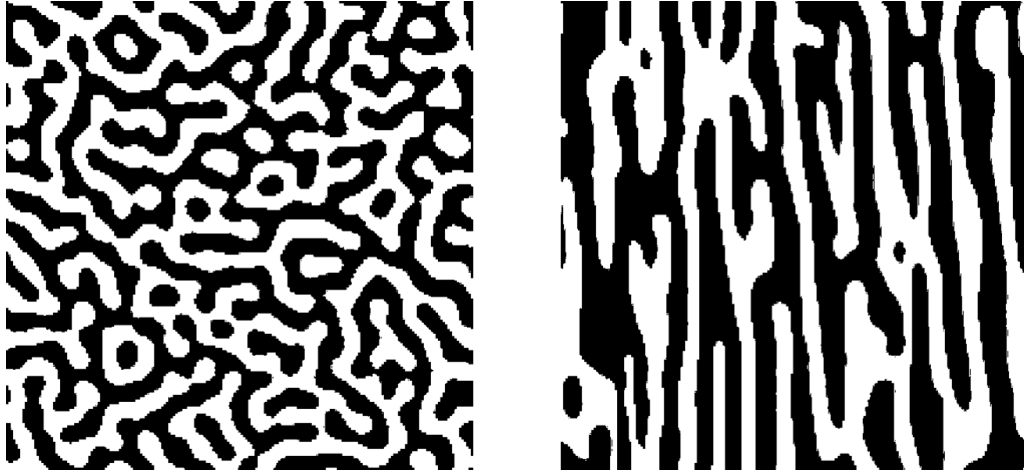


Figure 4.20: Sharing-Caring-Game pattern formation on 2D regular grid

(a) $\alpha = 0$ isotropic pattern formation. (b) $\alpha = 1$ self-generated anisotropy across tissue.



Figure 4.21: Self-generated tissue-wide anisotropy, $\alpha = 1$

singular attractor is proposed as a dynamics that results in morphogen to get concentrated at a singular peak on the tissue. This peak, and its counter peak in the case of dipolar dynamics, form the polar axis system for position information used in embryogenesis. In this section, we further discretized the state space, i.e. assign finite states to morphogen levels which moves the formalism to that of cellular automata. In this discretized state-space-time model, our reaction-diffusion-drift equations become simple cell level automata computations which are now represented as pseudo-code of the automata given in Algorithm 1. This algorithm in effect converges to well studied distributed computing problem of *leader election*.

The leader election is the process of designating a single computational agent as the organizer of some task distributed among several others in a distributed computing environment where the computational agents are typically considered as nodes on a graph, and their peer-to-peer communication links denoted as edges of the

graph. Initially, all network nodes are equivalent and post leader election, one node emerges as the recognized leader throughout the network. Given the applicability of this core subroutine in a wide range of distributed computing applications, it is well studied both from engineering and computation theory perspectives. The algorithms range based on the topology of underlying network and timing schemes. Ring networks with known size to all computational agents with synchronous communications is the most basic one. The problem has been first generalized to 2D mesh structures, then to arbitrary graphs and eventually to dynamical generalized graphs where connectivity fluctuates in time. Any leader election protocol requires some means of symmetry breaking from the initial state of nodes. In id-based systems each node has a unique identifier, hence the system has no symmetric global state, each node is fully-differentiated by the identifier. Leader-election algorithms in such networks is less demanding. In *uniform* leader election protocols, all nodes are identical, the initial state is symmetric. In these algorithms symmetry is broken by randomization. The singular attractor procedure given in Algorithm 1 applies to a general mesh that is growing over time where nodes (i.e. a cell of the tissue) do not possess a unique id. Hence, it is a uniform leader election protocol which is primed with randomization of morphogen concentrations as the initial system condition.

We developed our singular attractor concept using reaction-diffusion-drift formalism in continuous space-time systems with continuous morphogen concentrations as state variables. This is a mathematical domain quite apart from distributed computing protocols. Yet, upon discretizing morphogen states and cellularizing space-time, we arrived at an algorithmic representation of the singular attractor dynamics which is equivalent to leader election protocol used in distributed computing.

The research in leader election on distributed computing understandably focuses on designing efficient protocols from engineering point of view and correctness from computing theoretical point of view. In contrast, our singular attractor is an extension of a chemical system and therefore simplistic in protocol terms. Our singular attractor algorithm is not proposed as an alternative to existing leader-election protocols in distributed computing, but as an outcome of reaction-diffusion-drift dynamics that converged to protocols in distributed computing. Our goal in providing some selected research below is to note the similarities to our resultant algorithm

but not to provide a comprehensive review of leader election protocols in distributed computing.

In 1990, Korach *et al.* [72] presented a general technique for efficient leader finding in distributed asynchronous networks which mapped the problem to efficient serial traversing of the network and provided a bound on the message complexity of the proposed algorithm. In 1992, Dolev *et al.* [73] provided a leader election and ranking protocol for dynamic networks (i.e. topology changing) with uniform node processors. They applied a random ingredient to computation at nodes to break the symmetry of the system at each step of computation, rather than our randomization approach of initial condition.

In 2013, Ingram *et al.* [74] presented an algorithm for leader election in a network with dynamical communication topology and asynchronous clocks. The algorithm's heuristic is as follows: Each computational node is endowed with a *height*. The edges of the graph are set to point to higher neighbors by local comparison. Maximum height is communicated and edge pointers are updated. If due to topology changes some nodes lose their links to emerging leader, then such node reverses all of its incident edges which triggers the start of search mechanism. They also provided a correctness proof and stability condition for their algorithm. The resultant method is very similar to our leader election method where local up-gradient directions are stored at each cell as pointers to cell with highest local gradient.

We now provide the finite-state version of our singular attractor. This is the domain of cellular automata where each cell executes a look-up table based on its current state (finitely many) and its neighbors' states. Our goal is not to provide cellular automata formalism, but to show that the singular attractor and dipolar attractor dynamics can be implemented with finite cell states leading to self-organized cellular automata.

Let us recap the singular attractor model with two continuous state variables Source and Field, $S, F \in [0, 1]$:

$$\begin{aligned}\partial_t F &= \mathcal{R}(S - F) + D\nabla^2 F \\ \partial_t S &= -w\nabla \cdot (S\nabla F)\end{aligned}$$

where $\mathcal{R}(x) = \max(x, 0)$. ∇ is a differential operator which works best when applied

to continuous state variables. If we were to do a straightforward discretization of state variables such as $S, F \in \{0, 1, \dots, P\}$, we would quickly fall into indifference problem where the differential operator at small values would always give zero output. This leads to localization of the dynamics with characteristic spatial length equal to P which becomes a sensitivity delimiter for the dynamics. We do not get tissue-wide singular attractor with this simplistic state discretization approach. The governing equations need to be carefully modified to avoid the indifference problem yet preserving the heuristics of the original dynamics. Instead of keeping the analog Field value as a state-variable for cell i and executing a drift of Source on up-gradient of Field, we store $FieldMax$ $F_i^* \in \{0, 1, \dots, P\}$ and gradient direction $GradDir$ $G_i \in \{0, 1, \dots, |N_i|\}$ at each cell. G_i is simply a pointer to the neighbor that has the highest F^* level. The $FieldMax$ level is diffused digitally throughout the tissue with the logic that if a cell's neighbor has higher $FieldMax$ than the cell itself, it inherits that value and sets the $GradDir$ pointing at that cell. $Dist$ D^* is a local variable that tracks distance to local peak. The pseudo-code of the automata is given in Algorithm 1.

Algorithm 1 is a digital version of our singular attractor model that generates tissue-wide singularity. This simple automata executed by each cell in parallel results in cellular automata to self-organize into a single pole across the whole tissue. The simulation results are shown in Figure 4.22.

Lastly, one can easily build the dipolar attractor model in automata form by having two types of Source, $_rS$, $_gS$, each running independently Algorithm 1 and $_rS$ drifting away from $_gS$ as given in Algorithm 2. The simulation results for automata running Algorithm 2 are shown in Figure 4.23. Two poles, at maximally separated position on the tissue is self-organized (upper right figure). Each cell have positional information based on distances to the poles, $_rD$ and $_gD$. We can differentiate cells using this positional information. Various markings shown in Figure 4.23 are self-organized based on self-generated positional information, without reference to an extrinsic coordinate system of the tissue.

```

initialize  $S$  to random  $\in \{0 \dots P\}$ ;
initialize  $F^*$  and  $D^*$  to 0;
while simulation runs do
  for all cells do
    /* Diffusion here */
    if Neighbors have higher  $F^*$  then
      Determine the neighbor with highest  $F^*$ ;
      Inherit his  $F^*$ ;
      Inherit his  $D^* + 1$ ;
      Point GradDir to him;
    end
    /* Reaction here */
    if  $S$  is higher than  $F^*$  then
      Pump  $F^*$  up to  $S$  level;
      Set GradDir to self;
      Set  $F^*$  and  $D^*$  to 0;
    end
    /* Drift here */
    Transfer  $S$  to neighbor pointed by GradDir
  end
end

```

Algorithm 1: Digital singular attractor

```

initialize  ${}_rS, {}_gS$  to random  $\in \{0 \dots P\}$ ;
initialize  ${}_rF^*, {}_gF^*$  to 0;
while simulation runs do
  /* Singular attractor dynamics here */
  for all cells do
    run Reaction-Diffusion-Drift on  ${}_rS$ ;
    run Reaction-Diffusion-Drift on  ${}_gS$ ;
  end
  /* Dipolar drift here: Red drifts away from green */
  for every  $T$  time steps do
    for all cells do
      Transfer  ${}_rS$  to neighbor with highest  ${}_gD$ ;
    end
  end
end

```

Algorithm 2: Digital dipolar attractor

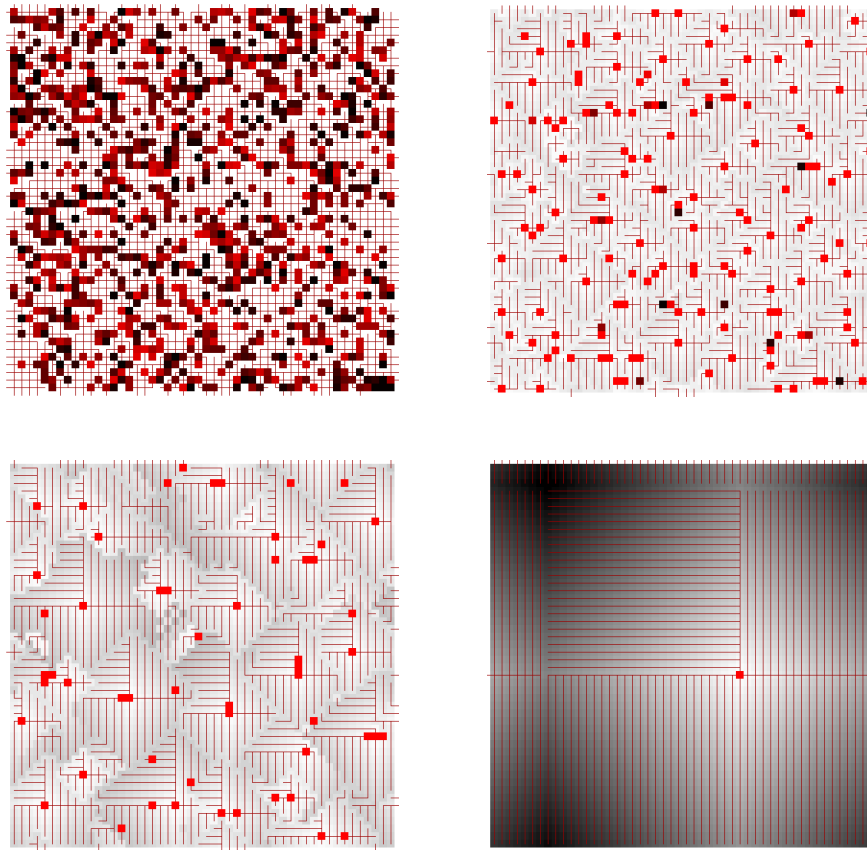


Figure 4.22: Singular attractor on 2D regular grid

Shades of red dots corresponds to Source level at each location. If Source level is zero, there is no dot. The dash lines shows the direction of the local gradient pointer. If one follows these dash lines, he reaches a local peak. Shades of gradient depict the distance to local peak field values.

4.10 Discussion

Turing's instability which caused a locally stable spatially homogeneous system to become unstable and generate patterning was a counter intuitive concept at the time. Pattern seemed to arise out of an undifferentiated state due to spatial fluctuations being amplified and aggregated. We see a similar phenomenon in leader election. Locally communicating uniform agents with no differentiation, other than randomized initial state fluctuations, result in a global order being self-organized, a leader emerging throughout the network. We have shown with analytical formalism that such systems with non-linearity and spatial positive feedback loops can self-organize randomness into order.

In this chapter, we introduced cellular systems executing dynamics in discrete

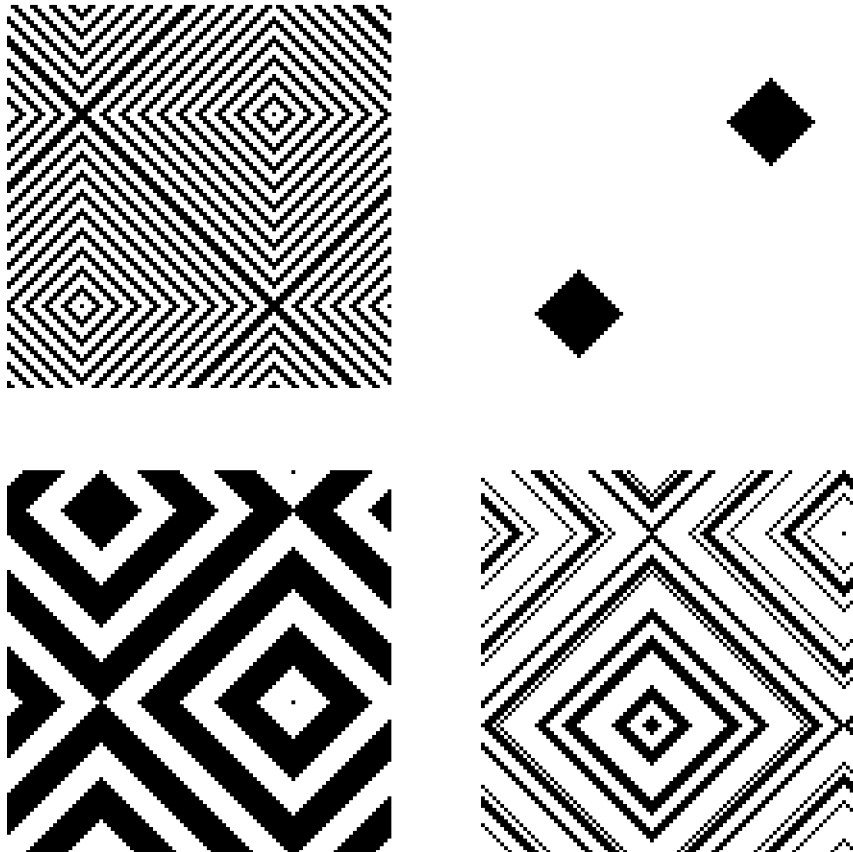


Figure 4.23: Self-organized cellular automata

Digital dipolar attractor on 2D regular grid results in self-organized poles. Distance-to-poles positional information used for various cell differentiation schemes.

space-time that self-organize into periodic patterns and singular poles with dipole field extending between them. These polar peaks and the dipole field are called *organizers* and *morphogen fields*, respectively, in developmental biology. In cellular systems analyzed so far, each cell executes the same cellular dynamics with the same parameter set; i.e. all cells are of same type. What if the cells could have different types (i.e. execute different dynamics and growth functions) and such types were selected by the cell itself based on morphogen cues. Starting from a common *omnipotent* type, each cell could determine its fate into new possible types based on such cues. In developmental biology this is called *cell differentiation*. Upgraded with such regulatory circuitry, the cells can now self-organize and self-differentiate across the tissue generating what we call the *organism*. This is the topic of next chapter.

Chapter 5

Gene regulation

“Things do not exist other than in arrangements”

Tractatus – Ludwig Wittgenstein

In Chapter 4, we proposed cellular systems in discrete space-time that self-generated organizing poles which led to positional information across the tissue. It was shown that the dipolar attractor system can robustly generate self-organized body plans. In this chapter, we offer a cell regulatory model where each cell can differentiate into possible types using positional cues generated by the dipolar attractor. This regulatory model finds its biological realization in *gene regulatory networks* (GRN). The central construct developed and analyzed is our *computron* model which is a finite state automaton where each state is defined as the configuration pattern of the underlying gene expressions. The attractor configurations of the machine correspond to cell types. Depending on positional cues given by the morphogenic field, computron’s attractor configurations change. This is the epigenetic pathway for gene expressions being altered by inter cellular dynamics. This interplay between gene expression and morphogen field formation can result in self-organization of tissue in nested complex forms.

5.1 Terminology

In biology, a set of molecular regulators (DNA, RNA, or proteins) interact with each other and with substances in the cell to set the gene expression levels of proteins. Some of these expressed proteins, called transcription factors, activate other genes,

creating a network of regulation chain which is called the *gene regulatory network* (GRN). Genes can be viewed as nodes in the network, with inputs being proteins such as transcription factors. The gene is either turned ON (expressing its protein) or OFF. This is called the *gene expression state*. This binary state of the gene is determined as a function of the value of the regulators at such time. In *Boolean network* formalism for GRNs, the state transitions are defined as logic functions of the boolean states of the impacting genes with all genes updating their states synchronously.

We use the terminology *computron* for a Boolean network with the extension that computron can have an external input driving its state transitions and also can have probabilistic state transition, rather than only deterministic. We represent GRNs as computrons where each gene represents a node on the *generating graph* of the computron and each directional edge of the generating graph shows a relation with the impacting gene to the impacted. The combined impact of expression states of all impacting genes of a given gene is defined as a boolean logic function which we call a *connector*. The ensemble of all binary states of each gene is called the *configuration state*.

A *finite state automaton* (FSA) is an abstract computational machine where the machine can be in exactly one of a finite number of states at any given time, and changes its state in response to some external inputs. We show that any FSA is equivalent to a computron with the configuration states of the computron corresponding to the finite states of the automaton.

In this chapter, we diverge from the central theme of the Thesis, cellular self-organization, and seek to characterize the regulation of a cell by its GRN from a dynamical system theory point of view. Typically dynamical systems are defined on metric spaces using vector field operators. A finite state automaton on the other hand does not exist in a metric space and state transitions are not expressed by vector field operators but as transitions tables, driven by external inputs. Therefore, as far as we know, there does not exist a theory for bridging the state transition dynamics of an FSA into a metric dynamical systems formalism. We seek to provide such a bridge in this chapter.

This challenge pushes us to use set-theoretic concepts which can be technical.

The reward is that we embed the FSA into a metric space, called the *diversity space*, that captures the similarities of gene expression patterns of the equivalent computron. A *metric space* consists of a set of elements (in our case each expression pattern of all genes is an element) and a distance between any two elements satisfying the intuitive notion of distance (i.e. it is positive; satisfies triangle inequality; and zero for distance to self). Pattern similarity on the generating graph of the computron of each gene expression pattern is such distance in our formalism. With the similarity metric in hand, we start using dynamical systems theory concepts to characterize the dynamics of the FSA.

The typical characterization of a dynamical system is through its attractors and basins of attraction, as well as structure of the transient states. The Lyapunov characterization specifies the future of a system starting from close-by initial states. A system is called chaotic if close-by states would eventually diverge. The entropy of the system is the richness of entanglement of such initial states that end up in different attractors. We will be providing formal definitions of these characterizations and then apply them to dynamics of computron as the model for gene regulatory networks.

Whilst chaoticity is a well-defined concept in dynamical systems theory, complexity is not. Within this chapter, we further divulge into seeking a definition of complexity that captures the entanglement of the basins of attraction. Clearly, a system that maps all initial states to a single final state is not complex, nor is a system that maps each initial state to itself. We present a novel complexity definition that captures the information required to describe the basins of attraction of a system, higher the more complex. With this definition, we classify the complexity of some basic GRNs.

Finally, we embark on characterizing the behavior of probabilistic GRNs. This opens up a whole new domain of mathematics described through Markov chains. Although admittedly lost in the overall theme of this Thesis, the section on probabilistic GRNs has very interesting concepts and results, including fuzzy partitions and ergodic decomposition, which are formally defined. These new concepts allow us to define chaoticity and complexity of probabilistic GRNs.

5.2 Gene regulatory networks

In Equation 4.54, we specified the body plan of a specie with the parameter set Ψ that defined the segment growth, given the positional information. To be able to self-organize a tissue into further detailed forms, we implement cell differentiation by a regulatory model. Our regulatory model consists of a finite number of abstract cell types, each having its own unique parameter sets. *DNA* consists of the vector of parameter sets $\Psi = [\Psi_0, \Psi_1, \dots]$ each corresponding to a cell type. Whilst all cells have the same DNA that carries the full vector of parameters for all cell types, depending on differentiation dynamics, each cell would be executing one of the type parameters at a given time.

Type of a cell is the attractor configuration state of the cell's GRN. Figure 5.1 depicts the cell type description as configuration of ON-OFF states of genes as an ensemble. Starting from cell type 0 (which is the omnipotent cell that can transition into any other cell type), the cell types change driven by the external inputs which are local morphogen levels. We consider each cell type to be a state of a finite state automaton. States of the automaton changes based on the external inputs, the morphogen levels. Such state transition function Φ is called the fate function of the given cell state.

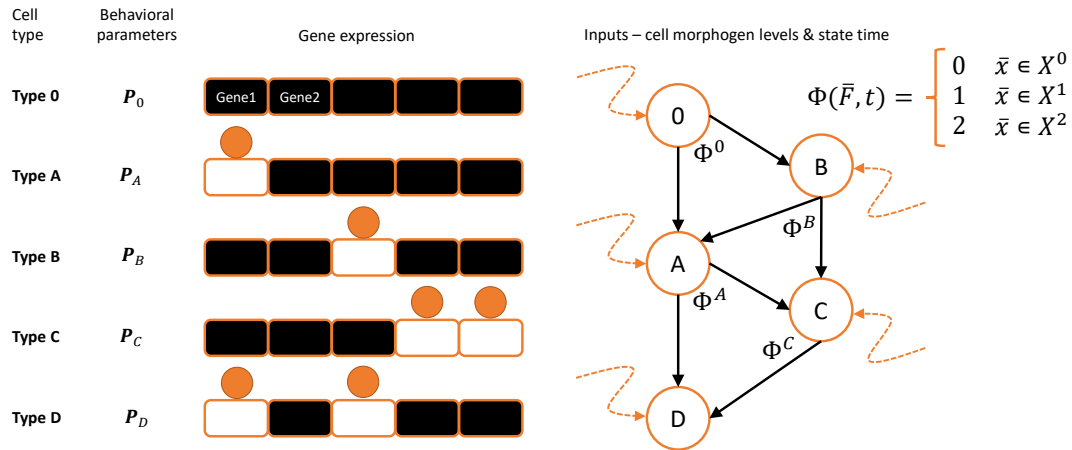


Figure 5.1: Cell regulation by gene expression switching

Various theoretical models for cell differentiation using biochemical networks have been proposed. The *Boolean network* model was introduced by Kauffman [2], and have since been used extensively for modeling GRN dynamics, including

fission yeast cell cycle, *Arabidopsis thaliana* floral morphogenesis and mammalian cell cycle [75]. In these models, differentiated cell types correspond to attractors of the underlying non-linear dynamical system. Globally coupled metabolic reaction network with active transport of chemicals has been shown to generate clustering of differentiated cells [76]. Neural-network type models were also considered [12], [15]. Spatially-extended GRN models have been recently used for pattern generation [77]. A network of morphogens reacting through neural-network type function where spatial connectivity is either a traditional diffusion or direct contact induction (effectively a discretized cellular model) have categorical pattern generating capabilities [78].

Complex pattern formations do not need to be due to complicated hierarchical and layered GRNs. Cells that have the same genome can differentiate into clusters of different behavior based on their developmental history, indicating that epigenetic lineage is likely to be switching the genetic stage of a cell [79]. Spatial complexity emerges when epigenetic information provides feedback to the GRN. Developmental mechanisms that have modular re-use of genetic tool-set, coupled with epigenetic feedback can generate a large variation of adaptive morphologies. Stuart Kauffman was amongst the first biologists to use Boolean networks to model gene regulatory networks [9].

1. Each gene, each input, and each output is represented by a node in a directed graph in which there is an arrow from one node to another if and only if there is a causal link between the two nodes.
2. Each node in the graph can be in one of two states: ON or OFF.
3. For a gene, ON corresponds to the gene being expressed. For inputs and outputs, ON corresponds to the substance being present.
4. Time is viewed as proceeding in discrete steps. At each step, the next state of a node is a Boolean function of the current states of the nodes with arrows pointing towards it.

The global state of the Boolean network is the configuration pattern of ON-OFF local states of its nodes. We use the terminology *computron* for boolean networks representing the gene regulation.

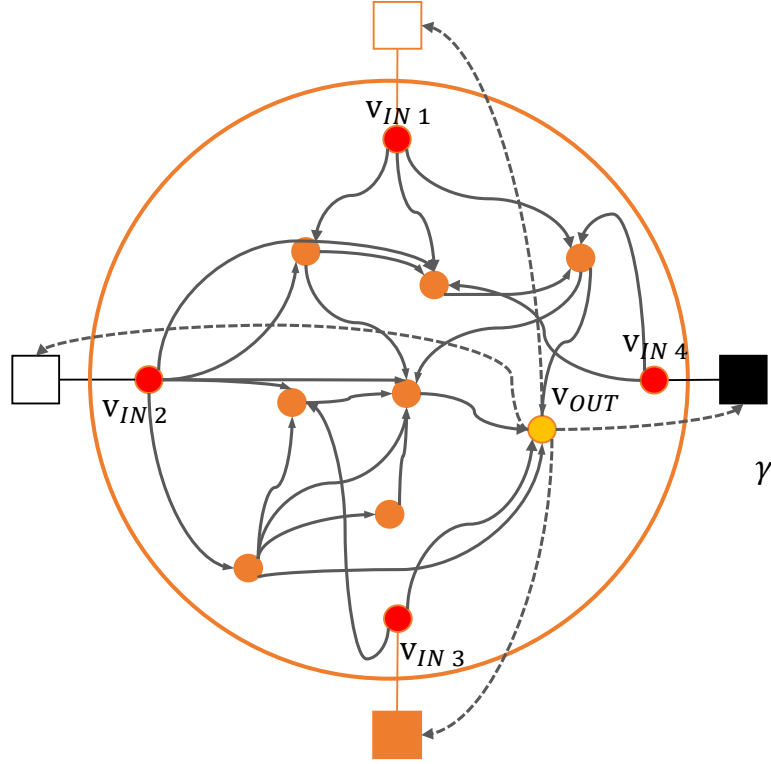


Figure 5.2: Cell regulation by computron

Each node is a gene with binary state. Inputs are linked to input nodes, and output is read from output nodes. Directional edges between the nodes show information flow. Nodes set their next state based on current states of incoming connected nodes based on local connector logic specific to each node.

5.3 Computron

A computron $M = [G, C, v_{IN}, v_{OUT}]$ is an arrangement of N units with binary states ω_i which is formally constructed with:

1. a *generating graph* $G(V, E)$ where each unit is assigned to a node $v_i \in V$. A unit v_j is said to be a neighbor of unit v_i if there exists the directional edge $e_{ji} \in E$. The ordered set of neighbor units of v_i is called its neighborhood denoted as $\delta_i = [v_a, v_b, \dots]$. Each unit is assumed to be also its own neighbor, i.e. $e_{ii} \in E, \forall i$.
2. a set of *connectors* $C = \{c_i\}$, each assigned to a node. A connector $c_i(\Omega_i)$ is a binary valued look-up table based on the state configuration of neighbors, $\Omega_i = [\omega_j | v_j \in \delta_i]$.

3. an *input unit* at node v_{IN} which acts as interface for the external binary input σ , if the computron is driven with an external signal, $\omega_{IN} = \sigma$.
4. a nominated *output unit* at node v_{OUT} which provides a binary output γ if required, $\gamma = \omega_{OUT}$.

The local states are updated synchronously. The global state of computron $s = [\omega_0, \omega_1, \dots, \omega_{N-1}]$ is the configuration of local states on the generating graph. Local dynamics induce the global dynamics of the computron as follows:

$$\begin{aligned}\omega_i^n &\rightarrow \omega_i^{n+1} = c_i(\Omega_i^n) \\ s^n &\rightarrow s^{n+1} = [\omega_0^{n+1}, \omega_1^{n+1}, \dots, \omega_{N-1}^{n+1}] =: T(s^n)\end{aligned}\tag{5.1}$$

An autonomous computron performs computation by mapping an initial configuration to an attractor (fixed or periodic). Such mapping partitions the configuration space. A driven computron generates an output string for a given input string based on an initial starting configuration.

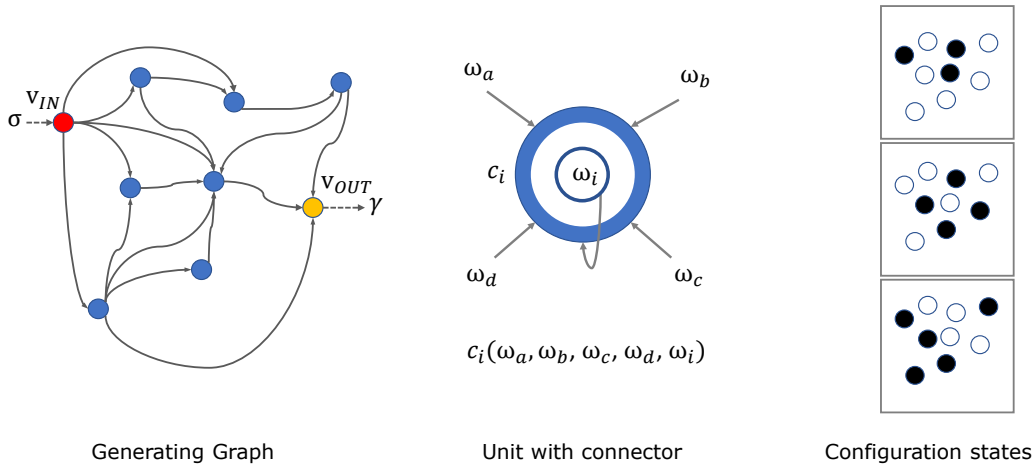


Figure 5.3: Computron construction on a generating graph with gene units.

Gene units are placed at nodes of the graph. Each unit has a binary state ω_i and is endowed with a local look-up table, called a connector. Global state of the computron is the configuration of local unit states on the generating graph. v_{IN} is the special unit that interfaces the external input and v_{OUT} is the nominated unit for reading the output.

A *finite state automata* (FSA) is a computational machine that maps an input string to an output one with no memory. Formally, the machine $M = [S, s_0, \Sigma, \Gamma, T, U]$ is specified by a set of states S , an initial state s_0 , an input alphabet Σ , an output al-

phabet Γ , the *Transition Map* $T : S \times \Sigma \rightarrow S$, and the *Output Map* $U : S \times \Sigma \rightarrow \Gamma$. FSA generates an output string $y = \gamma_0\gamma_1\gamma_2\ldots \in \Gamma^*$ driven by the input string $x = \sigma_0\sigma_1\sigma_2\ldots \in \Sigma^*$. Without loss of generality we assume that input and output alphabets are binary.

Lemma 1 *For any FSA with 2^N states there is an equivalent computron of N units plus the input unit.*

Figure 5.4 shows the state transition diagram of 32-state machine and its equivalent computron representation with 5 genes. We note that state transition diagram of finite state automata lacks intuitive description of the inherent structure provided by the equivalent computron. The generating graph of the computron defines ‘*local flow of information*’ whereas connectors of each unit describe ‘*local processing of information*’.

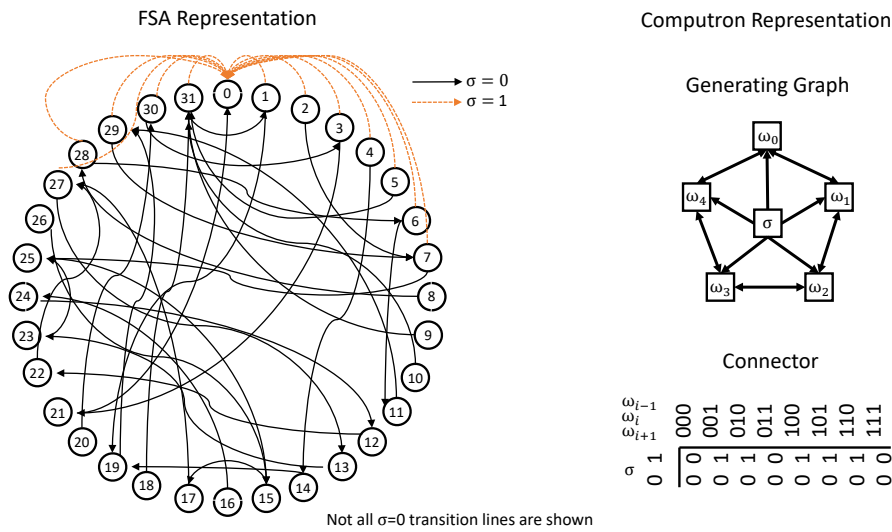


Figure 5.4: 32-state machine on linear cyclic graph

Wolfram cellular automata Rule 110 on 5-circle and its equivalent computron with external input used as Reset (0) or Step (1)

5.4 Regulatory dynamics

The theory of dynamical systems and that of computational machines are mostly disconnected areas of research. The dynamical systems typically live in metric state spaces and evolve in time by iterating the rules of dynamics. The system is said to be complex if the information required to describe the evolving paths is high, a notion formally defined as Kolmogorov-Sinai entropy. On the other side of the world where computational machines live, the aim is to map a set of inputs to a set of outputs. A computron is a computational machine composed of a finite number of units placed on nodes of a directional graph. The units have binary states which are updated according to local transition rules as a function of connected units' states. Computron is the generalization of a cellular automata where local rules are not necessarily uniform and graph not required to be a regular grid. Furthermore, a computron can have an external input, in which case it is called *driven*, otherwise *autonomous*. The computron's global state is the configuration of its units' local states on its graph. It performs computation by mapping an initial configuration to an attractor, be it fixed or periodic by iterative dynamics. Such mapping partitions its configuration space.

A metric space is a set-theoretic concept that consists of a set of elements and distance between any two elements that satisfies the intuitive distance concept: it is always non-negative; satisfies triangle inequality (i.e. given three elements, A,B,C, sum of distance between AB and BC is greater than or equal to distance between AC); and distance AA is zero. A system embedded in a metric space can be analyzed with concepts of dynamical systems as one would have a proper notion of distance. In our computron model of finite state automata, the state of the system is the ensemble of binary activation states of the genes which are nodes of the computron. Therefore, a system state can be thought of as a pattern of active nodes on the generating graph of the computron. If two patterns are similar, say they only differ by one node which is active in one system state vs not on the other with the rest of activations being same, we should intuitively call these two system states similar. Also, if there is more than one difference in activation patterns of two system states, where those differences are with respect to each other is also intuitively important for defining similarity of activation patterns. The closer the differences

lie together vs distributed around, the more similar the activation patterns are. In other words, the notion of similarity of two activation patterns should capture the information required to describe the differences between the two activation patterns. The diversity-induced-metric concept we will formally propose below is designed to support this similarity notion between the gene activation patterns.

By embedding the computron in this metric space, we move into world of dynamical systems to do analysis of the dynamics of a finite state automata, represented as a computron. We develop notions of dynamical chaoticity using Lyapunov characterization of the machine's evolution and computational complexity by analyzing the basin partitioning of this proposed metric space. Our work contributes to literature in multiple ways:

1. Computron is introduced as a computational machine which is equivalent to a finite state automaton, but has the novelty that *flow of information* and *processing of information* is separated.
2. A new distance measure is introduced for configuration space on automata graphs. It is different than Hamming distance used in literature since it captures pattern similarity of configurations on the underlying graph as opposed to solely counting mismatch between them.
3. A new complexity measure is proposed which captures the intrinsic structural complexity of a partitioning of the configuration space, not its randomness.
4. The above concepts are applied to quantify dynamics of generalized graph automata where as literature is mainly focused on dynamics of cellular automata.
5. The framework is expanded to probabilistic automata using fuzzy-basins and recurrent classes which enables us to quantify chaoticity and complexity of probabilistic machines, as well as their susceptibility to noise.

In this section we provide background on dynamical systems, including Lyapunov characterization and entropy concepts. Section 5.5 introduces new set-theoretic concepts such as diversity measure and entanglement. Section 5.6 characterizes the dynamics of computron by embedding the machine into a metric space that captures similarity of configurations. Section 5.6.4 describes our basin-centric complexity

notion and describes how it compares to path-centric, entropy based, notions of dynamical system complexity. Section 5.7 expands into probabilistic computation where each unit performs one of many rules probabilistically, driven by a random external input. This is a major step which pulls the formal domain into Markov chains and recurrent classes. Probabilistic computation lends itself to analysis of susceptibility of computrons, including effects of noise. Proofs of Lemmas introduced in various sections are provided at the end of the chapter.

In his “*A Philosophical Essay on Probabilities*” (1814), Laplace states that knowing the present and evolution laws gives full knowledge of the future. Yet, even if laws are truly known and are truly deterministic, the chaos could prevail:

“Chaos is when the present determines the future, but the approximate present does not approximately determine the future”

Lorenz (1972)

The sensitivity of future to present is formalized by Lyapunov characterization. The richness of what future may bring is quantified by Kolmogorov-Sinai entropy.

5.4.1 Lyapunov characterization

Lyapunov exponents measure the typical rate of exponential divergence of nearby trajectories in a dynamical system. Consider the following continuous-time system:

$$\partial_t \mathbf{x} = \mathbf{F}(\mathbf{x}) \tag{5.2}$$

where $\mathbf{x} \in \mathbb{R}^d$ specifies the system state, and $\mathbf{F}$, a differentiable function, defines the evolution law. Let $\mathbf{x}(t)$ and $\mathbf{x}'(t)$ be two trajectories that start from close-by initial states $\mathbf{x}'(0) = \mathbf{x}(0) + \delta\mathbf{x}(0)$. The evolution of difference in trajectories, $\mathbf{z}(t) = \mathbf{x}'(t) - \mathbf{x}(t)$, can be approximated as a vector in tangent space:

$$\partial_t z_i = \sum_{j=1}^d \partial_{x_j} F_i|_{\mathbf{x}(t)} z_j(t) \tag{5.3}$$

and there exists an orthonormal basis $\{\mathbf{e}_i\}$ such that for large t , the difference in trajectories can be represented as exponential growth or decay:

$$\mathbf{z}(t) = \sum_{i=1}^d c_i \mathbf{e}_i e^{\lambda_i t} \quad (5.4)$$

The Lyapunov exponents $\{\lambda_i\}$ are independent of the initial state $\mathbf{x}(0)$ if the system is ergodic. Systems with a positive Lyapunov exponent are typically chaotic and error of prediction grows exponentially if the initial state is slightly off.

5.4.2 Kolmogorov–Sinai entropy

If we were to discretize time and coarsen the state-space into d -dimensional cubes of each side ϵ (indexing each cube with a symbol), then a trajectory of evolution could be approximated with a string of symbols based on the index of each cube visited at each time step. For a simple system with predictable trajectories, the information needed to represent the ensemble of such strings is less than that of a system where trajectories disperse rapidly. Let $K(\epsilon, t)$ be the total number of different strings generated by the system in t steps in a coarse-grained state space. The *topological entropy* of the system is defined as:

$$h = \lim_{\epsilon \rightarrow 0} \lim_{t \rightarrow \infty} \frac{1}{t} \log K(\epsilon, t) \quad (5.5)$$

Note that in this definition, each different string generated by the system is counted as one. If we take into account the frequency of observing a given finite string of length T (a word w^T), and apply the Shannon entropy formulation to this stream of words, we get the information content of T -symbol paths generated by the system as:

$$H^T = - \sum_{\{w^T\}} P(w^T) \log P(w^T) \quad (5.6)$$

Kolmogorov-Sinai entropy is then defined as the average information content per time step of the path, chosen as the highest one over all possible coarse graining partitions of state space $\{\mathcal{G}\}$:

$$h^{KS} = \sup_{\mathcal{G}} \lim_{T \rightarrow \infty} \frac{1}{T} H^T \quad (5.7)$$

5.4.3 Basin entropy

Dynamical systems theory has traditionally been *path-centric*. Chaoticity is described as divergence of close-by paths, and entropy is conceptualized as richness of paths emanating from a given present. Where paths end up in the long-term is called the *attractor* of a dynamical system. In systems with fixed point or periodic attractors, future is rather dull. In chaotic systems however paths are entangled in such a way that bundles which are close-by soon end up far separated and the attractor is called *strange*. Some dynamical systems, and most computational automata, have multiple simple attractors. In these systems while the future is dull, present can be exciting if knowing the now gives little indication on which final simple destiny one ends up in. This is the *basin-centric* way of looking at the dynamics.

Consider a discrete-time dynamics $x \rightarrow x^2 - y^2 + a$ and $y \rightarrow 2xy + b$ with (a, b) system parameters. If one marks all possible initial states $x(0), y(0)$ according to two ultimate destiny scenarios (escape to infinity or not), we get the fractal Julia sets as given in Figure 5.5, and quite interesting ones for different system parameters (a, b) . A basin of attraction are the set of initial states that end up in a given attractor. The *basin entropy* measures the entanglement of these basins.

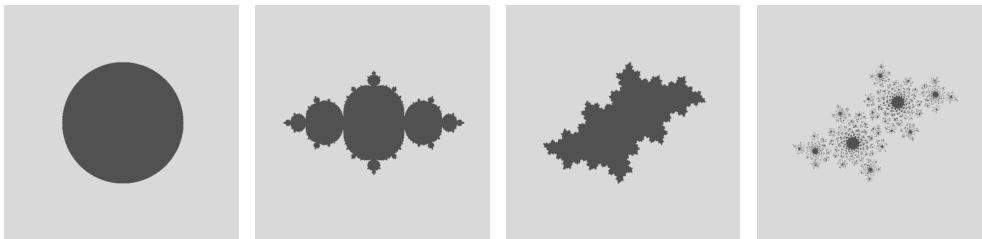


Figure 5.5: Basin partitioning in quadratic Julia sets

System parameters (a, b) from left to right: $(0, 0)$, $(-0.75, 0)$, $(-0.35, -0.55)$, $(-0.40, -0.60)$. The last one tends to a disconnected Cantor-set as time goes to infinity.

5.4.4 Entropy in discrete state-space

A *cellular automata* (CA) is a dynamical system that consists of a grid network of units where each unit run the same evolution rule that tells it how to change its current state (which is drawn from a finite number of elements) into the next, given the neighbors' and its current state. The global state of CA is the configuration of local states of each units on the grid. The local evolution rule and state information

exchange between neighbors synchronously update all local states in each time step, changing the configuration of CA to its next ,hence generating the dynamics on discrete configuration state space. A CA is called elementary if the grid is 1D, each unit takes binary state values and the neighborhood is just the left-right nearest units.

Entropy for CA can be defined with the same spirit as Kolmogorov-Sinai entropy given above. Assuming a 1D grid, consider a window of L units and ignore the rest. This is effectively a coarse-graining of the configuration space since all configurations that have the same pattern within the window are grouped together. Now, let us look into strings of window patterns generated by the system in T time steps w_L^T with observation frequency $P(w_L^T)$. The entropy of this block (i.e. entropy of the system at this coarse-graining) is:

$$H_L^T = - \sum_{\{w_L^T\}} P(w_L^T) \log P(w_L^T) \quad (5.8)$$

The equivalent Kolmogorov-Sinai entropy of CA is then given as the average information content per time step of the evolution path with no coarse graining, i.e. when the unit space window is extended to full grid size:

$$h^{KS} = \lim_{L \rightarrow \infty} \frac{1}{L} \lim_{T \rightarrow \infty} \frac{1}{T} H_L^T \quad (5.9)$$

5.4.5 Dynamics of cellular automata

As per Equation 5.3, Lyapunov exponents rely on differentiability in state space. Being a Cantor set, the configuration space of CA does not lend itself to differentiability. Shereshevsky [80] in 1992, proposed the average propagation speed along the CA grid of a single unit disturbance as a proxy of CA's Lyapunov characterization. Shereshevsky showed that for ergodic 1D CA almost all configurations have unique left and right propagation velocities, λ^-, λ^+ , and further established the inequality linking entropy of the CA to entropy of the shift operator σ and the propagation speeds as: $h(F) \leq h(\sigma) (\lambda^- + \lambda^+)$. Later in 2000, Tisseur [81] expanded the Shereshevsky's Lyapunov characterization by introducing average propagation velocities.

Entropy given in Equation 5.9 is finite and calculable for trivial CAs such as

left-shifting CA. However, Hurd, Kari and Culik [82] (1992) proved that it is uncomputable for an arbitrary CA. The undecidability result does not preclude the existence of large classes of examples, such as *linear* CA, for which explicit calculations are possible. A CA is called linear if the local state set S is endowed with the sum and product operations that make it a commutative ring, and the local evolution function f is given as the linear combination of the neighboring units' local states, $f(x_{-r}, \dots, x_{+r}) = \sum_i a_i x_i$ with $\{a_i\} \in S$. With sum and product operations defined on S , it is possible to describe a given configuration c with characteristic power series $P_c(z) = \sum_{i=-\infty}^{\infty} c_i z^i$ and the local function as a kernel $A_f(z) = \sum_{i=-r}^r a_i z^i$. For linear CA dynamics reduces to $P_{F(c)}(z) = P_c(z)A_f(z)$. Manzini *et al.* published a series of articles [83], [84], [85], [86] on topological dynamics of linear 1D CA and explicitly calculated the entropy.

Topological entropy is a characterization of the richness of time streams of configuration patterns generated by the CA. This richness is closely linked to compressibility of these pattern streams. Excellent progress has been made on practical implementation of compression algorithms for patterns. Lempel-Ziv coding [87] introduced in 1976 is a universal lossless data compression algorithm which is at the core of GIF image format used today. Zenil (2010) [88] proposed to calculate entropy of CA based on compressed length of output patterns with Lempel-Ziv like compression.

5.4.6 Back to basins

Basin entropy defined in 5.4.3 is a relatively new concept and plays a central role in this chapter. Daza *et al.* (2016) [89] calculated basin entropy of two well known physical dynamical systems with path chaoticity: (i) Duffing oscillator which is a periodically driven rod with a nonlinear elasticity, and (ii) Hénon-Heiles system which describes motion of a star around a galactic center. They defined basin entropy by coarse-graining the basin into boxes and marking distribution of ultimate destinies' probability in each box, to which they applied an entropy measure.

Finite size discrete systems have been mostly dismissed in the entropy world since paths are of finite size and ultimately fixed or periodic. However, these systems as computational machines are very rich in the structure of their basins of attraction.

In fact, mapping initial states to ultimate destinies is what computation means for these machines. As we shall now formally see, our definition of chaoticity and more importantly complexity of such finite state automata is all about quantifying the structure of basin partitioning, and as such fundamentally differs from previous work on complexity and entropy for automata.

5.5 Diversity space

5.5.1 Purpose

Dynamical systems live on metric spaces whereas computation does not. Computron as a model of computation is constructed with global states defined as configurations of local states on its generating graph. In this section we formally define a set measure called *diversity measure* which gives the cardinality of a set adjusted for similarity of its elements. Diversity measure induces a similarity distance between two sets as the measure of their symmetric difference. This similarity metric generate a metric space for the states of the computron called *diversity space*. We later perform all our dynamics investigations for the computron on this diversity space.

Various distance functions between sets with elements embedded in a metric space have been proposed in literature. *Hamming* distance is arguably the simplest one which counts the number of different elements in two sets; it lacks the concept of similarity among the elements. *Hausdorff* distance between two sets is calculated as the furthest separation of one set's elements from closest of the other; final result reduces to distance between two most significant elements disregarding the rest.

$$d_h(S_1, S_2) = \max\left\{\max_{e \in S_1} \min_{f \in S_2} \Delta(e, f), \max_{e \in S_2} \min_{f \in S_1} \Delta(e, f)\right\} \quad (5.10)$$

The *sum-of-min-distances* incorporates all elements, but it is an aggregation of closest peers, hence presence of non-closest peers are shadowed, thus fails to capture an ensemble closeness.

$$d_m(S_1, S_2) = \frac{1}{2} \left(\sum_{e \in S_1} \Delta_m(e, S_2) + \sum_{e \in S_2} \Delta_m(e, S_1) \right) \quad (5.11)$$

Other distance measure between sets have been proposed to capture the overall

similarity of the elements of two sets given the underlying metric space such as *Surjection* distance and *Link* distance [90]. Bennett, Vitanyi and Zurek *et al.* [91] introduced *absolute information distance* metric between two objects. Consistent with definition of absolute information content of an object as length of the shortest binary program $K(x)$ running on a universal machine, generating the given object x , they described the distance metric between two given objects x and y as length of the shortest binary program that generates one given the other: $d^*(x, y) = \max K(x|y) + \max K(y|x)$. They proved that absolute information distance is a universal pattern similarity metric. This theoretical metric is similar to that of algorithmic complexity in that whilst universal, it is not computable in its pure form.

5.5.2 Diversity measure

We introduce *diversity measure* as the size of a set adjusted for the similarity of its elements. The intuition behind set theoretic formulation below is as follows: Assume we are asked to form a *diverse team* from a pool of students. Each student has a profile which allows us to define how similar two students are. We build the team by adding one student at a time. First member contributes an *individuality* of 1. Second member's individuality is reduced by her similarity to the first member. Third member's is reduced by his similarity to the first two. Diversity of the team is sum of individualities.

Let (Q, f) be a metric space where Q is a finite set and f is a metric on Q , i.e. a function $f : Q \times Q \rightarrow \mathbb{R}^{0+}$ such that for any $x, y, z \in Q$, (i) $f(x, y) = 0 \Leftrightarrow x = y$, (ii) $f(x, y) = f(y, x)$, and (iii) $f(x, z) \leq f(x, y) + f(y, z)$. Let $\rho : \mathbb{R}^{0+} \rightarrow [0, 1]$ be a concave monotonic function with $\rho(0) = 0$ (*similarity kernel*) and define $\rho(x, y) = \rho(f(x, y))$, then (Q, ρ) is also a metric space.

Individuality of an element x given a set of constraining elements A is defined as:

$$d(x|A) = \prod_{y \in A} \rho(x, y) \quad (5.12)$$

if $\rho(x, y) = 0$, i.e. there exists an element $y \in A$ that is exactly the same as x , then $d(x|A) = 0$, meaning x has no individuality given the constraining set A .

Diversity of an ordered set $\vec{A} = [x_0, x_1, x_2, \dots]$ given a set of constraining elements, B is defined as:

$$\begin{aligned} d(\vec{A}|B) &= d(x_0|B) \\ &+ d(x_1| \{x_0\} \cup B) \\ &+ d(x_2| \{x_0, x_1\} \cup B) \dots \end{aligned} \quad (5.13)$$

Clearly, the ordering of the tuple $\vec{A}$ matters for the calculation of diversity, therefore we define the *diversity measure* of a set A constrained by set B as the supremum over all possible orderings of the tuple $\vec{A}$. For numerical experiments, we devised a routine whose computation load grows quadratically rather than combinatorially which made calculation of diversity measure feasible.

$$\|A|B\| = \sup_{\vec{A}} d(\vec{A}|B) \quad (5.14)$$

$$\begin{aligned} d(\vec{A}) &= d(\vec{A}|\emptyset) \\ \|A\| &= \|A|\emptyset\| \end{aligned} \quad (5.15)$$

Section 5.7 of this paper introduces probabilistic computron which generates probability distributions on states. Therefore, we extended diversity measure to handle *weighted-sets*. Let $w : S \rightarrow \mathbb{R}^{0+}$ be a weight function. *Individuality* of a weighted element x given a set of constraining weighted elements A is given as:

$$d_w(x|A) = w(x) \prod_{y \in A \setminus x} \rho(x, y) \quad (5.16)$$

Diversity and diversity measure for weighted sets follow the same definitions as in Equations (5.13) and (5.14) using individuality for weighted sets. We use the following extension for union $\tilde{\cup}$, symmetric difference $\tilde{\Delta}$, and subset $\tilde{\subseteq}$ definitions for weighted sets. In the case of union and symmetric difference operations between two sets A, B , the resultant weighted set is C with its element weights being $w_C(x)$:

$$\begin{aligned} C = A\tilde{\cup}B &= \{x : x \in A \cup B, w_C(x) = (w_A(x) + w_B(x))\} \\ C = A\tilde{\Delta}B &= \{x : x \in A \cup B, w_C(x) = |w_A(x) - w_B(x)|\} \\ A\tilde{\subseteq}B &\Leftrightarrow (x \in A \Rightarrow x \in B, w_A(x) \leq w_B(x)) \end{aligned} \quad (5.17)$$

Lemma 2 (Diversity measure) *Diversity is a sub-additive measure on Q .*

$$A \tilde{\subseteq} B \implies \|A\|_w \leq \|B\|_w \quad (5.18)$$

$$\|A_0 \tilde{\cup} A_1\|_w \leq \|A_0\|_w + \|A_1\|_w \quad (5.19)$$

Diversity-induced distance (DiD) is an induced metric on Q^* that captures dissimilarity between two sets, $A, B \in Q^*$:

$$\varphi(A, B) = \frac{\|A \tilde{\Delta} B\|}{|Q|} \quad (5.20)$$

where $\varphi : Q^* \times Q^* \rightarrow [0, 1]$, $\tilde{\Delta}$ is the symmetric difference operator as defined in Equation 5.17 and $|Q| = \sum_Q w(x)$ is size of weighted space.

Lemma 3 (DiD metric) *(Q^*, φ) is a metric space.*

5.5.3 Entanglement

We introduce the entanglement concept which will later be used in defining partitioning complexity of a computron. *Entanglement* of a set of sets $\mathbf{A} = \{A_0, A_1, A_2 \dots\}$ is defined as:

$$\|\mathbf{A}\| = \sup_{\vec{\mathbf{A}}} D(\vec{\mathbf{A}}) \quad (5.21)$$

where

$$D(\vec{\mathbf{A}}) = \|A_0\| + \|A_1|A_0\| + \|A_2|A_1, A_0\| \dots \quad (5.22)$$

Entanglement captures the total diversity of a collection of sets where each set's diversity is constrained by the presence of the other sets. The more the sets are within each other's proximity entanglement increases. On the other hand as one given set's elements get separated from each other, that set's diversity decreases. Entanglement captures the balancing of these two factors in calculating the total diversity of the set of sets. This is illustrated in Figure 5.6.

Lemma 4 (Maximal entanglement) *Let $\mathbf{P} = \{B_0, B_1 \dots\}$ be a partitioning of Q (i.e. $\cup_i B_i = Q$, and $B_i \cap B_j = \emptyset, i \neq j$) with $\mathbf{P}^0 = \{Q\}$ (null partition) and*

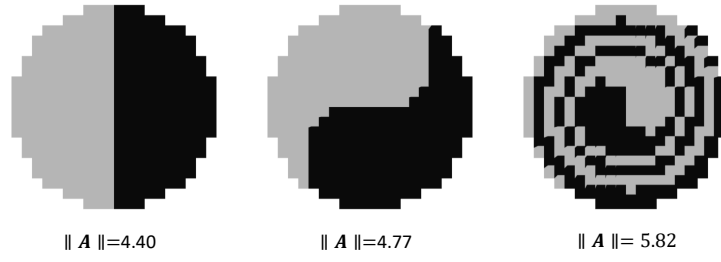


Figure 5.6: Illustration of entanglement

Each element is assumed to be a pixel with Manhattan distance metric. Space is partitioned into two sets, $\mathcal{A} = \{A_0, A_1\}$. Three different partitions with increasing entanglement measure is provided.

$\mathbf{P}^\infty = \{\{x_0\}, \{x_1\}, \dots\}, \forall x_i \in Q$ (full partition), then:

$$\|\mathbf{P}^0\| = \|\mathbf{P}^\infty\| \geq \|\mathbf{P}\| \quad (5.23)$$

5.6 Dynamics of autonomous computrons

5.6.1 Embedding in metric space

Given a computron $M = [G, C]$ with a generating graph G and connectors set C , we embed it in a metric space that captures similarity of its configuration states as follows: The distance between two units is set as the *shortest-path-length* on generating graph between the two nodes occupied by these units mapped to a unit interval. The intuition of this distance definition between two units is that a state change in one of the units can possibly impact the state of other one in compute steps that are no less than the shortest-path-length.

$$d(v_i, v_j) := \rho(\text{SPL}(v_i, v_j)) \quad (5.24)$$

$\rho : \mathbb{R}^{0+} \rightarrow [0, 1]$ concave, monotonic with $\rho(0) = 0$. We then define the configuration state of the computron as set of units that have local state of ON.

$$s \in S := \{v_i | \omega_i = \text{ON}\} \quad (5.25)$$

This leads to diversity-induced distance as a metric for dissimilarity of the states

for computron:

$$\varphi(s_0, s_1) := \frac{\|s_0 \triangle s_1\|}{|S|} \quad (5.26)$$

By Lemma 3, (S, φ) is a metric space with each configuration of the computron an element of the space. This is the *diversity space* that we embed the computron where distance between two global states of the machine is based on the similarity of the corresponding configurations. Figure 5.7 illustrates the concept on a computron where generating graph is assumed to be regular 2D grid. Each configuration is a set of pixels that are have state 1 (a binary image). Diversity-induced distance between configuration s_0 and s_1 is less than that of s_0 and s_2 , capturing their similarity, although Hamming distances are the same.

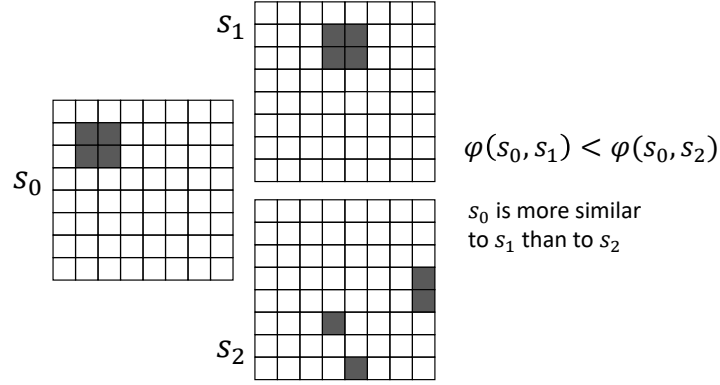


Figure 5.7: Diversity-induced distance between configurations

5.6.2 Computron as a dynamical system

An autonomous computron has no external input. Starting from an initial configuration, it moves to next based on the transition map given in Equation 5.1. The dynamics is governed by discrete time iteration on configuration metric space (S, φ) :

$$\begin{aligned} s^{n+1} &= T(s^n) \\ T^{n+1} &:= T \circ T^n, T^0 = I, n \in \mathbb{N} \end{aligned} \quad (5.27)$$

An *attractor* of the computron is a tuple of configurations $\vec{A} = [s^0, s^1, \dots, s^{p-1}]$ where $s^{i+1} = T(s^i)$, $s^0 = T(s^{p-1})$ and $s^n \neq s^m \Leftrightarrow n \neq m$. For $p = 1$, it is called a fixed-point attractor with the corresponding state a '*fixed-state*', else a

periodic attractor. States that are elements of a periodic attractor are called *periodic states*. States that are not element of an attractor are called *transient states*. The *basin* of an attractor are those states that iterate into, or a part of, that attractor $B(\vec{A}) = \{s \in S : \exists k \ni T^k(s) \in \vec{A}\}$.

Lemma 5 (Basin partition) *The set of all basins of an autonomous computron M is a partition of its configuration space S , denoted as $\mathcal{P}_M = \{B_0, B_1, \dots\}$, with the corresponding attractors $\mathcal{A}_M = \{\vec{A}_0, \vec{A}_1, \dots\}$ where $B_i \subset S$, $\cup_i B_i = S$ and $B_i \cap B_j = \emptyset, i \neq j$.*

Figure 5.8 gives a recap: A computron consists of a set of units placed at nodes of a graph G with binary local states $\{\omega_i\}$. The configuration of the machine is specified as set of units that have local state 1. A basin is a set of configurations. As per Lemma 14, the configuration space is partitioned into basins by the computation performed by the computron.

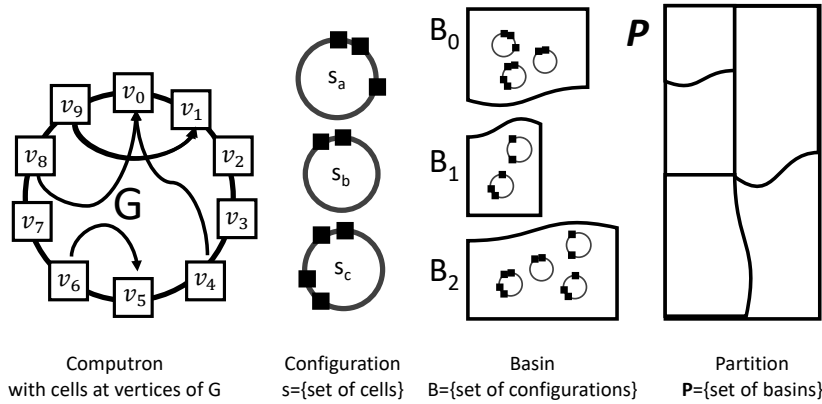


Figure 5.8: Hierarchy of computron dynamics

A partition is a set of basins; a basin is a set of configurations; a configuration is a set of units that have local state 1.

5.6.3 Chaotic computation

As defined in Equation 5.3 Lyapunov exponents characterize the rate of separation (or convergence) of infinitesimally close trajectories in a dynamical system. In discrete-time systems (maps), $x_{n+1} = f(x_n), n \in \mathbb{N}$, for an orbit starting with initial

point x_0 , Lyapunov exponent becomes:

$$\lambda(x_0) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \quad (5.28)$$

where $f'(x_i) = \frac{d}{dx} f(x)|_{x=x_i}$ is the derivative of f . For a discrete-time discrete state-space dynamical system (i.e. dynamics of autonomous computron) on a metric space (S, φ) , we define Lyapunov number for an initial configuration $s_0 \in S$ as follows: Let $\mathcal{B}_\epsilon(s) = \{q \in S : \varphi(s, q) < \epsilon\}$ denote the epsilon-ball around state s , where ϵ is set as the lowest value where epsilon-ball is non-empty. Let $\mathcal{A}_M = \{A_0, A_1, \dots\}$ be the attractors of computron, and $\mathcal{P}_M = \{B_0, B_1, \dots\}$ the corresponding basins of attraction. We calculate Lyapunov number of a state as the average separation of its neighboring states as iteration time goes to infinity. Since, each state converges to its attractor, we measure the dissimilarity of attractors using an induced DiD metric.

$$\lambda(s) = \frac{1}{|\mathcal{B}_\epsilon(s)|} \sum_{q \in \mathcal{B}_\epsilon(s)} \frac{\Phi(A_k, A_l)}{\Phi(\{s\}, \{q\})} \quad (5.29)$$

$$s \in B(A_k), q \in B(A_l)$$

Lyapunov number for the computron is the average over the configuration space $\lambda_M := \langle \lambda(s) \rangle_S$.

5.6.4 Complex computation

Whilst chaos is a well-defined concept in dynamical systems theory, complexity has been more elusive with wide range of definitions as reviewed by Grassberger [92]. Observer's reference base using existing information is relevant for defining complexity yet it leads to observer subjectivity. This can be avoided by either dealing with an equivalence class of the object as the reference base, or by treating the object on a self-referential basis using a generative description. In the former case, we are in the domain of information theoretical definition of complexity, whereas the latter is in the domain of algorithmic complexity. Either way, the complexity becomes a measure of difficulty in describing the object in question with regards to a reference

base.

We introduce *diversity-based complexity* of a computron as the disentanglement of its basin partition. Computron performs its computation by act of mapping the input configurations to attractors, generating a partitioning of configuration space into basins of attraction. We consider the complexity of this partitioning as the defining characteristic of computron as the computational machine. Equation 5.30 gives the complexity measure:

$$\Psi_M = 1 - \frac{\|\mathcal{P}_M\|}{\|\mathcal{P}_M^\infty\|} \quad (5.30)$$

with $\Psi_M \in [0, 1]$ since $\|\mathcal{P}_M\| \leq \|\mathcal{P}_M^\infty\|$ by Lemma 4.

This definition means that a complex computron has a basin partition where each basin consists of maximally diverse set of configurations with respect to other configurations in that basin, and furthermore basins themselves are maximally diverse with respect to each other. As depicted in Figure 5.9 complexity of a partitioning increases initially with the number of partitions, then it starts to decrease such that no partitioning and full partitioning have zero complexity. Therefore, high complexity partitioning of the configuration space requires maximum information to describe such partitioning.

Our complexity definition is basin-centric, not path-centric such as Kolmogorov-Sinai entropy described in Section 5.4.2. Path-centric descriptions seek to capture the information required to describe the long term behavior of the dynamical system, discarding the initial and transient states. As such, they provide a measure of information content of the attractors of the system, i.e. the *strangeness of the attractors*. However, the act of computation performed by a finite state automata, or the computron, is a mapping of initial systems states to some final system states. The information required to capture partitioning of all states to basins of attraction (which includes the transients as well as the final attractors) is what we are after when we try to describe the complexity of the machine. This is the reason we are using a complexity measure that seeks to quantify the information required to describe the basins of attraction, i.e. *the strangeness of the attracted*.

Lemma 6 (Minimal complexity) *Absorbing and idle computrons have the least*

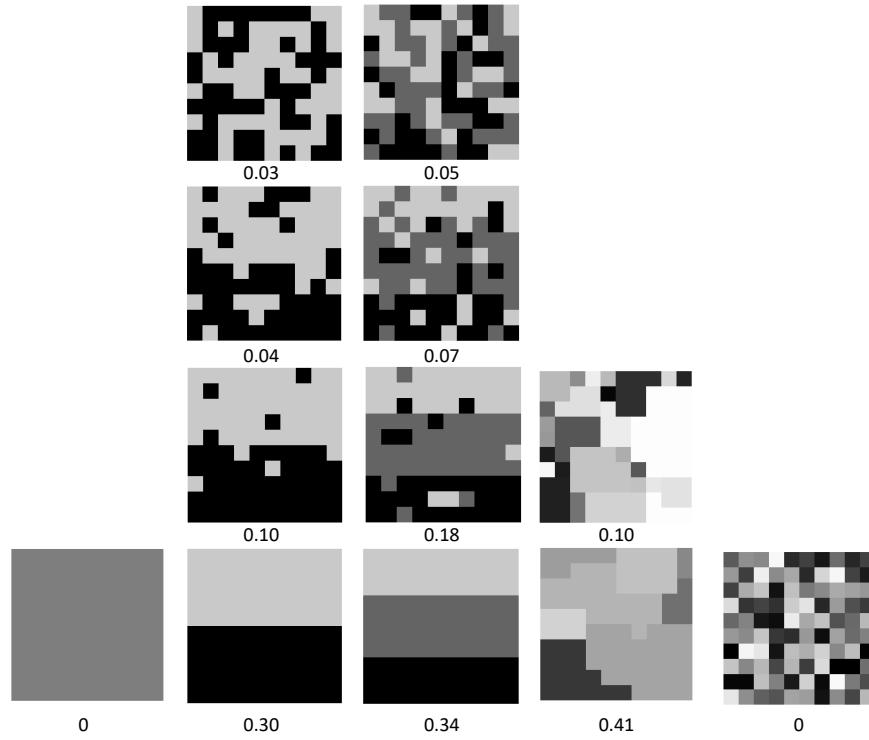


Figure 5.9: Illustration of diversity-based complexity

Null partitioning and infinite partitioning has zero complexity. Complexity first increases with increasing number of partitions then starts to decrease. For a given number of partitions increasing entanglement reduces complexity. Complexity is maximized somewhere between order and chaos.

complexity (null) compared to any other computron.

$$\Psi_{M^0} = \Psi_{M^\infty} = 0 \leq \Psi_M \quad (5.31)$$

An *absorbing* computron has a single fixed-point attractor which all configurations iterate into denoted as $M^0 : \forall s \in S, \exists k \in \mathbb{N} \ni T^k(s) = s^*$, where T is the transfer map of computron and s^* is the absorbing state. An *idle* computron does nothing, i.e. each state is mapped to itself; and is denoted as $M^\infty : \forall s \in S, T(s) = s$.

5.6.5 Case Study – Linear cyclic generating graph

We analyzed complexity and chaoticity of autonomous computrons with linear cyclic generating graph and homogeneous connectors, i.e. elementary cellular automata. As expected, absorbing and idle computrons have minimal complexity and chaoticity. With same generating graph (linear cyclic of 5 units), we analyzed different

connector types. Chaoticity and complexity of each computron is plotted in Figure 5.10. There are several elementary CA that have very low complexity and not chaotic (less than 1 chaoticity). However, there is a group of machines that demonstrate high complexity while not chaotic.

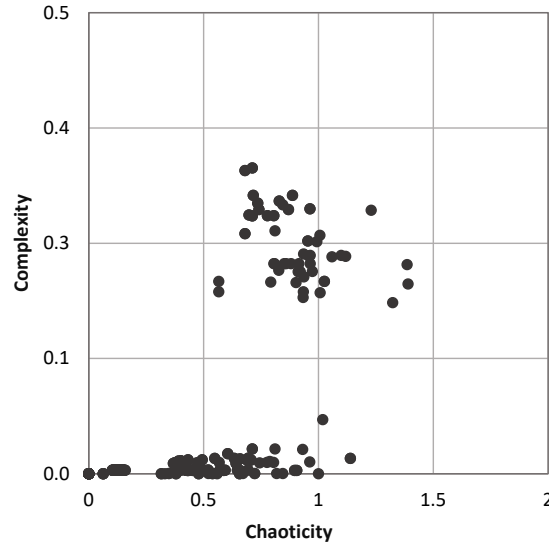


Figure 5.10: Chaoticity λ vs complexity Ψ

32 state computrons with linear cyclic generating graph and homogeneous connectors

5.7 Dynamics of driven computrons

5.7.1 Computron as a Markov Chain

A *driven computron*, unlike an autonomous one, has a transition-map that is a function of its current configuration and the external input drawn from binary alphabet. Dynamics is governed by discrete-time iteration:

$$s^{n+1} = T(s^n, \sigma^n) = \begin{cases} T^A(s^n) & \sigma^n = 0 \\ T^B(s^n) & \sigma^n = 1 \end{cases} \quad (5.32)$$

where σ^n is the external input at time n . We assume no apriori information about the input string other than observed frequency of alphabet letters, resulting in representing the input string as a sequence of independent identically distributed random variables on binary alphabet Σ with $p = Pr\{\sigma = 0\}$.

A *noisy computron* is a computron exposed to thermal fluctuations in the binary states of its units. The dynamics is governed by probabilistic discrete-time iteration:

$$s^{n+1} = T^\varepsilon(s^n, \sigma^n) = \begin{cases} T^A(s^n) & \text{with prob. } p(1 - \varepsilon) \\ T^B(s^n) & \text{with prob. } (1 - p)(1 - \varepsilon) \\ s_r \in \mathcal{B}_\varepsilon(s^n) & \text{with prob. } \varepsilon \end{cases} \quad (5.33)$$

where $\mathcal{B}_\varepsilon(s^n)$ is epsilon-ball neighborhood of configuration s^n and ε is the probability of malfunction due to thermal noise.

We now enter into domain of Markov chains to describe the probabilistic dynamics of the driven and noisy computrons. A *discrete Markov chain* is a sequence of random variables with domain S , $\{X_n, n \in \mathbb{N}\}$, where each random variable X_n is independent of prior random variables in the sequence given its previous one.

$$Pr\{X_n = j \mid X_{n-1} = i, X_{n-2} = k, \dots\} = P_{ij} \quad (5.34)$$

Probabilistic computrons are equivalent to discrete Markov chains. They are represented with directed labeled graphs where each node is assigned a configuration $s_i \in S$, and an edge from configuration s_i to s_j with transition probability $P_{ij} \neq 0$. It can also be represented by a $M \times M$ *transition probability matrix* $\mathbf{P}$ where $M = |S|$ is the number of all configurations and $\mathbf{P}_{ij} = P_{ij}$. From matrix representation, it follows that probability of reaching k from i in exactly n steps is $\mathbf{P}_{ik}^n$

Terminology

1. n -step *walk* is an ordered set of states $(i_0, i_1, \dots, i_n)$ in which there is a directed edge from i_{m-1} to i_m and no states are repeated
2. State j is *accessible* from state i if there exists a walk from i to j , shown as $i \rightarrow j$
3. Two states *communicate* if $i \rightarrow j$ and $j \rightarrow i$
4. *Class* is a set of states where each pair of states in the set communicate, and none communicate with a state outside the class

5. State i is called *recurrent* if $i \rightarrow j \Rightarrow j \rightarrow i$
6. *Transient* state is a state that is not recurrent
7. *Period* of a state i is the greatest-common-divisor of those values of n for which $P_{ii}^n > 0$. If the period is 1, the state is called *aperiodic*.
8. It follows from definitions that for a given class, either all states are recurrent or they are all transient. It can also be shown that all states in the same class have the same period.
9. A class is called *ergodic* if it is recurrent and aperiodic.
10. A Markov chain consisting of single ergodic class is called an *ergodic chain*.
11. A *unichain* contains a single recurrent class plus, possibly, some transient states. An *ergodic unichain* is a unichain for which the recurrent class is ergodic.
12. A *steady-state distribution* for an M state Markov chain with transition matrix $\mathbf{P}$ is a row vector $\boldsymbol{\pi}$ that satisfies:

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}, \text{ where } \pi_i \geq 0, \sum_i \pi_i = 1 \quad (5.35)$$

For a given Markov chain, Equation 5.35 always has a solution, and the solution is unique if and only if the Markov chain is a unichain. If there are c recurrent classes, then there are c linearly independent solutions each having non-zero entries only for the states of corresponding recurrent class. $\mathbf{P}^n$ converges if and only if recurrent classes are aperiodic with each row of $\mathbf{P}^n$ converging to $\boldsymbol{\pi}$ of one class.

We compare dynamical systems concepts for autonomous computation to that of Markov chain. An attractor is analogous to a recurrent class. An ergodic dynamical system has a single attractor, and similarly an ergodic chain has a single recurrent aperiodic class. In autonomous computation a transient state feeds into a single attractor, therefore the concept of basin of attraction is well-defined. However, in driven automaton, a transient state can feed into multiple recurrent classes with overlapping basins.

Let P_{ij} be the probability of transition from state s_i to s_j as described in Equation 5.33, and $\mathbf{P}$ the corresponding transition probability matrix. Let $p_n(s_i)$ be the

probability of computron being in configuration s_i at time n , and $\mathbf{p}_n$ be the corresponding probability distribution vector over configuration space S . Its evolution starting from an initial distribution $\mathbf{p}_0$ is governed by the *master equation*:

$$\mathbf{p}_{n+1} = \mathbf{p}_n \mathbf{P} \quad (5.36)$$

Impact set R_s of a given configuration of computron is a set weighted by the impulse response distribution π_s defined as follows:

$$\begin{aligned} \pi_s &= \lim_{n \rightarrow \infty} \mathbf{p}_s^\delta \mathbf{P}^n \\ p_s^\delta(x) &= \begin{cases} 1 & \text{if } x = s \\ 0 & \text{else} \end{cases} \end{aligned} \quad (5.37)$$

$$R_s = \{x \in S : \pi_s(x) > 0\} \quad (5.38)$$

Since Equation 5.36 is linear, steady-state distribution of a driven computron can be written as linear combination of the impulse responses of all states weighted by the initial probability distribution. Let $\mathbf{p}_0$ be an arbitrary initial distribution stated as linear combination of delta-dirac distributions over all configurations:

$$\mathbf{p}_0 = \sum_{s \in S} p_0(s) \mathbf{p}_s^\delta \quad (5.39)$$

Then, steady-state distribution π becomes:

$$\pi = \sum_{s \in S} p_0(s) \pi_s \quad (5.40)$$

5.7.2 Ergodic decomposition

A computron is said to be ergodic if its Markov chain representation is ergodic (i.e. single recurrent aperiodic class). Typically computrons are non-ergodic, i.e. they have more than one recurrent class, and also can be periodic. A periodic computron would not have a stationary steady-state distribution as $\mathbf{P}^n$ would not converge.

Stationarity can be introduced by imposing ‘*asynchronicity*’ where we modify each state’s transition-map with non-zero probability λ to *stay idle*.

$$\tilde{\mathbf{P}} = (1 - \lambda)\mathbf{P} + \lambda\mathbf{I} \quad (5.41)$$

As per the definition of periodicity (see Terminology), the greatest-common-divisor of n for which $\tilde{P}_{ii}^n$ becomes 1 for all states (since they all have a transition to self with λ probability), and therefore the machine would become aperiodic. An aperiodic computron would have $\tilde{\mathbf{P}}^n$ converge with each row becoming $\tilde{\boldsymbol{\pi}}$ of one recurrent class.

The Coloring Lemma provides a technique for identifying the recurrent classes $\mathcal{C}_M = \{C_1, C_2, \dots, C_K\}$ of a computron. Let $\mathbf{p}_a, \mathbf{p}_b$ two random initial distributions with non-zero entries for each state, and let $\boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_b$ be the corresponding steady-state distributions in an asynchronous computron. Define, the ‘*gray color*’ of a state $c(s)$ as:

$$c(s) = \begin{cases} \pi_b(s)/\pi_a(s) & \text{if } \pi_a(s) > 0 \\ 0 & \text{else} \end{cases} \quad (5.42)$$

Lemma 7 (Coloring) *Two states will, almost always, have the same shade of gray if and only if they are elements of same recurrent class, provided all classes of the computron are ‘distinguishable’ (as defined in the proof); i.e. for two random non-zero initial distributions, $c(s) = c(q) > 0 \Leftrightarrow s, q \in C_i$, almost always.*

Having identified the recurrent classes of a computron using the Coloring Lemma, we now use them as basis for ergodic decomposition:

Lemma 8 (Ergodic bases) *Recurrent classes form a non-overlapping spanning set (basis) for the impact sets of all states. Let $\mathcal{C}_M = \{C_1, C_2, \dots, C_K\}$ be the set of recurrent classes of a computron and $\boldsymbol{\chi}_s = \{\chi_1(s), \chi_2(s), \dots, \chi_K(s)\}$ the ergodic decomposition of impulse response $\boldsymbol{\pi}_s$:*

$$\chi_k(s) = \sum_{q \in C_k} \pi_s(q) \quad (5.43)$$

then:

$$\sum_k \chi_k(s) = 1, \text{ with } \chi_k(s) \geq 0, \forall s \in S \quad (5.44)$$

We can now formally define *basin of attraction for a recurrent class* as a weighted-set $\tilde{B}(C_k)$:

$$\tilde{B}(C_k) = \{s : \chi_k(s) > 0, w_{\tilde{B}}(s) = \chi_k(s)\} \quad (5.45)$$

Lemma 9 (Fuzzy partition) *The basins of attraction of a probabilistic computron is a fuzzy partition of its configuration space, $\tilde{\mathcal{P}}_M = \{\tilde{B}(C_1), \tilde{B}(C_2), \dots, \tilde{B}(C_K)\}$. A fuzzy partition of S is the set of weighted-sets $\tilde{\mathcal{P}} = \{\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_K\}$, where $\cup_i \tilde{B}_i = S$, and $\sum_k w_{\tilde{B}_k}(s) = 1, \forall s \in S$.*

5.7.3 Chaoticity – Complexity of probabilistic automata

A computron acting on a configuration space transforms each configuration into its impact set. Configurations that are similar to each other (based on diversity-induced-distance) can be transformed to similar or different impact sets. We define the ratio of DiD distance among the impact sets of two configuration to distance between them as the ‘*bending ratio*’. Computron’s action on the configuration space can be conceptualized as transforming the distance between each pairs of configurations with the bending ratio.

$$\tau(s, q) = \frac{\tilde{\Phi}(R_s, R_q)}{\Phi(\{s\}, \{q\})} \quad (5.46)$$

Lyapunov number for a state is the average bending of the ϵ -ball around such state:

$$\lambda(s) = \frac{1}{|\mathcal{B}_\epsilon(s)|} \sum_{q \in \mathcal{B}_\epsilon(s)} \tau(s, q) \quad (5.47)$$

Chaoticity of the machine is measured by the average Lyapunov number over configuration space. $\lambda_M := \langle \lambda(s) \rangle_S$.

As an extension of the complexity definition we used for autonomous computrons, Equation 5.30, complexity of a probabilistic computron is the disentanglement of its fuzzy basin partition.

$$\tilde{\Psi}_M = 1 - \frac{\|\tilde{\mathcal{P}}_M\|}{\|\tilde{\mathcal{P}}_M\|} \in [0, 1] \quad (5.48)$$

As a case study we analyzed the complexity of hybrid elementary CA that execute

two alternative rules based on the input signal. Figure 5.11 shows the variation in complexity of hybrid CAs with the driver mix $p(\sigma = 1)$ changing in $[0, 1]$. For $p = 0$ and $p = 1$ the machine is regular elementary CA. We plotted the deviation of complexity from a linear mix, so-called synergy, as p is varied. Creating hybrid CAs using high and low complexity constituents results in drop in overall complexity (i.e. non-synergistic mix). The drop happens at even small mixing around the higher complexity CA. This compares to synergistic case when both constituents are of high complexity.

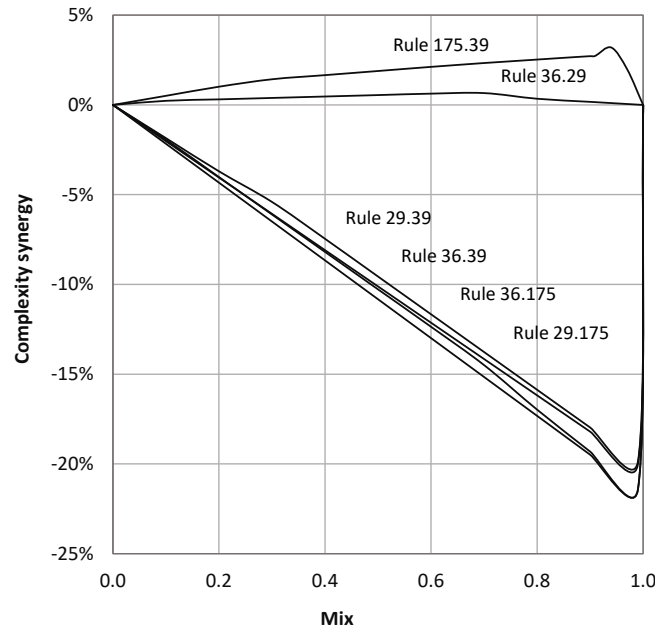


Figure 5.11: Complexity synergy of driven cellular automata

Increase (decrease) in complexity compared to linear combination of autonomous CA, with changing intensity of driving signal $p(\sigma = 1) \in [0, 1]$.

5.7.4 Susceptibility

Let M^A, M^B be two computrons operating on the same state space. We provide a measure of topological similarity between them by comparing their bendings of the configuration space. The bending ratio $\tau(s, q)$ was defined in Equation 5.46. The similarity of two computrons is given as:

$$\Phi(M^A, M^B) = \sum_{s, q \in S \times S} |\tau_A(s, q) - \tau_B(s, q)| \quad (5.49)$$

Equipped with a measure for similarity of two computrons, we can now analyze distortions in a machine upon small structural changes. As an application, we investigate distortion of a computron due to noise which is given as:

$$\xi(\mu) = \Phi(M_{p_\mu=\mu}, M_{p_\mu=0}) \quad (5.50)$$

Figure 5.12 shows noise distortion curves on log-log scale for a computron executing Rule 29.175 with varying levels of driver intensity. The machine exhibits linear noise distortion curves with saturation at high noise levels when it is driven in region Rule 29 towards Rule 175. But as it gets closer to Rule 175, noise distortion become exponential with power $\gamma \approx 1.45$. This characterization is consistent with the sudden change in complexity around Rule 175 as was provided in Figure 5.11.

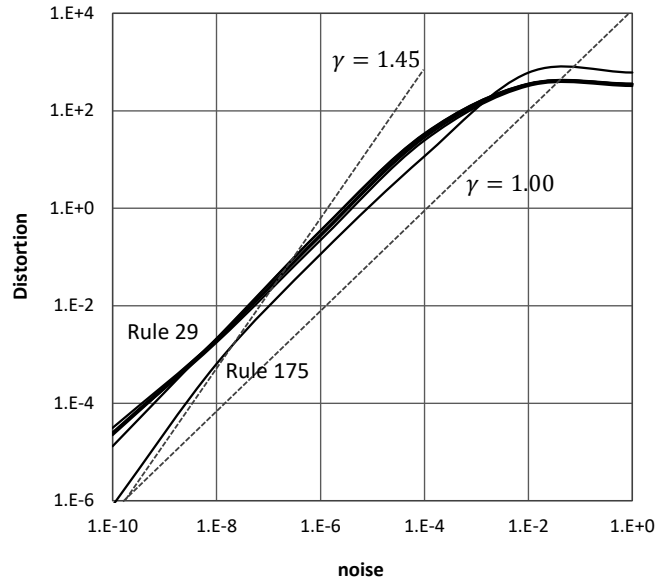


Figure 5.12: Topological distortion to driven computron due to noise

Distortion vs noise levels are plotted in log-log scale. Curves correspond to different levels of driver intensity which results in varying the machine's behavior from Rule 29 to Rule 175 gradually.

5.8 Discussion

This chapter was an investigation into dynamics and complexity of gene regulatory networks represented as a computational machine, the computron. Having embedded the machine into a metric space that captures similarity of configurations as

patterns on the gene expressions, we analyzed its chaoticity and complexity. Our approach to complexity was basin-centric rather than path-centric. We also expanded the framework into non-autonomous machines with external drivers which led to Markov chain formalism. The theory was applied to hybrid elementary cellular automata with noise. Case studies showed that machines with high complexity are more susceptible to externalities, such as mixing and noise.

Our conceptualization of gene regulatory network as computron in diversity space opens up a series of avenues for future research, including the study of robustness and complexity of various gene expression dependencies. The evolution of gene regulatory networks can be studied by ‘*growing*’ (attaching genes to an existing DNA) and ‘*mutating*’ (altering generating graph or the connectors) computrons.

5.9 Proof of Lemmas

Lemma 10 *For any finite state automaton with 2^N states there is an equivalent computron of N units plus the input unit.*

Two machines $M = [S^M, s_0, \Sigma, \Gamma, T^M, U^M]$ and $\Omega = [S^\Omega, s_0, \Sigma, \Gamma, T^\Omega, U^\Omega]$ are said to be *equivalent* if there exists a one-to-one mapping between the states $\mu : S^M \leftrightarrow S^\Omega$, and for all $s \in S^M$ and $\sigma \in \Sigma$:

$$\begin{aligned}\mu(T^M(s, \sigma)) &= T^\Omega(\mu(s), \sigma) \\ U^M(s, \sigma) &= U^\Omega(\mu(s), \sigma)\end{aligned}\tag{5.51}$$

A generating graph $G(V, E)$ is said to be *fully-connected* if $\forall v_i, v_j \in V, \exists e_{ji} \in E$. A state of a computron is the configuration of its unit states $s^\Omega = [\omega_0 \omega_1 \dots \omega_{N-1}]$, $\omega_i \in \{0, 1\}$ with *base indexing* defined as $k(s^\Omega) = \sum_{i=0}^{N-1} 2^{\omega_i}$, where $N = |V_G|$ is the number of nodes of the graph. following 3 propositions construct the proof for the Lemma.

Proposition 10.1 *For any computron Ω with an arbitrary generating graph G there exists an equivalent computron Ω^* with a fully-connected generating graph G^* with same number of nodes.*

We pair up units of Ω and Ω^* as $v_i \leftrightarrow v_i^*$ and we use base indexing to enumerate

configurations s^Ω and s^{Ω^*} . This establishes the one-to-one mapping between the two configuration spaces $\mu : S^\Omega \leftrightarrow S^{\Omega^*}$. Then we construct the connectors $[C^*]$ of Ω^* as follows: Let $\delta_i = [v_{i_0} \dots v_{i_{n(i)}}]$ be the neighbors of unit v_i , and let δ'_i be the rest of the units (i.e not-connected to v_i), then we set the loop-up tables of connectors C_i^* for each unit as follows:

$$C_i^*([\delta_i, \delta'_i], \sigma) = C_i([\delta_i], \sigma) \quad (5.52)$$

With output maps also set identical, $U^*(s, \sigma) = U(s, \sigma)$, then the implied Transition Maps $T^\Omega(s)$ and $T^{\Omega^*}(s)$ satisfy Equation 5.51. Therefore, $\Omega \equiv \Omega^* \square$

Proposition 10.2 *There is one-to-one correspondence between finite state automata M of 2^N states with binary input and fully-connected computrons Ω^* with $N + 1$ units, including the input unit.*

Let $\mathbb{U}_{\Omega^*}$ denote set of all fully-connected computrons with $N + 1$ units (including the input unit).

$$|\mathbb{U}_{\Omega^*}| = (2^N)^{(2^N)} \times (2^N)^{(2^N)} \quad (5.53)$$

Each unit of Ω^* (other than the special input unit) has $N + 1$ inputs from other units and itself (fully-connected) with 2^{N+1} input configurations. With each input configuration having a binary output set by the connector, the number of different connector types are $2^{(2^{N+1})}$. Each unit, other than the input unit, can have any of the connector types (input unit has a unique connector that solely acts as interface to external input). Therefore, total number of different computrons with fully-connected generating graph is $(2^{2^{N+1}})^N = (2^N)^{(2^N)} \times (2^N)^{(2^N)}$.

Let M be a finite state automaton with 2^N states, and let $\mathbb{U}_M$ denote set of all such machines. Since each state can go to any state based on the binary external input, we have:

$$\begin{aligned} |\mathbb{U}_M| &= (2^N)^{(2^N)} \times (2^N)^{(2^N)} \\ &= |\mathbb{U}_{\Omega^*}| \end{aligned} \quad (5.54)$$

Base indexing provides the one-to-one map between the configurations of Ω^* and (excluding the input unit) and the states of M . From Proposition 10.1, for each computron there is an equivalent fully-connected computron. From 10.2, there is a

one-to-one mapping between finite state automata of 2^N states with binary input and a fully-connected computron of $N + 1$ unit, including the input unit. Therefore, for any finite state automata there exists an equivalent computron. $\square$

Lemma 11 (Diversity measure) *Diversity is a sub-additive measure on Q .*

Monotonicity:

$$A \tilde{C} B \implies \|A\|_w < \|B\|_w \quad (5.55)$$

Let $A = \{a, b, c\}$, $B = \{a, b, c, x\}$ with corresponding weights $\{w_a^A, w_b^A, w_c^A\}$ and $\{w_a^B, w_b^B, w_c^B, w_x^B\}$, respectively, satisfying subset definition given in Equation 5.17. Let $\vec{A}^* = [bac]$ be the optimal ordering of A-tuple that maximizes $\sup_{\vec{A}} \|\vec{A}\|_w$, and construct $\vec{B}^* = [bax]$.

$$\begin{aligned} \|\vec{A}^*\| &= w_a^A + w_b^A \rho_{ab} + w_c^A \rho_{cb} \rho_{ca} \\ \|\vec{B}^*\| &= w_a^B + w_b^B \rho_{ab} + w_c^B \rho_{cb} \rho_{ca} + w_x^B \rho_{xb} \rho_{xa} \rho_{xc} \end{aligned} \quad (5.56)$$

Therefore, $\|A\|_w = \|\vec{A}^*\|_w < \|\vec{B}^*\|_w < \sup_{\vec{B}} \|\vec{B}\|_w = \|B\|_w \square$

Sub-additivity:

$$\|A_0 \tilde{\cup} A_1\|_w \leq \|A_0\|_w + \|A_1\|_w \quad (5.57)$$

Let $A = \{a, x\}$, $B = \{b, x\}$ and $C = A \tilde{\cup} B = \{a, b, x\}$ with corresponding weights $\{w_a^A, w_x^A\}$, $\{w_b^B, w_x^B\}$, and $\{w_a^C, w_b^C, w_x^C = w_x^A + w_x^B\}$, respectively. Let $\vec{C}^* = [bxa]$ be the optimal ordering of C-tuple that maximizes $\sup_{\vec{C}} \|\vec{C}\|_w$. Construct $\vec{A}^* = [xa]$ and $\vec{B}^* = [bx]$ as lifting from $\vec{C}^*$ with same ordering within. Then:

$$\begin{aligned} \|\vec{C}^*\|_w &= (w_b^B) + (w_x^A + w_x^B) \rho_{xb} + (w_a^A) \rho_{ab} \rho_{ax} \\ \|\vec{A}^*\|_w &= \quad \quad \quad + (w_x^A) \quad \quad \quad + (w_a^A) \rho_{ax} \\ \|\vec{B}^*\|_w &= (w_b^B) + (w_x^B) \rho_{xb} \end{aligned}$$

Since $\rho \in [0, 1]$, $\|C\|_w = \|\vec{C}^*\|_w \leq \|\vec{A}^*\|_w + \|\vec{B}^*\|_w \leq \|A\|_w + \|B\|_w \square$

Lemma 12 (Diversity-Induced Distance) *'DiD' defined as $\varphi(A, B) = \frac{\|A \tilde{\Delta} B\|_w}{|Q|}$ is a metric on Q^**

Let $A, B, C \in Q^*$ and diversity-induced distance $\varphi(A, B) = \frac{\|A\tilde{\Delta}B\|_w}{|Q|}$, then from definitions $\varphi(A, A) = 0$, and $\varphi(A, B) = \varphi(B, A) \geq 0$. We need to show the triangle-inequality property to establish DiD as a metric: $\varphi(A, B) \leq \varphi(A, C) + \varphi(C, B)$. First, we will show that $A\tilde{\Delta}B \subseteq (A\tilde{\Delta}C)\tilde{\cup}(C\tilde{\Delta}B)$. Let $A = \{a, x, y, p\}$, $B = \{b, x, x, p\}$, and $C = \{c, y, z, p\}$ with corresponding weights:

$A\tilde{\Delta}C$	$C\tilde{\Delta}B$	$A\tilde{\Delta}B$	$(A\tilde{\Delta}C)\tilde{\cup}(C\tilde{\Delta}B)$
w_a		w_a	w_a
	w_b	w_b	w_b
w_c	w_c		w_c
w_x^A	w_x^B	$ w_x^A - w_x^B $	$(w_x^A + w_x^B)$
$ w_y^A - w_y^C $	w_y^C	w_y^A	$ w_y^A - w_y^C + w_y^C$
w_z^C	$ w_z^C - w_z^B $	w_z^B	$w_z^C + w_z^C - w_z^B $
$ w_p^A - w_p^C $	$ w_p^C - w_p^B $	$ w_p^A - w_p^B $	$ w_p^A - w_p^C $ $+ w_p^C - w_p^B $

For each row, weight for $A\tilde{\Delta}B$ is less than or equal to the weight for $(A\tilde{\Delta}C)\tilde{\cup}(C\tilde{\Delta}B)$. Therefore, $A\tilde{\Delta}B \subseteq (A\tilde{\Delta}C)\tilde{\cup}(C\tilde{\Delta}B)$. Then the rest follows from Lemma 11:

$$\begin{aligned}
A\tilde{\Delta}B &\subseteq (A\tilde{\Delta}C)\tilde{\cup}(C\tilde{\Delta}B) \\
\|A\tilde{\Delta}B\|_w &\leq \|(A\tilde{\Delta}C)\tilde{\cup}(C\tilde{\Delta}B)\|_w \\
&\leq \|A\tilde{\Delta}C\|_w + \|C\tilde{\Delta}B\|_w
\end{aligned} \tag{5.58}$$

□

Lemma 13 (Maximal entanglement) *Let $\mathbf{P} = \{B_0, B_1, \dots\}$ be a partitioning of Q (i.e. $\cup_i B_i = Q$, and $B_i \cap B_j = \emptyset, i \neq j$) with $\mathbf{P}^0 = \{Q\}$ and $\mathbf{P}^\infty = \{\{x_0\}, \{x_1\}, \dots\}$, $\forall x_i \in Q$, then:*

$$\|\mathbf{P}^0\| = \|\mathbf{P}^\infty\| \geq \|\mathbf{P}\| \tag{5.59}$$

The proof is based on a combinatorics argument. Entanglement of $\mathbf{P}^0$ is calculated by shuffling all the elements of universe Q within a single set, whereas

entanglement of $\mathbf{P}^\infty$ is calculated by shuffling sets of all single elements, to find the supremum.

$$\begin{aligned}
\|\mathbf{P}^0\| &= \sup_{\vec{P}} D(\vec{P}) = \|Q\| = \sup_{\vec{Q}} d(\vec{Q}) \\
\|\mathbf{P}^\infty\| &= \sup_{\vec{P}} D(\vec{P}) = \|\{x_i\}\| + \|\{x_j\}|\{x_i\}\| + \dots \\
&= d(x_i) + d(x_j|x_i) + \dots \\
&= \sup_{\vec{Q}} d(\vec{Q})
\end{aligned} \tag{5.60}$$

Therefore, $\|\mathbf{P}^0\| = \|\mathbf{P}^\infty\|$. For any other partition $\mathbf{P}$, entanglement is calculated by shuffling the set blocks of the partition together with shuffling the elements within each set block. Therefore, its supremum is less than or equal to that of $\mathbf{P}^0$ which seeks the supremum through all possible orderings. As a simple case consider $Q = \{a, b, c\}$. Let $\mathbf{P}^0 = \{\{a, b, c\}\}$, $\mathbf{P}^\infty = \{\{a\}, \{b\}, \{c\}\}$, and $\mathbf{P}^1 = \{\{a, b\}, \{c\}\}$

$\ \mathbf{P}^0\ $	$\ \mathbf{P}^\infty\ $	$\ \mathbf{P}^1\ $
$\sup_A d(\cdot)$	$\sup_B d(\cdot)$	$\sup_C d(\cdot)$
A	B	C
abc	abc	abc
acb	acb	
bac	bac	bac
bca	bca	
cab	cab	cab
cba	cba	cba

Therefore, $\|\mathbf{P}^0\| = \|\mathbf{P}^\infty\| \geq \|\mathbf{P}\| \square$

Lemma 14 (Basin partition) *The set of all basins of an autonomous computron is a partition of its configuration space, denoted as $\mathcal{P}_M = \{B_0, B_1, \dots\}$, with the corresponding attractors $\mathcal{A}_M = \{\vec{A}_0, \vec{A}_1, \dots\}$.*

A partition of S is $\mathcal{P} = \{B_0, B_1, \dots\}$, where $B_i \subset S$, $\cup_i B_i = S$ and $B_i \cap B_j = \emptyset, i \neq j$. So we need to show that each state belongs to some basin and no state belongs to more than one basin. Due to finite state nature of the computron this is trivial. Let $P_n(s_0) = \{s_0, s_1, \dots, s_n\}$ be the n -step forward path of state s_0 . Since

$|S| < \infty, \exists k \leq |S|$ such that $s_{k+1} \in P_k(s_0)$ and $s_{i+1} \notin P_i(s_0), \forall i < k$. Denote $s_{k+1} = s_q (q \leq k)$. Then, $\vec{A} = [s_q, s_{q+1} \dots s_k]$ is an attractor of the computron, as per definition, and $s_0 \in B(\vec{A})$. Therefore each state is an element of some basin.

Now we show that no state belongs to more than one basin. Assume $s \in B(\vec{A}_a)$ and $s \in B(\vec{A}_b)$ with $\vec{A}_a = [s_{q_a} \dots s_{k_a}]$ and $\vec{A}_b = [s_{q_b} \dots s_{k_b}]$. If $k_b < k_a$ then $\exists i = k_b \ni s_{i+1} \in P_i(s)$ which means $\vec{A}_b$ is not an attractor as it violates attractor definition, and similarly if $k_a < k_b$ then $\vec{A}_a$ is not an attractor. Therefore $k_a = k_b$ and $\vec{A}_a = \vec{A}_b \square$

Lemma 15 (Minimal complexity) *Absorbing and idle computrons have the least complexity (null) compared to any other computron.*

$$\Psi_{M^0} = \Psi_{M^\infty} = 0 \leq \Psi_M \quad (5.61)$$

Absorbing computron M^0 has a single fixed-point attractor which all states iterate into therefore its basin partitioning is $\mathcal{P}_{M^0} = \{S\} = \mathbf{P}^0$, and idle computron M^∞ each state is mapped to itself, therefore its basin partitioning is $\mathcal{P}_{M^\infty} = \{\{s_0\}, \{s_1\} \dots\} = \mathbf{P}^\infty$. Hence from definition of complexity, $\Psi_{M^0} = \Psi_{M^\infty} = 0 \leq \Psi_M \square$

Lemma 16 (Coloring) *Let $\mathbf{p}_a, \mathbf{p}_b$ two random initial distributions with non-zero entries for each state, and let $\boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_b$ be the corresponding steady-state distributions in an asynchronous computron. Define, the ‘gray color’ of a state $c(s)$ as:*

$$c(s) = \begin{cases} \pi_b(s)/\pi_a(s) & \text{if } \pi_a(s) > 0 \\ 0 & \text{else} \end{cases} \quad (5.62)$$

Two states will, almost always, have the same shade of gray if and only if they are elements of same recurrent class, provided all classes of the computron are ‘distinguishable’; i.e. for two random non-zero initial distributions, $c(s) = c(q) > 0 \iff s, q \in C_i$, almost always.

Let $\mathcal{C}_M = \{C_1, C_2, \dots, C_K\}$ be the recurrent classes of a computron, and $\boldsymbol{\pi}_M = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots, \boldsymbol{\pi}_K\}$ the corresponding steady-state distributions of the class. Then:

$$\boldsymbol{\pi}_a = \sum_{k=1}^K \chi_k(\mathbf{p}_a) \boldsymbol{\pi}_k, \quad \boldsymbol{\pi}_b = \sum_{k=1}^K \chi_k(\mathbf{p}_b) \boldsymbol{\pi}_k \quad (5.63)$$

First we show, $s, q \in C_m \implies c(s) = c(q)$. By definition recurrent (see Terminology) classes are non-overlapping. If $s, q \in C_m$, then:

$$\begin{aligned} \pi_a(s) &= \chi_m(\mathbf{p}_a) \pi_m(s), \quad \pi_a(q) = \chi_m(\mathbf{p}_q) \pi_m(q) \\ \pi_b(s) &= \chi_m(\mathbf{p}_b) \pi_m(s), \quad \pi_b(q) = \chi_m(\mathbf{p}_b) \pi_m(q) \end{aligned} \quad (5.64)$$

Therefore,

$$c(s) = \pi_a(s)/\pi_b(s) = \frac{\chi_m(\mathbf{p}_a)}{\chi_m(\mathbf{p}_b)} = \pi_a(q)/\pi_b(q) = c(q) \quad (5.65)$$

We define ‘*distinguishable*’ as follows: Let $\vec{W}_{x \rightarrow y} = x_0 x_1 \dots x_k$ be a *walk* state s_x to state s_y . We define the walk probability as $p(\vec{W}_{x \rightarrow y}) = \prod_{i=0}^{k-1} \mathbf{P}_{x_i x_{i+1}}$. Let $\tilde{W}_{x \rightarrow y}$ denote set of all different walks from state s_x to state s_y . We define reach probability from s_x to s_y as $p_{x \rightarrow y} = \sum_{\vec{W} \in \tilde{W}} p(\vec{W}_{x \rightarrow y})$. We define reach probability from a state s_x to an attractor A as $p_{x \rightarrow A} = \sum_{s_y \in A} p_{x \rightarrow y}$, where the walk from $s_y \in A$ there does not have any other element of A . Two recurrent classes A and B are said to be ‘*equidistant*’ with respect to state s_x , if $p_{x \rightarrow A} = p_{x \rightarrow B}$. Two recurrent classes A and B are said to be not ‘*distinguishable*’ if they are equidistant with respect to all states in S . Define the $|S| \times |\mathcal{C}|$ matrix $\mathbf{P}^{\mathcal{C}}$ as follows:

$$\mathbf{P}^{\mathcal{C}} = \begin{bmatrix} p_{0 \rightarrow C_1} & p_{0 \rightarrow C_2} & \dots & p_{0 \rightarrow C_K} \\ p_{1 \rightarrow C_1} & p_{1 \rightarrow C_2} & \dots & p_{1 \rightarrow C_K} \\ \vdots & & & \end{bmatrix} \quad (5.66)$$

If any two columns of $\mathbf{P}^{\mathcal{C}}$ are equal then the corresponding recurrent classes are indistinguishable. Let $\boldsymbol{\chi}(\mathbf{p}_a) = [\chi_1(\mathbf{p}_a) \dots \chi_K(\mathbf{p}_a)]$, then:

$$\boldsymbol{\chi}(\mathbf{p}_a) = \mathbf{p}_a \mathbf{P}^{\mathcal{C}} \quad (5.67)$$

Now, we show that, almost always, for any two recurrent classes C_m, C_n that are ‘*distinguishable*’, $s \in C_m, q \in C_n \implies c(s) \neq c(q)$. Assume the contrary:

$$\begin{aligned}
c(s) = c(q) &\implies \frac{\chi_m(\mathbf{p}_a)}{\chi_m(\mathbf{p}_b)} = \frac{\chi_n(\mathbf{p}_a)}{\chi_n(\mathbf{p}_b)} \\
&\implies \frac{\mathbf{p}_a \cdot \mathbf{P}^c[m]}{\mathbf{p}_a \cdot \mathbf{P}^c[n]} = \frac{\mathbf{p}_b \cdot \mathbf{P}^c[m]}{\mathbf{p}_b \cdot \mathbf{P}^c[n]}
\end{aligned} \tag{5.68}$$

The set of distinct initial distributions pairs $\mathbf{p}_a, \mathbf{p}_b$ that are solution to Equation 5.68 represent a plane in $\mathfrak{R}^{|S|}$ which may or may not be a probability distribution. The ratio of size of this set to size of all possible initial distribution pairs is $\frac{\mathcal{O}(|S|-1)}{\mathcal{O}(|S|)} \approx 0$, hence contrary assumption leads to contradiction almost always $\square$

Lemma 17 (Ergodic bases) *Recurrent classes form a non-overlapping spanning set (basis) for the impact sets of all states. Let $\mathcal{C}_M = \{C_1, C_2, \dots, C_K\}$ be the set of recurrent classes of a computron and $\chi_s = \{\chi_1(s), \chi_2(s), \dots, \chi_K(s)\}$ the ergodic decomposition of impulse response π_s :*

$$\chi_k(s) = \sum_{q \in C_k} \pi_s(q) \tag{5.69}$$

then:

$$\sum_k \chi_k(s) = 1, \text{ with } \chi_k(s) \geq 0, \forall s \in S \tag{5.70}$$

$$\chi(\mathbf{p}_a) = \mathbf{p}_a \mathbf{P}^c \tag{5.71}$$

$$\chi_k(s) = \mathbf{p}_s^\delta \mathbf{P}^c[k] = \mathbf{P}^c[s][k] = p_{s \rightarrow C_k}$$

$$\sum_k \chi_k(s) = \sum_k p_{s \rightarrow C_k} = 1 \tag{5.72}$$

$\square$

Lemma 18 (Fuzzy Partition) *The basins of attraction of a driven computron is a fuzzy partition of its configuration space, $\tilde{\mathcal{P}}_M = \{\tilde{B}(C_1), \tilde{B}(C_2), \dots, \tilde{B}(C_K)\}$.*

Basins of attraction for an asynchronous computron is a weighted-set $\tilde{B}(C_k)$:

$$\tilde{B}(C_k) = \{s : \chi_k(s) > 0, w_{\tilde{B}}(s) = \chi_k(s)\} \tag{5.73}$$

A fuzzy partition of S is the set of weighted-sets $\tilde{\mathcal{P}} = \{\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_K\}$, where $\cup_i \tilde{B}_i = S$, and $\sum_k w_{\tilde{B}_k}(s) = 1, \forall s \in S$. The proof follows directly from definition of basin, Equation 5.73 and Lemma 17 $\square$

Chapter 6

Morphogenesis

“It is not birth, marriage, or death, but gastrulation which is truly the most important time in your life.”

Lewis Wolpert

The previous two chapters provided analysis of inter-cellular dynamics and intra-cellular regulation. In this chapter, we combine the two building blocks and expand inter-cellular dynamics into vector of morphogens. This combined model is shown to be capable of generating life-like organisms through self-organization. This chapter is less technical mathematically, and is designed to be read on a standalone basis, particularly for readers with engineering or biology background.

The development of form in an embryo is the result of a series of topological and informational symmetry breakings. We introduce the Vector-Reaction-Diffusion-Drift (VRDD) system where the limit cycle of spatial dynamics is morphogen concentrations with Dirac delta-type distributions. This is fundamentally different from Turing reaction-diffusion system, as VRDD generates system-wide broken symmetry. We developed *fundamental forms* from spherical blastula with a single organizing axis (rotational symmetry), double axis (mirror symmetry), and triple axis (no symmetry operator in 3D). The dynamics for cell differentiation is introduced, where gene regulatory states are modeled as a finite state automata (FSA). The state-switching of FSA is based on local morphogen concentrations as epigenetic information that changes dynamically. We grow complicated forms hierarchically in spatial sub domains using the FSA model coupled with VRDD system. Using the

integrated simulation model with four modules (topological, physical, chemical and regulatory), we generated life-like forms such as *hydra*.

6.1 Terminology

In biology, *morphogenesis* is the process that causes an organism to develop its form. It controls the organized spatial growth of cells during the embryonic development of the organism. Morphogenesis can also take place in mature organisms such as regeneration of tissue after damage or normal maintenance of tissue that is being shed from the body continually.

Embryonic development, also known as *embryogenesis* is the morphogenesis of an embryo. It starts with the fertilization of the egg cell by a sperm cell, which results in the single cell called *zygote*. The zygote goes through mitotic divisions with no significant change in the total volume of the divided cells compared to the original zygote and no cell differentiation at this stage. These undifferentiated cells are *totipotent*, meaning they can differentiate into all possible cells of the organism. As we shall see in our model, totipotency corresponds to being at State-0 of the finite state automata model of the gene regulatory network.

These divided mini cells are called blastomeres. After the 7th mitotic division with 128 resultant blastomere cells, the embryo is called a *blastula*. At this stage blastomeres are typically arranged as a layer on a ball, surrounding a fluid-filled cavity. Mitotic division continues for another 5-7 times, resulting in c.5000-10000 blastomere cells lined up on the surface a sphere. At this stage *gastrulation* starts. During gastrulation cells migrate to the interior of the blastula and polar proliferation starts. The cells at the polar ends of the gastrula divide faster generating head-toe elongation. During this stage cells start major differentiation, losing their totipotentiality.

Our model starts from blastula stage with undifferentiated cells represented as nodes on the surface of a spheric mesh graph with each cell's state in their finite state automata model of GRN set to State-0, i.e. totipotency. The cells start changing their state in the finite state automata model of the gene regulatory network due to inter-cellular morphogenic cues fed to the fate-functions. Cell differentiation

and differential growth start shaping the blastula into form. Differential growth is modeled as surface membrane expanding incrementally to the normal of the surface at the node and then mesh graph undergoing node splitting once the mesh faces connecting to the cell becomes larger than a threshold which corresponds to cell volume expansion followed by mitotic division.

Morphogens are signaling molecules that act directly on cells to produce cellular responses depending on the local concentration levels. Some examples of well-studied signaling molecules are *Sonic Hedgehog* and *Bicoid*. The non-uniform spatial distribution of morphogens on the tissue governs the embryonic development. Typically, morphogens are produced by *source cells* located at so-called *organizing poles* within the tissue. Morphogens diffuse throughout the surrounding tissue (either by chemical diffusion or by cell-to-cell signaling called *juxtacrine signaling*), setting up *morphogen gradients* across the tissue between the polar source cell regions. These gradient levels act as positional cues for the cells to know where they are located in the tissue with regards to the organizing poles. This is called the *positional information* in developmental biology. Such positional information is used by the cells for type differentiation as well as differential growth, result of which is the formation of the embryo's physical form.

This chapter will introduce our morphogenesis model that uses vector-reaction-diffusion-drift dynamics, which is an extension of the dipolar attractor model that was introduced in Chapter 4. Four morphogen source-field pairs are used with their interaction set through the *interaction matrix*. These four morphogen sources self-organize on a blastula forming polar pairs placed orthogonally with corresponding fields gradients stretching across the pole pairs linearly. Hence, the system self-organizes into organizing regions and morphogen gradients observed in biology. The positional cues obtained from the gradient fields trigger differential growth and cell differentiation. *Fundamental forms* with single axis (*rotational symmetry*), dual axis (*mirror symmetry*) and even triple axis (no biological form known with triple axis) are generated. Differentiated regions reorganize polar regions within the subregions in an hierarchic orchestration which allows limb growths at prescribed positions possible.

6.2 Background

The morphogenesis of an embryo is the emergence of its form from the singularity of a zygote. The theory of embryonic development has been a long-standing issue in biology and can be traced back to Aristotle. In Book II of *“On the Generation of Animals”* [93], Aristotle rejects preformationism and introduces the concept of epigenesis and orderly development towards a final cause. Meckel and Serres, in the 18th century, proposed the recapitulation theory [94]. The embryogenesis of the higher-order organisms was a progression through the formation stages of lower life forms. In the 19th century, Haeckel expanded recapitulation to biogenetic law. An animal embryo’s development was the replay of its species evolutionary forms; ontogeny recapitulated phylogeny. Later, Roux introduced the mosaic theory. The embryonic development is a well-choreographed unfolding of form; after a few divisions of the zygote, the cells attained fate that would generate specific parts of the developing organism. With the advances in genetics, the debate of “neo”-preformationism and epigenetics intensified. Currently, we seek a theory of embryogenesis that would explain the unfolding of form from the singularity of a zygote.

Pattern formation is a process of building spatial structures out of none. In his seminal paper *“Morphogenesis”* 1952 [1], Alan Turing argued that a non-linear dynamical system of the two reactants diffusing on a tissue at different rates can spontaneously generate pattern starting from a homogeneous initial condition if provided with some perturbation. Meinhardt and Gierer published their equivalent of reaction-diffusion formulation for pattern formation [3], which was based on interacting morphogens that had short-range activation and long-range inhibition, very much similar to morphogen reactants proposed by Turing. However, their theory was further expanded with morphogen sources that had shallow gradients across the tissue. Such source gradients were not generated by the morphogen dynamics but were driving it exogenously. Meinhardt and Gierer claimed that the source densities were expected to change slowly as an effect of cell differentiation. While the proposed theory could explain generation of morphogen peaks or spatial periodicity, it did not explain how the driving shallow source gradients were formed in the first place. Since then, various types of reaction-diffusion models have been introduced [95]. Although mathematically elaborate, reaction-diffusion as a mechanism

for pattern generation in animals is criticized due to its need for the tuning of system parameters, limited integration with tissue mechanics, and scarce *in vivo* evidence.

Although Turing's reaction-diffusion system constitute an important framework for explaining self-organization through local interactions, it generates patterns, not form. As Turing allegedly said [96] regarding the formation of zebra skin patterns:

"Well, stripes are easy, but what about the horse part?"

The generation of form requires spatial symmetry-breaking at the organism length scale, whereas pattern formation is based on breaking the local symmetry, which results in spatial quasi-periodicity. Form generation in a living object, starting with the establishment of a body plan, requires cells to know their positions. This is the basis of positional information theory that was developed by Wolpert [11]. Cells are assumed to acquire positional identities through a coordinate system formed by morphogen gradients that span the organism or its sub-region. Subsequently, they differentiate based on such positional information.

In stark difference to Turing's reaction-diffusion systems, the positional information theory was built on *in vivo* experimental results rather than mathematics. Morphogen gradients that provide positional information are observed in many developing systems [97]. Early studies on chick, hydra and insect limbs showed the existence of morphogen gradients along the main body axes [98], [99]. Recent experimental data show that Dpp, Sonic hedgehog (Shh) and activin exhibit gradient distribution across the organism during embryonic development [100]. Such gradient thresholds trigger the expression of target genes [101]. Other studies showed that the hedgehog family of proteins have spatial gradients on *Drosophila*'s larval cuticle and wing imaginal disc, as well as on the vertebrate neural tube [102]. In *Xenopus* embryo, activin gradients correlate with the expression of *goosecoid* and *Xbra*.

In the vertebrate neural tube, Shh protein acts as the gradient forming morphogen and drives the spatial patterns of gene expression, resulting in subdivision of the ventral neuroepithelium into five domains with distinct neuronal subtypes. [103], [104]. Recent *in vivo* studies, supported by mathematical modeling of non-linear dynamics, has shown that Shh morphogen gradient is interpreted by the downstream gene regulatory network by changing the intracellular Gli activity that triggers transcriptional circuit resulting in generation of spatial patterns of gene

expression [105].

Despite this evidence of gradient distribution and corresponding gene activation, there are limited theories on the way such axes-defining gradients are formed and converted into precise positional information [106]. Possible mechanisms such as self-enhanced morphogen degradation and competition between the morphogens for binding to inhibitors have been suggested [107]. Although the pre-patterning of an organizer is assumed, it is not clear how such an organizer would establish a smooth gradient distribution across the tissue. Typically, such gradients are based on secreted molecules that can diffuse extracellularly. Alternative transport mechanisms based on cell-to-cell transport have been proposed [108], [109]. Moreover, it is challenging to maintain the established gradient against noise, degradation and fluctuations at the organizer. Any acceptable model must ensure reproducibility and robustness [110], [111], [112].

Turing’s reaction-diffusion theory and Wolpert’s positional information theory were perceived as opposing camps in explaining morphogenesis [113]:

“This [positional information-based] view of pattern formation must be contrasted with those views which explicitly or implicitly claim that in order to make a pattern it is necessary to generate a spatial variation in something which resembles in some way the pattern... View of pattern formation is characterized by the work Turing (1952) and is the antithesis of positional information.”

Lewis Wolpert (1971)

Meinhardt introduced exogenous source gradients into his equations, which provided positional information to otherwise isotropic reaction-diffusion system [4]. This was the first step towards amalgamating reaction-diffusion with positional information theory. However, attempts to establish positional information through reaction-diffusion systems faced difficulties such as (i) the generation of secondary or periodic peaks as tissue size grows, (ii) exponential decay of gradients, (iii) difficulty in establishing orthogonal gradients for primary and secondary body axes, and (iv) forming an exact number of segments [114]. Several elaborate remedies were proposed within the framework of the reaction-diffusion systems [5], [6], [115], [8].

Need for ‘self-organized’ organizers Typically, early blastula is symmetrical topologically and information-wise. A complete embryo can be formed from a fragment of the early blastodisc not containing the posterior marginal zone [7]. These experimental observations indicate that the organizer regions of the tissue that determine the body plan of an embryo is emergent and self-organized. Despite the elegant mathematics of Turing’s reaction-diffusion systems for pattern formation, and notwithstanding the well-established experimental evidence of positional information based on gradients in the embryo, to date, there does not exist a theory that can robustly explain the self-organization of the organizers, which would reproducibly and reliably generate the immense variety and detail of forms in phyla.

6.3 Theory

6.3.1 Vector-Reaction-Diffusion-Drift

Pattern-forming reaction-diffusion systems were introduced in Chapter 3, and analyzed in cellularized systems in Chapter 4. In continuous partial differential equation formalism, a reaction diffusion system with two morphogens have the following form:

$$\partial_t U = D_U \nabla^2 U + f(U, V)$$

$$\partial_t V = D_V \nabla^2 V + g(U, V)$$

where $U(\mathbf{x}, t), V(\mathbf{x}, t), \mathbf{x} \in \Omega$ are chemical reactants whose concentration levels across the tissue Ω change over time according to the reaction-diffusion equations above.

Some well-studied reaction-diffusion systems include the Gray-Scott, Barrio-Varea-Aragon-Maini and Gierer-Meinhardt models. These models generate pattern but not form. Pattern is a spatial organization that is locally differentiated but globally repeated, where as form is a spatial organization that is globally differentiated. Generating form requires the tissue to have unique differentiation such as anterior-posterior and dorso-ventral axis formation. Reaction-diffusion systems can not generate global symmetry breaking; they act as local spatial filters that feed on random fluctuations or disturbances to generate spatial differentiation within the characteristic distance of the underlying diffusion dynamics. Form of an organ-

ism, on the other hand, is a juxtaposition of topological symmetry breakings, which can be achieved in a system that incorporates a local sense of directionality, which traditional reaction-diffusion systems lack. In the Gierer-Meinhardt model, spatial isotropy is extrinsically broken by the introduction exogenous source densities.

Our *Vector-Reaction-Diffusion-Drift* (VRDD) system is the extension of the dipolar attractor system that was introduced in Chapter 4 in a vector format with more than one dipole pairs or morphogens. VRDD can create self-organized tissue-wide symmetry breaking and, thus, is capable of generating form. It has three building blocks: (i) self-organization through percolation of locally generated information (*source-field coupling*); (ii) local directionality introduced as drift on field gradients (*symmetry-breaking by drift*); and (iii) topology encoded in the interaction matrix of the morphogen vector, which sets the organizing axes (*pillars of form*).

Source-Field coupling The *source-field* concept is introduced herein where *source*, denoted by state variable U is the “organizer” in Wolpert’s nomenclature of positional information theory [57] and *field*, denoted by state variable V , corresponds to *morphogen field*. *Source-field* coupling is defined by Equation 6.1. Any change in *source* ripples through the *field* based on the spatio-temporal solution.

$$\partial_t V = \mathcal{R}(U - V) - \beta V + D \nabla^2 V \quad (6.1)$$

$$\mathcal{R}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

where $U(\mathbf{x}, t)$ is the *source* concentration across the tissue, and $V(\mathbf{x}, t)$ is the *field* concentration across the tissue. $\mathcal{R}$ is the ramp function through which *source* pumps-up the *field*. β and D are the degradation and diffusion rates of the *field*, respectively. In the proposed model, *field* is generated by the localized *source* and dispersed through the tissue by inter-cellular diffusion.

Symmetry breaking by drift The concept is motivated by the following “Lego Game” problem that we posed in Chapter 4, recapped here:

A group of N players join in a circle; each player is given a bag of a random number of Lego bricks. Each member can exchange Legos with

members on their left or right in the circle. Members are undifferentiated, and they all execute the same rule set. The goal is to combine all Legos at one member.

Such a system, starting from a random initial condition with no spatial correlation, ending in a unique spatial distribution of Dirac delta-type function, would be achieving global symmetry-breaking with local interactions. We propose that such a system can be implemented by a drift of *source* on the gradient of *field*. The drift of a morphogen *source* U based on a generic gradient vector field $\mathbf{F}$ is given by $\partial_t U = -\nabla \cdot (U\mathbf{F})$. We want *source* to drift up gradient of the morphogen *field*. Hence, our *source-field* coupled Reaction-Diffusion-Drift system is given in Equation 6.2:

$$\partial_t U = -w\nabla \cdot (U\nabla V) + D_U \nabla^2 U \quad (6.2a)$$

$$\partial_t V = \mathcal{R}(U - V) - \beta V + D_V \nabla^2 V \quad (6.2b)$$

where $\mathcal{R}$ is the ramp function and w is the drift rate (positive as attraction, negative as repulsion). Source profile is initialized with a positive random noise level and Field is initialized to zero. The limit cycle of this spatially distributed dynamic system is a Delta dirac-type singular peak with a system-wide broken symmetry. Figure 6.1 shows the initial and final condition of the *source* and *field* distribution on a 1D cyclic domain running Equation 6.2.

Pillars of form We expand the Lego Game into multi-colour. The goal is to have pillars of the same colour with specified relative positioning to other colours. This is the conceptual framework for our morphogen vector model. The VRDD system is based on an interacting network of M morphogens of *source-field* pairs. Each morphogen *source* reacts to others through a drift on their gradient fields with weights set by the interaction matrix $\mathbf{W}$. The vector formulation of the VRDD system is given in Equation 6.3.

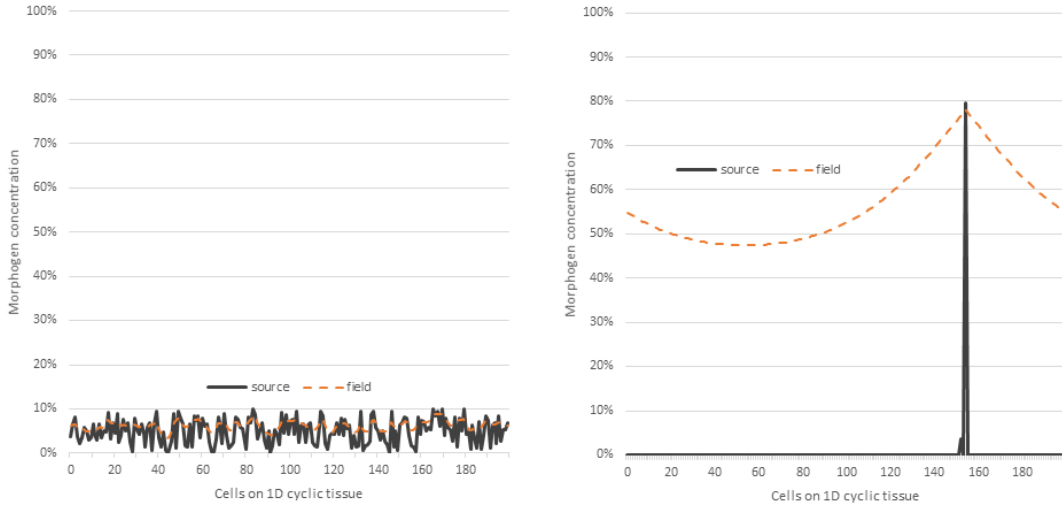


Figure 6.1: Initial vs limit-cycle distribution

1D cyclic tissue with VRDD dynamics consisting of 200 cells. Box on the left shows the initial condition with each cell having a random (noise level) morphogen *source* concentration. Box on the right shows the steady-state attractor of the system. After c.100 steps, *source* concentration falls to zero in all cell, except the organizer cell that manages to aggregate all morphogen sources. It is important to note that organizer cell was not preset exogenously but emerged as self-organization of this non-linear spatially distributed dynamical system. The dashed line is the corresponding steady-state morphogen *field* distribution. Parameters used: $w = 0.1, D_U = 0, \beta = 0, D_V = 0.2$

$$\mathbf{U} = \begin{pmatrix} {}_0U & \dots & {}_{M-1}U \end{pmatrix} \quad {}_iU(\mathbf{x}, t) \in [0, 1] \quad \mathbf{U} : \text{Source morphogen vector}$$

$$\mathbf{V} = \begin{pmatrix} {}_0V & \dots & {}_{M-1}V \end{pmatrix} \quad {}_iV(\mathbf{x}, t) \in [0, 1] \quad \mathbf{V} : \text{Field morphogen vector}$$

$$\partial_t {}_iU = -\nabla \cdot \left({}_iU \sum_j W_{ij} \nabla_j V \right) + D_{iU} \nabla^2 {}_iU \quad (6.3a)$$

$$\partial_t {}_iV = \mathcal{R}({}_iU - {}_iV) - \beta_i {}_iV + D_{iV} \nabla^2 {}_iV \quad (6.3b)$$

VRDD is a spatially-distributed non-linear dynamical system where the limit-cycle attractors are Dirac delta distribution of *source* types with relative positioning set by the interaction matrix.

Let us consider a VRDD system run on a spheric surface with two morphogens (R,G); and the interaction matrix is set as $\mathbf{w} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$. Then R *source* will be drifting away from G *source* based on the gradient of the G *field* (because $\mathbf{W}_{0,1} = -1$), and similarly G *source* will be drifting away from R *source* based on the

gradient of the R *field* (because $\mathbf{W}_{1,0} = -1$). The steady-state spatial configuration will be R and G *sources* concentrated as polar opposite peaks on the sphere, forming anterior-posterior body axis of the spheric tissue in a self-organized manner.

Now let us consider the case where VRDD system is again run on a spheric surface with 4 morphogens (R, G, B, Y), and the interaction matrix is set as $\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -0.5 \\ -1 & -1 & -0.5 & 0 \end{pmatrix}$. R and G will be again drifting away from each other based on each others' *field* gradient; and they would be ambivalent to *fields* of B and Y (because $\mathbf{W}_{0,2}, \mathbf{W}_{0,3}, \mathbf{W}_{1,2}, \mathbf{W}_{1,3}$ are zero), aligning themselves as two polar opposites similar to the previous case. B and Y, on the other hand will be drifting away from both R and G (because $\mathbf{W}_{2,0}, \mathbf{W}_{2,1}, \mathbf{W}_{3,0}, \mathbf{W}_{3,1}$ are -1), hence they would be pushed to the equator of R, G poles. Secondly, B and Y are being pushed away from each other (because $\mathbf{W}_{2,3}, \mathbf{W}_{3,2}$ are -0.5), aligning as polar opposites on the equator of R, G. The steady-state spatial configuration will be dual body axis (anterior-posterior and dorso-ventralis) formed by RG and BY poles, respectively.

6.3.2 Biological implications of VRDD

Here we discuss the biological implications of our VRDD theory on its three building blocks. We conceptualize Sonic Hedgehog protein (Shh) and Bicoid protein (Bcd) and their encoding genes as basis for biological relevance to our VRDD model. Shh is a well-studied signaling molecule in development that has crucial role in embryogenesis [102]. Shh is known to be generated in organizing regions and form long-range gradients that act as positional information for development. For example, the notochord, found ventral to the neural tube (transient during vertebrate development) plays a key role in signaling and coordinating development during gastrulation and provides the organizing region for Shh generation which subsequently diffuses to generate morphogen gradient. Also, in early tooth development, the region called primary enamel knot generates Shh which then leads to morphogen gradient that provides positional information. Similarly, Bicoid protein (Bcd) and its encoding gene has been widely studied as morphogen gradients [116] and is relevant for conceptualizing our VRDD theory's biological implications.

- i *Source-field coupling*: Our VRDD model would predict existence of an activator of Shh gene expression which would correspond to the *source*. High concentrations of *source*, i.e. high concentration of the Shh gene activator would drive the level of Shh protein expression. The *field* would be the Shh protein concentration itself. The ramp function in Equation 6.1 essentially says that *source* drives field generation until *field* concentration exceeds *source* concentration. Therefore, our VRDD model would predict existence of an inhibitor of Shh gene expression which would be triggered once field concentration exceeds that of source. This catalytic system would effectively generate the source-field coupling. Numerical solution to Equation 6.1 shows rapid decay close to *source*, but more gradual linear-like decline away from the *source*, which has been a proposed requisite for morphogen based positional information to have robustness and long-range action[111].
- ii *Symmetry-breaking by drift*: The key concept of our VRDD model is organism-wide topological symmetry breaking achieved by local gradient based drift (hence the name reaction-diffusion-drift model). In its continuous-space formulation, drift requires a mechanism that would generate directional diffusion based on the direction of an underlying gradient field. Such directional diffusion is difficult to generate in a chemical system. However, once we consider discretizing our equations in space as provided in Chapter 4, the results lead naturally to cell-to-cell transport based mechanism. In discretized formulation, field gradient operator becomes a difference of concentration across the set of neighboring cells of a given cell. Hence, gradient based drift is an active transport of *source* based on *field* concentration differences of neighboring cells. This can be biologically realized by a juxtacrine signaling based mechanism. Our VRDD model would predict that ligand-receptor concentration on the boundary of two cells on such a juxtacrine signaling system should be proportional to the *field* difference between such cells. Hence, higher differential cell pairs would have higher transport resulting in discretized gradient drift. Such discretized models based on cell-to-cell signaling for morphogen transport and pattern formation have been proposed [117], [118].
- iii *Interaction matrix of morphogen vector*: The morphogen vector in our VRDD

model consists of a set of morphogens with their respective gradients. The drift is based on the perceived single field which is a linear combination of the fields of morphogen vector, which is defined by the corresponding row of the interaction matrix. To biologically realize such linear combination, our VRDD model would predict that a kernel of activators (or inhibitors) should exist that would drive the ligand-receptor concentration on the cell boundaries based on their corresponding field differences of each morphogen. The weights in this kernel would define the interaction matrix row components. Such a system would generate Equation 6.3 of VRDD model.

6.3.3 GRN as a Finite State Automaton

Various theoretical models for cell differentiation using biochemical networks have been proposed. As presented in Chapter 5, the *Boolean Network* model was introduced by Kauffman [2], and have since been used extensively for modeling the gene regulatory network (GRN) dynamics, including fission yeast cell cycle, Arabidopsis thaliana floral morphogenesis and mammalian cell cycle [75]. In these models, differentiated cell-types correspond to attractors of the underlying non-linear dynamical system. Globally coupled metabolic reaction network with active transport of chemicals has been shown to generate clustering of differentiated cells [76]. Neural-network type models were also considered [12], [15]. Spatially-extended GRN models have been recently used for pattern generation [77]. A network of morphogens reacting through neural-network type function where spatial connectivity is either a traditional diffusion or direct contact induction (effectively a discretized cellular model) have categorical pattern generating capabilities [78].

Complex pattern formations do not need to be due to complicated hierarchical and layered GRNs. Cells that have the same genome can differentiate into clusters of different behavior based on their developmental history, indicating that epigenetic lineage is likely to be switching the genetic stage of a cell [79]. Spatial complexity emerges when epigenetic information provides feedback to the GRN. Developmental mechanisms that have modular re-use of genetic tool-set, coupled with epigenetic feedback can generate a large variation of adaptive morphologies and shall be selected by evolution [78].

We recap the dynamics for cell differentiation where a gene regulatory network is modeled as a finite state automata (FSA) that was presented in Chapter 5. FSA is an abstract mathematical model of computation where the machine can be in one of a finite number of states at any given time and transitions from that state in response to external inputs.

Definition: *a finite state automaton is defined by a 5-tuple $\{Q, q_0, X, \Sigma, \delta\}$, where*

- Q is the set of FSA states
- $q_0 \in Q$ is the start state
- $X \subseteq Q$ is the set of final states
- Σ is the set of symbols representing input to FSA
- $\delta : Q \times \Sigma$ is the transition function of FSA

As depicted in Figure 6.2, FSA can be represented as a directed graph where each node constitutes a state of the machine, and the edges depict possible transition from such state to the next connected state. In this representation, FSA's transition function δ is the set of fate functions Φ_q for each state. The outgoing edge of a node is selected based on its fate function and external inputs to FSA at any given time.

The FSA model of a gene regulatory network, each cell state is defined by the expression of underlying genes. The input set for fate functions are dynamically changing epigenetic information consisting of local morphogen concentrations plus the “state-timer” which is a time decaying local morphogen that gets reset upon state transition. The state-timer morphogen provides time dimensionality to switching logic of FSA. It is a crucial part of FSA design through which stasis and aging are introduced to morphogenesis. Without losing generality, FSA can be reduced to a situation where each state has two connections, and therefore, fate functions become a tripartite complete and non-overlapping partitioning of the input set Σ into $\Sigma_0, \Sigma_1, \Sigma_2$. Let $\sigma = (f_0 \dots f_{M+1})$ represent the vector of local morphogen concentrations, including the state-timer. The fate function is given as:

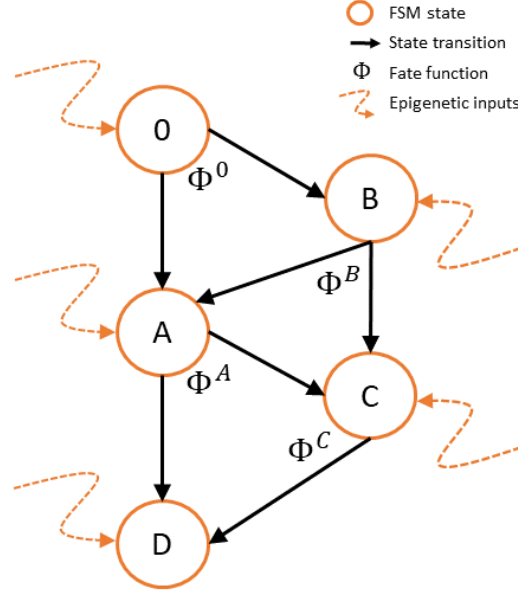


Figure 6.2: FSA model of a gene regulatory network

Each cell can be at one of a finite number of states, defined by its gene expressions, at any given time. The cell can change its state and transition to another state based on its fate-function. Fate-function is a function of epigenetic information that the cell has at any given time such as morphogen concentration levels at the cell location. Embryonic stem cells start from State 0. Each cell state is a differentiation. If a cell is reset to State 0, it will start acting as a stem cell.

$$\Phi(\sigma) = \begin{cases} 0 & \sigma \in \Sigma_0 \quad \text{STAY} \\ 1 & \sigma \in \Sigma_1 \quad \text{SWITCH TO A} \\ 2 & \sigma \in \Sigma_2 \quad \text{SWITCH TO B} \end{cases}$$

We can further reduce fate function Φ so that the tripartition of the input space can be scripted as a logic function. Let φ_α denote a threshold gate:

$$\varphi_\alpha(x) = \begin{cases} 0 & x < \alpha \\ 1 & x \geq \alpha \end{cases}$$

Let $\mathbf{P} = (p_0 \dots p_{M+1})$ denote logic values of input set σ , where $p_i = \varphi_{\alpha_i}(f_i)$, and let $L_{stay}(\mathbf{P})$ and $L_{AB}(\mathbf{P})$ be logic functions on $\mathbf{P}$. Then, the fate function becomes:

$$\tilde{\Phi}(\mathbf{P}) = \begin{cases} \text{STAY} & \text{if } L_{stay} \\ \text{SWITCH TO A} & \text{if } \neg L_{stay} \wedge L_{AB} \\ \text{SWITCH TO B} & \text{if } \neg L_{stay} \wedge \neg L_{AB} \end{cases} \quad (6.4)$$

Definition: a reduced fate function $\tilde{\Phi}$ is defined by a list of threshold values and 2 logic functions $\{\alpha_0, \dots, \alpha_{M+1}, L_{stay}, L_{AB}\}$.

Definition: a GRN state S is defined by its reduced fate function and two connected “next” states $\{\tilde{\Phi}, S_A, S_B\}$

In this model, the GRN of the zygote is at the start state of FSA. All other states of FSA, by construct, are reachable through a set of state transitions. As cells divide, each new cell begins to run its own FSA commencing from the state of the parent cell. Cells in the start state are totipotent cells; they can transition into any further cell type given a particular sequence of external inputs triggering state transitions.

The input to fate functions are the local morphogen concentrations that develop over time through the underlying VRDD systems. This model allows for us to generate spatial sub-regions consisting of cell neighborhoods with common FSA states and grow complicated forms hierarchically in spatial sub-domains. Cells that are at a given FSA state have common gene expressions that determine the functional parameters of the cell; in our case, these are the parameters of underlying VRDD systems, including interaction matrix $\mathbf{W}$ and growth kernel $\mathbf{g}$.

The FSA model allows morphogen gradients to be re-established in smaller sub-regions in a dynamical fashion, which allows for hierarchical form generation. Dynamically changing morphogen gradients, rather than a static one, has been experimentally observed and been proposed as a dynamical system based generalization to Wolpert’s static positional information [119]. The FSA model incorporates such dynamically (and sub-spatially) changing gradient concept.

6.4 Model

Integration of reaction-diffusion dynamics to tissue mechanics requires chemical state to drive cell physics such as growth, division, motion, adhesion and apoptosis; and mechanical state, such as convexity, deformation, compression to provide reverse cues to chemical reactions [120]. Coupling tissue mechanics with morphogen dynamics can result in robust pattern generation [121]. In the last decade, simulation environments that model tissue mechanics and chemical dynamics have been introduced, such as **MecaGen** [23], **SynBioTIC** [122] and **COMPUCELL** [123], [124].

We developed an integrated simulation model to demonstrate self-organization of form in embryonic development starting from the blastula stage and advancing to gastrulation using the VRDD and FSA concepts outlined above. The term blastula stage refers to the period between 128 cells to c. 10,000 cells (14th zygotic cell cycle), where the early embryo is a sphere of blastomeres that surround the fluid-filled inner cavity. During this stage of development, a significant amount of activity occurs to establish cell polarity, cell specification, and axis formation. After the blastula stage, gastrulation commences where the spheric topology starts to change under the impact of organizing axes. This is the onset of morphogenesis in the embryo.

Our simulation model consists of four modules: topological, physical, chemical, and regulatory, with causation starting from the regulatory module, cascading down to the topology module. There is feedback from the chemical module to the regulatory module. This means regulatory logic of FSA is driven by chemical cues at each cell location, which is epigenetic feedback of local morphogen concentrations to gene regulation. In the current version of the model, there is no feedback from physical module to the regulatory one.

- i Topology module: Blastula-stage embryo is modeled as a spherical mesh with each node representing a cell. This module sets the spatial form of the organism and manages cell-to-cell connectivity, cell growth, and cell splitting.
- ii Physical module: Tissue mechanics of the embryo, including differential growth and elasticity, run at this module with minimally descriptive physics capturing the essential dynamics.
- iii Chemical module: Vector-Reaction-Diffusion-Drift system is simulated at this

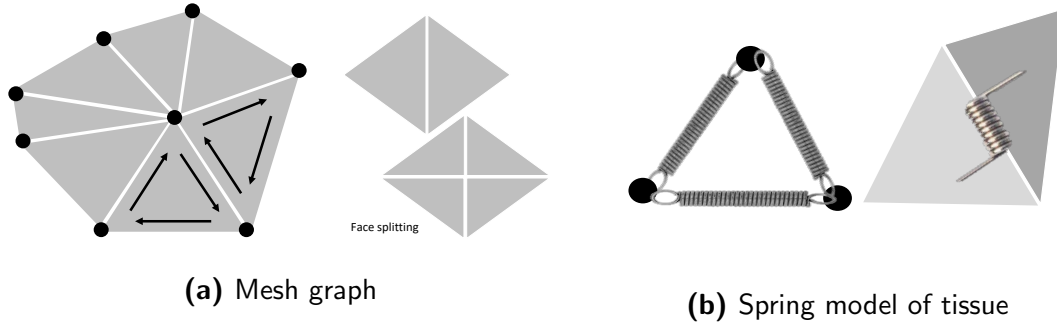


Figure 6.3: Mesh topology

module. Morphogen levels act as cues that feed downward to the physical module generating differential growth. Morphogen levels also act as cues that feed upward to the regulatory module determining state switching of FSA.

- iv Regulatory module: Finite state automata model of the gene regulatory network is maintained in this module.

6.4.1 Topology module

The topology module constitutes a mesh graph where each cell of the embryo is a node of the graph, and cell-to-cell contact is represented by edges between the nodes. The form of the embryo is represented as embedding of the mesh graph in R^3 . Each **node** has a position vector $\mathbf{p}_a$, and each **face** has a unit normal vector $\hat{\mathbf{n}}_A$ (defined as pointing away from the organism). We use the following notation:

Definition: $\mathbf{G}(\mathbf{V}, \mathbf{F}, \mathbf{E})$ denotes the mesh graph of the blastula, where **nodes** are $\mathbf{V} = \{a, b, c, \dots\}$; **faces** are $\mathbf{F} = \{A, B, C, \dots\}$; and **edges** are $\mathbf{E} = \{a, b, c, \dots\}$. $\mathbf{V}(a)$ denotes the set of **nodes** of *edge* a ; $\mathbf{V}(A)$ denotes the set of **nodes** of *face* A ; and $\mathbf{F}(a)$ denotes the **faces** of *edge* a . Additionally, let $\mathbf{V}(\mathbf{a})$ denote the set of first-degree neighbor **nodes** of **node** a , which is defined as having a common edge; i.e., $c \in \mathbf{V}(\mathbf{a}) \iff \exists b \in \{a, c\} \subset \mathbf{V}(b)$. For the triangular closed-surface mesh topology, we have the following cardinality rules: $|\mathbf{V}(a)| = 2$, $|\mathbf{F}(a)| = 2$, $|\mathbf{V}(A)| = 3$.

We used the so-called half-edge data structure to represent the mesh graph where each edge of the graph is composed of two oppositely directed half edges. This data structure puts connectivity information of the mesh into half edges, instead of faces.

The half-edge data structure is preferred, as it minimizes the amount of data storage and allows for an effective way of modeling cell splits in the embryo through edge splitting. Our data structure was built in C++; no external mesh library was used in this project. A key feature of the topology module implementation is dynamical cell splitting. As nodes change their spatial position during differential growth, edges get stretched. Cell splitting is implemented by dividing edges, which creates new nodes in the middle of split edges. The mesh data structure dynamically updates itself, maintaining a typical cell size as the organism grows.

6.4.2 Physical module

The stretchiness and bendiness of the blastula is modeled by two spring types on each edge, one connecting its two nodes and the other one its two faces: node-to-node spring with rest length of $\tilde{l}$, which regulates the stretchiness of cell, and face-to-face spring with rest angle of $\tilde{q}$, typically 0, which regulates bendiness of the tissue. We acknowledge that the physical module we are proposing oversimplifies the tissue mechanics of a real embryo at the gastrulation stage. Our purpose in this model is to create a simplistic topological medium that allows for the morphogen concentrations generated by the VRDD dynamics to generate differential growth. The topology of our model is a 2D mesh-membrane surface, which is clearly a significant deviation from biological tissue reality of 3D. Our modeling of tissue boundary as a 2D mesh-membrane has two reasons.: Firstly, tissue growth arguably is predominantly on the surface, and secondly, *in silico* implementation of a model that runs on the 2D surface rather than 3D body decreases computing implementation load (memory and processing time) by an order of magnitude 3/2 (for example 10,000 cells on the tissue boundary would correspond to 1,000,000 cell simulation in 3D body).

The energy of an edge is defined as the sum of the Hooke energies of its two springs:

$$\mathcal{U}_a = k_1(l_a - \tilde{l})^2 + k_2(q_a - \tilde{q})^2 \quad (6.5)$$

where l_a is the linear distance between the two nodes of edge a ; q_a is the angular distance between the two faces of the edge a ; and k_1, k_2 are the respective spring

constants.

$$l_a = \|\mathbf{p}_b - \mathbf{p}_c\| \quad (6.6)$$

$$q_a = \frac{1 - \hat{\mathbf{n}}_B \cdot \hat{\mathbf{n}}_C}{2} \quad (6.7)$$

where $\mathbf{V}(a) = \{\mathbf{b}, \mathbf{c}\}$ and $\mathbf{F}(a) = \{\mathbf{B}, \mathbf{C}\}$. The energy of a node is defined as the sum of its edges' energies:

$$\mathcal{U}_b = \sum_{a \in \mathbf{E}(\mathbf{b})} \mathcal{U}_a \quad (6.8)$$

where, $\mathbf{E}(\mathbf{b})$ means set of edges emanating from node $\mathbf{b}$. The differential growth of the tissue is modeled as a movement of the cellular node normal to the tissue surface governed by Equation 6.9.

$$\partial_t \mathbf{p}_a = -\nabla \mathcal{U}_a + \rho_a \hat{\mathbf{n}}_a \quad (6.9)$$

where $\hat{\mathbf{n}}_a$ is the surface unit normal vector at the position of node $\mathbf{a}$, defined as an average of the unit normal vectors of the faces that connect at $\mathbf{a}$ given as:

$$\hat{\mathbf{n}}_a = \langle \hat{\mathbf{n}}_A \rangle_{\mathbf{F}(\mathbf{a})}$$

where $\hat{\mathbf{n}}_A$ is the unit normal vector of face A , and $\mathbf{F}(\mathbf{a})$ means set of faces emanating from node $\mathbf{a}$. Differential growth is driven by spatially distributed growth factor ρ , which is an output of the chemical module of our model. In biology, embryonic tissue growth is driven by various mitogenic growth factors, such as IGFs (insulin-like growth factors) and EGFs (epidermal growth factors), which, in our model, are simplified to a mitosis-inducing, spatially distributed, dynamical generated variable ρ , given by Equation 6.11.

6.4.3 Chemical module

At the core of form generation is the VRDD system, which are managed at the chemical module. The following section explains the manner in which the VRDD is spatially discretized and the way they run on the mesh with each node representing a cell of the tissue. Each cell has M morphogens of the *source-field* pair. Let ${}_i u_a, {}_i v_a$

denote the *source* and *field* concentrations of morphogen type i at node $\mathbf{a}$.

$$\mathbf{u}_{\mathbf{a}} = \begin{pmatrix} u_{\mathbf{a} \ 1} u_{\mathbf{a} \ 2} \dots \end{pmatrix}, \mathbf{v}_{\mathbf{a}} = \begin{pmatrix} v_{\mathbf{a} \ 1} v_{\mathbf{a} \ 2} \dots \end{pmatrix},$$

are the morphogen vectors of *source* and *field*. The Vector-Reaction-Diffusion-Drift system is denoted as:

$$\begin{aligned} \mathbf{u}_{\mathbf{a}} &\rightarrow (\mathbf{I} - \Lambda)\mathbf{u}_{\mathbf{a}} + \Lambda \langle \mathbf{u} \rangle_{N(\mathbf{a})} - \sum_{N(\mathbf{a})} \mathbf{u}_{\mathbf{b}} \odot (\tilde{\mathbf{v}}_{\mathbf{b}} - \tilde{\mathbf{v}}_{\mathbf{a}}) \\ \mathbf{v}_{\mathbf{a}} &\rightarrow (\mathbf{I} - \Gamma)\mathbf{v}_{\mathbf{a}} + \Gamma \langle \mathbf{v} \rangle_{N(\mathbf{a})} + R(\mathbf{u}_{\mathbf{a}} - \mathbf{v}_{\mathbf{a}}) - \beta \mathbf{v}_{\mathbf{a}} \end{aligned} \quad (6.10)$$

$$\begin{aligned} \tilde{\mathbf{v}}_{\mathbf{a}} &= \mathbf{W} \mathbf{v}_{\mathbf{a}} \\ \rho_{\mathbf{a}} &= \mathbf{g} \cdot \mathbf{u}_{\mathbf{a}} \end{aligned} \quad (6.11)$$

where $\odot$ denotes element-by-element multiplication of two vectors. Λ and Γ are diagonal matrices of diffusion constants of *source* and *field* morphogens. $R()$ is the ramp function and β is the decay rate of *field* morphogens. $\mathbf{W}$ and $\mathbf{g}$ are the *interaction matrix* and *growth kernel*, respectively. Growth factor ρ is the linear combination of local morphogen concentrations with weights set by the growth kernel $\mathbf{g}$. Biological implication would be that local morphogen concentration levels act as activators or inhibitors to expression of the growth factors, set by the weights in the kernel $\mathbf{g}$.

6.4.4 Regulatory module

Definition: *the genotype of an organism is the set of FSA states $\{S_i\}$, where each S_i consists of a fate function, two connected states, an interaction matrix and the growth kernel, $S_i = \{\tilde{\Phi}, S_A, S_B, \mathbf{W}, \mathbf{g}\}$.*

The finite state automata model of the gene regulatory network is run at the regulatory module based on the organism's genotype. This module feeds state parameters to chemical and physical modules and receives back morphogen concentrations from the chemical module as epigenetic information to be used for input to fate functions. In our current model, there is no feedback from physical or topology modules upwards to the regulatory module. Local information such as the stretch

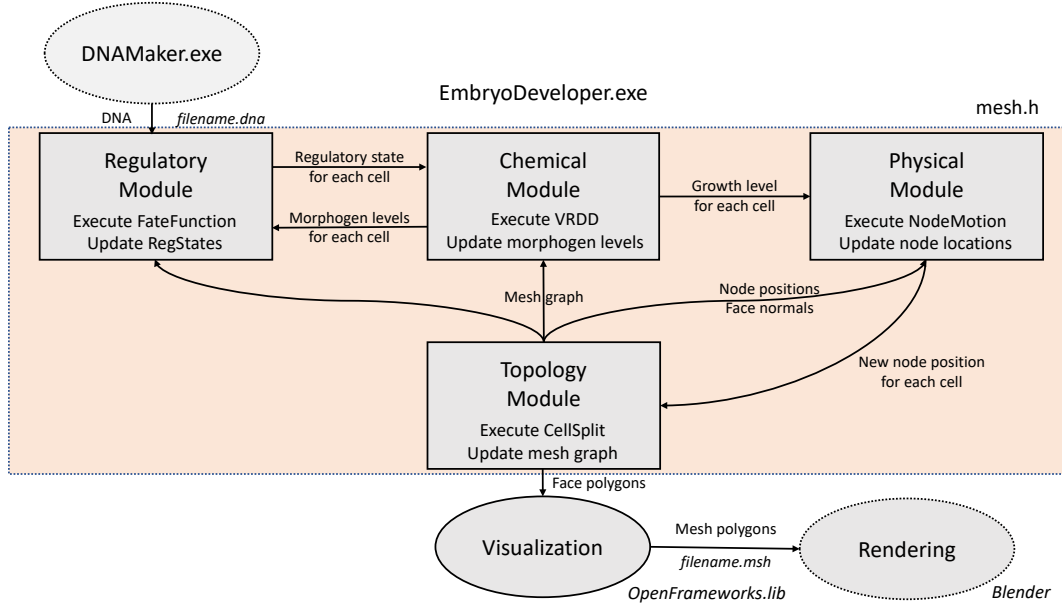


Figure 6.4: Simulation model architecture

Rectangular and elliptic units denote software modules or tools internal and external to **EmbryoDeveloper**, respectively.

and bend of tissue, as well as local convexity, may be used as mechanical cues to the regulatory model in future work.

6.5 System architecture

All simulations were done using the code written by us in C++ using the standard library only. Visual Studio 2015 was used for code development. Mesh Class that has mesh graph data structure as well as VRDD and FSA-related methods, is provided under `mesh.h` and `mesh.cpp`. Two main tools for genotype design and phenotype generation were also coded in C++ (**DNAMaker** and **EmbryoDeveloper**). Figure 6.4 shows the architecture of the core library developed under `mesh.cpp` used in **EmbryoDeveloper**. There are four modules:

Regulatory Module executes the `FateFunction` for each cell. `FateFunction` determines the next state for a cell given its current state and its morphogen levels which are provided from **Chemical Module**, based on the **DNA** of the specie provided from **DNAMaker** with `.dna` proprietary file format. The **Regulatory Module** provides updated cell state information to **Chemical Module**.

Chemical Module executes the Vector-Reaction-Diffusion-Drift morphogen dynamics. The parameters of VRDD, including Interaction Matrix and Growth Kernels are cell-state dependent. Each cell uses the relevant set of parameters given its current cell state as set by **Regulatory Module**. The **Chemical Module** provides updated morphogen levels for each cell back to **Regulatory Module**. This creates the *epigenetic loop*, where chemical information drives regulatory switching and regulatory states drive chemical reactions. The **Chemical Module** also provides growth level for each cell which is passed to **Physical Module**.

Physical Module executes Equation 6.9, determining the motion of each node, given its edge and face tensions and the growth perpendicular to membrane surface, based on growth level for each cell provided from **Chemical Module**. The updated node positions are then fed to **Topology Module**.

Topology Module maintains the underlying mesh graph. As the nodes move away from each other as dictated by the **Physical Module**, once any face size grows beyond a size limit, **Topology Module** executes CellSplit for that face. This is a core functionality that generates cellular growth, maintaining an average cell size. Upon execution of face split for a given cell, new nodes and edges are created, old ones are deleted and mesh graph connectivity is updated. The **Topology Module** supplies the underlying Mesh Graph structure to all three modules (Regulatory, Chemical, Physical). These modules use the graph structure to know which cell is neighboring which. The **Topology Module** also provides node position and face normal vectors data to **Physical Module** to be used in node motion calculations.

Synchronization Each module and within each module each cell is executed serially, but updated in parallel. This generates effective parallel execution on a serial hardware. We note that cell by cell execute and immediate update of cell states and variables, could generate superfluous patterning based on execution sequence. This can be generally avoided by random shuffling execution sequence or parallel updates. We used the latter approach.

Visualization of embryo development in real time is done by the openFrameworks library which is a wrapper on standard OpenGL graphics library. **Topology Module** provides face polygon data to openFrameworks for visualization.

Rendering of embryo for artistic illustrations is done by Blender, an open source animation environment. Mesh polygon data is provided as an output of **EmbryoDeveloper** in raw .msh format.

System resources Simulations were run on Intel Core i7 CPU of a DELL XPS laptop with 16 GB of RAM and a 64-bit Windows operating system. Real-time graphic output and high-resolution rendering were done using Nvidia GeForce GTX 1050 GPU. Simulations of embryonic development starting with c. 10.000 cells of blastula stage, ending up with c. 100.000 cells of final form, take c. 5 minutes.

6.6 Results

6.6.1 Fundamental forms

All forms that are topologically homeomorphic to a sphere can be transformed from it by a series of stretchings that are normal to surface at the organizing pole points. With no organizers, all points on the sphere are undifferentiated and equivalent. We identified three fundamental forms based on the symmetry groups on a sphere:

1. Single organizer: all points on the sphere can be grouped in terms of distance to the single organizing pole, resulting in equivalent sets of concentric rings. This is a single-axis body plan (anterior-posterior) with rotational symmetry.
2. Double organizer: two organizing poles (that are not trivially on the same line with the center) subdivide the sphere into two mirror halves. Any point on the half sphere can be uniquely identified in terms of its distance to the two organizing poles. This is a dual-axis body plan (anterior-posterior and dorso-ventral) with mirror symmetry.
3. Triple organizer: the three organizing poles (that are not trivially on the same line with the center) fully break the symmetry of the sphere, and all points

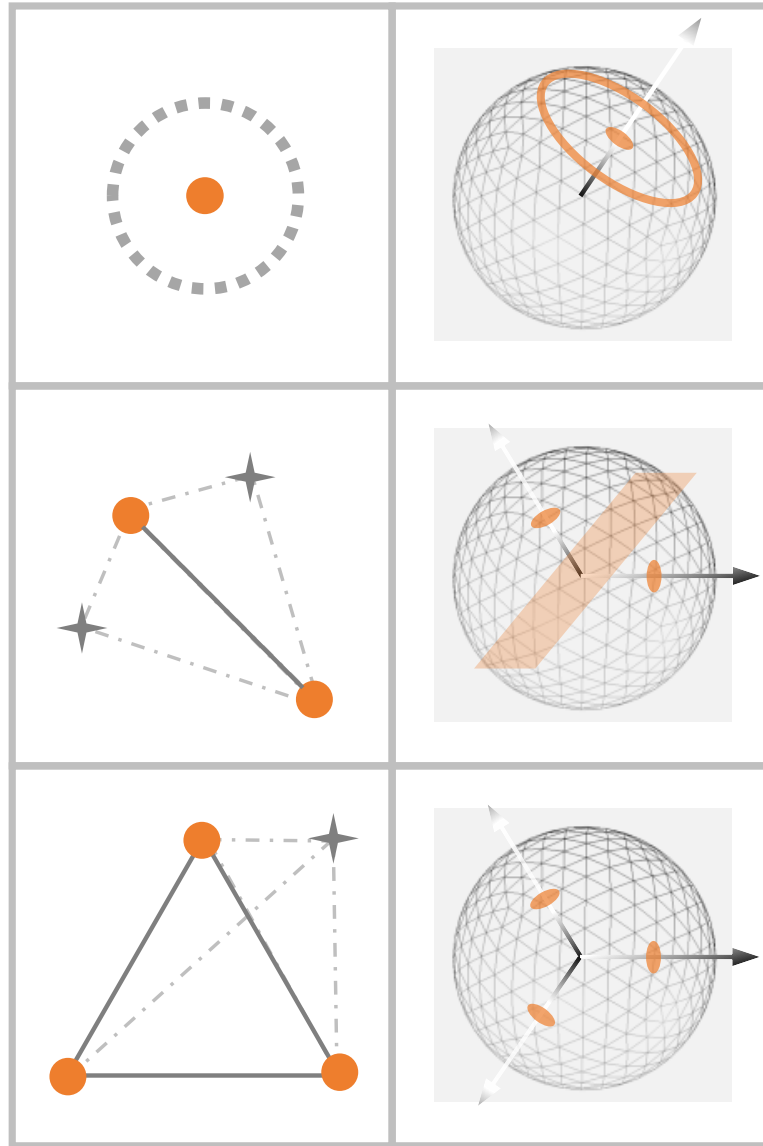


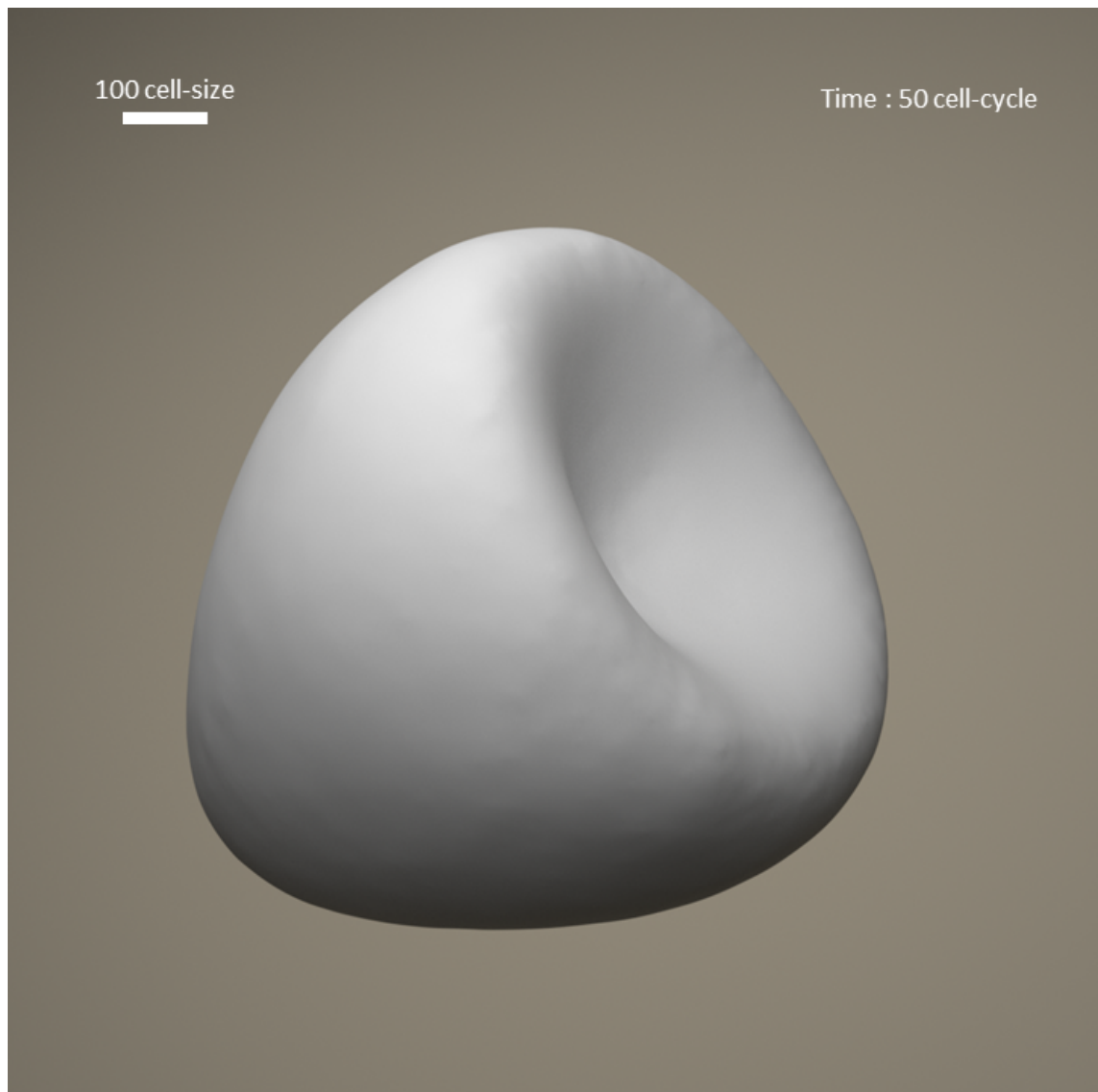
Figure 6.5: Symmetries in fundamental forms

Left column depicts the single, double and triple organizers on an infinite plane. Right column shows the corresponding organizers on a spheric surface.

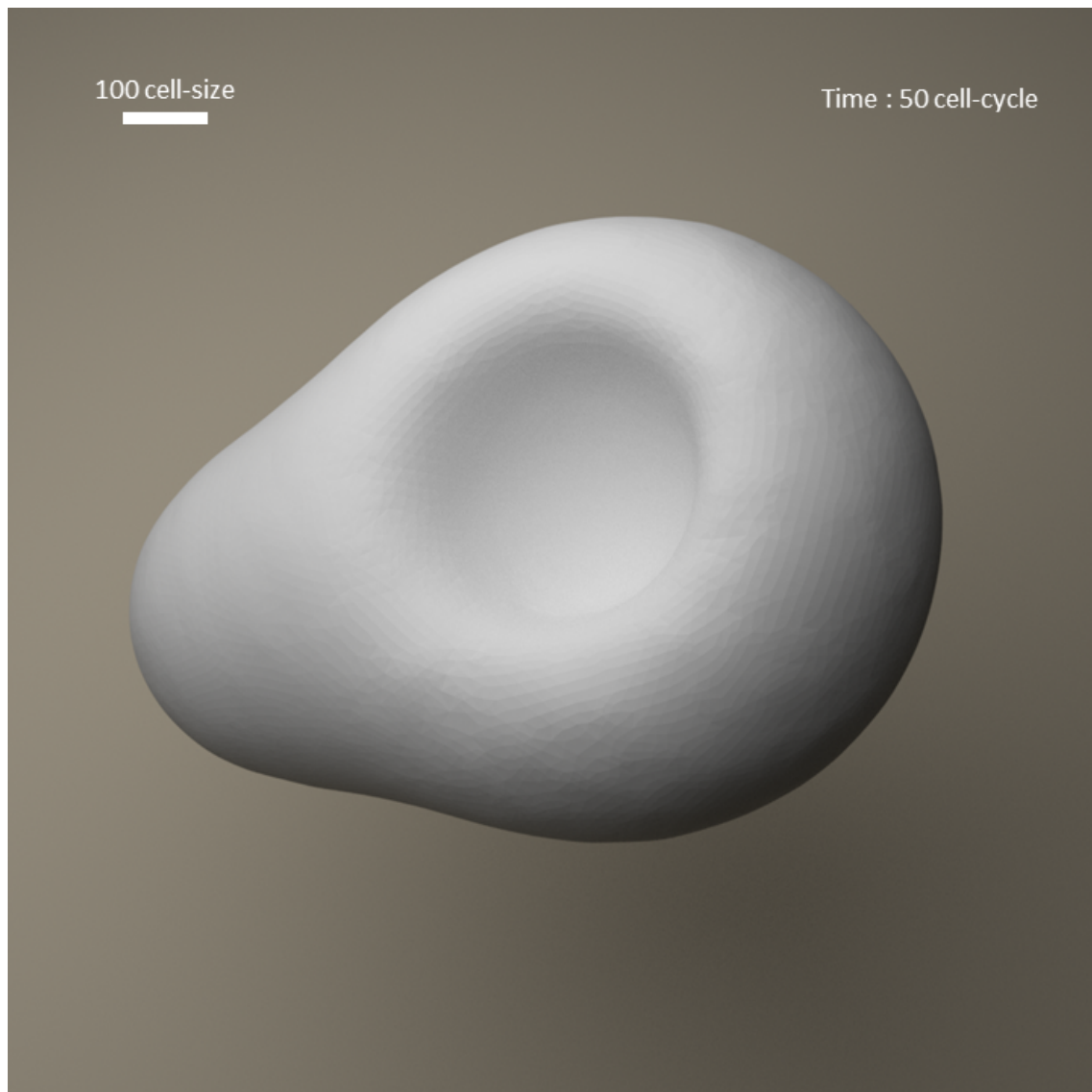
can be uniquely identified in terms of distance to such three poles. This is a triple-axis body plan. There are no known life forms on earth with a triple-axis body plan.

We generated the three fundamental forms using Vector-Reaction-Diffusion-Drift system. The results are shown in Figure 6.6. These forms are generated using a single-state finite state automaton and are the building blocks for more complicated forms.

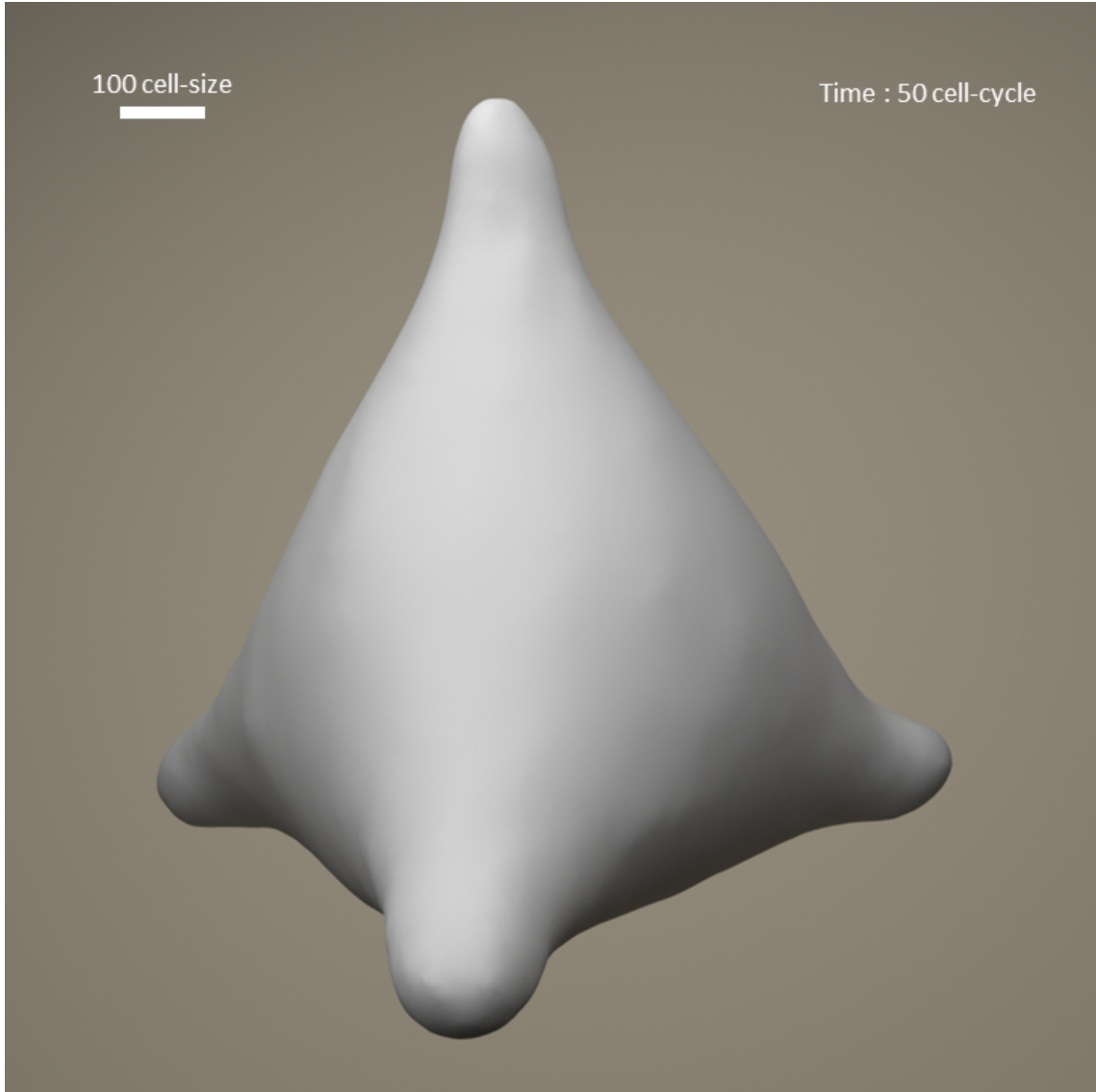
Single body axis is formed with two morphogens (R,G) with interaction matrix



Single Body Axis



Dual Body Axis



Triple Body Axis

Figure 6.6: Fundamental forms

Initial conditions in all simulations are spheric blastula with c.2000 cells. Gastrulation simulation time is 50 cell-cycle. Resulting embryonic form has c.5000 cells. Organizing polar regions are self-organized as emergent property of the specified VRDD. **(a)** Single body axis (anterior-posterior) formed using a single state FSA and 2-morphogen VRDD. **(b)** Dual body axis (anterior-posterior and dorso-ventralis) formed using a single state FSA and 4-morphogen VRDD. **(c)** Triple body axis is formed using a single state FSA and 4-morphogen VRDD. Interaction matrices and growth kernels are as follows, respectively:

$$\mathbf{W}_a = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, \quad \mathbf{W}_b = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -0.5 \\ -1 & -1 & -0.5 & 0 \end{pmatrix}, \quad \mathbf{W}_c = \begin{pmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 \\ -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{pmatrix},$$

$$\mathbf{g}_a = (+1 \quad -1), \quad \mathbf{g}_b = (+1 \quad 0 \quad -1 \quad 0), \quad \mathbf{g}_c = (+1 \quad +1 \quad +1 \quad +1).$$

set as $\mathbf{w}_a = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$. With $\mathbf{W}_{0,1} = -1$ and $\mathbf{W}_{1,0} = -1$, the two morphogens will be repelling each other, hence their concentrations would accumulate at polar opposites of the 2D spheric blastula surface forming a single body axis. With growth kernel set at $\mathbf{g}_a = \begin{pmatrix} +1 & -1 \end{pmatrix}$, head grows, tail stays tucks in, with the emergent form given in Figure 6.6, *Single Body Axis*.

Dual body axis is formed with four morphogens (R,G,B,Y) with interaction matrix set as $\mathbf{w}_b = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -0.5 \\ -1 & -1 & -0.5 & 0 \end{pmatrix}$. As in single body axis formation, $\mathbf{W}_{0,1} = -1$ and $\mathbf{W}_{1,0} = -1$, causes morphogens R and G to drift and accumulate at polar opposites, forming the anterior-posterior primary axis. With $\mathbf{W}_{2,0} = -1$ and $\mathbf{W}_{2,1} = -1$, morphogen B would be repelled by morphogens R and G, and similarly with $\mathbf{W}_{3,0} = -1$ and $\mathbf{W}_{3,1} = -1$, morphogen Y would also be repelled by morphogens R and G. This would result in morphogens B and Y migrating towards the equator of the anterior-posterior polar axis. Finally, with $\mathbf{W}_{2,3} = -0.5$ and $\mathbf{W}_{3,2} = -0.5$, B and Y would mutually repel each other, which would result them in aggregating at opposite sides on the equator, forming the dorso-ventralis body axis. With growth kernel set at $\mathbf{g}_b = \begin{pmatrix} +1 & 0 & -1 & 0 \end{pmatrix}$, head grows, tail stays flat, tummy tucks in, back stays flat, with the emergent form given in Figure 6.6, *Dual Body Axis*.

Triple body is formed with four morphogens (R,G,B,Y) with interaction matrix set as $\mathbf{w}_c = \begin{pmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 \\ -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{pmatrix}$, which simply means all repel each other. Therefore, each morphogen accumulates on a single peak that is maximally distant from the other three, i.e on the vertices of a tetrahedron inside the spheric blastula. With growth kernel set at $\mathbf{g}_c = \begin{pmatrix} +1 & +1 & +1 & +1 \end{pmatrix}$, four poles grow outwards, with the emergent form given in Figure 6.6, *Triple Body Axis*.

6.6.2 Making a hydra

Figure 6.7 illustrates the sequential form making with the finite state automata model. For illustration purposes, a 1D tissue with cells arranged on a line segment are depicted. Initially, all cells are at State 0. The chart illustrates formation of an

organism with mirror-symmetry, and another organism with no-symmetry axis in 1D, in columns left and right, respectively.

In the left column, single symmetry breaking VRDD is used with one morphogen type 'A', resulting in a single pole with mirror symmetry. Growth is driven by diffused field of such pole (head formation). Secondary poles of morphogen type 'B' are positioned at specified distance to primary pole 'A', based on the weights of interaction matrix $\mathbf{W}$. Fate-function that uses morphogen 'B' concentration level creates two symmetric bands of tissue switching to State 1. Lastly, the two bands of State 1 have their sub-spatial growth driven by the diffused field of morphogen 'B' poles, resulting in a final form of arm-head-arm with mirror symmetry.

On the right column, double symmetry breaking VRDD is used with two morphogens types 'A', 'B' having their respective source peaks as primary poles. Growth is driven by diffused field of pole 'A' (head formation), 'B' is silent in terms of growth generation at this stage. Secondary pole of morphogen 'B' is positioned at specified distances to two primary poles 'A' and 'B'. Fate-function that uses morphogen 'A' and 'B' concentration levels creates a band of tissue switching to State 1, asymmetric to head (beak formation). Second fate-function that uses morphogen 'B' concentration level creates a band of tissue switching to State 2 (tail formation). Lastly, the cells in State 1 have their sub-spatial growth driven by the diffused field of morphogen 'B' poles, whereas the band of cells in State 2 execute a system with periodic peak formation, resulting in a final form of beak-head-tail with no mirror symmetry.

Hydra is an interesting model organism for morphogenesis studies because its body plan is simple with single axis from head (hypostome and tentacles) to glandular foot, and it continuously renews its body cells from stem cells migrating based on ever-available morphogen gradients for positional information. Hydra is therefore "*an immortal and perpetual embryo*". Hydra can reproduce sexually with few eggs produced in weeks, or if well fed, it can reproduce by budding every 1.5-2 days, giving cloned offspring. Hydra has been extensively studied with transplantation experiments due its regeneration capability and continual axial morphogen gradients. In our Hydra model, we modeled the embryonic development from single egg; not the budding process for reproduction.

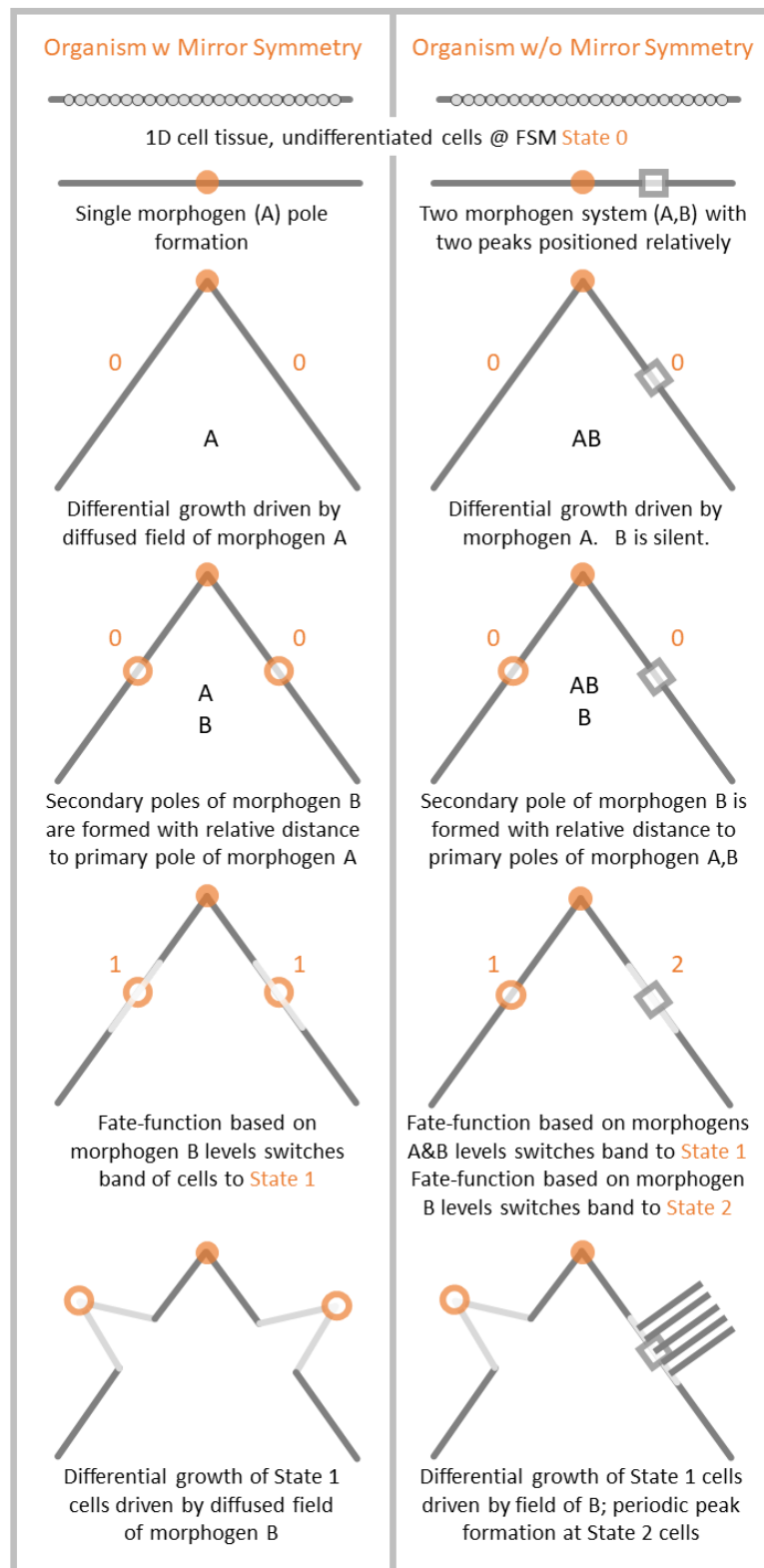


Figure 6.7: Sequential form making with FSA

Left and right columns depict formation of organisms with and without mirror symmetry, respectively on a 2D space, starting from initial condition of undifferentiated linearly placed tissue cells. Left column results in arm-head-arm formation whereas right one generates asymmetric beak-head-tail.

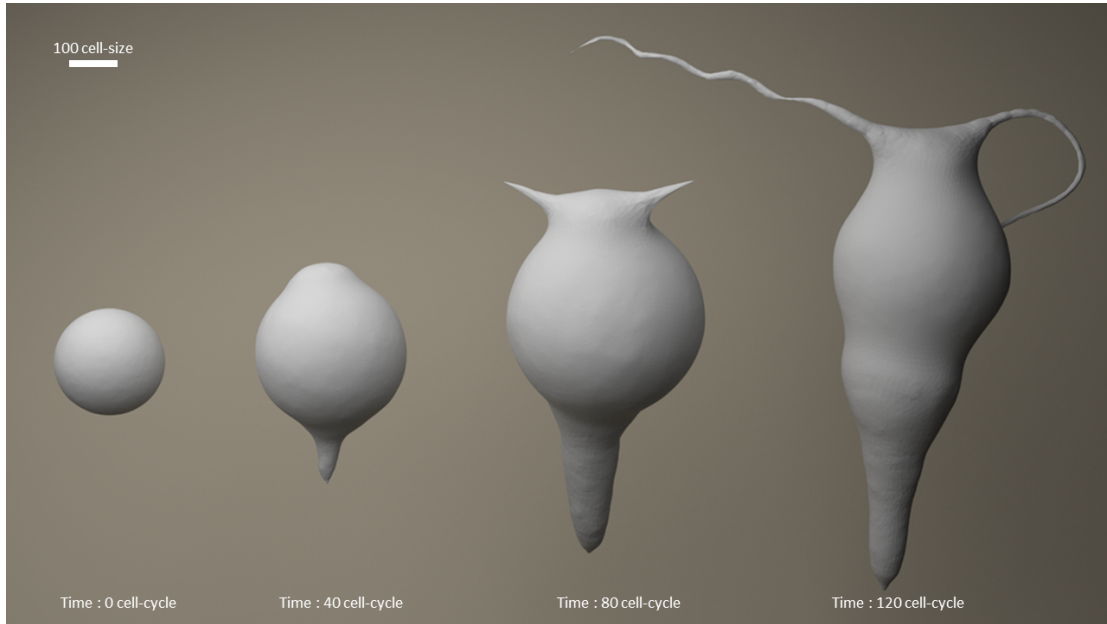


Figure 6.8: Hydra embryogenesis stages

7-State FSA with 4 morphogens are used to generate the Hydra. Initial condition is a blastula with c.2000 cells. After 40 cell-cycle simulation time anterior-posterior axis is formed using 2-morphogen VRDD (R,G), followed by head and tail region cell state switching. A secondary pole formation at 'head' sub-spatial domain is achieved by a separate 2-morphogen (B,Y) VRDD running on cells with 'head polarize' state. Tentacle growth results from narrow band of cells switching to 'tentacle growth' state, triggered by fate-function of B,Y morphogen concentration that defined the tentacle poles. Final form of the embryo is achieved in 120 cell-cycles of simulation time with c 8000 resultant cells on the topological surface.

Using the sequential form-making technique, we generated a toy organism in the shape of a hydra, albeit with two tentacles for demonstration purposes. We implemented a seven-state finite state automaton that consists of 5 developmental states, plus the totipotent ground state and the final stasis state. The descriptions of states, including VRDD parameters, growth kernel, and most importantly fate function with state transition connections, are provided in Supplementary Materials. The simulation results of the developmental stages of our hydra are shown in Figure 6.8.

6.6.3 Mutating a hydra

In-vivo developmental mutants of Hydra have been produced by selective breeding [125] that had mini strains, maxi strains, multi-headed strains, nematocyst-deficient strains, regeneration deficient strains and male sterile strains. These strains were transmitted to progeny by budding and cloned for developmental studies. It is chal-

lenging to generate transgenic mutant Hydra. [126] and [127] reported a successful method for generating stable transgenic Hydra lines which have since then been used for signaling pathway studies. These studies showed that morphogenesis in Hydra involves paracrine and juxtacrine signaling of Wnt and β -catenin [128], as well Notch-Delta activity [129] for creation of tissue boundaries.

For *in-silico* mutation experiments, we varied some of the structural parameters of our synthetic genome which resulted in fat-hydra and long-tentacled hydra. These mutants are topologically homologous to the wild-type as shown in Figure 6.9 (Mutant A and C).

More interestingly our jump-mutation type alterations of the genotype resulted in double head formation. In this experiment cell-fate functions that differentiate cells to tail-type were rewired to switch to head-type in the synthetic genome which resulted in stably generated mutants with two heads in opposite ends of the primary body axis as shown in Figure 6.9 (Mutant B). These *in-silico* mutants are similar to two-headed mutants that were bred by [125] and resemble homeotic mutants in *Drosophila* with legs emanating from head [130]. Multi-headedness is also observed in transgenic mutants that were produced by [126], however these are not based on generation of two heads in axial opposites but rather generation of multi heads throughout the body. The genotype to phenotype mapping is not smooth. A small alteration of FSA connectivity resulted in radical change in phenotype. Random walk on genotype space with small variations can result in discontinuity on phenotype space. This may be an explanation of observed discontinuity in fossil records.

6.6.4 Other organism forms

We experimented with generating various other organism forms. To streamline the design of genotypes for such forms, we developed a drag-and-drop design tool, **DNAMaker**. FSA state table generated by **DNAMaker** was then injected to the embryogenesis simulation environment **EmbryoDeveloper**. Development results for jellyfish and urchin variants are provided in Figures 6.10, 6.11 and 6.12. Other simulation results with parameter sets and FSA are provided in Appendix B.

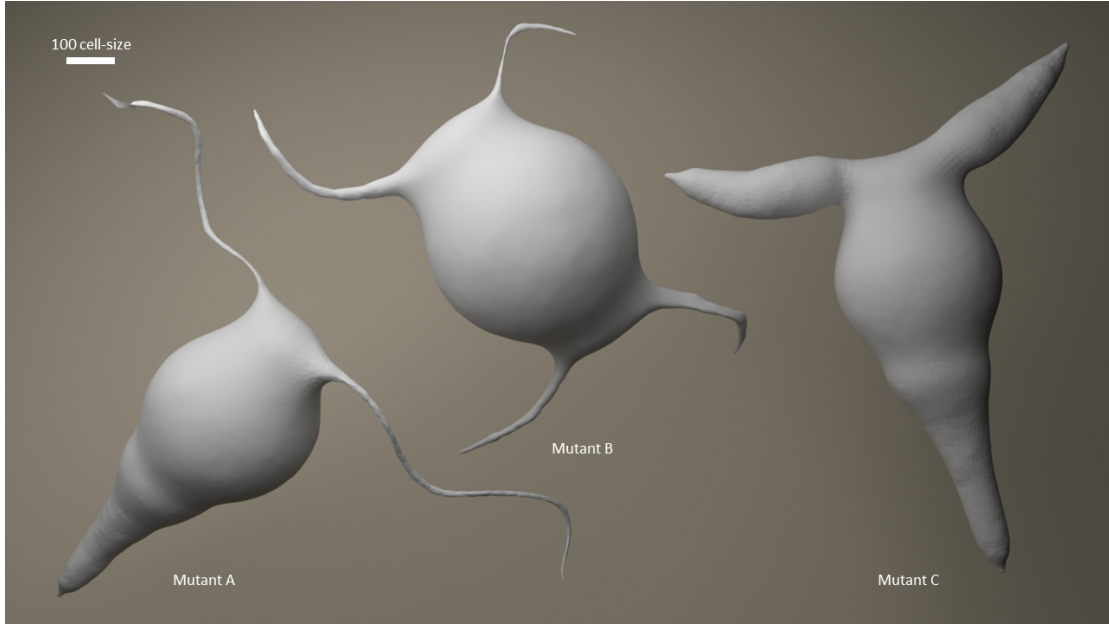


Figure 6.9: Hydra mutants

Mutants A and C are generated by varying growth kernel values as a type of continuous mutation. Mutant B is generated as a jump mutation where state transition of FSA is rewired leading to double head formation.

6.7 Discussion

In this chapter, we introduced Vector-Reaction-Diffusion-Drift system as a novel concept which can generate body plans of fundamental forms by self-organization and elaborated on a finite state automata model of the gene regulatory network. The result of VRDD combined with FSA model is spatial cell differentiation during embryogenesis that can be used for hierarchical modeling of complicated forms. Our current implementations generate forms of lower animals such as hydra or jellyfish. However, VRDD model is capable of generating dual-axis body plans (anterior-posterior and dorso-ventral) of vertebrates by self-organization. We are cognizant that the physical module of our simulation model, which is currently based on elastic-membrane, needs to be expanded to incorporate aggregation dynamics for hard tissue formation.

The Four Color Theorem in mathematics says that any partitioning of a topologically spheric surface can be colored with four colors without two neighboring patches having the same color [131]. We conjecture that this would lead to a lower bound in the number of morphogen types that can be used in generating any arbitrary form using an FSA model with the morphogen re-use in the spatial sub-domains.

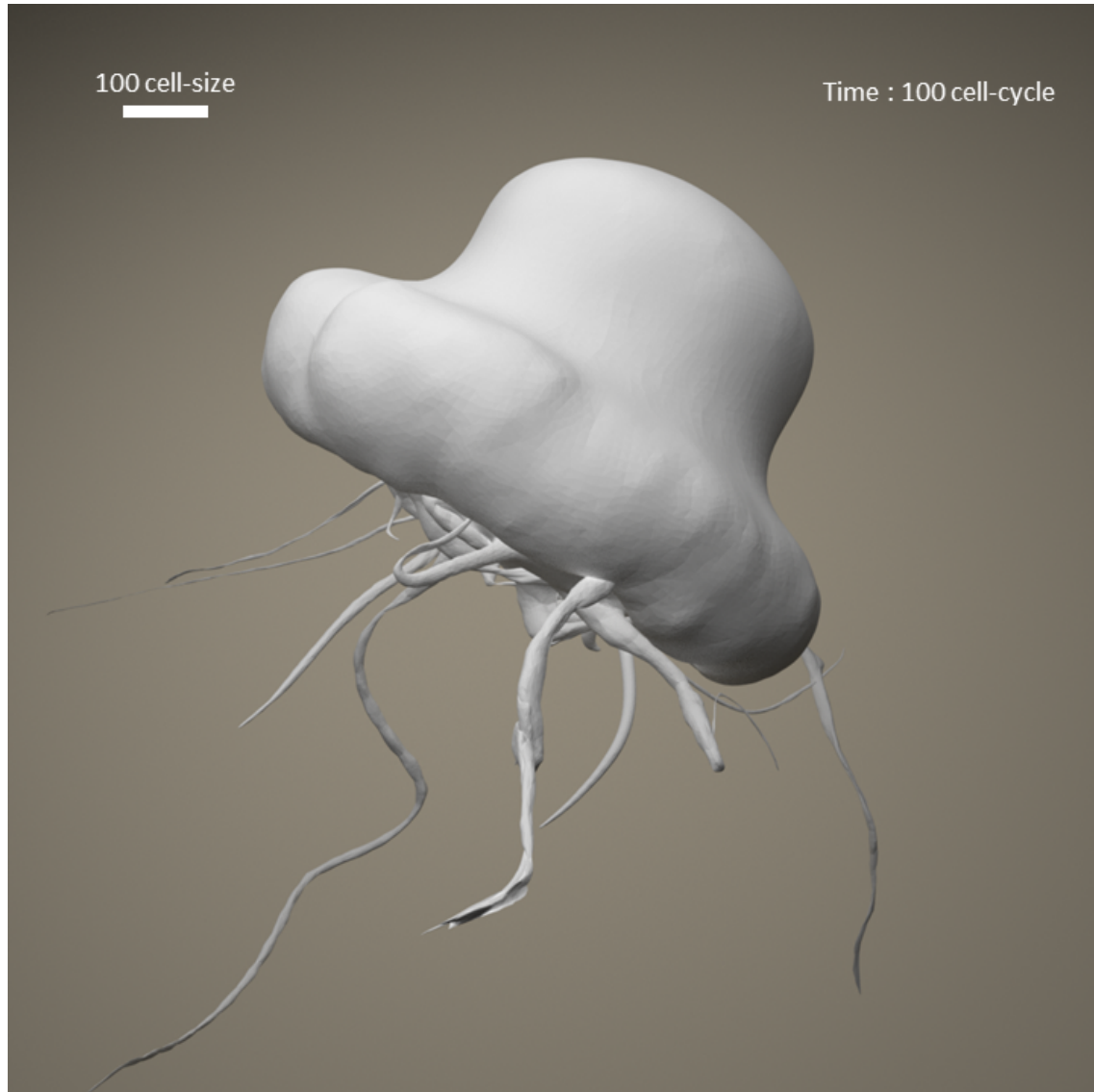


Figure 6.10: Jellyfish

Jellyfish was generated using a 10-state FSA; primary single axis body plan was generated in State 1 using 2-morphogen (R,G) VRDD followed by head and tail state changes. Head part was further differentiated to belly-skirt with another state change based on fate-function driven by morphogen R and G levels. Narrow tentacle growth was the result of further state change of a narrow band around each tentacle point which were then grown with a high valued growth kernel. Parameters and finite state automata of this specie is given in detail under Appendix B, Simulations, as specie Jelly23.



Figure 6.11: Sea anemone

Parameters and finite state automata of this specie is given in detail under Appendix B, Simulations, as specie Urchin22.

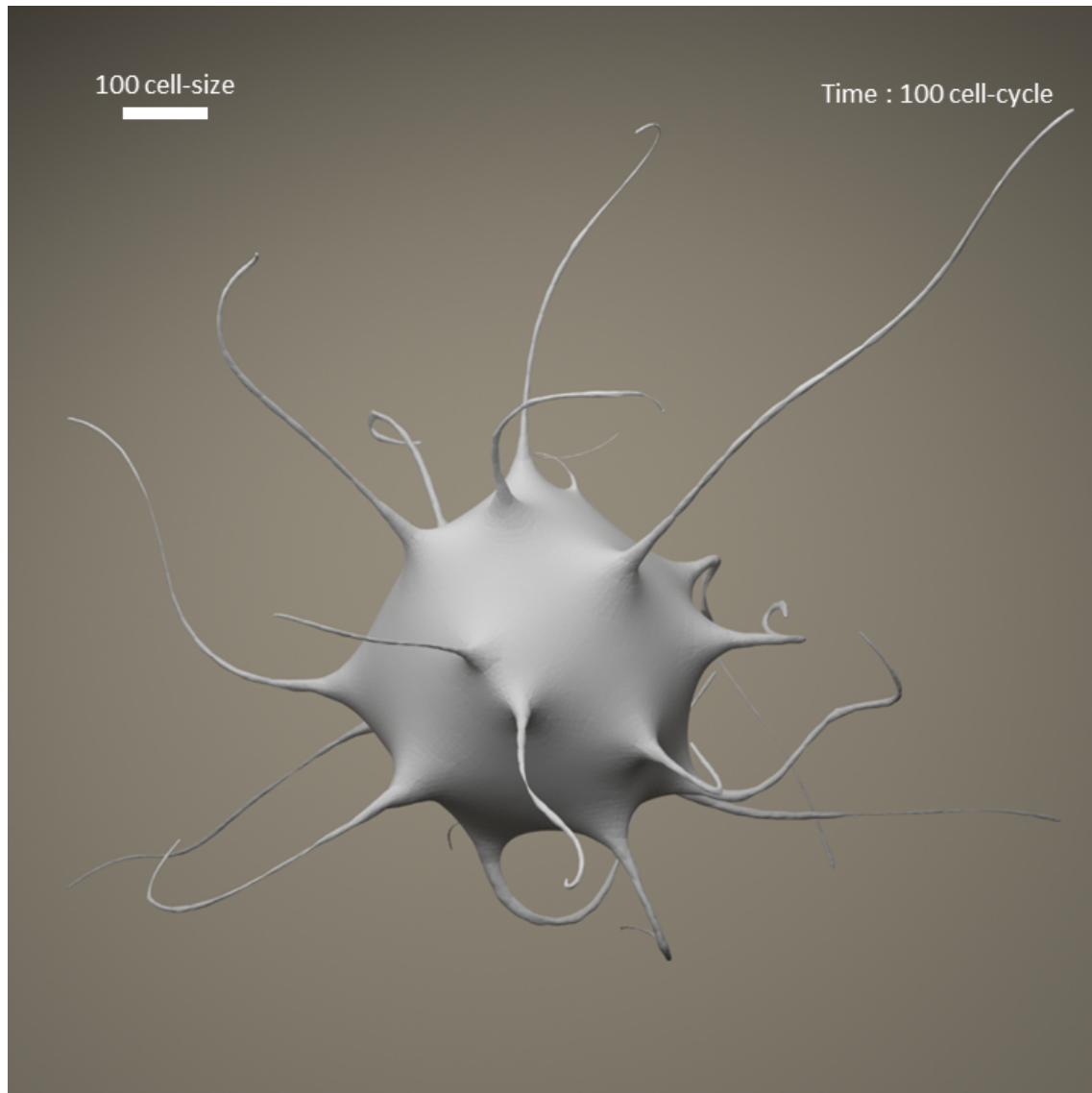


Figure 6.12: Tendrils

Parameters and finite state automata of this specie is given in detail under Appendix B, Simulations, as specie Spikeynew3.

Referring back to Figure 6.7, we have outlined the conceptual framework for defining spatial sub-domains in 1D tissue with two morphogens using VRDD dynamics. When the concept is expanded to 2D tissue, the topological dimensionality of the problem increases by two-fold. Hence, we believe that four morphogens could be sufficient to specify any spatial sub-domain in 2D. A key challenge to our conjecture would be that organisms are not 2D, but 3D. We would claim that embryonic growth is fundamentally a 2D phenomenon where tissue is layered as 2D shells. We acknowledge that linking Four Color Theorem to 2D tissue growth using four fundamental morphogens is a highly speculative proposition that will require significant further theoretical work.

We provide mechanisms of spatially distributed non-linear dynamical systems for embryogenesis, which have so far been restricted to Turing-like pattern formation systems [1] or the positional information theory of development by Wolpert [132]. Our concepts are capable of generating life-like forms replicably.

Chapter 7

Conclusion

7.1 Epilogue

In this Thesis, we investigated mechanisms for cellular coherence where local actioning cells ensembled over a tissue replicably generate self-organization. Nature is conspicuously populated with systems demonstrating self-organization, ranging from physical to chemical, biological and social. Patterning is the basic mode of cellular coherence observable in wide-range of phenomena such as Bénard thermal convection systems, Belousov-Zhabotinsky dissipative chemical systems, and Alan Turing's reaction-diffusion systems showing animal skin patterns. The mathematical models of patterning have been well studied. Form generation through cellular coherence is different than patterning. Whilst local symmetries are broken in a pattern, a form requires orchestrated tissue-wide symmetry breakings. Dynamics that can generate form have not been fully investigated. We believe we unlocked some possible such dynamics.

Our work was split into two building block: *Cellular dynamics* (Chapter 4) and *gene regulation* (Chapter 5). These building blocks were joined together under *morphogenesis* (Chapter 6), contributing a mathematical basis to biological theory of development.

In Chapter 4, we started with developing methods for cellularizing a general manifold and provided a framework for transferring vector calculus on the manifold to that on a discrete mesh. This tool set allowed us to analyze pattern formation in a cellular system. The investigation on Turing instability led us to conclude that

patterning was due to an interplay between attraction and repulsion in the state space. It is interesting to note that recent theoretical work on cosmology suggest that the formation of cosmic structures such as galaxies can be due to the interplay of gravitational attraction and repulsion [133].

The form generation is an important, yet understudied mode of cellular coherence. We proposed a simple novel dynamics, called *singular attractor*, that generates a singular pole on the tissue and analytically derived an upper bound on time to singularity event. This dynamics breaks the global symmetry of the tissue which is the first step in form generation. We introduced segmenting and body plan formation using a dipolar attractor as an extension of the singular attractor model. We have shown that a model embryo can replicably self-organize into precise body plans such as flies, insects and anachrids.

We extended the dynamics into generalized 2D manifolds and shown patterning and pole formation thereof, and also demonstrated that the dipolar attractor dynamics can be discretized in state space, leading to cellular automata set up. We proposed a simple automata algorithm that can generate self-organization of a single pole which is equivalent to a leader election algorithm of distributed computing.

In Chapter 5, we embarked on a totally different workstream, the gene regulatory networks, which are known to provide mechanisms for cell differentiation in biology. Building up on the Boolean network model of gene regulation that was first introduced by Stuart Kauffman, we developed the computron as a generalized computational model that is equivalent to a finite state automata but where the flow of information and processing of information are segregated. The genetic regulatory network found home in the computron where each gene was assigned to a binary unit. The reaction dependencies represented the information flow whereas reaction functions became the nodal information processing. We then embarked on the fundamental question of what it means to have a complex computation. Reflecting on work done on entropy in dynamical systems and complexity classification of cellular automata, we developed a technical framework on basin partitioning complexity performed by computrons. This line of work required us to develop a novel technique of embedding the gene regulatory network on a metric space, called the diversity space, and then calculating the entanglement of basin partitionings as a measure

of computational complexity. As further extension of the framework, we offered a probabilistic version of the automata using fuzzy-basins and recurrent classes which enabled us to quantify chaoticity and complexity of a genetic regulatory network, as well as its susceptibility to noise.

Combining cellular dynamics and gene regulation into morphogenesis in Chapter 6, we proposed the development of form in an embryo as a series of topological and informational symmetry breakings. We offered a vector form of dipolar attractor dynamics, named Vector-Reaction-Diffusion-Drift system, and provided a cell fate dynamics where cell type differentiation is performed on genetic regulatory network of the cell with cell types corresponding to attractors of the computron. Using a simulation environment that has physical, chemical, and regulatory modules, it was possible to generate life-like embryonic forms. In toy hydra model, it was shown that not only our morphogenesis model is capable of self-organizing a blastula into a precise form replicably, but that slight genetic mutations on the model can result in jumps in the phenotype.

Given that a detailed body form with mirror symmetry can possibly be generated by the two axis system with four morphogen types (double source type for each of the two dipolar attractor), we then conjectured whether *four* is a special number. Any partitioning of a spheroid manifold can be covered with four color patches due to recently proven Four Color Theorem and DNA is made up of four letters paired up in two.

In this Thesis, we provided several contributions to mathematical theories for biological development and gene regulation including (i) singular attractor dynamics that generates global symmetry breaking across the tissue, (ii) dipolar dynamics as an extension of singular dynamics that self-organizes an axis on spheric blastula and provides positional information across the tissue, (iii) computron as a computational machine equivalent to a finite state automata but with the novelty that flow of information and processing of information are segregated, (iv) diversity space as a novel metric space that captures the similarity distance of configuration states of graph automata, which allows us to bridge the theory of computation into theory of dynamical systems, (v) a new complexity concept for FSAs based on entanglement of basins of attractions generated by the equivalent computrons, (vi) extension of the

computron model to probabilistic automata with original concepts of fuzzy-basins and ergodic decomposition, (vii) Vector-Reaction-Diffusion-Drift as an extension of singular attractor dynamics providing orthogonal axis formation, and (viii) combining VRDD with gene regulation, a model for morphogenesis shown to be capable of generating organism forms replicably with testable propositions for biology.

7.2 Applications

Our current way of constructing (from physical structures, such as buildings, to functional objects, such as a vacuum cleaner, to software such as an operating system) is fundamentally different than that of nature. Ours is purpose driven, externally architected, precisely partitioned and rigorously assembled. Nature is also purposeful, but survival driven, intrinsically architected, distributed and self-organized. In the near future, we would expect to see a wave of nature-mimetic product design that self-assembles. This is particularly possible in nano-robotics where a swarm of nano-robots can be self-organizing for purposeful co-existence, generating form and function as an ensemble. Dynamical systems and models proposed in this Thesis can find applications in morphogenetic engineering and self-organizing architectures.

Biology is primarily an empirical science with utmost practical applications and theory of it typically comes afterwards. Development of an embryo from homogeneous ensemble of cells at blastula stage to differentiated cells organized in organs with definite form and function is a highly intriguing and important area. Vast amount of progress is made in understanding the progression during early embryonic development. Today cell level measurements of an embryo is possible, mapping the chemistry of development in immense detail. However, mathematical theories of such self-organized development is still lagging. Our proposed models to self-organize cells of a developing embryo with testable propositions for biology could help unlock theories for developmental mechanisms of embryogenesis.

7.3 Current limitations and future work

Our work is a *designed embryogeny* that maps information encoded in the artificial DNA to an embryonic form without external guidance, but intrinsically through

self-organization guided by the finite state automaton of gene regulatory network. We design the DNA of our species to get a desired form in an artisan way. Research can be undertaken where DNA is evolved with certain form novelty selection criteria on the resultant phenotypes.

We did not investigate regenerative implications of our embryogeny. Regeneration is an important morphogenetic process seen in tissue repair as well as, on the more extreme, as budding processes in certain species such as hydra. Research can be done to understand regenerative properties of our proposed models.

We did not analyze the impact of cell level defects on morphogenesis in our models. Tumour growth is a very important abnormal morphogenesis that results from cell level defects. A very interesting line of research can be performed to understand under what conditions defected finite state automata of a cell causes it to be a renegade cell, causing cancerous growth.

Case studies in our computron model for GRNs is rather limited at only 5 genes. Larger GRNs should be investigated in terms of complexity and robustness to noise using better computational platforms with much higher processing power. Also, gene level defects to GRN performance and characterization can be investigated in these case studies. Finally, GRNs can themselves be evolved by adding and amending nodes to the computron, observing the bifurcation in GRNs dynamical characteristics due to these incremental modifications.

Our simulation model is a 2D membrane defining the boundary surface of an organism's form. Full 3D model with cell mobility and adhesion can be developed, still using our VRDD dynamics with finite state automaton based cell regulation. Such 3D model can be then used in real-life nano-robotic implementations.

Finally, the biological propositions that come out of our theory, which we provided in Chapter 6, can be tested *in vivo*. In this thesis, we came up with possible mechanisms for developing fully fledged organism forms replicably through self-organization. Whether Nature does it the same way remains to be seen.

Appendix A

Codework

EmbryoDeveloper

Code that develops a mesh model of embryonic form using the Morphogenesis Model that was described in Chapter 6. Code, written in C++ using standard library is c.5000 lines long with with 5 min execution time on Intel Core i7 CPU for developing an embryonic form.

Object	Description
<code>mesh</code>	Class that defines the embryo.
<code>node, edge, face</code>	Struct objects that have pointers to each other that define mesh graph connectivity. A <code>mesh</code> consists of vectors of <code>node</code> , <code>edge</code> , <code>face</code> objects that defines its topology.
<code>regState</code>	Struct that defines the parameters of a regulatory state. A <code>mesh</code> has a vector of <code>regStates</code> that defines the physical and chemical module parameters as well as regulatory switching functions, which together comprises the DNA of the embryo. The <code>regState</code> vector is saved in proprietary <code>.dna</code> format.

Continued on next page

Table A.1 – *Continued from previous page*

Object	Description
addNode addFace deleteFace splitEdge splitFace	Functions that allow for topological local modifications to the mesh graph. These are core functions used in automatic cell splitting as the embryo grows.
faceNorm nodeNorm nodeConv edgeLength	Functions that provide local 3D geometric properties such as normal vectors, convexity, length.
runGeometry	Function that runs the topology module of the mesh. For all nodes and faces it calculates normals and convexities. For each edge it performs cell split if length is grown above preset size.
runChemistry	Function that runs chemical module of the mesh, including reaction, diffusion, and drift.
reactChemicalStates	Function that provides <i>field</i> formation, <i>field gradient</i> determination, <i>effective gradient</i> calculation based on <i>Interaction Matrix</i> of VRDD system at each node defined by the regulatory state of each node, and <i>Source</i> drift on effective gradient.

Continued on next page

Table A.1 – *Continued from previous page*

Object	Description
<code>diffuseChemicalStates</code>	Function that provides cellular diffusion among nodes of <i>Source</i> , <i>Field</i> and <i>Hormone</i> chemicals. Hormone chemical determines the growth of the tissue at a node location as defined in the biology module
<code>runPhysics</code>	Function that runs physical module of the mesh. Each node is moved locally to minimize its total energy, defined as sum of edge spring and face spring energies as well as growth hormone that drives growth normal to surface. It uses gradient descent logic to determine where each node moves incrementally in 3D space.
<code>runBiology</code>	Function that determines the growth hormone level at each node based on the local source chemicals and the growth kernel that is based on the current regulatory state.
<code>updateRegulatoryStates</code>	Function that determines the regulatory state switching of each node, based on the chemical levels at that node and its local timer. The cell state switching is the core of regulatory dynamics. It is determined by the <code>FATE_OF</code> function at each cell.
<code>FATE_OF</code>	Function that determines what regulatory state a cell switches to based on the <code>matrixPHI</code> vectors that are prescribed in the <code>regState</code> descriptors. This is the core function for regulatory switching of cell states based on the Finite State Automata logic and local chemical levels of each node.

Continued on next page

Table A.1 – *Continued from previous page*

Object	Description
meshGen	namespace that has several functions that allow mesh object manipulation as listed below:
surface	This is a core routine for cellularizing an analytically defined continuous 2D manifold into a mesh graph. It performs a stitching algorithm starting from a seed triangle and using a desired edge size.
clear	clears a mesh into null graph
move, scale, rotate	moves, scales and rotates a given mesh in 3D
smoothMesh	smoothes a given mesh in 3D splitting faces and reducing node convexity.
loadMesh, saveMesh	loads and saves a mesh in proprietary .msh format.
exportToBlender	exports mesh graph data in .raw format which is a delineated faces triangle list, used in Blender which is an opensource rendering software for 3D animation used in generating Artwork .
loadDNA	loads DNA of the embryo on to the mesh as specified in regState vectors, saved in proprietary .dna files, and sets all cells totipotent 0 state to initialize development. .dna files are generated using DNAMaker code described below.

DNAMaker

Code that provides a GUI based environment with drag and drop tools to generate the finite state automata design of regulatory switching logic of a given virtual specie's DNA, including all parameters and thresholds for the switching fate functions. The output is saved in the proprietary **.dna** format, which is then injected into a mesh blastula to develop the embryo with **EmbryoDeveloper** code described above. Code is written in C++ using standard library and openFrameworks library for GUI functionalities.

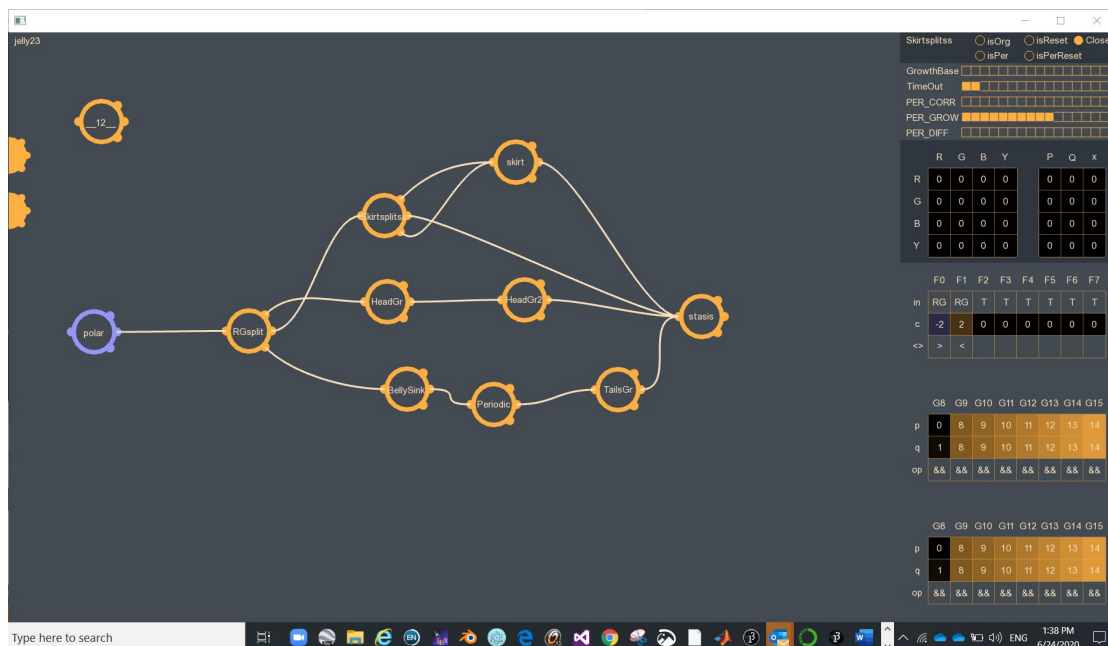


Figure A.1: Screenshot of DNAMaker showing DNA model of Jellyfish

The GUI environment provides for user to drag and drop new states of the finite state automata shown as circles with user given names. Each state can switch to three states based on parameter thresholds and input sets. The connectivity from one state to next is done by drag and dropping connection lines between states. The Vector-Reaction-Diffusion-Drift model's parameters, including the Interaction Matrix, is defined, for each state, on the left parameter design panel. Panel shows the parameter set for whichever state is clicked on. Parameters are set and changed by clicking on the relevant parameter boxes. Fate functions for switching out of each state is given on the three sub panels on the bottom left of the screen. Using click on GUI functionality, any p, q logic formula and threshold functions can be defined by the user. (l)oad and (s)ave keys allow for loading and saving of **.dna** files for DNA.

Appendix B

Simulations

Following is selected morphogenesis simulations and generating DNA (including finite state diagram and full parameter set) for some selected species:

Axis1only6

Urchin22

Spikeynew3

Jelly23

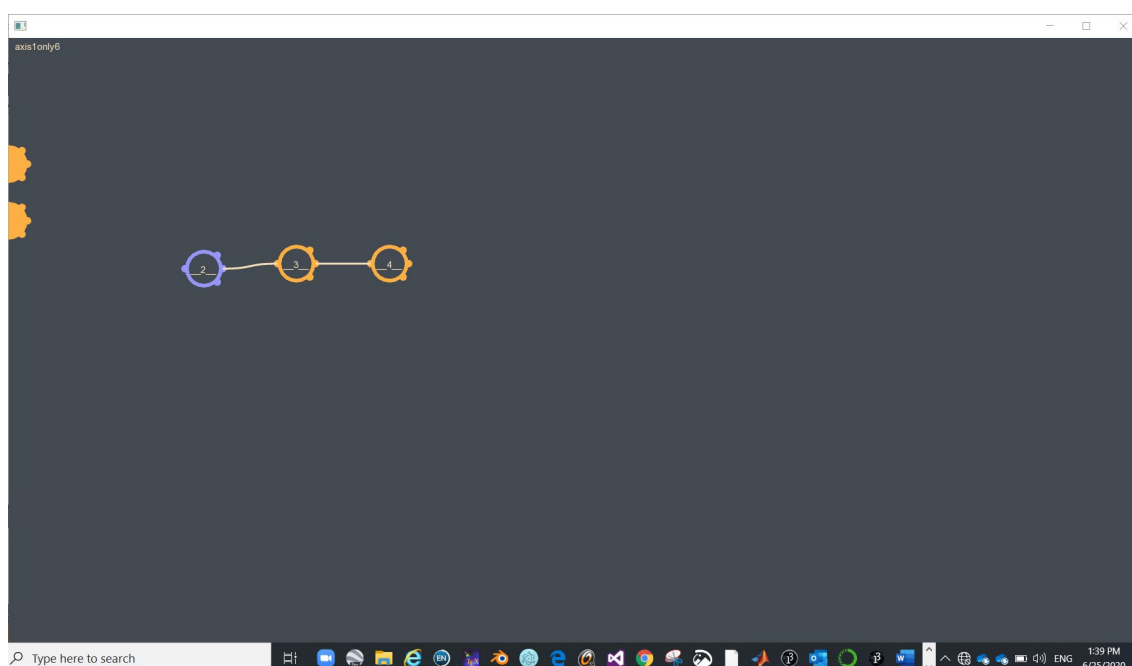
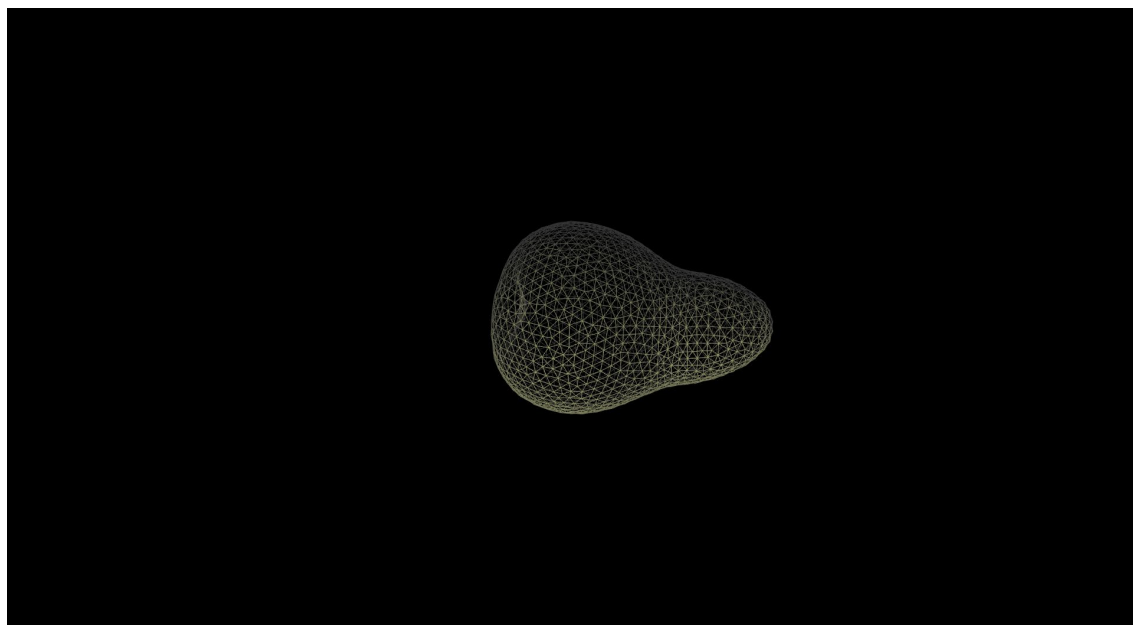
Specie: Axis1only6

Figure B.1: Axis1only6 morphogenesis simulation and DNA states

EOF

Figure B.2: Axis1only6.dna parameters

Specie: Urchin22

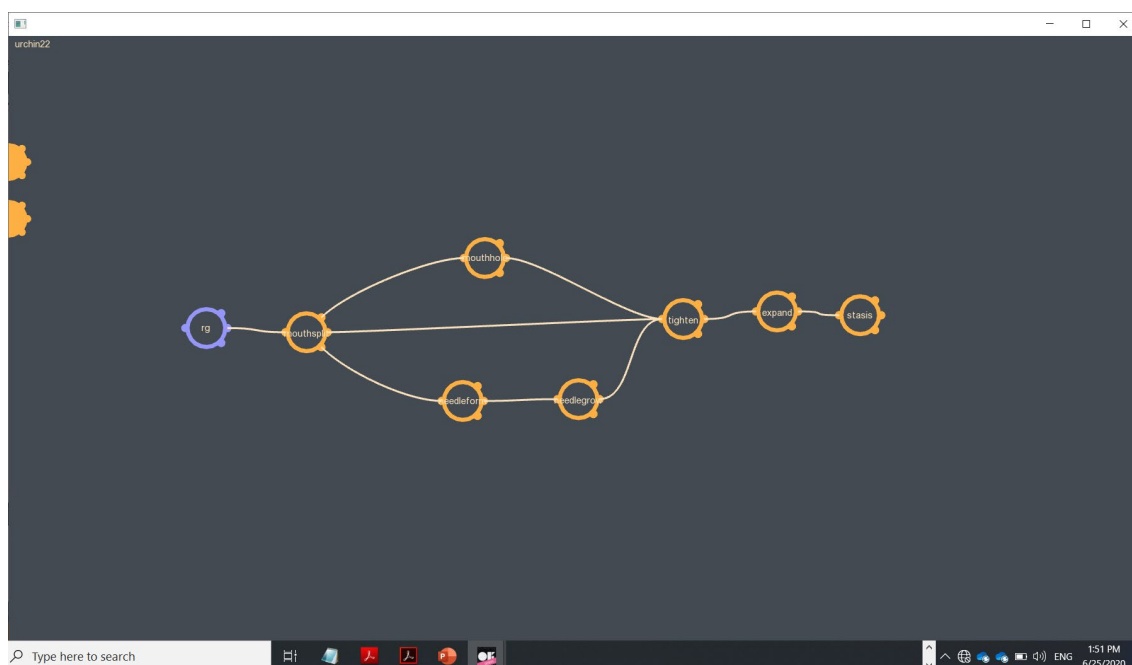
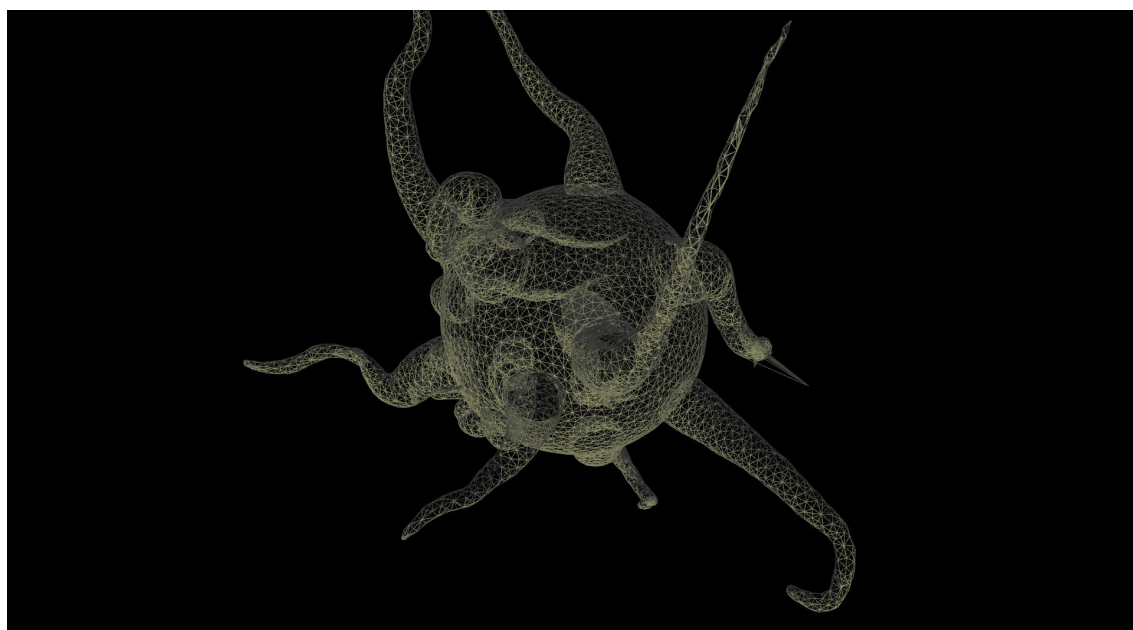


Figure B.3: Urchin22 morphogenesis simulation and DNA states

[illegible]

Figure B.4: Urchin22.dna parameters

Specie: Spikenew3

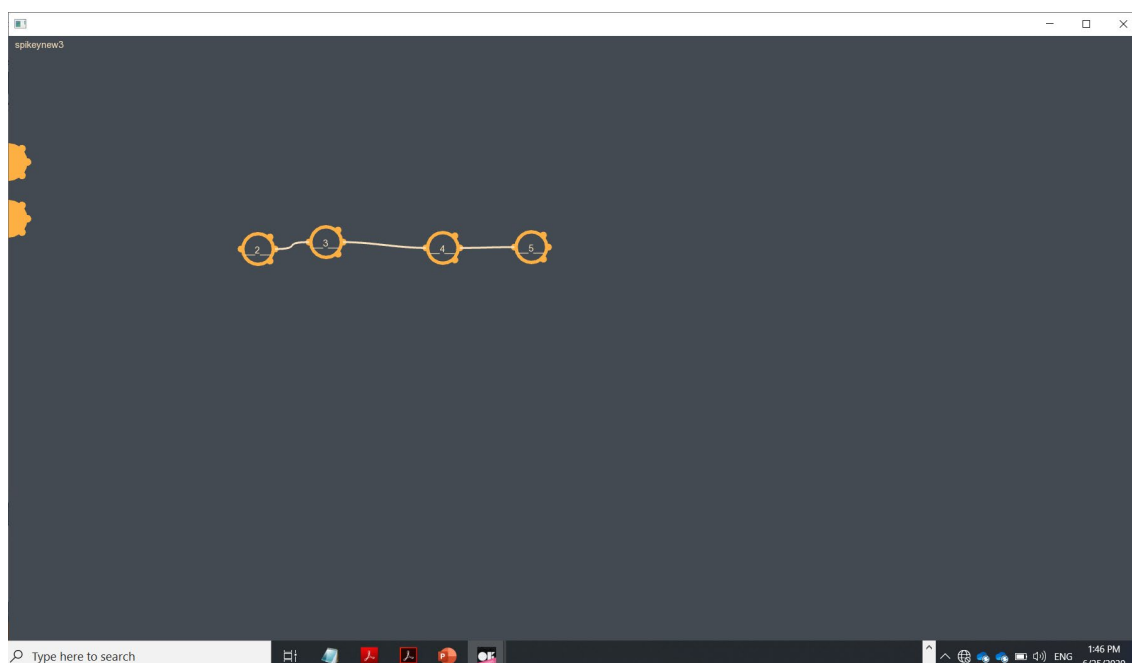
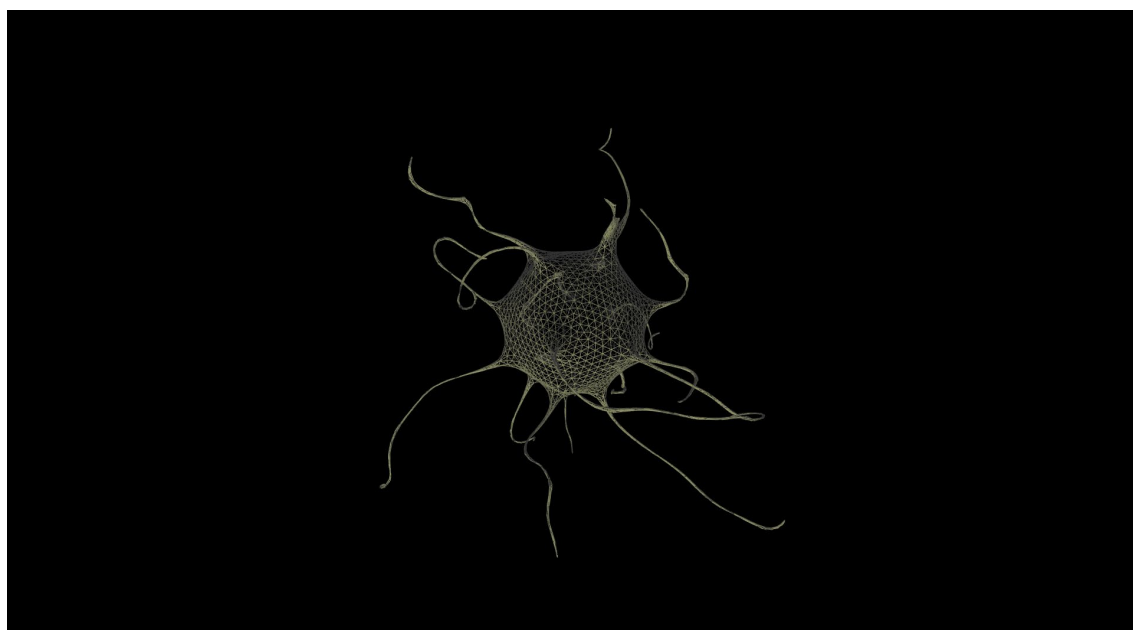


Figure B.5: Spikenew3 morphogenesis simulation and DNA states

Specie: Jelly23

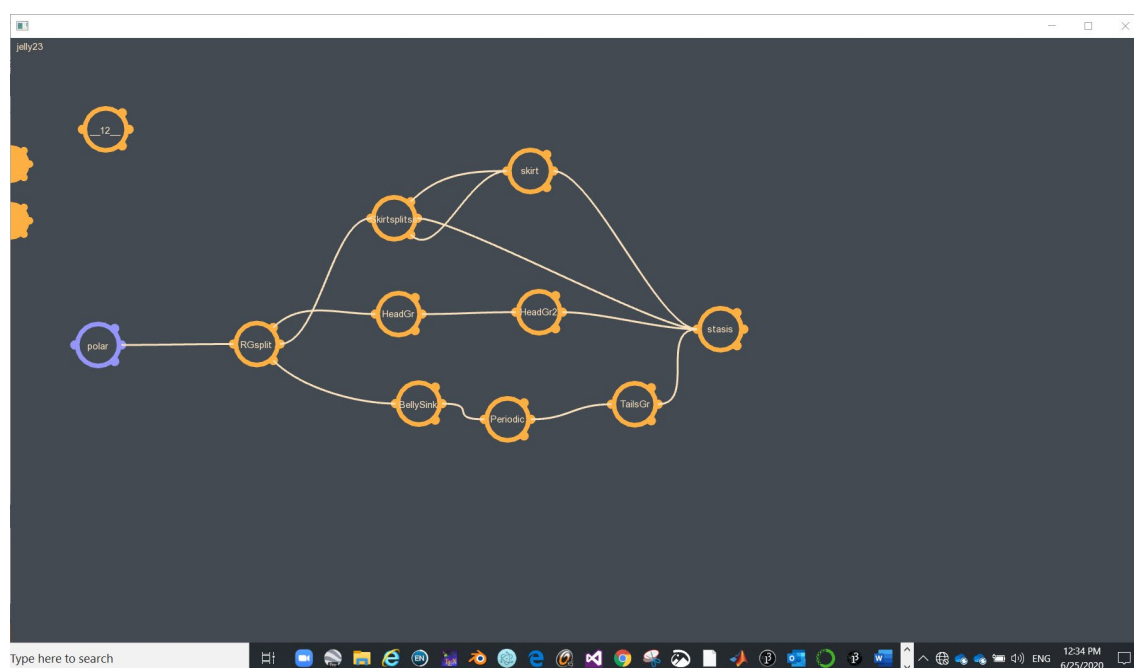


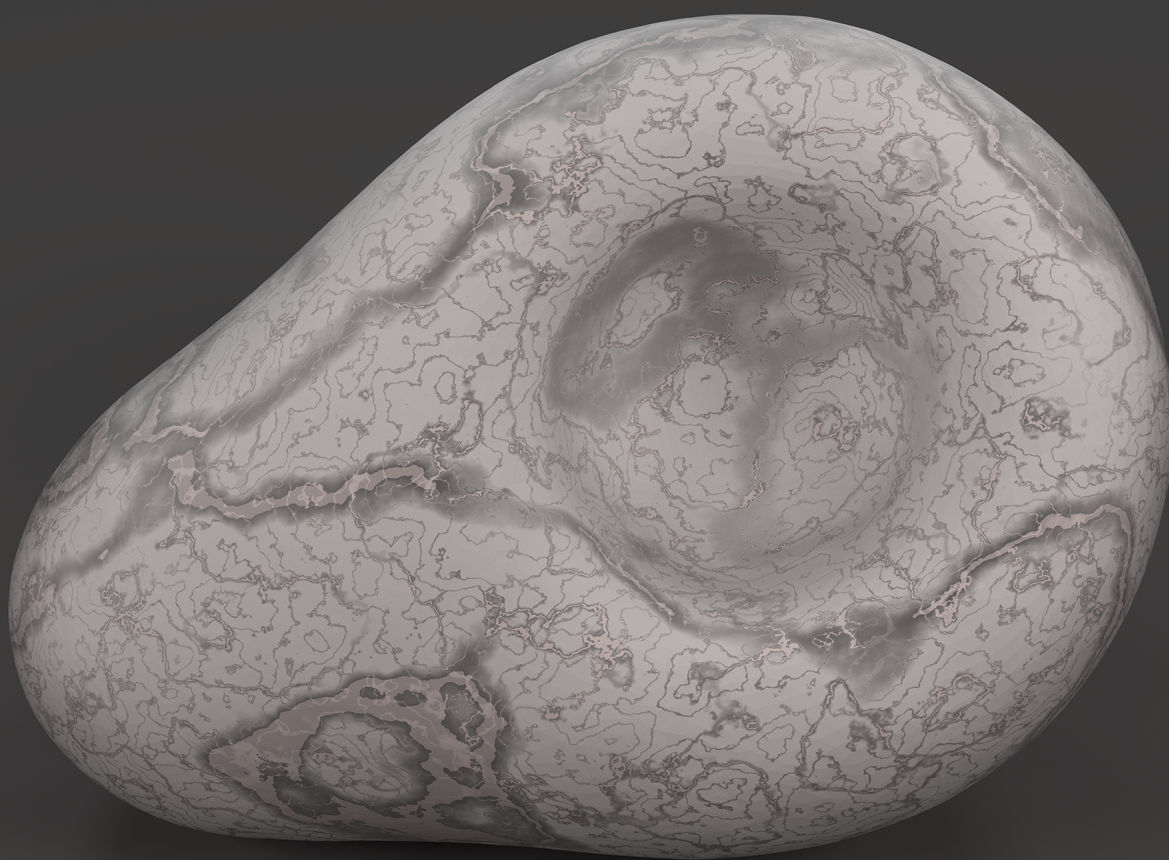
Figure B.7: jelly23 morphogenesis simulation and DNA states

[illegible]

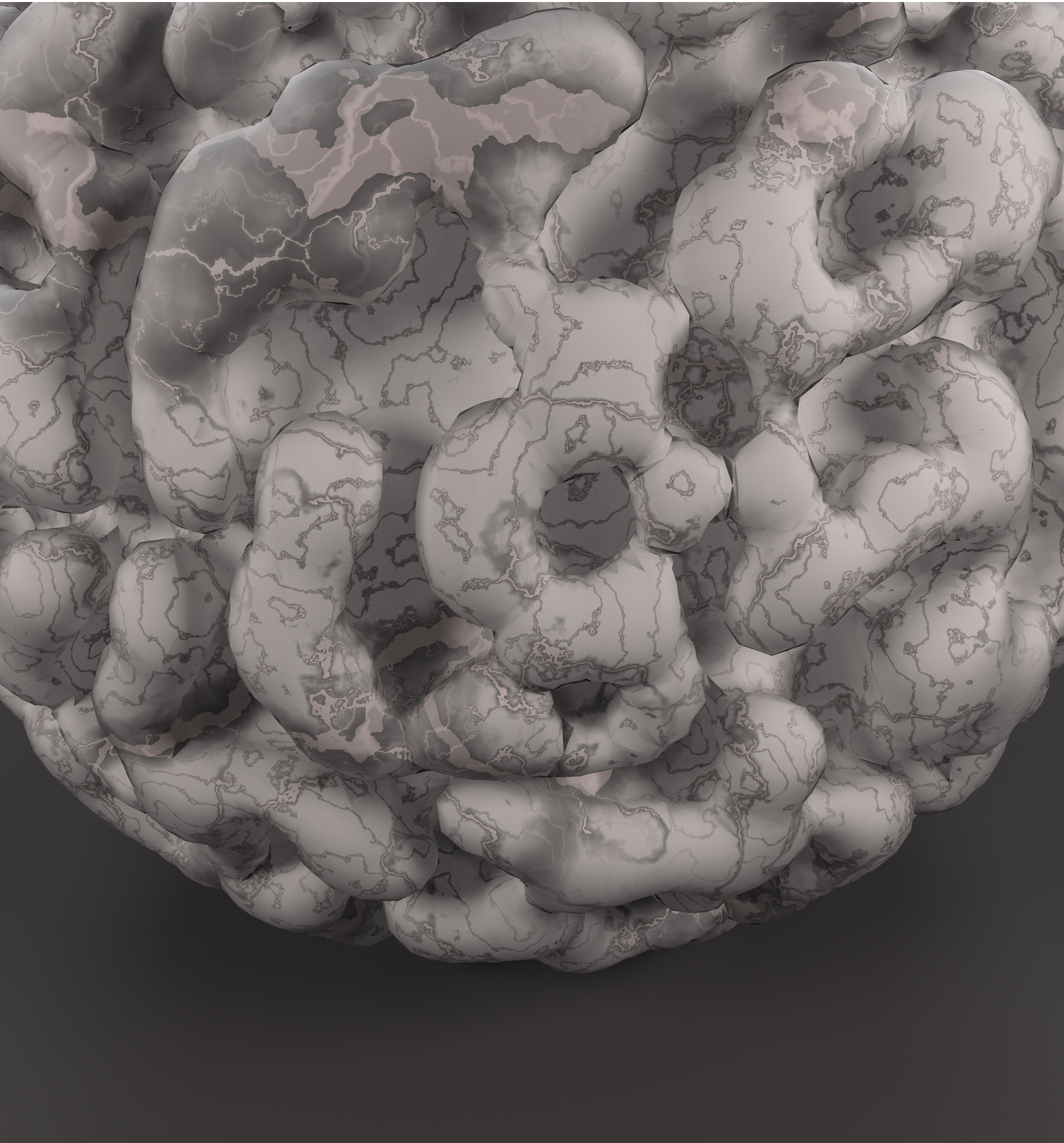
Figure B.8: jelly23.dna parameters page 1 of 2

Appendix C

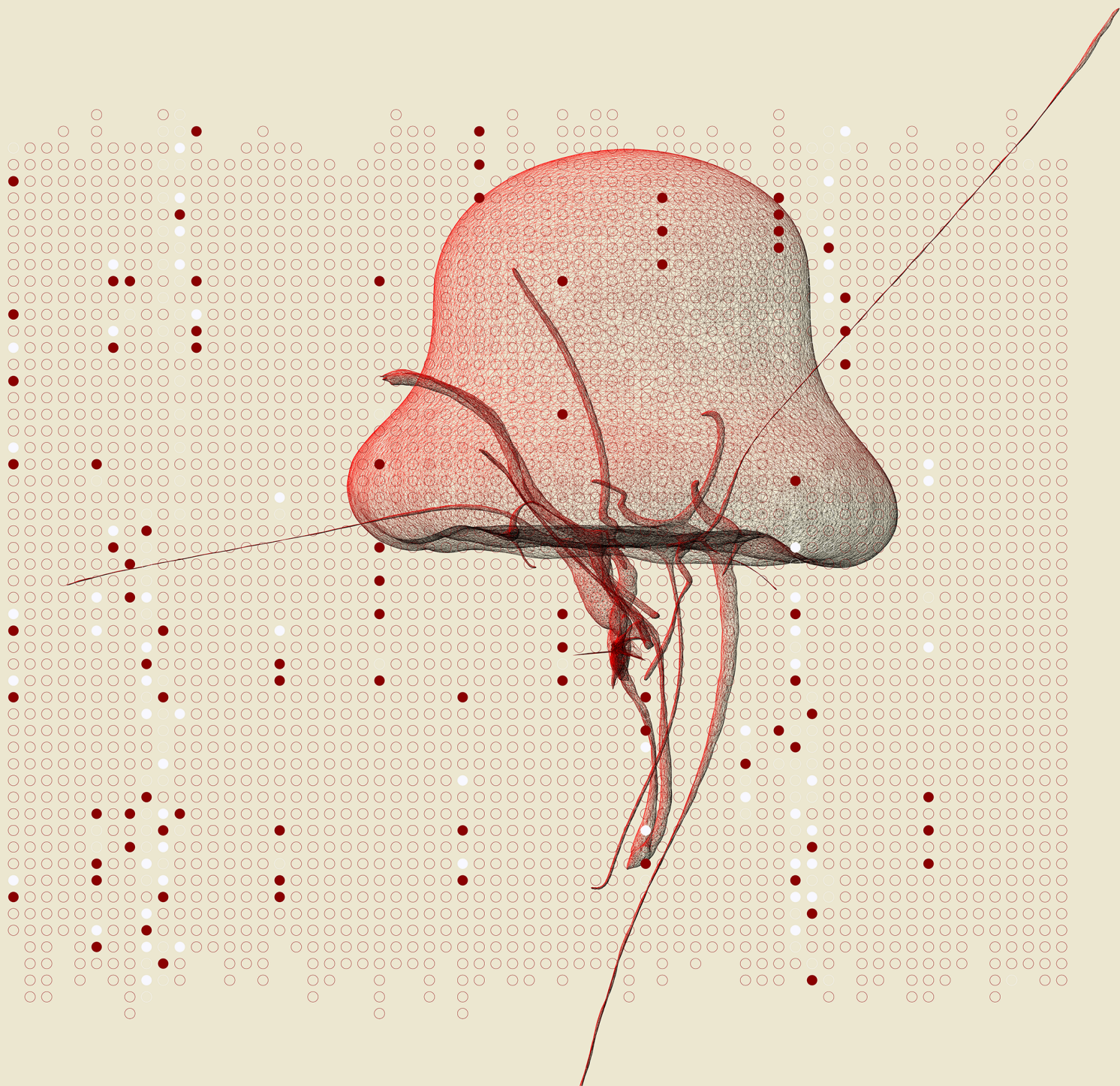
Artwork



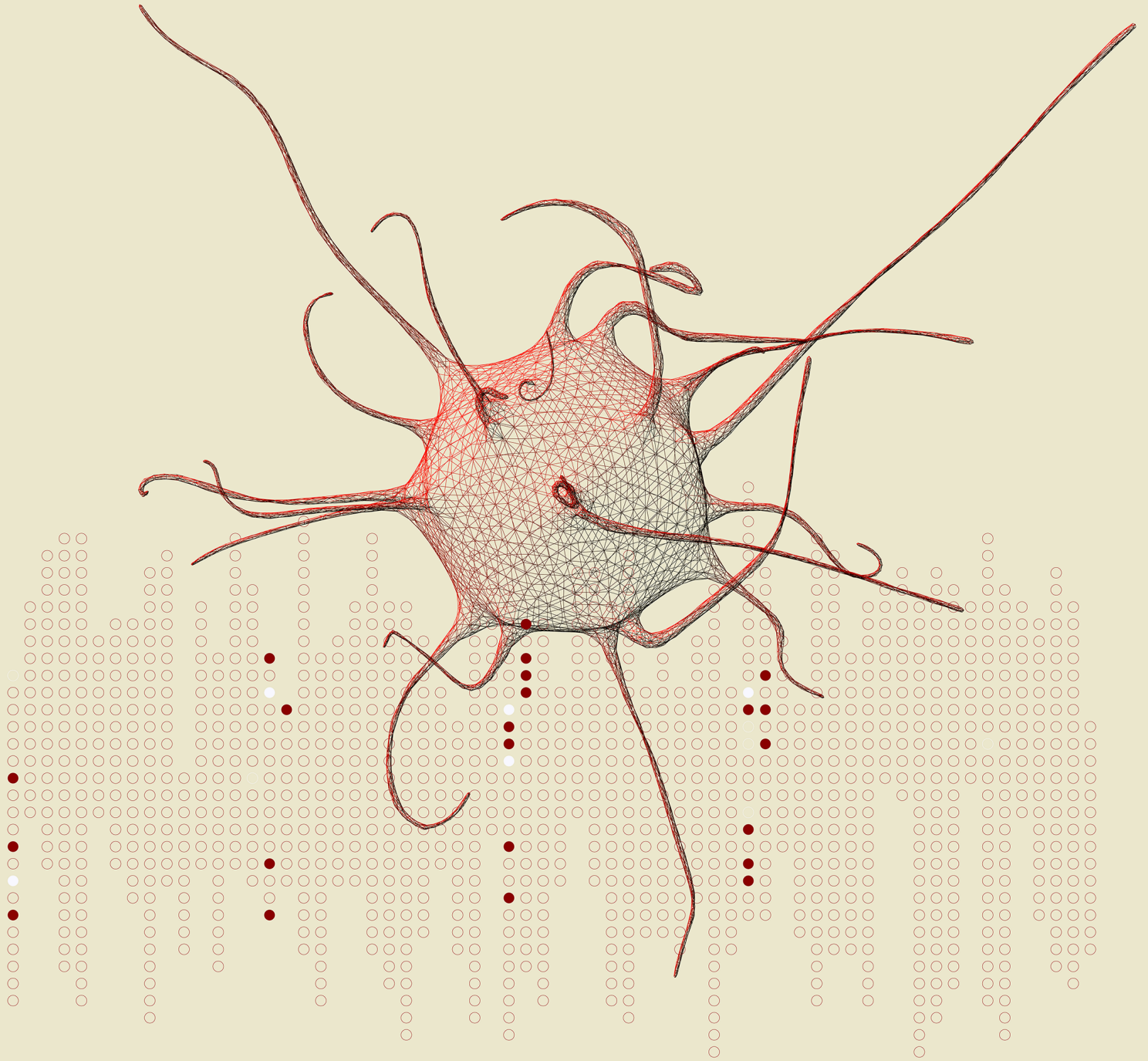
Fundamental Form B — 2018 — 120 x 120 cm — *Archival pigment print diasec*



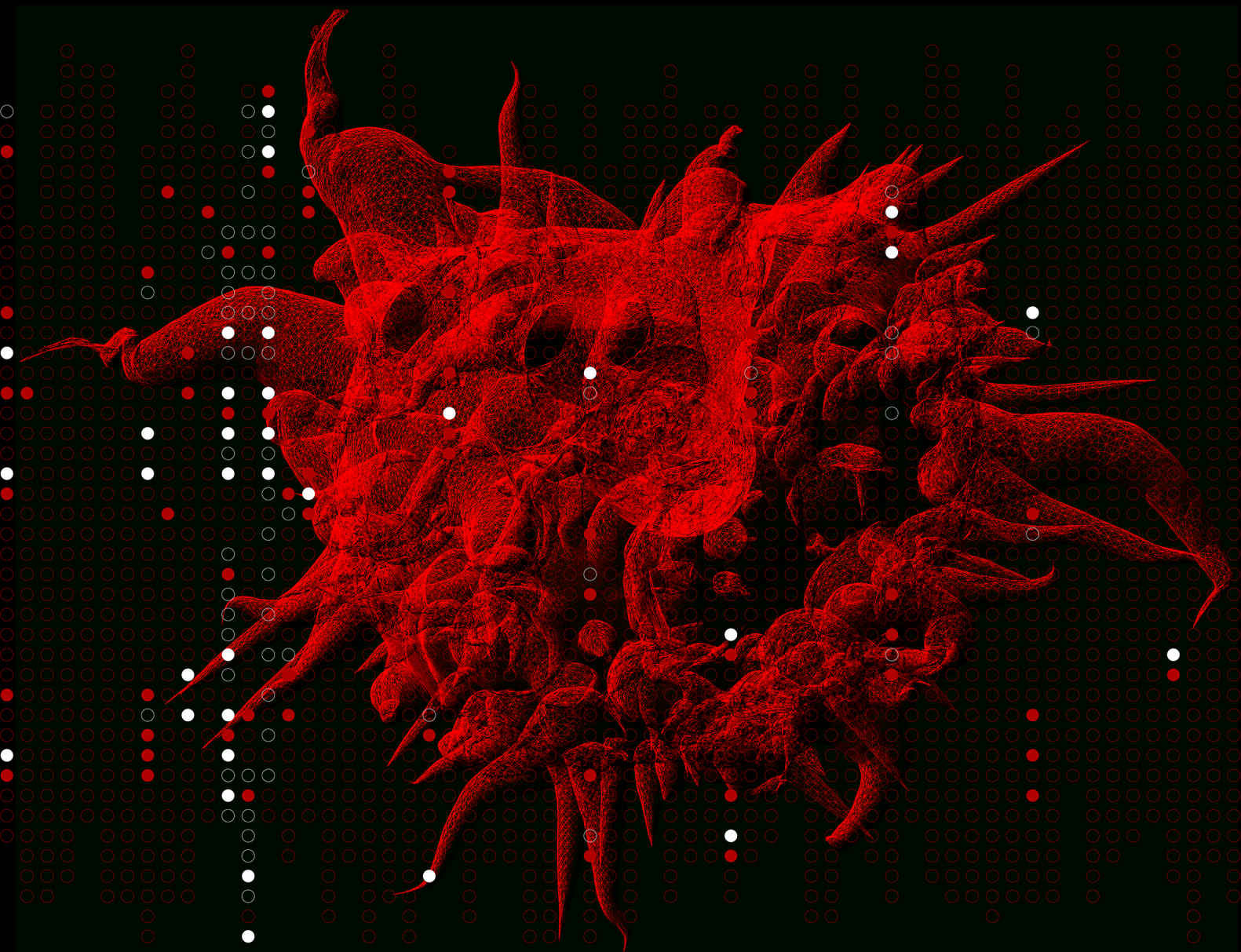
Fundamental Form D — 2018 — 120 x 120 cm — *Archival pigment print diasec*



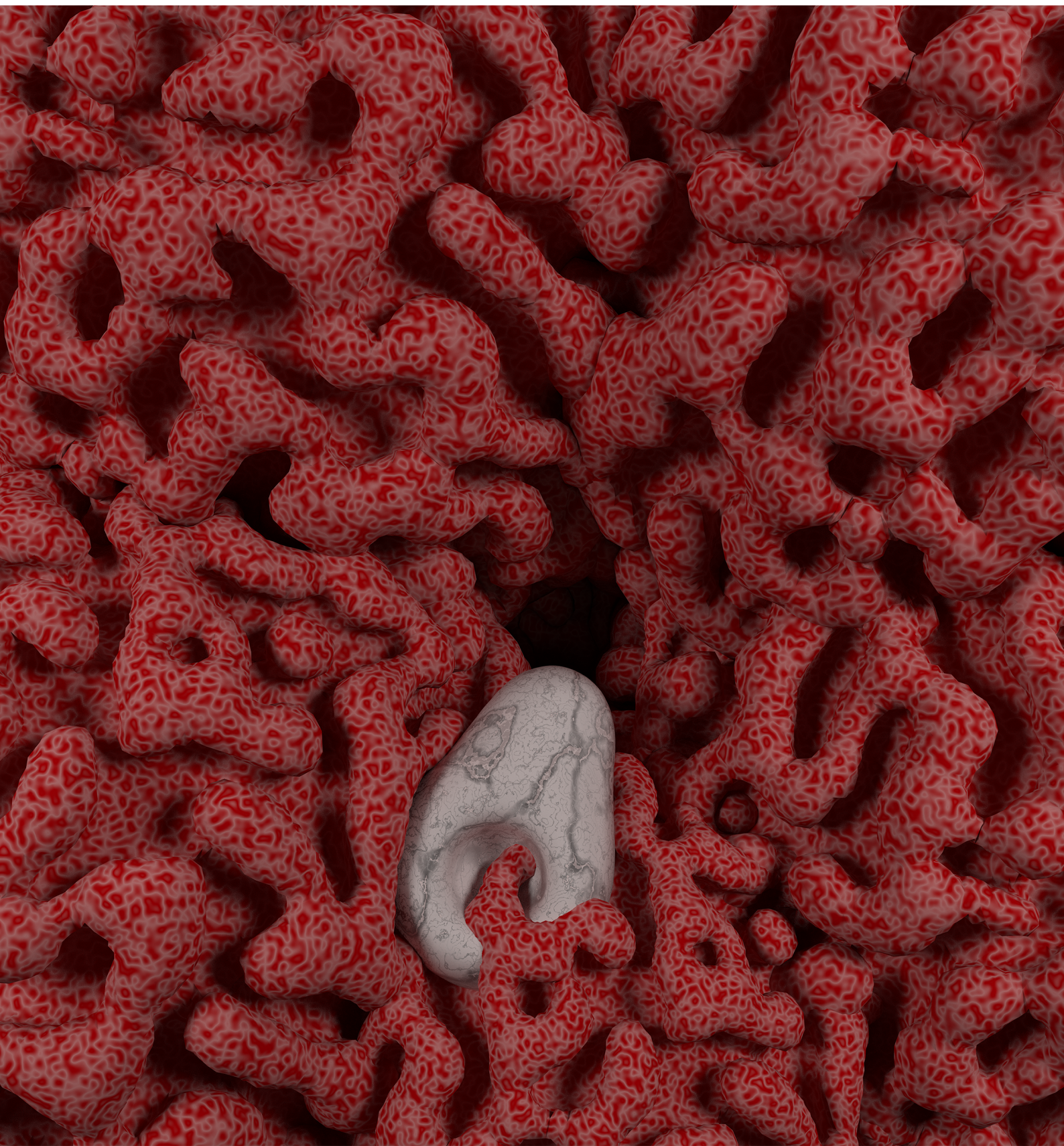
GenoPheno 1 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



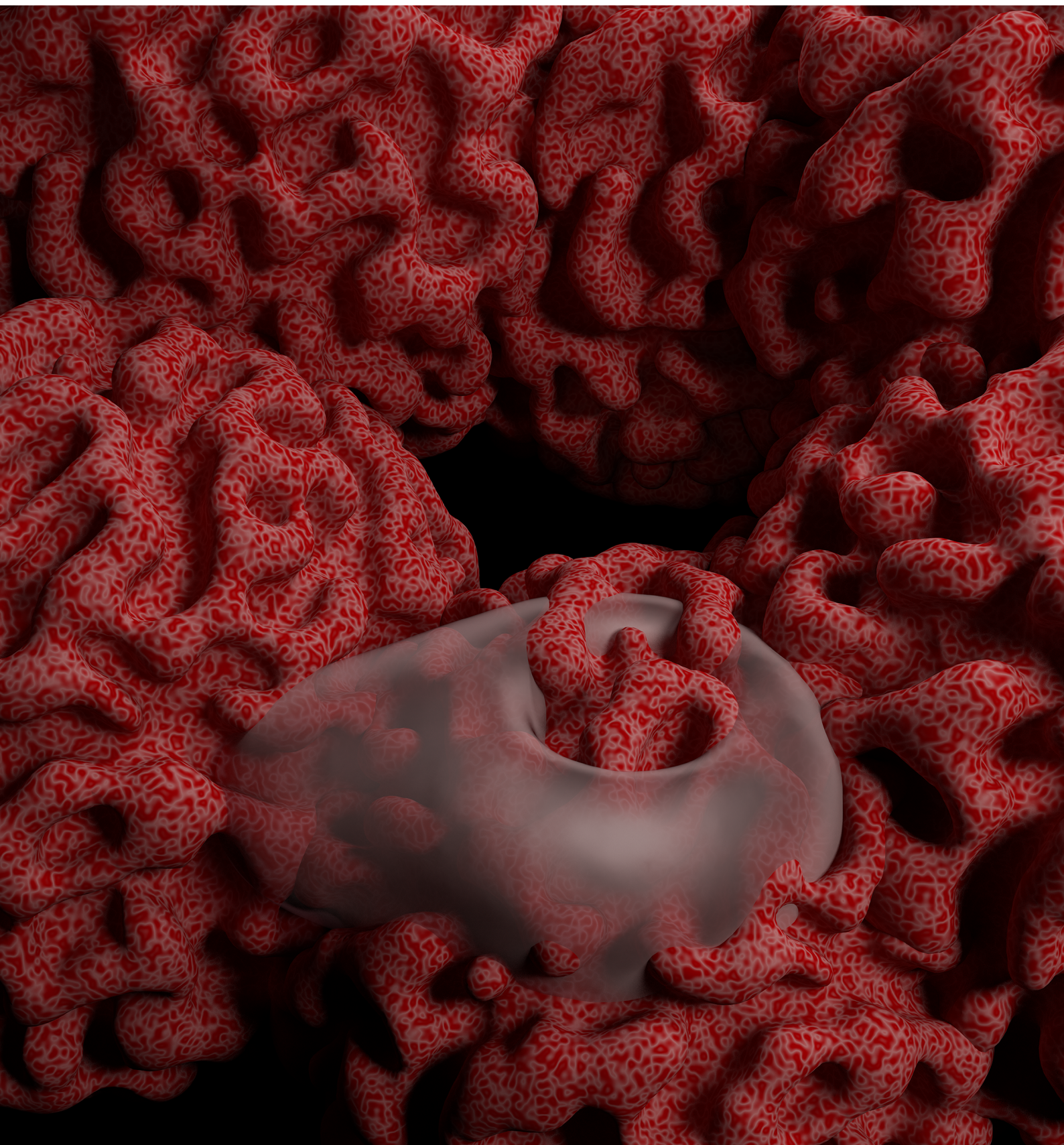
GenoPheno 2 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



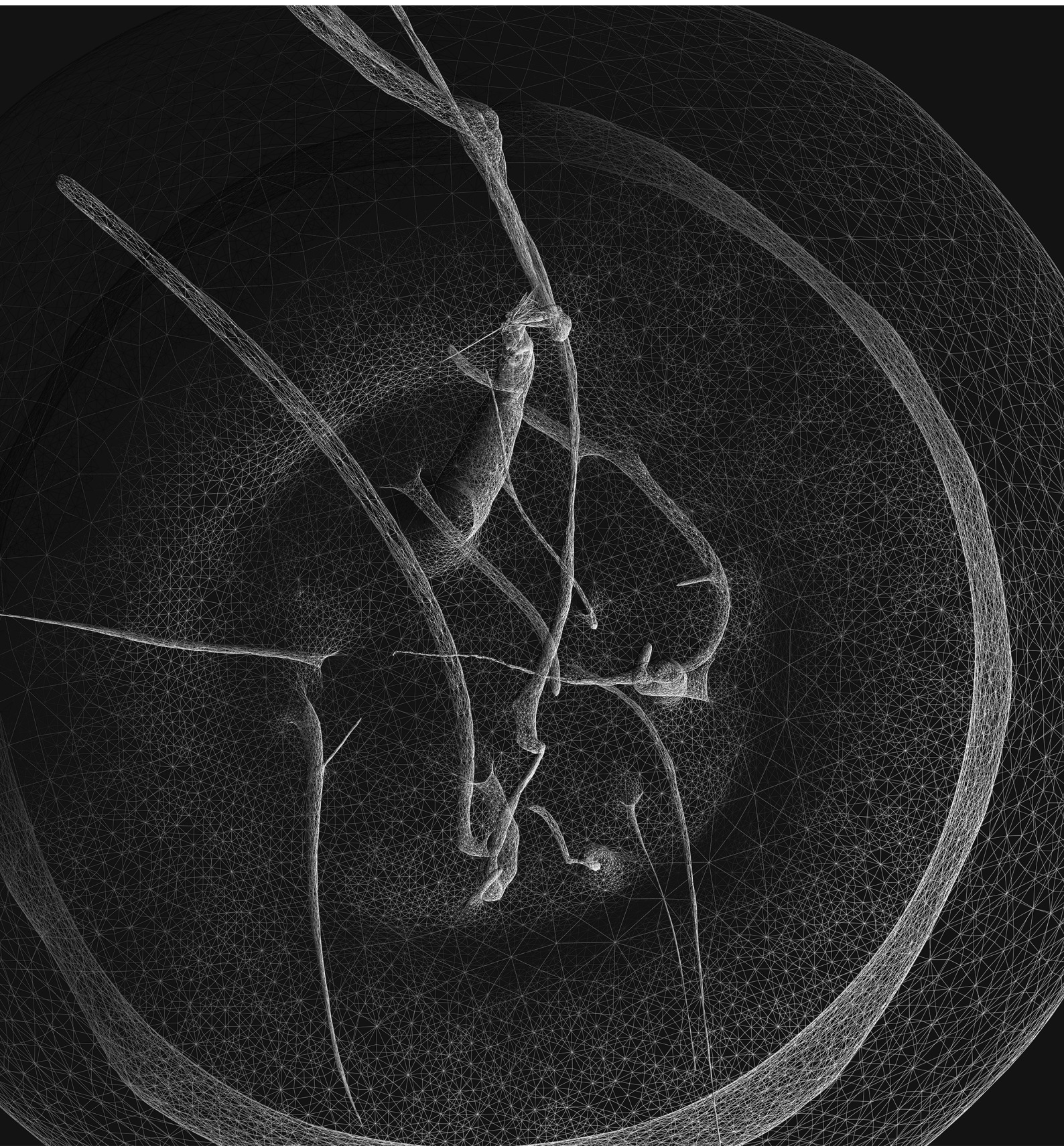
GenoPheno 3 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Form vs Pattern 1 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Form vs Pattern 2 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Origins 1 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Origins 2 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Origins 3 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



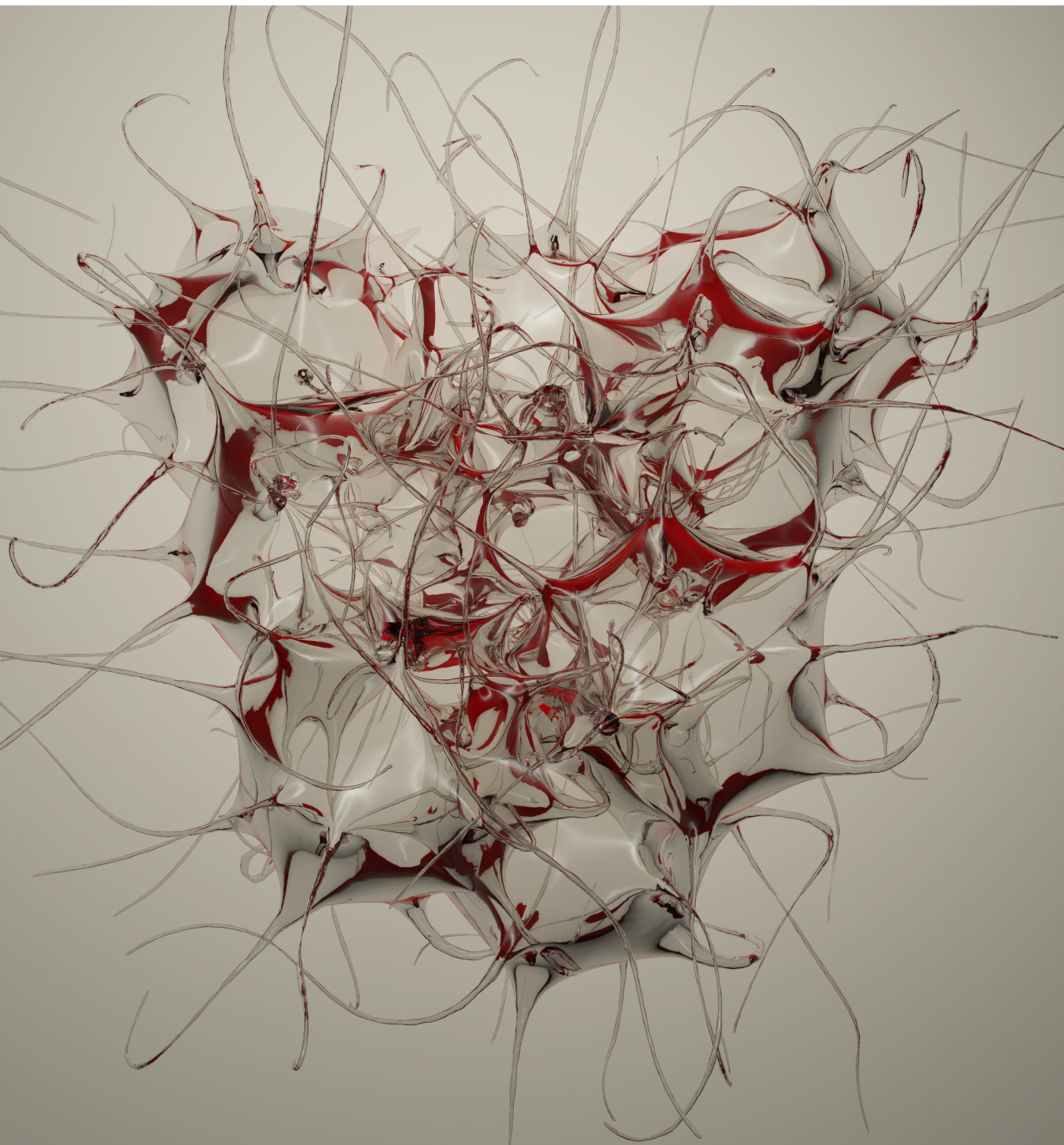
Abbys 1 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Abbys 2 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



Abbys 3 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



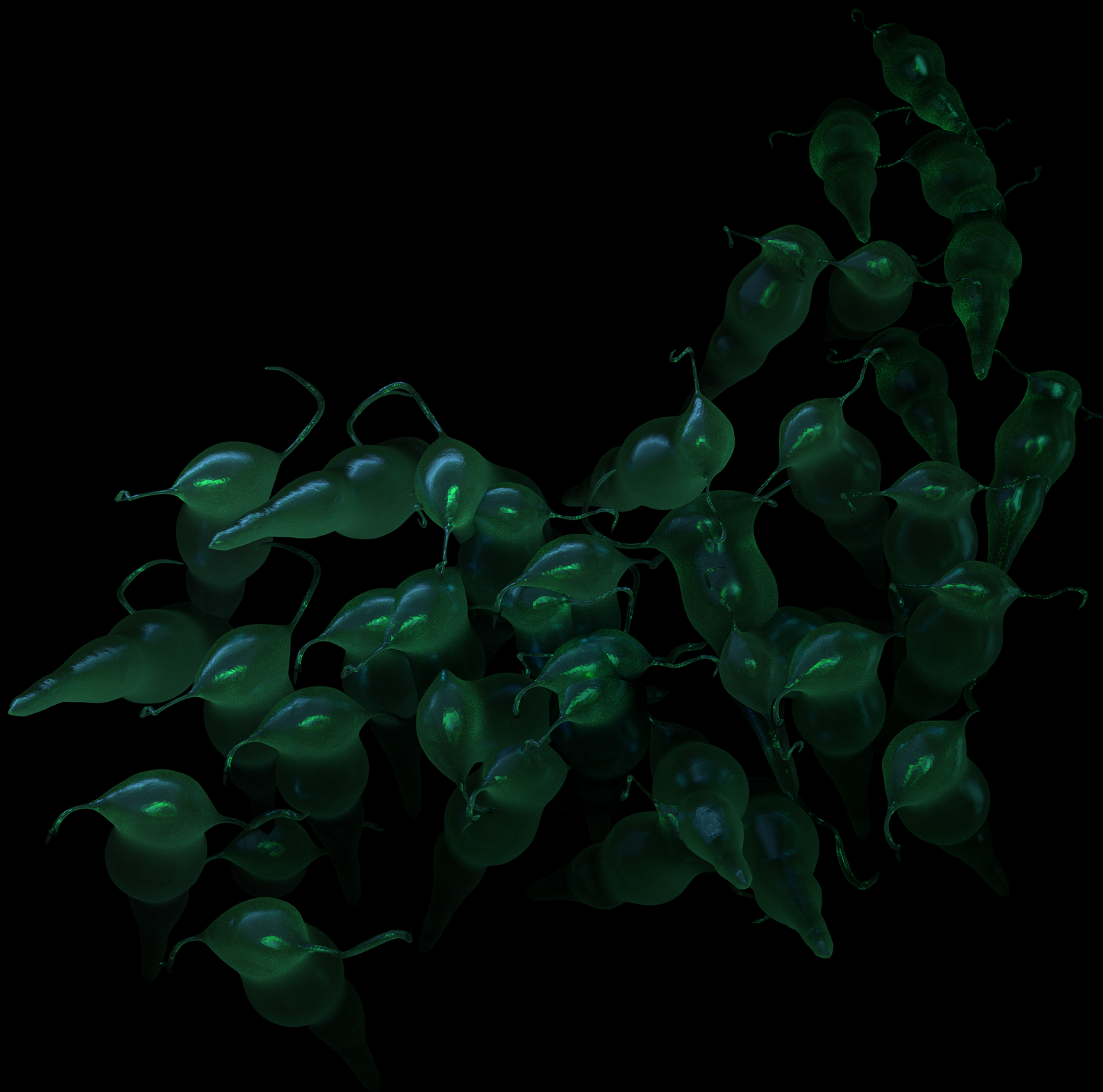
Abbys 4 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



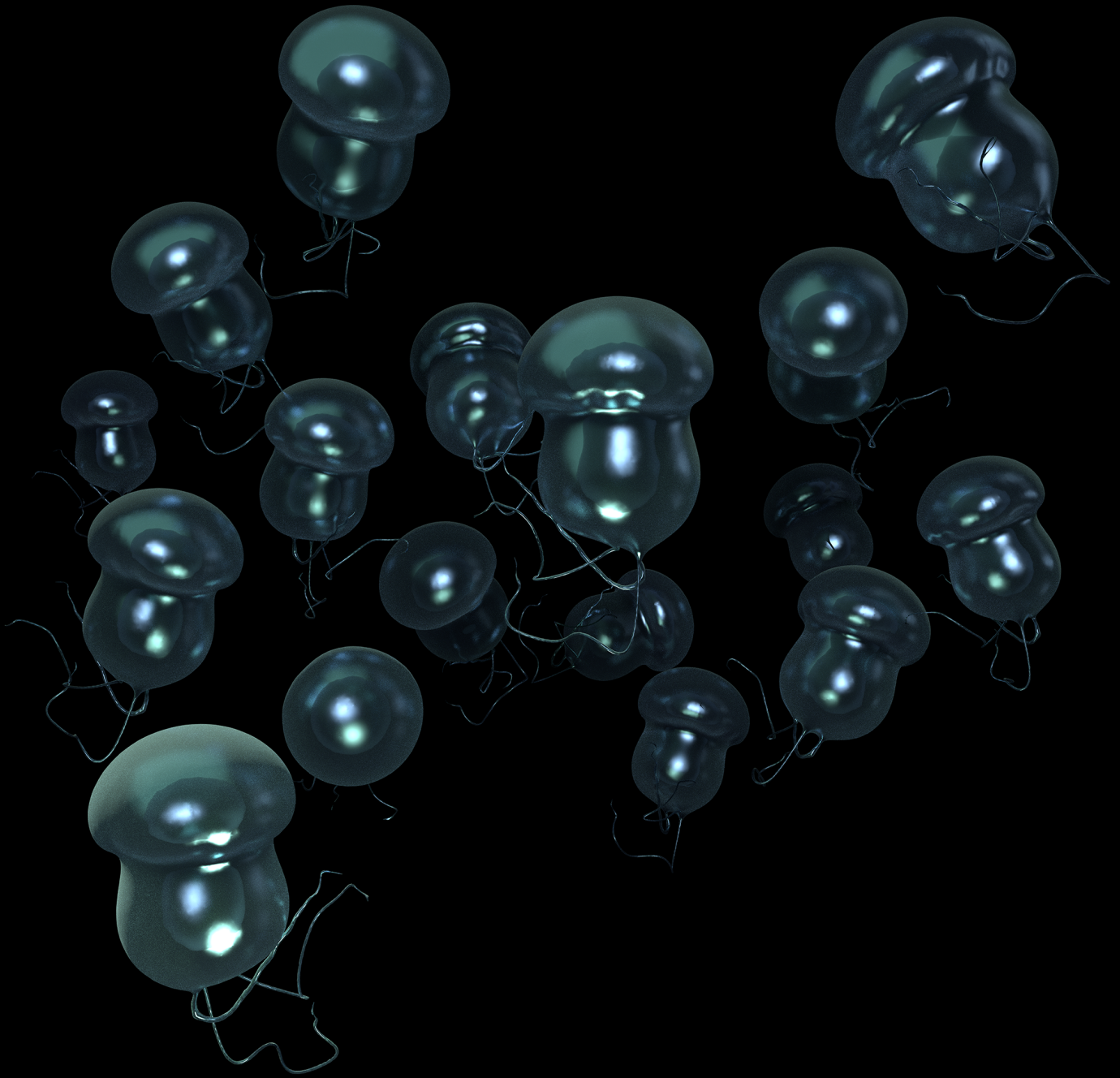
Abbys 5 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



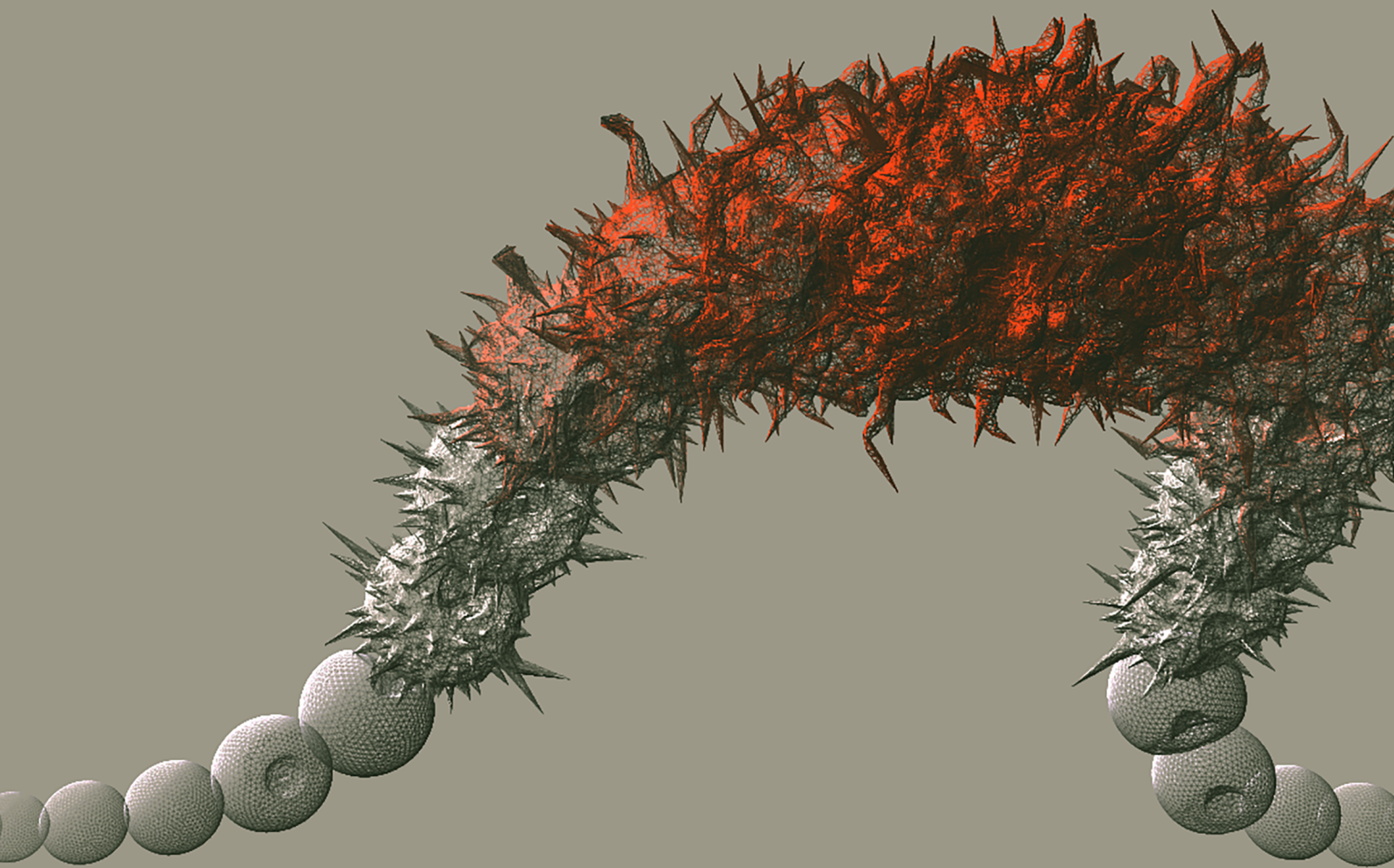
Encounters1 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



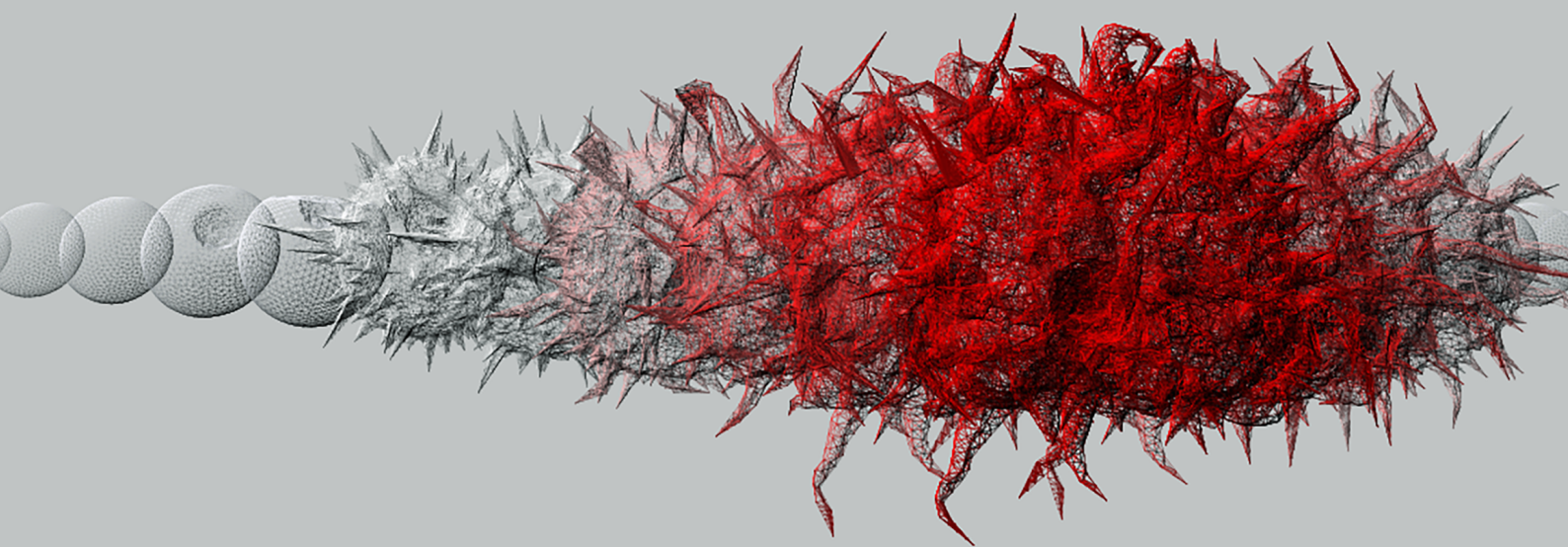
Encounters 2 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



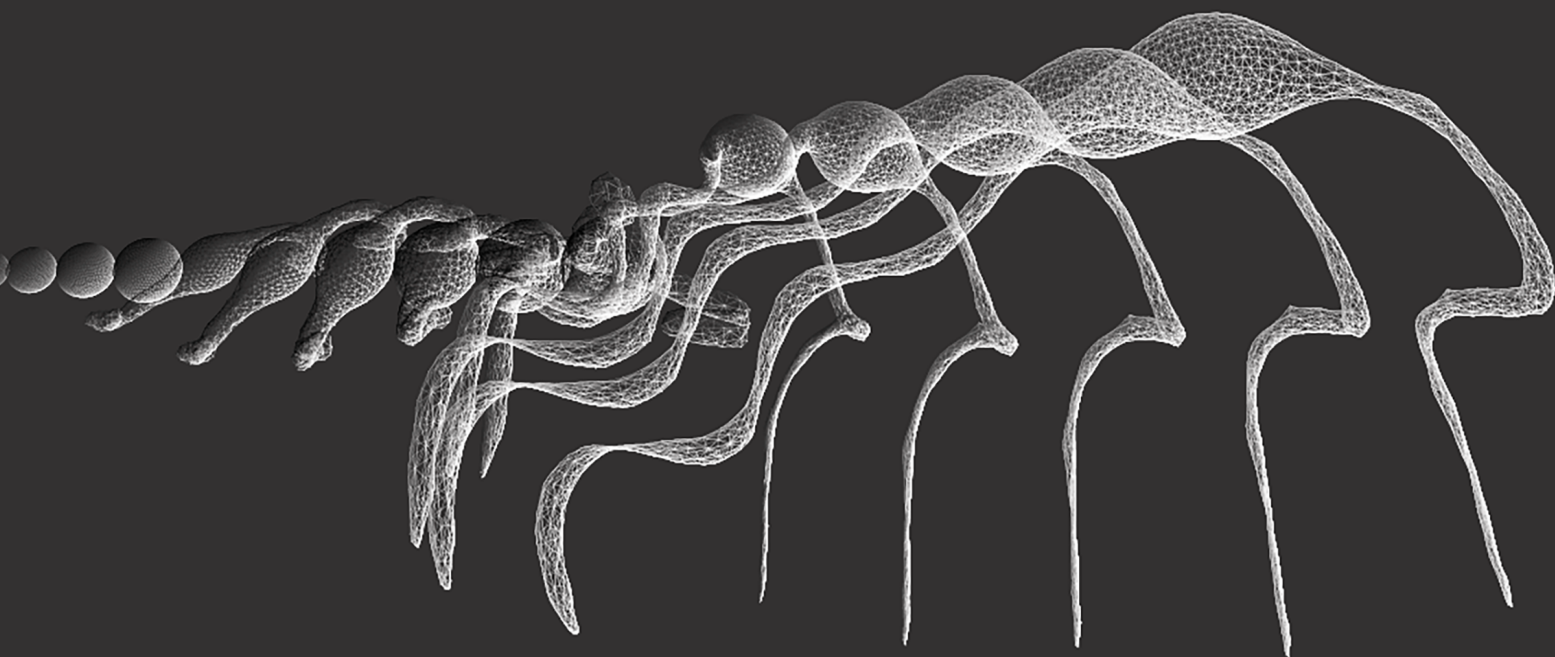
Encounters 3 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



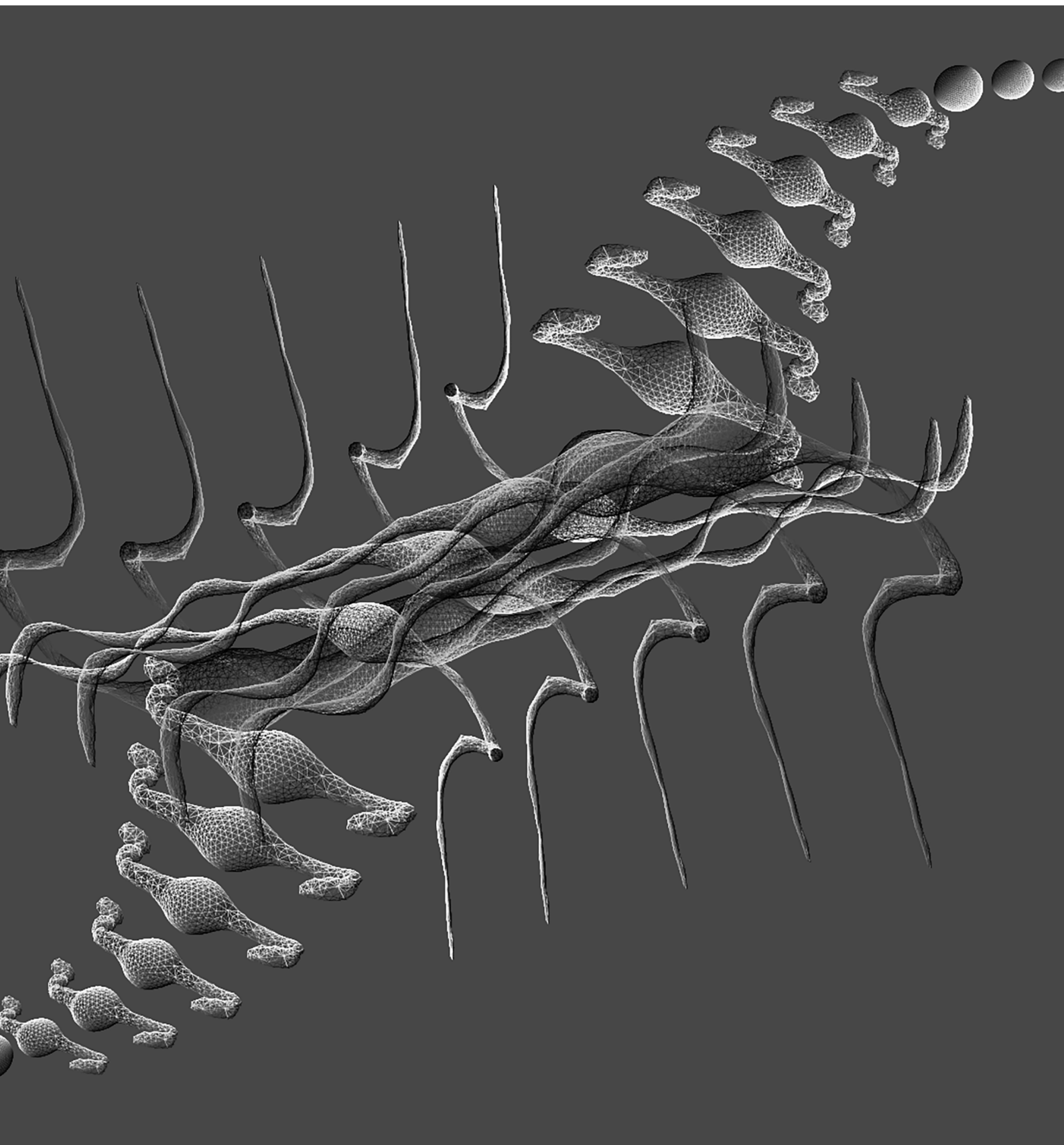
EvoDevo 1 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



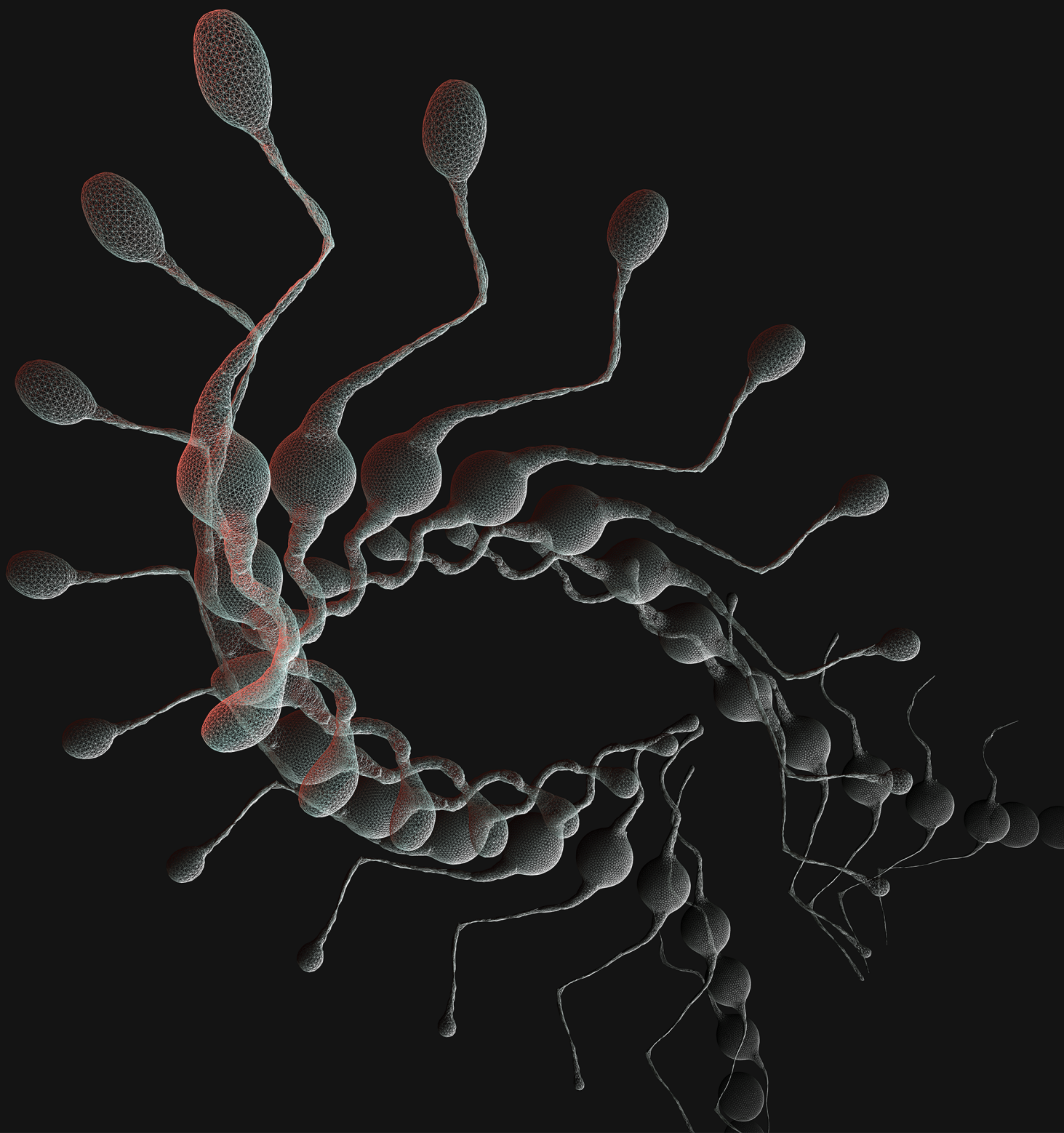
EvoDevo 2 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



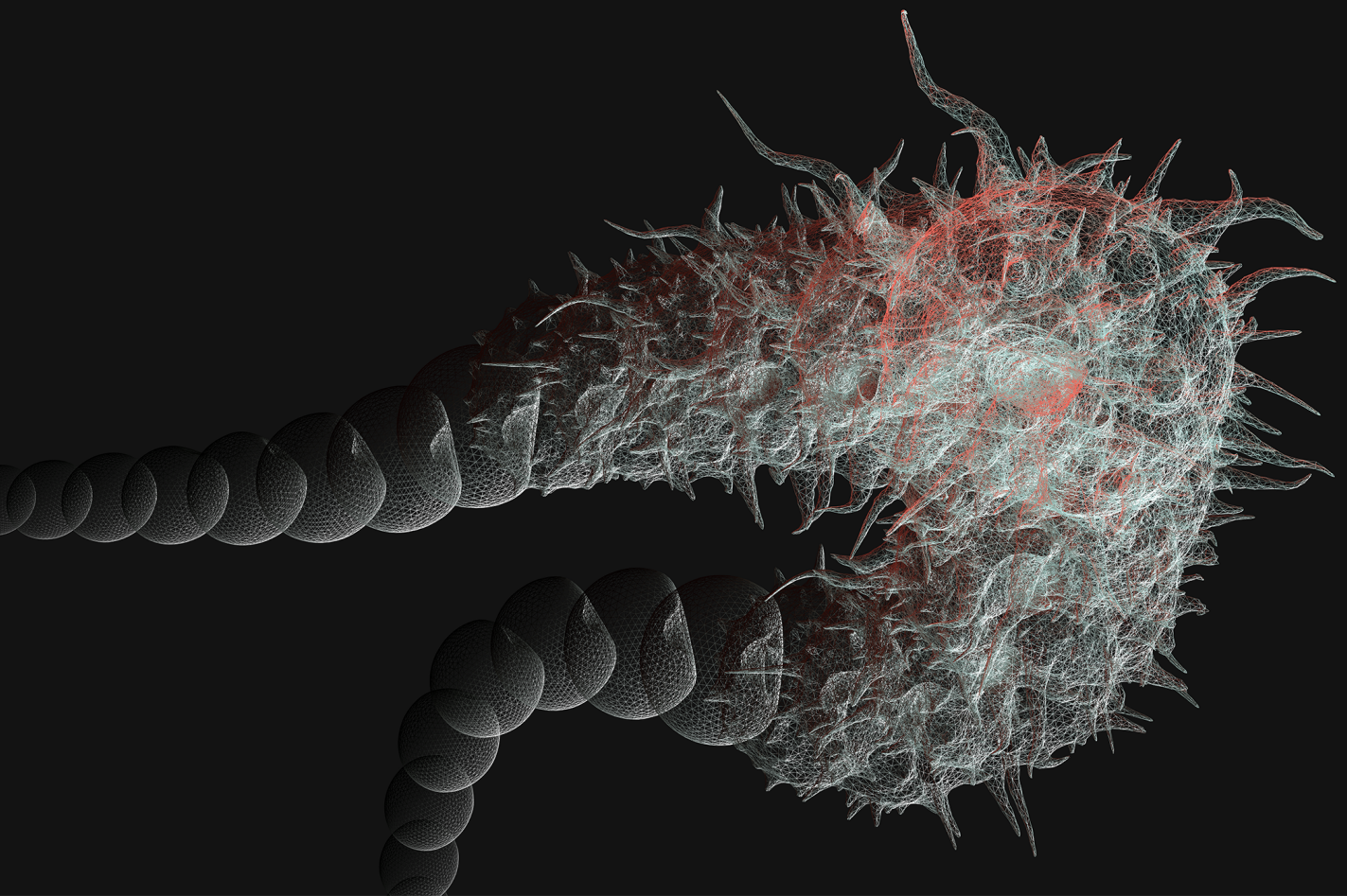
EvoDevo 3 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



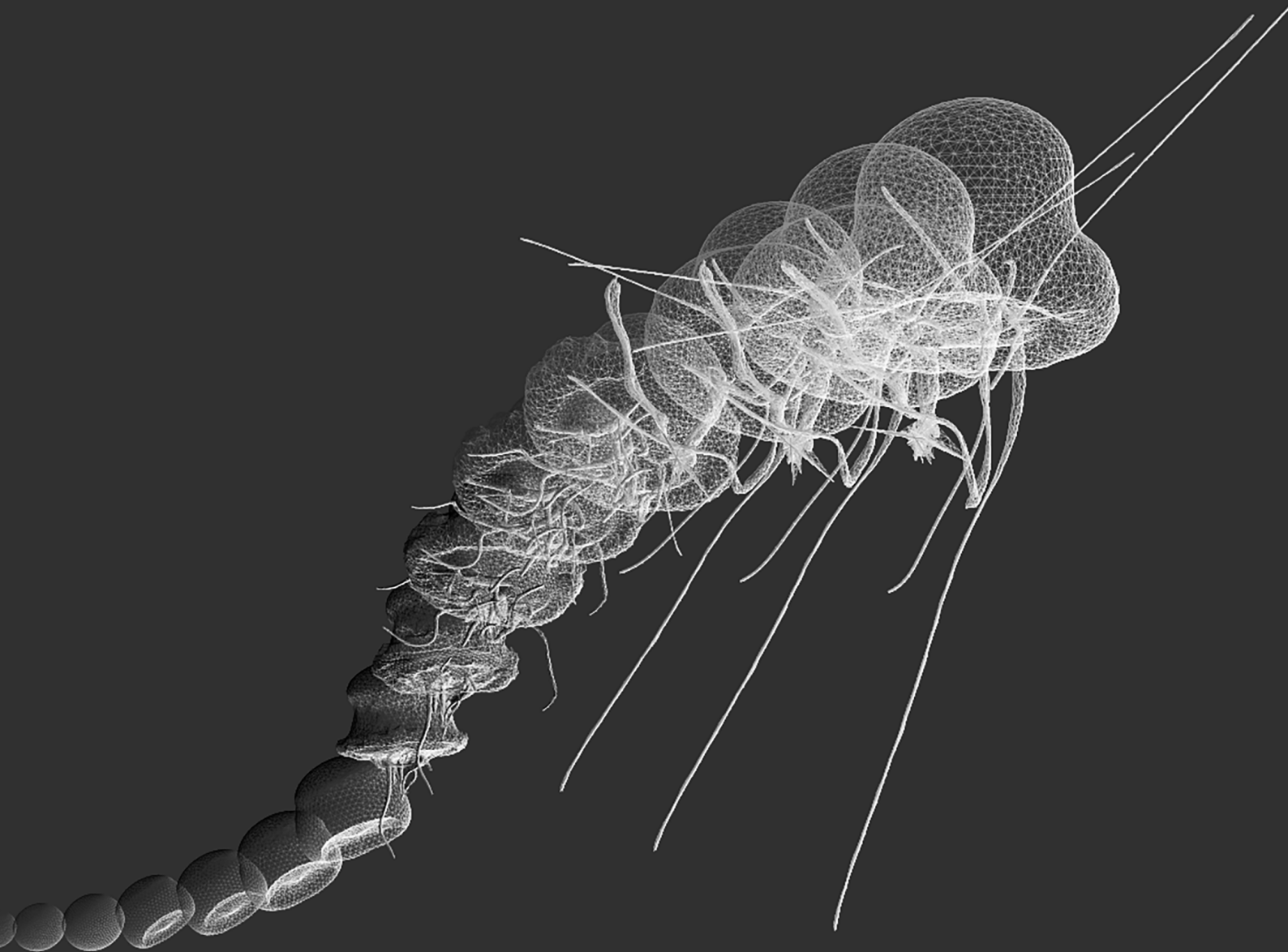
EvoDevo 4 — 2018 — 120 x 120 cm — *Archival pigment print diasec*

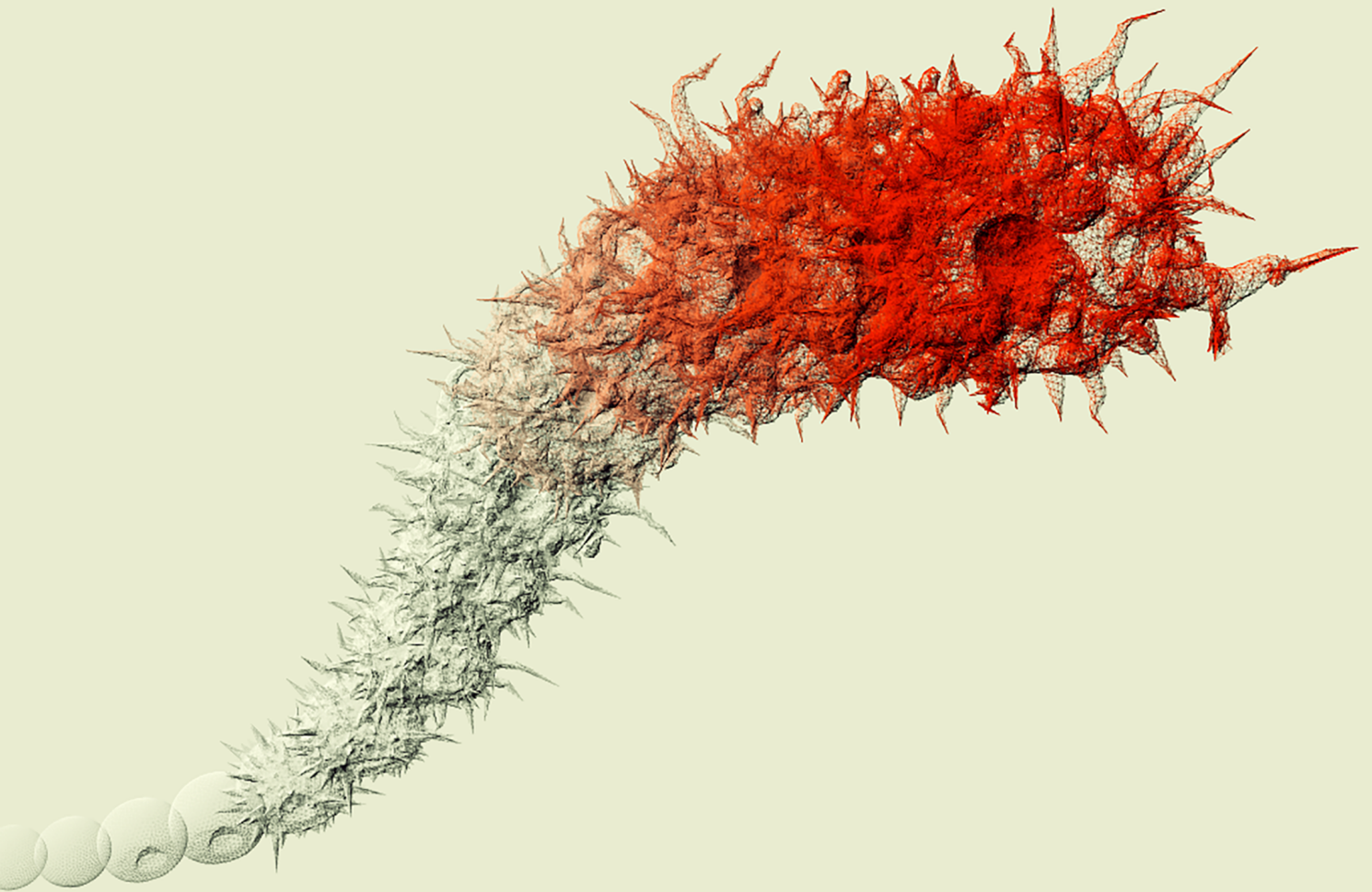


EvoDevo 5 — 2018 — 120 x 120 cm — *Archival pigment print diasec*



EvoDevo 6 — 2018 — 120 x 120 cm — *Archival pigment print diasec*





EvoDevo 8 — 2018 — 120 x 120 cm — *Archival pigment print diasec*

Bibliography

- [1] A. M. Turing. “The Chemical Basis of Morphogenesis”. In: *Royal Society of London Philosophical Transactions Series B* B 237 (1952), pp. 37–72.
- [2] S. Kauffman. “Gene regulation networks: a theory for their global structure and behaviors”. In: *Curr Top Dev Biol* 6.6 (1971), pp. 145–82.
- [3] A. Gierer and H. Meinhardt. “A theory of biological pattern formation”. In: *Kybernetik* 12.1 (1972), pp. 30–39.
- [4] H. Meinhardt. “Formation of Symmetrical and Asymmetric Structures during Development of Higher Organisms”. In: *Comput Math Appl-B* 12.1-2 (1986), pp. 419–433.
- [5] H. Meinhardt. “Tailoring and Coupling of Reaction-Diffusion Systems to Obtain Reproducible Complex Pattern-Formation during Development of the Higher Organisms”. In: *Appl Math Comput* 32.2-3 (1989), pp. 103–135.
- [6] H. Meinhardt. “Development of Higher Organisms - How to Avoid Error Propagation and Chaos”. In: *Physica D* 86.1-2 (1995), pp. 96–103.
- [7] H. Meinhardt. “Models for organizer and notochord formation”. In: *Cr Acad Sci Iii-Vie* 323.1 (2000), pp. 23–30.
- [8] H. Meinhardt. “Models for the Generation and Interpretation of Gradients”. In: *Csh Perspect Biol* 1.4 (2009).
- [9] S. A. Kuuffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993. ISBN: 978-0195079517.
- [10] D’arcy Thompson. *On Growth and Form*. Cambridge University Press, 1917. ISBN: 978-1107672567.

- [11] L. Wolpert. “Positional Information and Pattern-Formation”. In: *Philos T Roy Soc B* 295.1078 (1981), pp. 441–450.
- [12] E. Mjolsness, D. H. Sharp, and J. Reinitz. “A connectionist model of development”. In: *J Theor Biol* 152.4 (1991), pp. 429–53.
- [13] B. E. Shapiro et al. “Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations”. In: *Bioinformatics* 19.5 (2003), pp. 677–678. DOI: 10.1093/bioinformatics/btg042.
- [14] R. Dillon and H. G. Othmer. “A mathematical model for outgrowth and spatial patterning of the vertebrate limb bud”. In: *J Theor Biol* 197.3 (1999), pp. 295–330. DOI: DOI10.1006/jtbi.1998.0876.
- [15] I. Salazar-Ciudad, J. Garcia-Fernandez, and R. V. Sole. “Gene networks capable of pattern formation: From induction to reaction-diffusion”. In: *J Theor Biol* 205.4 (2000), pp. 587–603.
- [16] G. von Dassow et al. “The segment polarity network is a robust developmental module”. In: *Nature* 406.6792 (2000), pp. 188–192. DOI: Doi10.1038/35018085.
- [17] S. H. Robertson et al. “Multiscale computational analysis of *Xenopus laevis* morphogenesis reveals key insights of systems-level behavior”. In: *Bmc Syst Biol* 1 (2007). DOI: Artn4610.1186/1752-0509-1-46.
- [18] S. Hoehme and D. Drasdo. “A cell-based simulation software for multi-cellular systems”. In: *Bioinformatics* 26.20 (2010), pp. 2641–2642. DOI: 10.1093/bioinformatics/btq437.
- [19] B. Boehm et al. “The Role of Spatially Controlled Cell Proliferation in Limb Bud Morphogenesis”. In: *Plos Biol* 8.7 (2010). DOI: ARTNe100042010.1371/journal.pbio.1000420.
- [20] S. D. Hester et al. “A Multi-cell, Multi-scale Model of Vertebrate Segmentation and Somite Formation”. In: *PLoS Comput. Biol.* 7.10 (2011). DOI: ARTNe100215510.1371/journal.pcbi.1002155.

- [21] T. Glimm, R. Bhat, and S. A. Newman. “Modeling the morphodynamic galectin patterning network of the developing avian limb skeleton”. In: *J Theor Biol* 346 (2014), pp. 86–108. DOI: 10.1016/j.jtbi.2013.12.004.
- [22] P. Villoutreix et al. “An integrated modelling framework from cells to organism based on a cohort of digital embryos”. In: *Sci Rep-Uk* 6 (2016). DOI: ARTN3743810.1038/srep37438.
- [23] J. Delile et al. “A cell-based computational model of early embryogenesis coupling mechanical behaviour and gene regulation”. In: *Nat Commun* 8 (2017).
- [24] S. Tanaka, D. Sichau, and D. Iber. “LBIBCell: a cell-based simulation environment for morphogenetic problems”. In: *Bioinformatics* 31.14 (2015), pp. 2340–2347. DOI: 10.1093/bioinformatics/btv147.
- [25] E. H. Davidson. *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Academic Press, 2010. ISBN: ISBN-13 978-0120885633.
- [26] Iba H. and N. Noman. *Evolutionary Computation in Gene Regulatory Network Research*. Wiley, 2016. ISBN: ISBN-13 978-1118911518.
- [27] H. De Jong. “Modeling and simulation of genetic regulatory systems: A literature review”. In: *J Comput Biol* 9.1 (2002), pp. 67–103. DOI: Doi10.1089/10665270252833208.
- [28] S. Wolfram. “Computation Theory of Cellular Automata”. In: *Commun Math Phys* 96.1 (1984), pp. 15–57.
- [29] S. Wolfram. “Cellular Automata as Models of Complexity”. In: *Nature* 311.5985 (1984), pp. 419–424.
- [30] N. H. Packard and S. Wolfram. “Two-Dimensional Cellular Automata”. In: *J Stat Phys* 38.5-6 (1985), pp. 901–946.
- [31] K. Culik and S. Yu. “Undecidability of CA classification schemes”. In: *Complex Systems* 2 (1988), p. 177.
- [32] W. T. Li, N. H. Packard, and C. G. Langton. “Transition Phenomena in Cellular Automata Rule Space”. In: *Physica D* 45.1-3 (1990), pp. 77–94. DOI: Doi10.1016/0167-2789(90)90175-0.

- [33] P. Kurka. “Languages, equicontinuity and attractors in cellular automata”. In: *Ergod Theor Dyn Syst* 17 (1997), pp. 417–433.
- [34] L. O. Chua, S. Yoon, and R. Dogaru. “A nonlinear dynamics perspective of Wolfram’s new kind of science. Part I: Threshold of complexity”. In: *Int J Bifurcat Chaos* 12.12 (2002), pp. 2655–2766. DOI: Doi10.1142/S0218127402006333.
- [35] T. Ewert. “A Measure for the Complexity of Elementary Cellular Automata”. In: *Complex Syst* 28.2 (2019), pp. 219–+. DOI: 10.25088/ComplexSystems.28.2.219.
- [36] C. Marr and M. T. Hutt. “Topology regulates pattern formation capacity of binary cellular automata on graphs”. In: *Physica A* 354 (2005), pp. 641–662. DOI: 10.1016/j.physa.2005.02.019.
- [37] M. Tomassini. “Generalized automata networks”. In: *Cellular Automata, Proceedings* 4173 (2006), pp. 14–28.
- [38] G. Cattaneo et al. “Non-uniform Cellular Automata”. In: *Language and Automata Theory and Applications* 5457 (2009), pp. 302–+. DOI: Doi10.1007/978-3-642-00982-2_26.
- [39] P. Bentley and S. Kumar. “Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem”. In: *Gecco-99: Proceedings of the Genetic and Evolutionary Computation Conference* (1999), pp. 35–43.
- [40] K. O. Stanley and R. Miikkulainen. “A taxonomy for artificial embryogeny”. In: *Artif Life* 9.2 (2003), pp. 93–130. DOI: Doi10.1162/106454603322221487.
- [41] Rene Doursat. *Morphogenetic Engineering: Toward Programmable Complex Systems*. Springer, 2013. ISBN: ISBN-13: 978-3462339011.
- [42] L. G. Bai, M. Eyiurekli, and D. E. Breen. “An Emergent System for Self-Aligning and Self-Organizing Shape Primitives”. In: *Saso 2008: Second Ieee International Conference on Self-Adaptive and Self-Organizing Systems, Proceedings* (2008), pp. 445–454. DOI: 10.1109/Saso.2008.54.
- [43] R. Doursat. “Facilitating Evolutionary Innovation by Developmental Modularity and Variability”. In: *GECCO 2009* (2009).

-
- [44] R. Doursat and C. Sanchez. “Growing Fine-Grained Multicellular Robots”. In: *Soft Robot* 1.2 (2014), pp. 110–121. DOI: 10.1089/soro.2014.0014.
 - [45] Murat Erkurt. *Emergence of Form in Embryogenesis*. Private, SSM Museum, Istanbul, 2018.
 - [46] R. Jeremy Lent. *Patterning Instinct*. Prometheus Books, 2017. ISBN: 978-1633882935.
 - [47] Joseph Needham. *Moulds of Understanding - A Pattern of Natural Philosophy*. Allen Unwin, 1976. ISBN: 978-0049250116.
 - [48] A. N. Whitehead. *Process and Reality*. Macmillan USA, 1979. ISBN: 978-0029345702.
 - [49] Norbert Wiener. *The Human Use of Human Beings*. De Capo Press, 1988. ISBN: 9780786752263.
 - [50] Ilya Prigogine. *Non-Equilibrium Statistical Mechanics*. Interscience Publishers, 1962. ISBN: 978-0486815558.
 - [51] Scott Camazine. *Self Organization in Biological Systems*. Princeton University Press, 2015. ISBN: 978-0691116242.
 - [52] B. L. Partridge. “The Structure and Function of Fish Schools”. In: *Sci Am* 246.6 (1982), 114–.
 - [53] I. Aoki. “A Simulation Study on the Schooling Mechanism in Fish”. In: *B Jpn Soc Sci Fish* 48.8 (1982), pp. 1081–1088.
 - [54] J. T. Bonner. “Chemical Signals of Social Amebas”. In: *Sci Am* 248.4 (1983), 114–.
 - [55] P. Bak, C. Tang, and K. Wiesenfeld. “Self-Organized Criticality - an Explanation of $1/F$ Noise”. In: *Phys Rev Lett* 59.4 (1987), pp. 381–384.
 - [56] J. W. Barrett, H. Garcke, and R. Nurnberg. “Numerical computations of faceted pattern formation in snow crystal growth”. In: *Phys Rev E* 86.1 (2012).
 - [57] Alfonso Martinez Arias Lewis Wolpert Cheryll Tickle. *Principles of Development*. Oxford University Press, 2015. ISBN: 0199678136.

- [58] Irenaus Eibl-Eibesfeldt. “The Artist in the Scientist”. In: *Art Forms in Nature*. Munich: Prestel-Verlag, 1998.
- [59] Ernst Haeckel. *Kunstformen der Natur*. Leipzig and Vienna: Bibliographisches Institut, 1904.
- [60] Peter Grindrod. *Patterns and Waves, The theory and applications of reaction-diffusion equations*. Oxford University Press, 1991. ISBN: 019 8596766.
- [61] J. E. Pearson. “Complex Patterns in a Simple System”. In: *Science* 261.5118 (1993), pp. 189–192.
- [62] K. J. Lee et al. “Pattern-Formation by Interacting Chemical Fronts”. In: *Science* 261.5118 (1993), pp. 192–194.
- [63] A. Doelman, T. J. Kaper, and P. A. Zegeling. “Pattern formation in the one-dimensional Gray-Scott model”. In: *Nonlinearity* 10.2 (1997), pp. 523–563.
- [64] E. F. Keller and L. A. Segel. “Model for Chemotaxis”. In: *J Theor Biol* 30.2 (1971), 225–.
- [65] J. T. Oden and J. N. Reddy. *An Introduction to the Mathematical Theory of Finite Elements*. New York: Dover Publications, 2011. ISBN: ISBN-13: 978-0-486-46299-8.
- [66] Z. Q. Cai. “On the Finite Volume Element Method”. In: *Numer Math* 58.7 (1991), pp. 713–735.
- [67] S. Mazumder. *Numerical Methods for Partial Differential Equations: Finite Difference and Finite Volume Methods*. Elsevier, 2016. ISBN: 978-0128498941.
- [68] R. Eymard, T. Gallouet, and R. Herbin. *Finite Volume Methods*. Manuscript. 2019.
- [69] R. Vanselow. “Convergence analysis for the full-upwind finite volume solution of a convection-diffusion problem”. In: *J Math Anal Appl* 264.2 (2001), pp. 423–449. DOI: 10.1006/jmaa.2001.7677.

-
- [70] P. Deuring and R. Eymard. “L-2-Stability of a Finite Element - Finite Volume Discretization of Convection-Diffusion-Reaction Equations with Nonhomogeneous Mixed Boundary Conditions”. In: *Esaim-Math Model Num* 51.3 (2017), pp. 919–947. DOI: 10.1051/m2an/2016042.
 - [71] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids*. Oxford University Press, 1959. ISBN: 978-0198533689.
 - [72] E. Korach, S. Kutten, and S. Moran. “A Modular Technique for the Design of Efficient Distributed Leader Finding Algorithms”. In: *Acm T Progr Lang Sys* 12.1 (1990), pp. 84–101. DOI: Doi10.1145/77606.77610.
 - [73] S. Dolev, A. Israeli, and S. Moran. “Uniform Dynamic Self-Stabilizing Leader Election”. In: *Lecture Notes in Computer Science* 579 (1992), pp. 167–180.
 - [74] R. Ingram et al. “A leader election algorithm for dynamic networks with causal clocks”. In: *Distrib Comput* 26.2 (2013), pp. 75–97. DOI: 10.1007/s00446-013-0184-1.
 - [75] G. A. Ruz et al. “Dynamical and topological robustness of the mammalian cell cycle network: A reverse engineering approach”. In: *Biosystems* 115 (2014), pp. 23–32.
 - [76] K. Kaneko and T. Yomo. “Cell-Division, Differentiation and Dynamic Clustering”. In: *Physica D* 75.1-3 (1994), pp. 89–102.
 - [77] J. B. Hengeniuss et al. “Analysis of Gap Gene Regulation in a 3D Organism-Scale Model of the *Drosophila melanogaster* Embryo”. In: *Plos One* 6.11 (2011).
 - [78] I. Salazar-Ciudad. “Looking at the origin of phenotypic variation from pattern formation gene networks”. In: *J Biosciences* 34.4 (2009), pp. 573–587.
 - [79] A. G. Fisher. “Cellular identity and lineage choice”. In: *Nat Rev Immunol* 2.12 (2002), pp. 977–982.
 - [80] M. A. Shereshevsky. “Lyapunov Exponents for One-Dimensional Cellular Automata”. In: *J Nonlinear Sci* 2.1 (1992), pp. 1–8. DOI: Doi10.1007/Bf02429850.

- [81] P. Tisseur. “Cellular automata and Lyapunov exponents”. In: *Nonlinearity* 13.5 (2000), pp. 1547–1560. DOI: [Doi10.1088/0951-7715/13/5/308](https://doi.org/10.1088/0951-7715/13/5/308).
- [82] L. P. Hurd, J. Kari, and K. Culik. “The Topological-Entropy of Cellular Automata Is Uncomputable”. In: *Ergod Theor Dyn Syst* 12 (1992), pp. 255–265.
- [83] M. Finelli, G. Manzini, and L. Margara. “Lyapunov exponents versus expansivity and sensitivity in cellular automata”. In: *J Complexity* 14.2 (1998), pp. 210–233.
- [84] G. Manzini and L. Margara. “Attractors of linear cellular automata”. In: *J Comput Syst Sci* 58.3 (1999), pp. 597–610.
- [85] G. Cattaneo et al. “Ergodicity, transitivity, and regularity for linear cellular automata over $\mathbb{Z}(m)$ ”. In: *Theor Comput Sci* 233.1-2 (2000), pp. 147–164.
- [86] M. D’amico, G. Manzini, and L. Margara. “On computing the entropy of cellular automata”. In: *Theor Comput Sci* 290.3 (2003), pp. 1629–1646.
- [87] A. Lempel and J. Ziv. “Complexity of Finite Sequences”. In: *Ieee T Inform Theory* 22.1 (1976), pp. 75–81.
- [88] H. Zenil. “Compression-Based Investigation of the Dynamical Properties of Cellular Automata and Other Systems”. In: *Complex Syst* 19.1 (2010).
- [89] A. Daza et al. “Basin entropy: a new tool to analyze uncertainty in dynamical systems”. In: *Sci Rep-Uk* 6 (2016). DOI: [ARTN3141610.1038/srep31416](https://doi.org/10.1038/srep31416).
- [90] T. Eiter and H. Mannila. “Distance measures for point sets and their computation”. In: *Acta Inform* 34.2 (1997), pp. 109–133.
- [91] C. H. Bennett et al. “Information distance”. In: *Ieee T Inform Theory* 44.4 (1998), pp. 1407–1423.
- [92] A. Shreim et al. “Network analysis of the state space of discrete dynamical systems”. In: *Phys Rev Lett* 98.19 (2007).
- [93] Aristotle. *On the Generation of Animals, Book II*. Wikisource.org.
- [94] E. Mayr. “Recapitulation Reinterpreted - the Somatic Program”. In: *Q Rev Biol* 69.2 (1994), pp. 223–232. DOI: [Doi10.1086/418541](https://doi.org/10.1086/418541).

-
- [95] S. Kondo and T. Miura. “Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation”. In: *Science* 329.5999 (2010), pp. 1616–1620.
- [96] L. Marcon and J. Sharpe. “Turing patterns in development: what about the horse part?” In: *Curr Opin Genet Dev* 22.6 (2012), pp. 578–584.
- [97] K. W. Rogers and A. F. Schier. “Morphogen Gradients: From Generation to Interpretation”. In: *Annual Review of Cell and Developmental Biology, Vol 27* 27 (2011), pp. 377–407.
- [98] D. Summerbell and L. Wolpert. “Precision of Development in Chick Limb Morphogenesis”. In: *Nature* 244.5413 (1973), pp. 228–230.
- [99] V. French, P. J. Bryant, and S. V. Bryant. “Pattern Regulation in Epimorphic Fields”. In: *Science* 193.4257 (1976), pp. 969–981.
- [100] E. V. Entchev, A. Schwabedissen, and M. Gonzalez-Gaitan. “Gradient formation of the TGF-beta homolog Dpp”. In: *Cell* 103.6 (2000), pp. 981–991.
- [101] A. A. Teleman and S. M. Cohen. “Dpp gradient formation in the Drosophila wing imaginal disc”. In: *Cell* 103.6 (2000), pp. 971–980.
- [102] P. W. Ingham and A. P. McMahon. “Hedgehog signaling in animal development: paradigms and principles”. In: *Gene Dev* 15.23 (2001), pp. 3059–3087.
- [103] D. Stamatakis et al. “A gradient of Gli activity mediates graded Sonic Hedgehog signaling in the neural tube”. In: *Gene Dev* 19.5 (2005), pp. 626–641.
- [104] C. E. Chamberlain et al. “Notochord-derived Shh concentrates in close association with the apically positioned basal body in neural target cells and forms a dynamic gradient during neural patterning”. In: *Development* 135.6 (2008), pp. 1097–1106.
- [105] N. Balaskas et al. “Gene Regulatory Logic for Reading the Sonic Hedgehog Signaling Gradient in the Vertebrate Neural Tube”. In: *Cell* 148.1-2 (2012), pp. 273–284.
- [106] L. Wolpert. “Positional information and patterning revisited”. In: *J Theor Biol* 269.1 (2011), pp. 359–365.

- [107] A. D. Lander. “Morpheus unbound: Reimagining the morphogen gradient”. In: *Cell* 128.2 (2007), pp. 245–256.
- [108] M. Kerszberg and L. Wolpert. “Mechanisms for positional signalling by morphogen transport: a theoretical study”. In: *J Theor Biol* 191.1 (1998), pp. 103–114.
- [109] M. Ibanes and J. C. I. Belmonte. “Theoretical and experimental approaches to understand morphogen gradients”. In: *Mol Syst Biol* 4 (2008).
- [110] M. Kerszberg and L. Wolpert. “Specifying positional information in the embryo: Looking beyond morphogens”. In: *Cell* 130.2 (2007), pp. 205–209.
- [111] A. Eldar et al. “Self-enhanced ligand degradation underlies robustness of morphogen gradients”. In: *Dev Cell* 5.4 (2003), pp. 635–646.
- [112] A. Eldar, B. Z. Shilo, and N. Barkai. “Elucidating mechanisms underlying robustness of morphogen gradients”. In: *Curr Opin Genet Dev* 14.4 (2004), pp. 435–439.
- [113] J. B. A. Green and J. Sharpe. “Positional information and reaction-diffusion: two big ideas in developmental biology combine”. In: *Development* 142.7 (2015), pp. 1203–1211.
- [114] H. Meinhardt. “Pattern-Formation in Biology - a Comparison of Models and Experiments”. In: *Rep Prog Phys* 55.6 (1992), pp. 797–849.
- [115] H. Meinhardt. “Organizer and axes formation as a self-organizing process”. In: *Int J Dev Biol* 45.1 (2001), pp. 177–188.
- [116] E. Wieschaus. “Positional Information and Cell Fate Determination in the Early Drosophila Embryo”. In: *Curr Top Dev Biol* 117 (2016), pp. 567–579.
- [117] P. Muller et al. “Morphogen transport”. In: *Development* 140.8 (2013), pp. 1621–1638.
- [118] G. Vasilopoulos and K. J. Painter. “Pattern formation in discrete cell tissues under long range filopodia-based direct cell to cell contact”. In: *Math Biosci* 273 (2016), pp. 1–15.
- [119] J. Jaeger and J. Reinitz. “On the dynamic nature of positional information”. In: *Bioessays* 28.11 (2006), pp. 1102–1111.

-
- [120] R. Allena, J. J. Munoz, and D. Aubry. “Diffusion-reaction model for *Drosophila* embryo development”. In: *Comput Method Biomec* 16.3 (2013), pp. 235–248.
- [121] M. Mercker et al. “Beyond Turing: mechanochemical pattern formation in biological tissues”. In: *Biol Direct* 11 (2016).
- [122] J. Pascalie et al. “Developmental Design of Synthetic Bacterial Architectures by Morphogenetic Engineering”. In: *Acs Synth Biol* 5.8 (2016), pp. 842–861.
- [123] J. A. Izaguirre et al. “CompuCell, a multi-model framework for simulation of morphogenesis”. In: *Bioinformatics* 20.7 (2004), pp. 1129–1137.
- [124] T. M. Cickovski et al. “A framework for three-dimensional simulation of morphogenesis”. In: *Ieee Acn T Comput Bi* 2.4 (2005), pp. 273–288.
- [125] T. Sugiyama and T. Fujisawa. “Genetic-Analysis of Developmental Mechanisms in *Hydra* .1. Sexual Reproduction of *Hydra-Magnipapillata* and Isolation of Mutants”. In: *Dev Growth Differ* 19.3 (1977), pp. 187–200.
- [126] J. Wittlieb et al. “Transgenic *Hydra* allow in vivo tracking of individual stem cells during morphogenesis”. In: *P Natl Acad Sci USA* 103.16 (2006), pp. 6208–6211.
- [127] K. Khalturin et al. “Transgenic stem cells in *Hydra* reveal an early evolutionary origin for key elements controlling self-renewal and differentiation”. In: *Dev Biol* 309.1 (2007), pp. 32–44.
- [128] Tobias Lengfeld et al. “Multiple Wnts are involved in *Hydra* organizer formation and regeneration”. In: *Dev Biol* 330.1 (2009), pp. 186–199.
- [129] S. Munder et al. “Notch-signalling is required for head regeneration and tentacle patterning in *Hydra*”. In: *Dev Biol* 383.1 (2013), pp. 146–157.
- [130] C. Nussleinvohard and E. Wieschaus. “Mutations Affecting Segment Number and Polarity in *Drosophila*”. In: *Nature* 287.5785 (1980), pp. 795–801.
- [131] Kenneth I. Appel and Wolfgang Haken. “Every planar map is four colorable”. In: *Contemporary Mathematics*, xv (1989), 741 p.
- [132] L. Wolpert. “Positional Information and Pattern Formation”. In: *Curr Top Dev Biol* 117 (2016), pp. 597–608.

- [133] B. Green. *Until the End of Time: Mind, matter, and our search for meaning in an evolving universe*. Penguin Random House, 2020. ISBN: 978-0241295984.