Rule Derivation for Agent-Based Models of Complex Systems: Nuclear Waste Management and Road Networks Case Studies

By

Jorge Andrés García Hernández

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2018

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Ph.D. Ziad Kobti Professor and Director School of Computer Science, University of Windsor.
Supervisor(s)	Ph.D. Kumaraswamy Ponnambalam Professor, Systems Design Engineering, University of Waterloo.
Internal Member	Ph.D. Michele Bristow Adjunct Assistant Professor, Systems Design Engineering, University of Waterloo.
Internal Member	Ph.D. Shi Cao Assistant Professor, Systems Design Engineering, University of Waterloo.
Internal-external Member	Ph.D. Mark Crowley Assistant Professor, Electrical and Computer Engineering, University of Waterloo.

Author’s declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis explores the relation between equation-based models (EBMs) and agent-based models (ABMs), in particular, the derivation of agent rules from equations such that agent collective behavior produces results that match or are close to those from EBMs.

This allows studying phenomena using both approaches and obtaining an understanding of the aggregate behavior as well as the individual mechanisms that produce them. The use of ABMs allows the inclusion of more realistic features that would not be possible (or would be difficult to include) using EBMs.

The first part of the thesis studies the derivation of molecule displacement probabilities from the diffusion equation using cellular automata. The derivation is extended to include reaction and advection terms. This procedure is later applied to estimate lifetimes of nuclear waste containers for various scenarios of interest and the inclusion of uncertainty.

The second part is concerned with the derivation of a Bayesian state algorithm that consolidates collective real-time information about the state of a given system and outputs a probability density function of state domain, from which the most probable state can be computed at any given time. This estimation is provided to agents so that they can choose the best option for them. The algorithm includes a diffusion or diffusion-like term to account for the deterioration of information as time goes on. This algorithm is applied to a couple of road networks where drivers, prior to selecting a route, have access to current information about the traffic and are able to decide which path to follow.

Both problems are complex due to heterogeneous components, nonlinearities, and stochastic behavior; which make them difficult to describe using classical equation models such as the diffusion equation or optimization models. The use of ABMs allowed for the inclusion of such complex features in the study of their respective systems.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Ponnu, for giving me the opportunity to pursue a PhD is Systems Design Engineering, the knowledge shared, the continuous support, and for setting some interesting problems from which I learned a lot.

I would also like to express my gratitude to my Committee Members for all the time, commitment, and insightful commentaries: Prof. Ziad Kobti, the External Examiner, from the University of Windsor; Prof. Mark Crowley, the Internal-External Examiner, from the Department of Electrical and Computer Engineering; and Prof. Michele Bristow and Prof. Shi Cao, Internal Examiners, from the Department of Systems Design Engineering.

During my time at the University of Waterloo, I've met some amazing people. Christel & Guido Weber have welcomed me countless times in their house for wine-tasting classes and delicious dinners. Thank you for all the experiences shared and the warmth of your company. I'd like to acknowledge my peers at the Design Optimization Under Uncertainty Group for sharing thoughts, ideas, an occasionally a pint of beer: Shankai, Mythreyi, Vimala, Ahmed and many others. Thanks to everyone who has received me with the warmth of a smile.

The support of my family has been essential in the termination of this thesis; my wife Den, who has been a source of continuous motivation; my father and my siblings George and Mónica, who have always encouraged me to pursue my goals. Big motivation has come from three schnauzers that were and have been a source of joy: Oliver, Jozen, and Kyla.

This thesis has been possible thanks to the financial support of the following institutions: Consejo Nacional de Ciencia y Tecnología (CONACYT) through the scholarship CONACYT-Ciudad de Mexico; Beca SEP Complemento; my family; and the NSERC CRD and the Nuclear Waste Management Organization.

Dedication

To my family.

Table of contents

EXAMINING COMMITTEE MEMBERSHIP II

AUTHOR'S DECLARATION III

ABSTRACT IV

ACKNOWLEDGEMENTS V

DEDICATION VI

LIST OF FIGURES XI

LIST OF TABLES XIV

LIST OF ACRONYMS XVI

LIST OF SYMBOLS XVII

CHAPTER 1. INTRODUCTION 1

MOTIVATION 1
NUCLEAR WASTE MANAGEMENT 2
ROAD NETWORKS 3
RESEARCH OBJECTIVES 3
ORGANIZATION 3

PART I 5

CHAPTER 2 CELLULAR AUTOMATA AND NUCLEAR WASTE MANAGEMENT 6

2.0 LITERATURE REVIEW 7
2.1 NUCLEAR WASTE MANAGEMENT 8
2.1.1 Problem Description 8
2.1.2 Canadian DGR Design Concept 8
2.1.3 Normal Evolution Scenario 9
2.1.4 Definition of failure 10
2.1.5 Copper corrosion 10
2.1.6 Groundwater composition 11
2.1.7 Bentonite Density 11
2.2 CELLULAR AUTOMATA 12
2.2.1 Definition and development 12
2.2.2 Neighborhood 12
2.2.3 Rules of behavior 13
2.2.4 Collective behavior 15
2.2.5 Conway's Game of "Life" 15
2.2.6 Relation between CA and PDEs 16
2.2.7 Chemical Reaction 17
2.2.8 Diffusion Equation 17
2.2.9 Wave Equation 19
2.2.10 Reaction Diffusion Equation 20

CHAPTER 3 DIFFUSION USING CELLULAR AUTOMATA 21

3.1 DIFFUSION 22
3.2 DISCRETIZED DIFFUSION 22

3.3 Probability of Displacement 23
3.4 Initial and Boundary Conditions 26
3.5 Relation to Brownian Motion 27
3.6 Relation to Markov Chains 28
3.7 Particle Displacement 31
3.8 Diffusion in Mixed Mediums 33
3.9 Reaction Probability 34
3.10 Reaction-Diffusion Probability 37
3.11 Advection Term 38
3.12 Consistency, Order, Stability, and Convergence 38

CHAPTER 4 ESTIMATING LIFETIMES OF NUCLEAR WASTE CONTAINERS 42

4.0 Assumptions and limitations 43
4.1 Sulphate-Reducing Bacteria 43
4.1.1 Reaction Rate 44
4.1.2 Optimal Reduction Rate Coefficient 45
4.1.3 Sulphide Production 47
4.1.4 Density-Dependent Reaction Rate 49
4.2 Reaction – Diffusion Model 51
4.2.1 Cellular Automata Model 52
4.2.2 Diffusion 53
4.2.3 Reaction 54
4.2.4 Reaction-Diffusion 54
4.2.5 Initial Conditions 54
4.2.6 Boundary Conditions 56
4.3 Lifetime Calculation 57
4.3.1 Sulphide Flux 57
4.3.2 Corrosion depth 58
4.3.3 Determination of the lifetime of canisters 58
4.4 Verification and Validation 59
4.4.1 Validation Scenario I 59
4.4.2 Validation Scenario II 60
4.5 Scenarios 61
4.6 Results 62
4.6.1 Scenario I. SRB active in clays and rock interface 62
4.6.2 Results Scenario II. SRB active in rock interface 65
4.6.3 Results Scenario III. SRB active, homogeneous densities 68
4.7. Sensitivity Analysis 70
4.7.1 Scenario I. SRB active in clays and rock interface 71
4.7.2 Scenario II. SRB active only at rock interface 72
4.7.3 Scenario III. SRB active, homogeneous densities 73
4.8 Part I Conclusions 73

PART II 76

CHAPTER 5 AGENT-BASED MODELING 77

5.0 Literature Review 78
5.1 Agent-Based Modeling 81
5.1.1 Properties of agents 81

5.1.2 ABM implementation ... *82*
5.2 AGENT-BASED MODELS VS. EQUATION-BASED MODELS ... 82
5.2.1 ABM vs. ODEs: A Microeconomics Example ... *83*
5.3 ROAD NETWORKS ... 88
5.3.1 Traveling Time ... *88*
5.3.2 Road Segment Speed ... *89*
5.3.3 Road density ... *90*

CHAPTER 6 BAYESIAN STATE ESTIMATION USING COLLECTIVE INFORMATION ... 91
6.1 DEFINITIONS AND ASSUMPTIONS ... 92
6.2 AGENT PERCEPTION ... 93
6.3 ADDITION OF NEW INFORMATION ... 94
6.4 PROCESS OF FORGETTING INFORMATION ... 99
6.4.1 Diffusion-like process ... *99*
6.4.2 Diffusion process ... *101*
6.5 MOST PROBABLE STATE ... 103
6.6 NUMERICAL IMPLEMENTATION ... 106
6.6.1 Diffusion-like Process Implementation ... *106*
6.6.2 Diffusion Process Implementation ... *108*
6.6.3 Normalizing a Function ... *110*
6.7 AGENT LEARNING ... 110

CHAPTER 7 TRAVELING TIME ESTIMATION IN ROAD NETWORKS ... 113
7.0 ASSUMPTIONS AND LIMITATIONS ... 114
7.1 ROAD NETWORKS ... 114
7.1.1 Road network I (RN-I) ... *114*
7.1.2 Road network II (RN-II) ... *115*
7.1.3 Road network III (RN-III) ... *116*
7.2 ROAD NETWORK OPTIMIZATION MODEL ... 116
7.3 AGENT-BASED MODEL ... 117
7.3.1 Pseudo code ... *118*
7.4 DERIVATION OF AGENT RULES ... 119
7.4.1 Example I. Solving RN-I ... *120*
7.4.2 Example II Solving RN-II ... *122*
7.5 AGENT DECISION-MAKING ... 124
7.6 BAYESIAN STATE ESTIMATION IMPLEMENTATION ... 125
7.6.1 Input Values ... *126*
7.6.2 Comparison Bayesian vs. True values for RN-I ... *127*
7.6.3 Comparison between different types of agents ... *128*
7.6.4 Implementing Learning ... *129*
7.6.5 Implementing RN-III ... *130*
7.7 PART II CONCLUSIONS ... 132

CHAPTER 8. SUMMARY AND CONCLUSIONS ... 134
8.1 SUMMARY ... 134
8.2 CONTRIBUTIONS ... 135
8.3 CONCLUSIONS ... 135
8.4 FUTURE WORK ... 137

REFERENCES ... 138

APPENDIX A 144

APPENDIX B 149

APPENDIX C.. 151

APPENDIX D 154

APPENDIX E.. 156

List of figures

Figure 2.1 Canadian DGR design (Source: www.nwmo.ca Image by NWMO) 8

Figure 2.2 UFC dimensions 2506×556×556 mm (Source: [16] Image by NWMO) 9

Figure 2.3 Placement room for UFCs (Source: [16] Image by NWMO) 10

Figure 2.4 Examples of 2D cellular automata 12

Figure 2.5 Definitions of neighborhood for a square 2D CA (Image from Wolfram|Alpha) 13

Figure 2.6 Example of a behavior rule (Image from Wolfram|Alpha) 14

Figure 2.7 Plot of the evolution of a 1D CA from a single black cell 14

Figure 2.8 Plot of the evolution of a 1D CA from a random seed 14

Figure 2.9 Examples of different classes CA 15

Figure 2.10 Conway's Game of Life (blue = alive) random seed 16

Figure 2.11 Biochemical-Oxygen demand reaction using cellular automata 17

Figure 2.12 2D diffusion using cellular automata with a constant circle source 18

Figure 2.13 3D diffusion using cellular automata with a constant sphere source 18

Figure 2.14 2D cellular automata describing the wave equation 19

Figure 2.15 Fisher-Kolmogorov equation using 2D cellular automata 20

Figure 3.1 Discretized and continuous solute diffusion over a discretized space 23

Figure 3.2 Discretization of the domain space 23

Figure 3.3 Transition from states at time t to a single state at t+1 24

Figure 3.4 Neighbor cells (orange) of a reference cell (blue) 25

Figure 3.5 Boundary cells (red) in a 1D and 2D cellular automata 27

Figure 3.6 Adjacent cells with different diffusivities 33

Figure 4.1 Bisection iterative approximation to zero and convergence plot 46

Figure 4.2 SRB-driven chemical reaction 48

Figure 4.3 Linear and exponential adjustment of reaction rate 50

Figure 4.4 Convergence and error plot for parameter C_E 51

Figure 4.5 1D Cellular automata 53

Figure 4.6 2D Cellular automata (image not to scale) 53

Figure 4.7 Assumed sulphate content in groundwater 56

Figure 4.8 Histogram of UFC lifetimes for Validation Scenario I 60

Figure 4.9 Histogram of UFC lifetimes for Validation Scenario II (1D models) 61

Figure 4.10 Histogram of UFC lifetimes for Validation Scenario II (2D CA) 61

Figure 4.11. Profile of steady state concentration (distance from boundary) 63

Figure 4.12. Sulphide concentration at the left boundary and flux for at the right boundary for Scenario I 63

Figure 4.13 Scenario I lifetimes for 1D 64

Figure 4.14 Frame of concentrations at steady state: (a) sulphate, (b) sulphide, (c) ln(sulphide) 65

Figure 4.15 Profile of steady state concentration (distance from boundary) 66

Figure 4.16 Sulphide concentration at the left boundary and flux for at the right boundary for Scenario II 66

Figure 4.17 Scenario II lifetimes for 1D 67

Figure 4.18 Profile of steady state concentration (distance from boundary) 68

Figure 4.19 Sulphide concentration at the left boundary and flux for at the right boundary for Scenario III 69

Figure 4.20 Scenario III lifetimes for 1D 70

Figure 4.21 Lifetimes for Scenario I 71

Figure 4.22 Lifetimes for Scenario II 72

Figure 4.23 Lifetimes for Scenario III 73

Figure 5.1 Single *buyer*'s demand and single *seller*'s supply 84

Figure 5.2 Linear model: (a) Evolution of price, (b) Evolution of total demand and supply (c) Evolution of demand and supply with random terms 85

Figure 5.3 Nonlinear model: (a) Evolution of price, (b) Evolution of total demand and supply (c) Evolution of demand and supply with random terms 86

Figure 5.4 Linear model: Solution to the ODE system 87

Figure 5.5 Nonlinear model: Solution to the ODE system 87

Figure 5.6 BRP travel time function with *cap = 20* [vehicle] 89

Figure 5.7 Road segment speed function with *cap = 20* [vehicles] 90

Figure 6.1 Diagram of an agent's perception 93

Figure 6.2 Diagram of the evolution of PStk 93

Figure 6.3 Evolution of P(S) as new information is incorporated $\pi_A=1/2$ 97

Figure 6.4 Mesh plot of P(S) as observations are incorporated $\Lambda=\{5,5,5,5,5\}$ (left) and $\Lambda=\{3,4,5,6,7\}$ (right) , $\pi_A=1/2$ 97

Figure 6.5 Evolution of P(S) as new information is incorporated $\pi_A=1/20$ 98

Figure 6.6 Mesh plot of P(S) as observations are incorporated $\Lambda=\{5,5,5,5,5\}$ (left) and $\Lambda=\{3,4,5,6,7\}$ (right) , $\pi_A=1/20$ 98

Figure 6.7 Commutative property in PSt preserved when $\pi_A=1$ 99

Figure 6.8 The diffusion-like process increases values lower than 1/(b-a) and decreases values higher than that 100

Figure 6.9 Evolution of PSt using the diffusion-like process of eq. (6.7) 101

Figure 6.10 Diffusion spreads the function 102

Figure 6.11 Evolution of PSt using the diffusion process of eq. (6.9) 102

Figure 6.12 Evolution of pdf using the diffusion-like process 104

Figure 6.13 Contour plot of the solution using diffusion-like process....................104

Figure 6.14 Evolution of the pdf using the diffusion process....................105

Figure 6.15 Contour plot of the solution using diffusion process....................105

Figure 6.16 (a) Plot the parameter values (b) Error^2 plot....................112

Figure 7.1 Road network I....................115

Figure 7.2 Road network II....................115

Figure 7.3 Road network II (RN-II)....................116

Figure 7.4 ABM following rules: (a) Min Grad(Z_1), (b) Min Grad(Z_2), (c) Min Grad(Z_3)....................123

Figure 7.5 Road Network I using Bayesian State Estimation....................127

Figure 7.6 Probability density functions of roads (a) Bayesian diffusion (b) Bayesian diffusion-like....................128

Figure 7.7 Evolution of parameter values for the learning online procedure from estimates (a) BD, (b) BDL....................130

Figure 7.8 Simulation of RN-III (source nodes in blue; sink in red)....................130

Figure 7.9 Probability density functions of roads using Bayesian with diffusion....................131

Figure 7.10 Probability density functions of roads using Bayesian with diffusion-like....................132

List of tables

Table 2.1 Estimation of groundwater composition 11

Table 2.2 Properties of Bentonite clays 11

Table 4.1 Composition of groundwater before and after tests [mg/L] 44

Table 4.2 Comparison of optimal values 47

Table 4.3 Comparison of reduction rates of SRB in bentonite MX-80 50

Table 4.4 Water content in 1 [m^3] of clay 55

Table 4.5 Initial amounts per cell/patch [mg/L] 55

Table 4.6 Validation scenarios for CA model 59

Table 4.7 Lifetimes for Validation Scenario I [year] 59

Table 4.8 Lifetimes for Validation Scenario II [year] 60

Table 4.9 Lifetimes for Scenario I [year] 63

Table 4.10 Lifetimes for Scenario I, groundwater uncertainty [year] 64

Table 4.11 Lifetimes for Scenario II [year] 65

Table 4.12 Lifetimes for Scenario II, groundwater uncertainty [year] 67

Table 4.13 Lifetimes for Scenario III [year] 68

Table 4.14 Lifetimes for Scenario II, groundwater uncertainty [year] 69

Table 5.1 Input values for the microeconomics problem 85

Table 6.1 Results of Example 6.4 112

Table 7.1 Input values for Example I 121

Table 7.2 Value of Objective functions for Example I 121

Table 7.3 Optimal flow for Example 7.1 fI, fII, fIII in vehicles 121

Table 7.4 Optimal path times for Example I T(fI), T(fII), T(fIII) in time units 122

Table 7.5 Value of Objective functions for Example II 123

Table 7.6 Optimal flow for Example II fI, fII, fIII in % of F 123

Table 7.7 Optimal path times for Example II T(fI), T(fII), T(fIII) in time units 123

Table 7.8 Input values for network RN-I 126

Table 7.9 Input values the Bayesian State Estimation 126

Table 7.10 Objective functions for RN-I with Bayesian State Estimation 127

Table 7.11 Average flows after 500 time steps for RN-I fI, fII, fIII in vehicles 128

Table 7.12 Error between true values and estimates for RN-I 128

Table 7.13 Average path time per agent for RN-I 129

Table 7.14 Learning parameters after 150 trips 129

Table 7.15 Average minutes on network per driver for RN-III 130

Table 7.16 Average vehicles on network for RN-III 131

Table 7.17 Average flows per path in fI, fII, fIII, fIV in % (rounded) for RN-III 131

Table 7.18 Error between true values and estimates for RN-III 131

List of acronyms

ABM	Agent-based model.
BC	Boundary condition.
BD	Bayesian state estimation using diffusion process.
BDL	Bayesian state estimation using diffusion-like process.
BPR	Bureau of Public Road.
CA	Cellular automata.
Cu	Copper.
DGR	Deep geological repository.
EQB	Equation-based model.
GF	Gapfill, bentonite filling the gap between host rock and buffer boxes.
H^+	Hydron.
H_2	Hydrogen gas.
HCB	Highly compacted bentonite.
HS^-	Bisulphide.
IFD	Implicit finite difference.
MINLP	Mixed-integer nonlinear program.
MSE	Mean sure error.
NWMO	Nuclear Waste Management Organization.
O_2	Oxygen.
ODE	Ordinary differential equation.
PDE	Partial differential equation.
RN-I	Road network I.
RN-II	Road network II.
RN-III	Road network III.
S(s)	Sulfur.
SAE	Sum of absolute errors.
SD	Standard deviation.
SO_4^{2-}	Sulphate.
SRB	Sulphate-reducing bacteria.
SSE	Sum of squared errors.
SSLE	Sum of squared log errors.
UFC	Used fuel container.

List of symbols

$[SO_4]_{Gw}$	Sulphate concentration in the groundwater [mg/L].
$\boldsymbol{P}_{cond}{}^{i+1}$	Vector of conditional probability over a discretized state space after receiving i+1 observations.
$\{\mathcal{S}_t\}$	State evolution of the subsystem of interest over time.
$\hat{s}_t$	Most probable state at time t.
$\dot{\varphi}^{(.)}$	Time derivative of the concentration of chemical species (.) [mol/(L t)] or [mg/(L t)].
$\hat{\varphi}_i^{(SO_4)}$	Auxiliary decision variable of sulphate concentration in the reduction rate optimization problems [mg/L].
d_{Corr}^t	Corrosion depth per unit of time [m/t].
$d_i^{(SO_4)}$	Data point i of sulphate concentration in the reduction rate optimization problems [mg/L].
$\eth_i$	Normalized distance of path i.
$E_{\Delta x}[.]$	Expected value with respect to x-axis.
$E_{\Delta y}[.]$	Expected value with respect to y-axis.
$E\left[\varphi_{i\,j}^t\right]_{i,j\in N_{cell}}$	Expected value of the concentration at cell ij for the next time step t+1.
$f_i^{s\,z}$	Flow of path i with source s and sink z.
$g_x(.)$	Function at the boundary using Robin boundary condition.
$J^t(L_x)$	Sulphide flux at the boundary x = Lx [mg/(m^2 t)].
$k_A^{(i)}$	Auxiliary variable in the optimization problem to adjust the reduction rate [1/t].
$k_D^{(i)}$	Reduction rate as reported in the experiment used in the optimization problem to adjust the reduction rate [1/t].
$\mathcal{L}_i$	Loss function for path i.
$N_{HS^-}^t$	Amount of sulphide transferred to the copper layer [mol/t].
$p_{\pm i}$	Probability of moving one cell ahead or backwards for the next time step.
$p_{\pm j}$	Probability of moving up or downwards for the next time step.
p_0	Probability of remaining on the same cell for the next time step.
p_I	Probability of moving in between two cells with different diffusivity.
$\boldsymbol{P}_T$	Transition probability matrix describing diffusion.
$\check{\mathcal{S}}$	Linear adjustment to the estimations received by the agent.
$\mathcal{U}_i$	Utility of path i.
Λ_t	Set of observations received at time t.
τ_{Corr}	Total corrosion during the simulation [m].
$\varphi^{(.)}$	Concentration of chemical species (.) [mol/L] or [mg/L].
$\varphi_C^{(HS^-)}$	Sulphide concentration reacted with copper in the experiment to determine χ, [mg/L].
$\varphi_G^{(HS^-)}$	Sulphide concentration measured in the experiment to determine χ, [mg/L].

φ_I	Concentration at the interface of two cells.
$\varphi_N^{(HS^-)}$	Sulphide concentration not reacted in the experiment to determine χ, [mg/L].
$\boldsymbol{\varphi_t}$	Vector of concentrations at time t.
$\varphi_{xx}^{(.)}$	Second derivative with respect to x of chemical species (.).
$\mathcal{A}$	Set of agents that share information to a collective system and have access to collective estimations.
a_0	Initial concentration of sulphate in the solution of the reduction ODE and the reduction rate optimization problem [mg/L].
A_{Corr}	Area of the canister affected by corrosion.
a_R	Parameter of the Robin boundary condition.
$\mathcal{B}$	Set of agents that do not share any information nor have access to it.
b_R	Parameter of the Robin boundary condition.
cap_k	Capacity of road k in [vehicle/t].
C_E	Parameter of the reduction adjustment function.
D	Total demand in example of Chapter 5.
d(.)	Individual demand in example Chapter 5.
d_{Corr}	Depth of corrosion [m].
D_e	Effective diffusivity or diffusion coefficient [m^2/s].
d_k	Distance of road segment k.
d_{max}	Maximum individual demand in example Chapter 5.
D_p	Diffusion coefficient of probability.
D_x	Diffusion along x-axis.
D_y	Diffusion along y-axis.
$E\left[\varphi_{t,\ i,j}^{(.)}\right]_{i,j\in N_{cell}}$	Expected value of the concentration at cell ij for the next time step t+1 and for chemical (.) [mg/L].
F	Total flow on the road network.
f_{HS}	Stoichiometry factor of conversion [mol/mol].
$F^{s\ z}$	Total flow from source s to sink z.
H(.)	Differential entropy.
$k_A(\rho_d)$	Adjusted reaction coefficient as a function of dry density [1/t].
k_R	Unadjusted reaction coefficient [1/t].
k_D	Rate coefficient from experiment [1/t].
Lifetime	Lifetime of the canister [yr].
L_x	Length of the x-axis.
L_y	Length of the y-axis.
$\mathbf{M}$	Square matrix describing the diffusion of information.
M_{Cu}	Molar mass of copper [g/mol].
M_{HS}	Molar mass of sulphide [g/mol].
N_{cell}	Neighbor cells of a given cell, including itself.
N_{HS}	Amount of sulphide [mol].
n_t	Noise component at time t.
p	Unitary price of the good in example Chapter 5.
$P\left(\mathcal{S}_t{}^k\right)$	Probability density function of the state of subsystem $\mathcal{S}$ at time t, after receiving k observations.
p_L	Probability of displacement on the cell to the left.

p_{max}	Maximum unitary price in example Chapter 5.
p_R	Probability of displacement on the cell to the right.
$R^{(.)}$	Reaction term of chemical species (.).
S	Total supply in example of Chapter 5.
$\mathcal{S}$	State describing the subsystem of interest.
s(.)	Individual supply in example Chapter 5.
s_{max}	Maximum individual supply in example Chapter 5.
t	Time units in [s], [min], [month] or [yr].
t_f	Free flow travel time for a road segment in [t/segment] of [t/km].
T_k	BRP travel time for segment k in [t/segment] of [t/km].
u	Euclidian distance in the x and y Cartesian plane.
$V\left[\varphi_{i\,j}^{t}\right]$	Variance of the concentration at cell ij and time t.
v_f	Free flow speed in [segment/t] or [km/t].
v_k	Speed of road k in [segment/t] or [km/t].
V_x	Advection component on the x-axis.
w_k	Flow or volume of vehicles on segment k in [vehicle/t].
x	Space axis.
y	Space axis.
$\mathcal{Z}$	Set of autonomous agents
Z_i	Objective function i.
α_0	Parameter of linear bias correction.
α_1	Parameter of linear bias correction.
β_0	Parameter of the BRP travel time function.
β_1	Parameter of the BRP travel time function.
γ	Speed of diffusion-like process.
θ	Penalizing factor in [cost/(bit*vehicle)].
κ_{HS}	Mole conversion coefficient from sulphate to sulphate [unitless].
μ	Degree of selfishness in [cost*t/vehicle].
ξ	Speed of adjustment using gradient descent.
π_A	Speed of assimilation of new information.
ρ_{Cu}	Density of copper [kg/m^3].
ρ_D	Dry density of the clay [kg/m^3].
ρ_{jam}	Maximum density of road k.
ρ_k	Density of road k.
ρ_S	Specific density of MX-80 bentonite clay.
ρ_{water}	Density of water [kg/m^3].
σ_n	Standard deviation of the noise component.
τ	Time interval to derive the diffusion equation.
χ	Efficiency coefficient in the reduction of sulphate to sulphide [unitless].
ψ	Generic variable used to solve examples.
ω	Speed of price change in example Chapter 5.

Chapter 1. Introduction

Motivation

There are systems whose adequate performance have significant impacts on societies, e.g., transportation networks, economic markets, waste treatment; or whose understanding are essential for the preservation of life either human or otherwise, e.g., water pollution, carbon cycle, soil erosion. Having models that accurately describe them becomes important for designing, optimizing, or controlling such systems; but the validity of the solution largely depends on the accuracy and assumptions of the models. However, those systems are complex due to one or more of the following features

- Large number of elements,
- Heterogeneous components,
- Larger number of relations/interactions,
- Nonlinearity,
- Stochastic behavior,
- Contradicting goals among elements,

which makes them difficult to track. In general, there are two approaches to deal with these kind of systems: 1) simplify the real system into a more tractable one and solve it using an equation-based model (EBM) that describes the aggregate behavior, or 2) use a more sophisticated model like agent-based models (ABMs) that incorporates complex features as the ones listed above and describes the micro state behavior.

EBMs are useful to predict aggregate behavior and, in the case of differential equations, they are also concise and elegant. However, they often rely on assumptions such as homogeneity, continuity, or rationality that restrict the cases that can be studied by them. The treatment of stochastic components increases notably the difficulty of any equation-based solution procedure, so much that entire field branches have been created specifically to study probabilistic cases, such as stochastic programming or stochastic differential equations, with particular methods of solution. Another complication is the analyst's time invested in creating and verifying the code for solving EBMs, in particular, expanding the model may complicate significantly the script, e.g., expanding a partial differential equation to include two state variables or two dimensions.

The second approach to deal with complex systems is the use of ABMs. They offer an alternative paradigm where systems arise from the collective interaction of multiple agents. This bottom-up approach allows for the inclusion of multiple features such as heterogeneity, nonlinear responses, etc. The introduction of stochastic parameters does not increase significantly the computational cost of solving the model than that of deterministic parameters. Another advantage is the analyst's time require to expand the model, since many ABMs are coded in a modular form, it simplifies the addition or verification of new agents or methods. However, ABMs are computationally expensive to run, require the calibration of many parameters, and because of its stochastic nature, it would require the execution of multiple runs to create outputs statistically significant to describe the system.

The work developed in this thesis aimed at creating a bridge between both approaches, that is, from EBMs that describe the aggregate behavior of a system of interest, find the agent-based behavior that collectively will give rise to it. Having a way to relate macrostates to microstates allows for creating an ABM that is consistent globally to an EBM, whose agent rules can be expanded to include more complicated behavior, and then study how that translates to the overall behavior. This procedure creates an ABM that is grounded on an EBM.

Two cases are studied: 1) the estimation of lifetimes of nuclear waste containers and 2) the effect on drivers of using real-time travel times estimates in road networks. Both cases are similar in that they have heterogeneous elements whose behavior produce nonlinearities, they have stochastic parameters (or decisions in the case of road networks), and the aggregate system is sensitive to local interactions. They are different in that molecules are not goal-oriented as drivers are, and the goal of the nuclear waste problem is to describe how chemicals interact whereas that of the road network problem is to find a way to nudge drivers into an efficient behavior, i.e., to prescribe certain behavior.

Nuclear Waste Management

This problem is concerned with the calculation of lifetimes of nuclear waste containers that may potentially be affected by sulphide corrosion. The EBM that governs the movement of the chemicals and the reduction of sulphate to sulphide is the reaction-diffusion equation.

The ABM method developed may be implemented in two manners: 1) as molecule agents representing the chemicals as they move throughout a discretized medium, or 2) as cellular automata representing the solution/medium with state variables representing the solutes as they move; in either case, agents behave autonomously from the rest.

The goal of this ABM is exploratory of scenarios of interest that may describe different conditions under which the containers will be placed for permanent storage, and descriptive of the local molecule interaction and the aggregate behavior produced by it.

Road Networks

This section studies the effect of having drivers that have access to travel times estimates before the make a decision about the path to follow to arrive to their destination, and how that affects their individual decision-making and the aggregate efficiency of the network. Those estimates are calculated from measuring the speed of current drivers along road segments. Therefore, an algorithm was created to update information and to account for the passing of time.

Drivers are represented by autonomous agents that make simple and rational decisions in regard to the path to follow (minimal-time path or minimal-gradient-time path). The agent decision rule for route selection was derived from a standard optimization model for road networks. Although it is possible to expand the sophistication of the driver decision models, this thesis is limited to simple decision models.

The goal of this ABM is as an exploratory study about the benefit of using current travel time estimates, and to offer a possible implementation for nudging drivers into a more efficient configuration, i.e., a prescriptive behavior.

Research Objectives

The main objective of this thesis is to show the capabilities of agent-based models in representing and solving systems described with partial differential equations or optimization models by deriving agent rules of behavior that make agents behave collectively in a similar fashion as the descriptions obtained from such equation models.

In particular this thesis aimed at

- Building an agent-based model to describe the diffusion of chemical species with space-dependent parameters, heterogeneity of materials, and reaction terms while keeping the model simple to solve and explore.
- Developing a Bayesian algorithm to gather information from various agents and produce probability density functions describing the state of traffic networks.

Organization

The first part of the thesis outlines the problem of estimating lifetimes of used fuel containers and is divided in three chapters:

Chapter 2. Cellular automata and nuclear waste management. Here relevant literature is shown as well as a background review on cellular automata and the long-term nuclear waste solution design. Some examples of how to solve partial differential equations using cellular automata are presented.

Chapter 3. Diffusion using cellular automata. In this chapter, the derivation of displacement probabilities displacement is described as well as the inclusion of reaction and advection terms.

Chapter 4. Estimating lifetimes of nuclear waste containers. In this chapter, the ideas developed in Chapter 3 are applied to the problem of estimating lifetimes of nuclear waste containers.

The second part studies the use of real-time travel time estimates and its effect on route selection decisions for drivers on road networks; it is divided in three chapters:

Chapter 5. Agent-based modeling. This chapter contains a literature and background review on agent-based models and the use of information to improve decision-making.

Chapter 6. Bayesian state estimation using collective information. In this chapter, the algorithm to collect current observations from by agents is made, as well as the computation of the probability density function describing the current state. It is also described the process of forgetting information as it becomes obsolete over time.

Chapter 7. Traveling time estimation on road networks. This chapter contains the application of the algorithm developed in Chapter 6 to the problem of path selection for drivers wanting to go from a source to a destination node.

Chapter 8. Summary and Conclusions. This chapter presents a summary of the work done and highlights the contribution of the thesis. It also presents future directions of research.

PART I

Chapter 2 Cellular Automata and Nuclear Waste Management

INTRODUCTION

This chapter provides a literature review on cellular automata (CA) for diffusion or reaction diffusion systems, and the description its basic features and rules of behavior. Then, the chapter presents the deep geological repository solution for nuclear waste disposal and describes its main design features. A failure definition is provided, that will later be used for lifetime calculation. The corrosion reaction is presented. Finally, the relation between CA and partial differential equations (PDEs) is presented as well as some examples of typical PDEs solved using CA.

Cellular automata models in this chapter were created using Wolfram|Alpha Pro for Students® or Netlogo®.

2.0 Literature Review

The use of agent-based models (ABM), such as cellular automata, to solve ordinary or partial differential equations has been motivated by the need for including a higher level of detail into components' behavior that would be analytically difficult to solve deriving exact solutions or what it could be achieved using computationally expensive numerical methods [1] – [5]. Some examples of diffusion or reaction diffusion systems are the following.

Scalise and Schulman [6] used a reaction-diffusion system to derive a CA set of rules using logic gates. Their aim was to show that DNA networks may behave like a CA and potentially be programmed to sense and respond to dynamic signals, which could be the basis of intelligent biomaterials. However, the derivation they found is based on Boolean logic and requires at least 29 logic gates, making it difficult to implement. They found it is equivalent to *rule 110* or *60* of Wolfram's elementary CA rules.

Odagiri and Takatsuka [7] compared bacterial proliferation models using PDEs and CA. They derived diffusion coefficients from probabilities of movement, and probabilities of reaction from the chemical reactions. They found that CA exhibited a more detailed behavior due to its stochastic nature, which could be mimicked with stochastic differential equations.

Another example of diffusion using ABM is the work of Azimi et al. [8]. They developed a 3D model for the cytoskeletal diffusion using probabilities of movement. In addition to that, they introduced a crowding behavior that further limits the spreading of molecules down the gradient of concentration. In this work, molecules are the agents.

Kawamata et al. [9] also explored DNA chemical networks and derived rules for a simple cellular automaton. Tang and Benett [10] used diffusion for a non-biological system, which is the spatial spreading of opinions using GPUs. Faber et al. [11] applied diffusion to the demand of technologies in the Netherlands using ABM.

The problem of computational complexity in CA may be solved with parallel computation, especially for stochastic CA. Bandman [12] studied the efficiency of parallelization in CA for reaction diffusion processes. She found that an asynchronous CA can be properly parallelized; however, there is tradeoff between efficiency and stochasticity.

Applications to study potential threats to deep geological repositories (DGR) have not yet been explored. Diffusion of potential harmful chemicals may pose a risk to nuclear waste containers. The purpose of using CA here is to use a tool that allows modeling of multi-chemical species with some interactions among chemicals.

2.1 Nuclear Waste Management

2.1.1 Problem Description

Deep geological repositories (DGR) are considered to be the safest long-term nuclear waste disposal solution [13]. It involves the excavation of an underground site where used fuel containers (UFC) are placed for permanent storage. The basic idea of the design is to use barriers of different materials to protect and contain the containers. Several countries are currently developing DGRs, the most developed of which are Finland, Sweden, and Canada [14]-[16].

2.1.2 Canadian DGR Design Concept

The Canadian design is planned to manage more than 4.6 million fuel bundles in a suitable rock formation that may be crystalline or sedimentary. It will have underground facilities at ~500 m containing the UFCs and facilities on the surface of the site for operation, maintenance, and long-term monitoring [16].

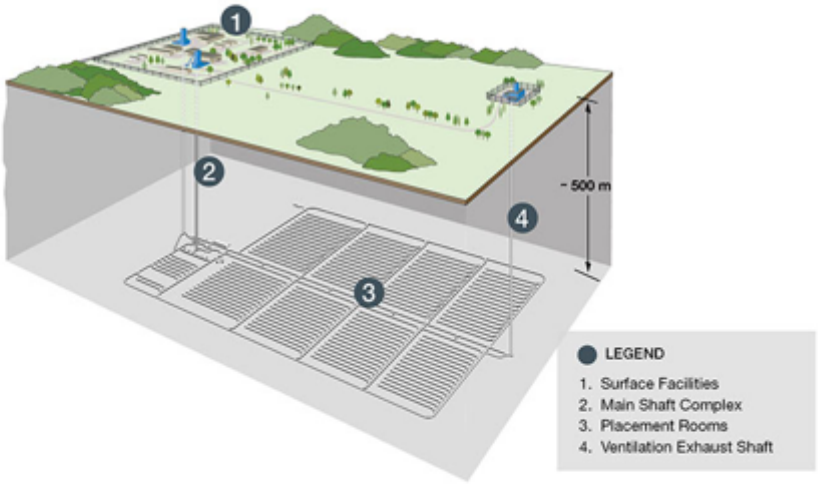


Figure 2.1 Canadian DGR design (Source: www.nwmo.ca Image by NWMO)

The design is based on a multiple-barrier system consisting of 5 barriers that contain and isolate the nuclear waste. The first barrier is the creation of ceramic fuel pellets from uranium dioxide powder which are durable, difficult to dissolve in water, and resistant to high temperatures. The second barrier are fuel bundles made of Zircaloy with a graphitecoated inside. Then bundles will be placed inside UFCs, the third barrier, whose design has been optimized to cope with thermomechanical forces. They are capsules made of 30 mm of steel with a 3 mm copper coating. The fourth barrier is the buffer box containing each UFC. It is made of highly compacted bentonite (HCB), a material made from volcanic ashes that is a natural water barrier and its swelling property makes it an excellent sealer. The last barrier is the geosphere, the ~500 m rock layer in between the underground facilities and the surface [17].

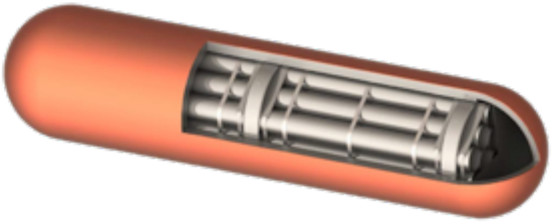


Figure 2.2 UFC dimensions 2506×556×556 mm (Source: [16] Image by NWMO)

2.1.3 Normal Evolution Scenario

Since the DGR is intended to last for a long-term period, it is fundamental to ensure it remains safe and secure for people, communities and the environment, and fair to current and future generations [18]. There are disruptive scenarios that may lead to penetration of barriers and abnormal loss of containment such as tectonic movement, glacial effects, human intrusion, or meteorite impact among others. For a thorough screening of extreme conditions see [19], [20].

For the present study it was assumed the **normal evolution scenario**, which “is based on a reasonable extrapolation of site and repository features, events and processes” [19], and the attention was limited to the breaching of the copper layer of UFCs caused by sulphide corrosion.

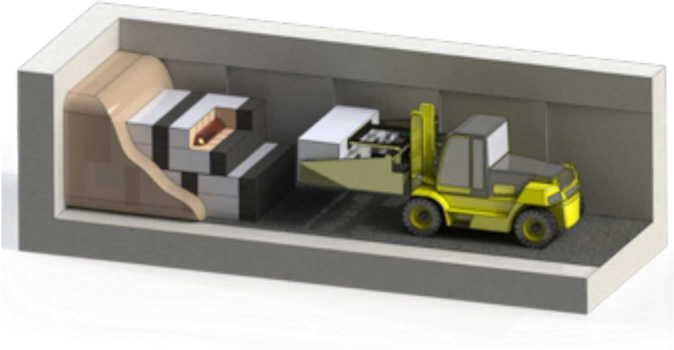


Figure 2.3 Placement room for UFCs (Source: [16] Image by NWMO)

2.1.4 Definition of failure

This work followed [22] defining failure of the UFC as a breach of the copper layer. From all the possible UFCs that will be inside the emplacement, those closest to the rock were selected, which are the first seven canisters to be placed per room [22]. The number of rooms is 318 [16], giving a total of 2,226 canisters. That is about 2% of all UFCs or 1 per 50 canisters. Our study determines lifetimes for those 2,226 UFCs.

2.1.5 Copper corrosion

The reaction describing copper corrosion by sulphide is

$$2Cu(s) \; + \; HS^{-} + H^{+} \rightleftarrows Cu_2S(s) + H_2 \qquad (2.1)$$

a conservative assumption is to assume copper reacts instantaneously with sulphide [22], [23] implying the rate of corrosion is determined by the rate at which sulphide reaches the copper surface (See Appendix A for a test on this assumption). From stoichiometry the depth of corrosion can be calculated using the formula proposed in [23]:

$$d_{Corr} = \frac{N_{HS} f_{HS} M_{Cu}}{A_{Corr} \rho_{Cu}} \qquad (2.2)$$

where N_{HS} is the amount of sulphide, f_{HS} is the stoichiometric factor (equal to 2), M_{Cu} is the molar mass of copper, A_{Corr} is the area exposed to corrosion, and ρ_{Cu} is the copper density. In Chapter 4 it is shown how this formula was applied to time-dependent results.

2.1.6 Groundwater composition

Data in Table 2.1 contains an estimation of groundwater composition in crystalline and sedimentary rock. Since there is no site of construction defined, the values are intended to represent plausible site conditions at a depth close to 500 m.

Table 2.1 Estimation of groundwater composition

Rock type	Crystalline	Sedimentary
pH	7.5	6.5
Environment	Reducing	Reducing
SO4 [mg/L]	1,000	310
HS^- [mg/L]	0	0

2.1.7 Bentonite Density

Bentonite clay MX-80 will be used for the buffer box protecting the UFC and to fill the gaps between the host rock and the buffer boxes. The former is referred to as highly compacted bentonite (HCB) and the latter gapfill (GF). However, dry densities are different producing different effective diffusivity coefficients.

The effective diffusivity for sulphide in HCB is assumed to be 1×10^{-11} [m^2/s] in [22], [24], [25] although that value is likely overestimated because it corresponds to a density of 1,590 [kg/m^3] as commented in [24].

Table 2.2 Properties of Bentonite clays

Clay	Dry density [kg/m^3]	Saturation [%]	Porosity [%]	Bulk density [kg/m^3]
GF	1,410	6	48.6	1439
HCB	1,700	67	38.2	1955

Instead, the effective diffusivity was estimated using the following formula from [24] for effective diffusivity of anions

$$D_e = 5.30087 \times 10^{-10} \times exp(-2.561 \times 10^{-3} \times \rho_{Dry}) \qquad (2.3)$$

where ρ_{Dry} is the dry density in [kg/m^3], D_e is in [m^2/s].

This work followed [25] estimating the effective diffusivity of sulphate as half that of sulphide. Table 2.2 shows some properties of the clay that will be useful to determine the initial conditions for our models.

2.2 Cellular Automata

2.2.1 Definition and development

A cellular automata (plural) or cellular automaton (singular) (CA) is a computational model represented by a structure of adjacent and interactive cells. The simplest CA have one discrete state variable whose domain is {0,1}; for a value of zero the cell is colored white and for a value of one it is colored black. The state changes according to the state of the surrounding cells (its neighborhood). Cellular automata can be defined in one, two or three spatial dimensions.

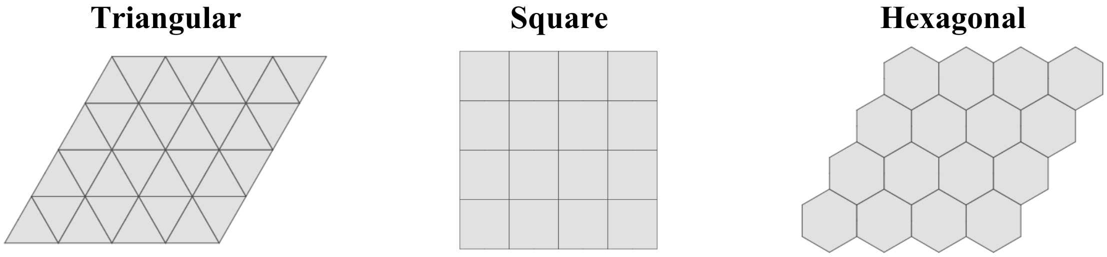


Figure 2.4 Examples of 2D cellular automata

Its formal development may be traced back to the work of John von Neumann when he tried to develop a self-reproducing automaton and later use the same framework to solve differential equations [26] around 1950s. The field grew steadily by the work of von Neumann and Stanislaw Ulam at the Los Alamos laboratory and by the 1960s there were attempts to link it with dynamical systems via evolving CA. On 1970 John Conway creates the '*The game of "Life"*' model [27]. At that time some connections between CA, parallel computing and neural networks were discovered. Around 1980s Stephen Wolfram [28] starts studying systematically the properties of CA since then the field has attracted wide attention [29].

2.2.2 Neighborhood

The neighbor of a reference cell (x_0, y_0) is the set of adjacent cells (including itself) with which the reference cell has some interaction. The von Neumann neighborhood of range *r* is defined as

$$N_{cell}^{N} = \{(x,y): |x - x_0| + |y - y_0| \leq r\} \tag{2.3}$$

the number of neighbors as r increases produces the following sequence 1, 5, 13, 25, 41, 61, 85,…, described by $2r(r+1)+1$. The Moore neighborhood of range *r* is defined as

$$N_{cell}^{M} = \{(x, y): |x - x_0| \leq r \ \wedge |y - y_0| \leq r\}$$

as r increases, we obtain the following number of neighbors 1, 9, 25, 49, 81,…, described by $(2r+1)^2$. The neighborhood will determine the degree of interaction (connectedness) of the CA. Both neighborhood definitions can be extended to 3D.

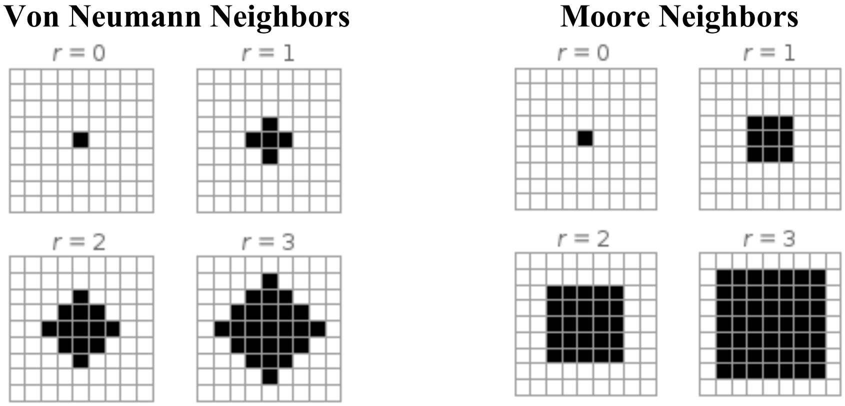


Figure 2.5 Definitions of neighborhood for a square 2D CA (Image from Wolfram|Alpha)

2.2.3 Rules of behavior

It was mentioned before the state of the cells changes according to some function of the state of its neighbors, e.g., *if more than two neighbor cells are black at time t, turn to black at time t+1; otherwise, turn white at time t+1.* This function may be deterministic or stochastic and defined over a von Neumann or Moore neighborhood of size *r*.

For a deterministic 2D square CA, the number of states of its Von Neumann neighborhood of range *r* is $2^{2r(r+1)+1}$ and the total number of rules is $2^{2^{2r(r+1)+1}}$, which produces 2^{32} different rules for *r = 1*.

Stephen Wolfram [29] has studied extensively the 1D square CA with *r=1*, also named **elementary cellular automata**, where the number of states of the neighborhood is 2^3 and the total number of rules is $2^{2^3} = 256$. He has categorized, studied its properties, and numbered (from 0 to 255) all the rules.

A convenient way to show the rules is by using the diagrams developed by Wolfram, where the 8 possible neighbor states are arranged in boxes. The figures inside indicate particular neighbor configurations and the next state for the following time step. Below we show an example of a behavior rule.

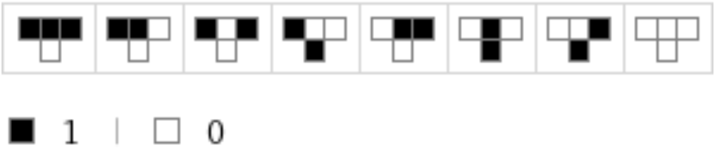
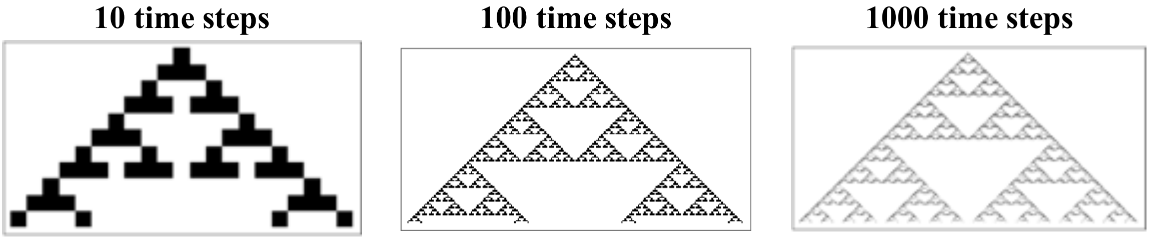


Figure 2.6 Example of a behavior rule (Image from Wolfram|Alpha)

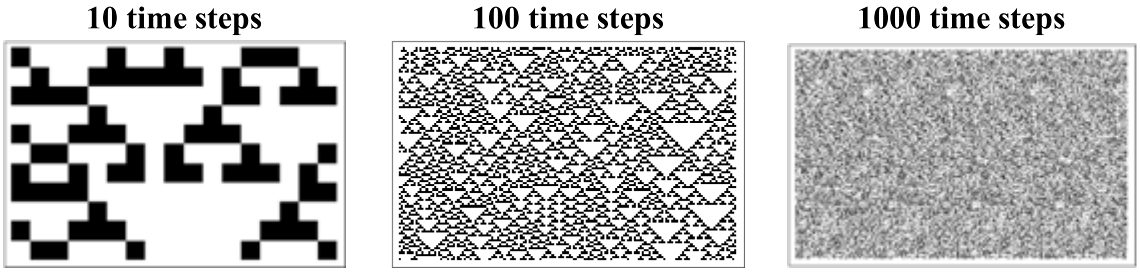
If exactly one cell in your neighborhood is black at time t, turn black at time t+1; otherwise, turn white at time t+1, we apply this rule (see Figure 2.6) to a single black cell, we will notice that a fractal pattern emerges after a few time steps, the Sierpinski triangle:



10 time steps **100 time steps** **1000 time steps**

Figure 2.7 Plot of the evolution of a 1D CA from a single black cell

If instead, the same rule is applied to a random initial state then it produces



10 time steps **100 time steps** **1000 time steps**

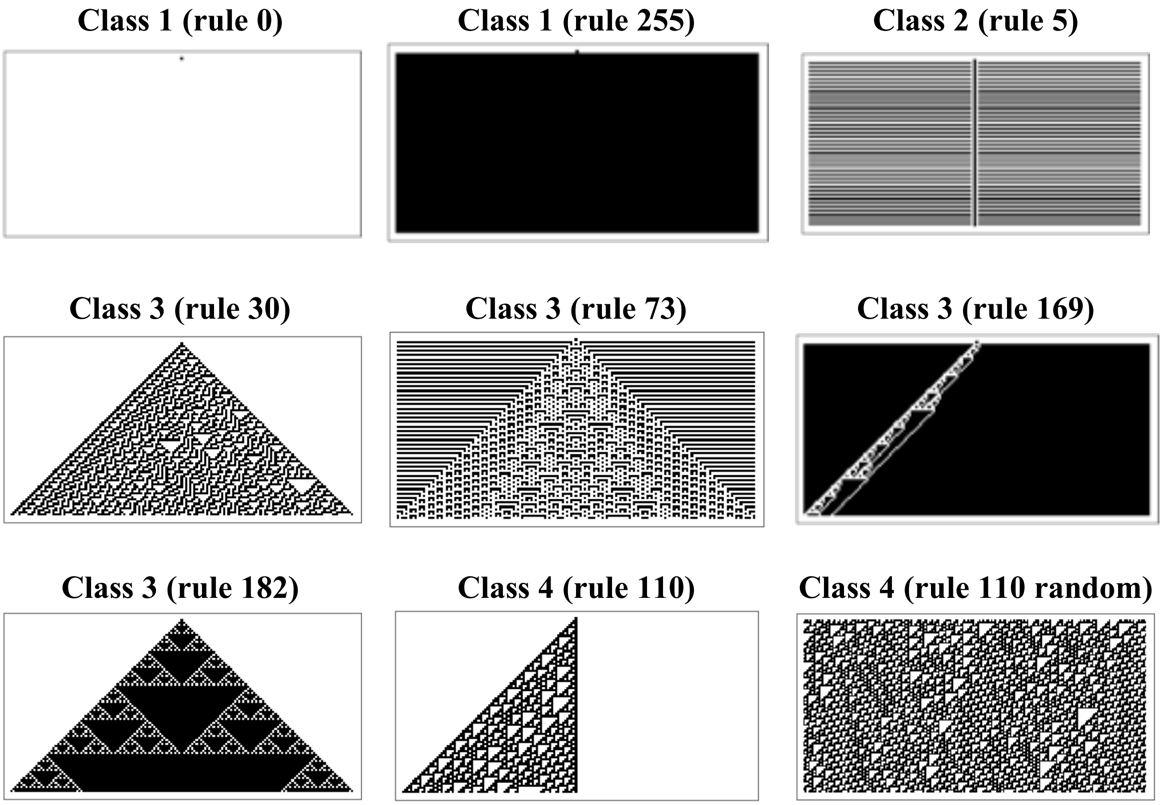
Figure 2.8 Plot of the evolution of a 1D CA from a random seed

Approximately one-third of the rules produce patterns that grow infinitely and about 14% of the rules produce complicated patterns. The author suggests consulting Wolfram's book for further information [29].

2.2.4 Collective behavior

The power of CA is that from simple rules, a complex collective behavior arises, i.e., emergent properties such as uniformity, self-similarity, recurrence, randomness. From observing the different patterns formed, Wolfram divided the behavior of the CA in four categories:

- Class 1. The cellular automaton evolves to a homogeneous state despite different initial states.
- Class 2. Oscillating patterns where initial perturbations remain local.
- Class 3. Chaotic structures, initial perturbations are propagated.
- Class 4. Complex patterns with persistent local structures, these are capable of universal computation (i.e., performing a finite set of instructions).



Class 1 (rule 0) **Class 1 (rule 255)** **Class 2 (rule 5)**

Class 3 (rule 30) **Class 3 (rule 73)** **Class 3 (rule 169)**

Class 3 (rule 182) **Class 4 (rule 110)** **Class 4 (rule 110 random)**

Figure 2.9 Examples of different classes CA

2.2.5 Conway's Game of "Life"

John H. Conway first designed this experiment in 1970 [27]. It describes the behavior of a cellular automata grid with simple rules of interaction. It is interpreted as an analogy of how species survive on a resource-constrained environment. Tiles are to represent living creatures, and its eight neighbors are the competitors. The possible states for each tile are: alive or dead. The rules to modify these states are:

1. If a tile has two or three live neighbors, then it survives to next time step.
2. If a tile has one or more than three alive neighbors, then it dies.
3. If a dead tile has exactly three alive neighbors, then it becomes alive.

These simple rules give raise to a non-trivial behavior on the whole grid. Nevertheless, future states are completely determined by the initial state. Typically, the initial state is determined randomly.

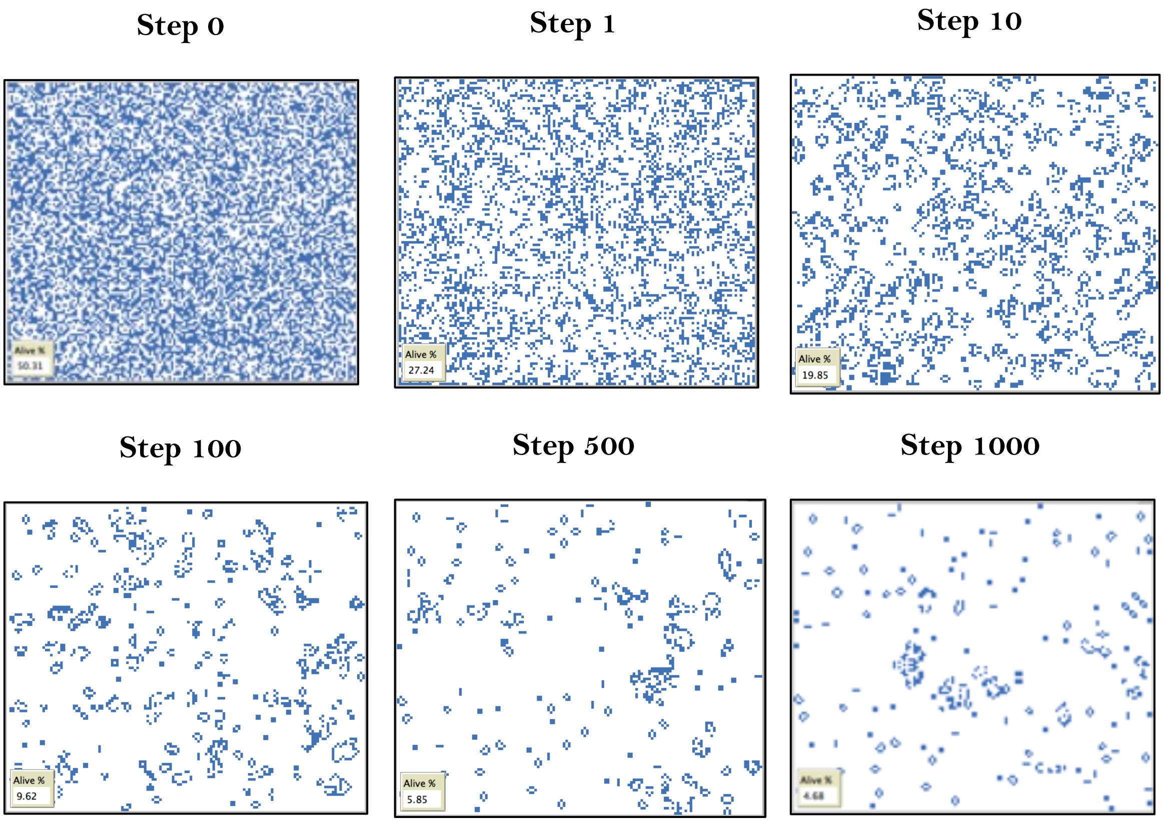


Figure 2.10 Conway's Game of Life (blue = alive) random seed

2.2.6 Relation between CA and PDEs

Cellular automata are by their own nature dynamic structures defined over a discretized space; therefore, it is possible to find a bridge to partial differential equations (PDEs) by means of the finite difference scheme. Each cell or patch may represent a state variable

Δx distance apart, and the change of state may be synchronized to take place every Δt. The state variable is then changed from a discrete to a real domain. Therefore, it is possible to convert a PDE into a CA, however the inverse problem is not as straightforward. Omohundro [30] studied a procedure to derive generic PDE from a CA rules; however, his procedure is not simple to implement.

Some examples of how to derive CA rules from some typical PDEs are presented next.

2.2.7 Chemical Reaction

Let us derive a cellular automata rule for the Biochemical-Oxygen demand reaction [32]:

$$\dot{\psi} = -k_B \psi$$

after applying a backward difference scheme, it is obtained

$$\frac{\psi^{t+1} - \psi^{t}}{\Delta t} = -k_B \psi^{t+1}$$

then solving for the concentration for the next time step, and the updating state rule is

$$\psi^{t+1} = \frac{\psi^{t}}{1 + \Delta t\, k_B}.$$

Figure 2.11 shows a frame of the CA simulation and a plot of concentration over time.

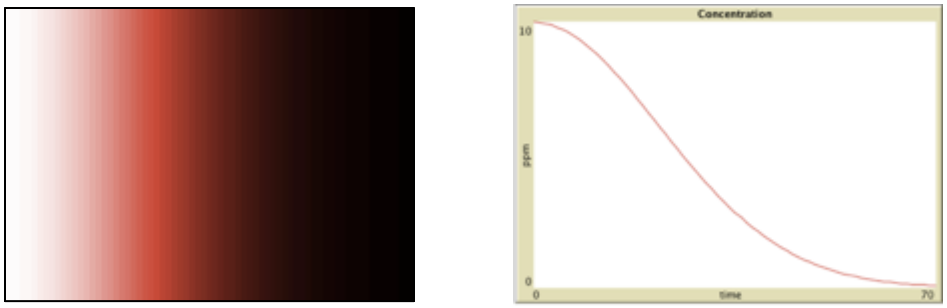


Figure 2.11 Biochemical-Oxygen demand reaction using cellular automata

2.2.8 Diffusion Equation

For the diffusion equation, let us derive the state rule for 1D space knowing that it can be extended to more dimensions

$$\dot{\psi} = D_x \frac{\partial^2 \psi}{\partial x^2}$$

after applying forward and central differences, it is obtained

$$\frac{\psi_i^{t+1} - \psi_i^t}{\Delta t} = D_x \left(\frac{\psi_{i+1}^t - 2\psi_i^t + \psi_{i-1}^t}{\Delta x^2} \right)$$

then, the updating state rule becomes

$$\psi_i^{t+1} = \psi_i^t + D_x \frac{\Delta t}{\Delta x^2} (\psi_{i+1}^t - 2\psi_i^t + \psi_{i-1}^t)$$

by choosing proper units such that Δx = 1 and $D_x \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$, and introducing initial and boundary conditions, the CA can evolve describing a diffusion process. In the next chapter we develop a procedure to derive displacement probabilities allowing us to solve the diffusion equation using CA or individual molecule agents.

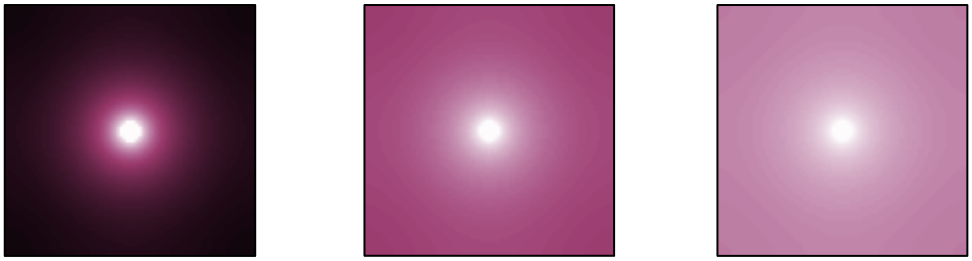


Figure 2.12 2D diffusion using cellular automata with a constant circle source

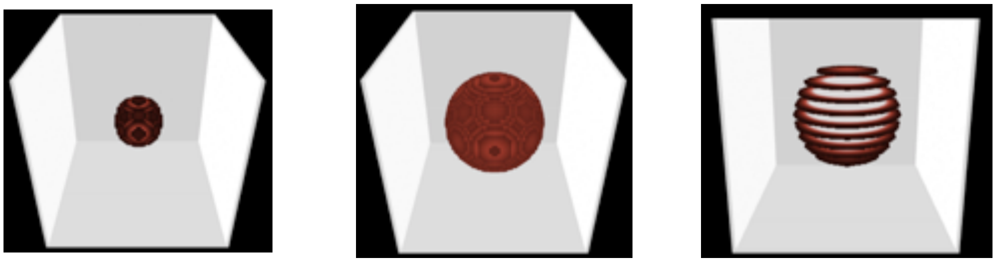


Figure 2.13 3D diffusion using cellular automata with a constant sphere source

2.2.9 Wave Equation

The wave equation is

$$\ddot{\psi} = c^2 \frac{\partial^2 \psi}{\partial x^2}$$

after applying central difference scheme, it is obtained

$$\frac{\psi_i^{t+1} - 2\psi_i^t + \psi_i^{t-1}}{\Delta t^2} = c^2 \frac{\psi_{i+1}^t - 2\psi_i^t + \psi_{i-1}^t}{\Delta x^2}$$

then, the CA state rule becomes

$$\psi_i^{t+1} = 2\psi_i^t + \left(c\frac{\Delta t}{\Delta x}\right)^2 (\psi_{i+1}^t - 2\psi_i^t + \psi_{i-1}^t) - \psi_i^{t-1}$$

the stability condition here is met by $c\frac{\Delta t}{\Delta x} \leq 1$. Figure 2.14 shows six frames of two waves colliding.

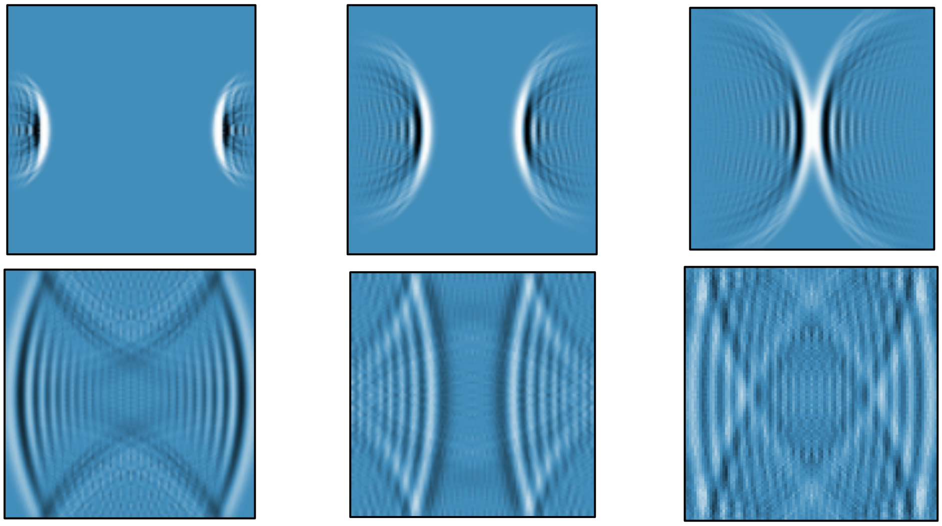


Figure 2.14 2D cellular automata describing the wave equation

2.2.10 Reaction Diffusion Equation

Let us consider the second-order nonlinear Fisher-Kolmogorov PDE (or Fisher-KPP) with convection; it is used to describe the spreading of biological populations:

$$\dot{\psi} + V_x \frac{\partial\psi}{\partial x} = D_x \frac{\partial^2\psi}{\partial x^2} + \mathrm{R}(\psi)$$

$$R(\psi) = \psi(1 - \psi)$$

after applying finite difference, it is obtained

$$\frac{\psi_i^{t+1} - \psi_i^t}{\Delta t} + V_x \frac{\psi_{i+1}^t - \psi_i^t}{\Delta x} = D_x \frac{\psi_{i+1}^t - 2\psi_i^t + \psi_{i-1}^t}{\Delta x^2} + \psi_i^t(1 - \psi_i^t)$$

then, the CA state rule becomes

$$\psi_i^{t+1} = \psi_i^t - V_x \frac{\Delta t}{\Delta x}(\psi_{i+1}^t - \psi_i^t) + D_x \frac{\Delta t}{\Delta x^2}(\psi_{i+1}^t - 2\psi_i^t + \psi_{i-1}^t) + \Delta t\, \psi_i^t(1 - \psi_i^t)$$

Figure 2.15 shows the evolution of five initial populations as they spread while moving to the right.

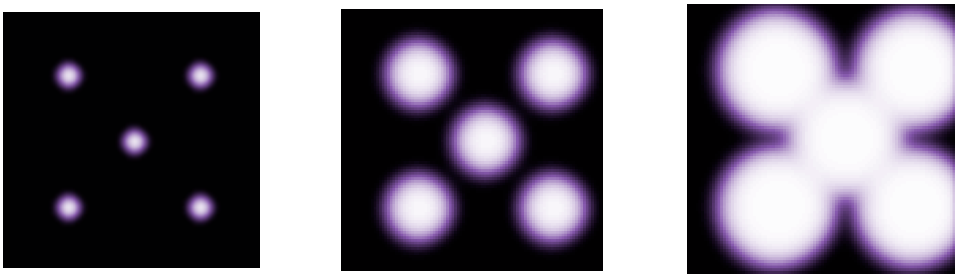


Figure 2.15 Fisher-Kolmogorov equation using 2D cellular automata

In the next chapter we will describe the derivation of diffusion and reaction terms for cellular automata in a probabilistic framework.

Chapter 3 Diffusion using Cellular Automata

INTRODUCTION

This chapter shows how to use cellular automata to describe diffusion by computing probabilities of displacement for particles. The methodology is described along with

- Its derivation from the diffusion PDE
- How to introduce initial and boundary conditions
- Its relation to Brownian motion and Markov chains
- The derivation of diffusion displacement
- Its consistency, order, stability, and convergence.

The method outlined in this chapter will be used to compute lifetimes of nuclear waste containers on Chapter 4.

3.1 Diffusion

Diffusion is the displacement of small particles of solute as they spread down the concentration gradient of a suspension or solution due to thermal molecular movement [32]. It is also used to describe heat propagation, groundwater flow among other applications in biology, sociology, economics, and finance. This work uses diffusion to describe the change in concentration of a solute along a solution/medium.

Fick's second law for a 2D space describes diffusion of solute φ throughout a solution as follows

$$\frac{\partial \varphi}{\partial t} = D_x \frac{\partial^2 \varphi}{\partial x^2} + D_y \frac{\partial^2 \varphi}{\partial y^2} \tag{3.1}$$

with initial conditions

$\varphi(x, y, 0) = h_0(x, y)$

and boundary conditions

$\varphi(x, 0, t) = h_1(x, t)$

$\varphi(0, y, t) = h_2(y, t)$

$\varphi(L_x, y, t) = h_3(y, t)$

$\varphi\left(x, L_y, t\right) = h_4(x, t)$

φ is the concentration; x and y are space dimensions; D_x and D_y are the diffusivity parameters along x and y; x = L_x and y = L_y are the right and upper boundaries; and h_1, h_2, h_3, h_4 are functions defining the values for the four boundaries. Common methods to solve PDEs are finding the analytical solution when it exists (it depends on the initial and boundary conditions), finite element, and finite differences. In the next sections an agent-based method is developed that has certain advantages over those methods but some restrictions as well.

3.2 Discretized Diffusion

Some particular problems are suitable and convenient for using a discretized space over which solute moves. For example: systems with heterogeneous materials with different physical properties, or systems where the focus is on modeling molecule interactions. The method of Section 3.3 accounts for discrete or continuous solute over a discretized space.

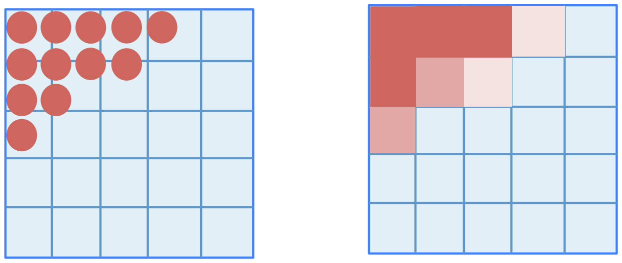


Figure 3.1 Discretized and continuous solute diffusion over a discretized space

3.3 Probability of Displacement

Diffusion on a discrete space is related to that on a continuous one by the probability of displacement of solute particles throughout the solution. We show next how to derive such probabilities for a 2D system from the diffusivity coefficients D_x and D_y.

1. Discretize the space domain using Δx and Δy as width and length of each panel or cell of the grid:

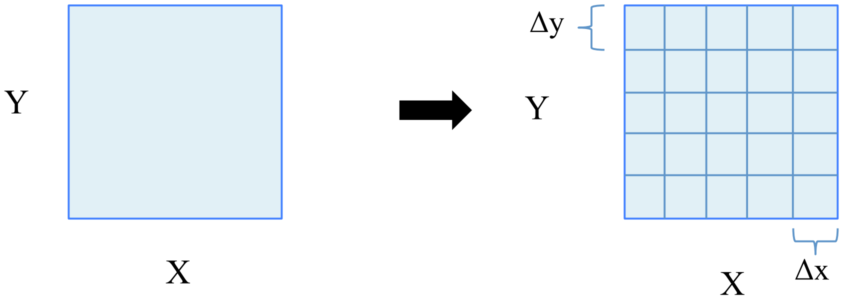


Figure 3.2 Discretization of the domain space

2. Replace the differentiation terms of eq. (3.1) by first-order difference terms using the explicit formulation:

$$\frac{\varphi_{i\,j}^{t+1} - \varphi_{i\,j}^{t}}{\Delta t} = D_x \left(\frac{\varphi_{i+1\,j}^{t} - 2\,\varphi_{i\,j}^{t} + \varphi_{i-1\,j}^{t}}{\Delta x^2} \right) + D_y \left(\frac{\varphi_{i\,j+1}^{t} - 2\,\varphi_{i\,j}^{t} + \varphi_{i\,j-1}^{t}}{\Delta y^2} \right)$$

3. Solve for the concentration for the next time step, i.e., φ_{ij}^{t+1}:

$$\varphi_{i\,j}^{t+1} = \left[1 - 2D_x\frac{\Delta t}{\Delta x^2} - 2D_y\frac{\Delta t}{\Delta y^2}\right]\varphi_{i\,j}^{t} + D_x\frac{\Delta t}{\Delta x^2}\varphi_{i+1\,j}^{t} + D_x\frac{\Delta t}{\Delta x^2}\varphi_{i-1\,j}^{t} + D_y\frac{\Delta t}{\Delta y^2}\varphi_{i\,j+1}^{t} + D_y\frac{\Delta t}{\Delta y^2}\varphi_{i\,j-1}^{t} \tag{3.2}$$

This equation may be interpreted as a Markov chain where the cells of the grid represent states and φ concentrations 'jump' from state to state with a probability equal to the coefficients of eq. (3.2).

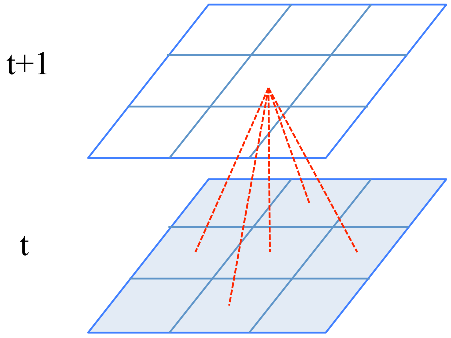


Figure 3.3 Transition from states at time t to a single state at t+1

4. Compute probabilities of displacement:

$$p_{i\pm1,j} = D_x\frac{\Delta t}{\Delta x^2}$$

$$p_{i,j\pm1} = D_y\frac{\Delta t}{\Delta y^2}$$

$$p_0 = 1 - 2D_x\frac{\Delta t}{\Delta x^2} - 2D_y\frac{\Delta t}{\Delta y^2}$$

p_0 is the probability that a particle in cell ij will remain in cell ij for the next time step and $p_{\pm i,j}$, $p_{i,\pm j}$ are the probabilities of a particle moving from neighbor cells to cell ij. According to the axioms of probability, the above distribution must meet the following requirements to be a pdf

1. $p_{ij} \geq 0, \forall i, j$
2. $p_0 + p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} = 1$

The second requirement is met if the following inequality holds

$$p_0 = 1 - 2D_x \frac{\Delta t}{\Delta x^2} - 2D_y \frac{\Delta t}{\Delta y^2} \geq 0 \tag{3.3}$$

assuming $D_x = D_y = D$ and $\Delta x = \Delta y = \Delta u$, then it yields

$$D \frac{\Delta t}{\Delta u^2} \leq \frac{1}{4} \tag{3.4}$$

The first requirement is met from the fact that diffusivity coefficients (D_x, D_y) are never zero or negative. However, in the majority of cases the computed probabilities do not meet (3.3). In that case, either the space or the time increment must be changed to a proper scale. Most time in ABMs the space increment is fixed and equal to one (i.e. $\Delta u = 1$), therefore the time increment must be modified to a proper value so that eq. 3.4 is met.

5. Compute the expected concentration for each cell and time as follows

$$\begin{aligned}
\varphi_{i\,j}^{t+1} &= p_0\, \varphi_{i\,j}^{t} + p_{i+1,j}\varphi_{i+1\,j}^{t} + p_{i-1,j}\varphi_{i-1\,j}^{t} + p_{i,j+1}\varphi_{i\,j+1}^{t} + p_{i,j-1}\varphi_{i\,j-1}^{t} \\
&= \sum_{i,j\in N_{cell}} p_{ij}\, \varphi_{i\,j}^{t} \\
&= E\left[\varphi_{i\,j}^{t}\right]_{i,j\in N_{cell}} \qquad (3.5)
\end{aligned}$$

which is the same equation as (3.2) but with probability notation instead, $N_{cell} = \{(\mathrm{i},\mathrm{j}), (\mathrm{i}+1,\mathrm{j}), (\mathrm{i}-1,\mathrm{j}), (\mathrm{i},\mathrm{j}+1), (\mathrm{i},\mathrm{j}-1)\}$ is the von Neumann neighborhood of cell ij (includes itself) . Section 3.5 shows this result is the same as the main assumption made in [1] to derive the diffusion equation from Brownian motion of particles.

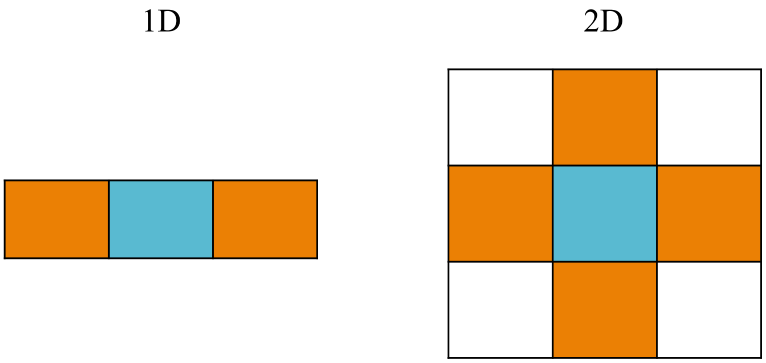


Figure 3.4 Neighbor cells (orange) of a reference cell (blue)

Concentration variance can be computed with

$$V[\varphi_{i\,j}^{t}] = E\left[\varphi_{i\,j}^{t\;2}\right] - E[\varphi_{i\,j}^{t}]^{2}$$

which may be used to measure the homogeneity of the solute inside the solution. Chapter 4 applies this discretized diffusion method to compute expected lifetimes of nuclear waste canisters.

3.4 Initial and Boundary Conditions

The initial condition is implemented by assigning an initial concentration to each cell of the cellular automata. Boundary conditions are typically implemented by applying Dirichlet or Neumman conditions; the implementation of a Robin condition which is a general case that includes both of them is shown next. Robin conditions are of the form

$$a_R \varphi(L_x, t) + b_R \frac{\partial \varphi(L_x, t)}{\partial x} = g(t), \qquad t \geq 0$$

where a_R and b_R are constants and g(t) is a function at the boundary $x = L_x$. After applying backward differentiation

$$a_R\, \varphi_{L_x}^{t} + b_R \frac{\left(\varphi_{L_x}^{t} - \varphi_{L_x-1}^{t}\right)}{\Delta x} = g(t), \qquad t \geq 0$$

and solving for concentration at time t:

$$\varphi_{L_x}^{t} = \frac{g(t)\, \Delta x + b_R\, \varphi_{L_x-1}^{t}}{a_R\, \Delta x + b_R}. \qquad t \geq 0 \tag{3.6}$$

To represent a physical barrier (von Neumann condition), set $a_R = 0$, $b_R = 1$, $g(t) = 0$; which yields

$$\varphi_{L_x}^{t} = \varphi_{L_x-1}^{t}, \qquad t \geq 0 \tag{3.7}$$

which affects the computation of all cells adjacent to L_x, which are φ_{ij} with $i = L_x\text{-}1$

$$\begin{aligned} \varphi_{L_x-1\,j}^{t+1} &= p_{ij}\, \varphi_{L_x-1\,j}^{t} + p_{i+1\,j} \varphi_{L_x\,j}^{t} + \cdots \\ &= p_{ij}\, \varphi_{L_x-1\,j}^{t} + p_{i+1\,j}\, \varphi_{L_x-1\,j}^{t} + \cdots \end{aligned}$$

$$= \left(p_{ij} + p_{i+1\,j}\right) \varphi^{t}_{L_x-1\,j} + \cdots$$

increasing the probability of remaining on the same cell.

To represent a constant concentration (Dirichlet condition), set $a_R = 1$, $b_R = 0$, $g(t) = c$; which yields

$$\varphi_{L_x} = \mathrm{c} \tag{3.8}$$

changes in the adjacent cells are

$$\varphi^{t+1}_{L_x-1\,j} = p_{ij}\, \varphi^{t}_{L_x-1\,j} + p_{i+1\,j} \varphi^{t}_{L_x\,j} + \cdots$$

$$= p_{ij}\, \varphi^{t}_{L_x-1\,j} + p_{i+1\,j}\, \mathrm{c} + \cdots$$

Similarly, other boundary conditions may be introduced using equation (3.6).

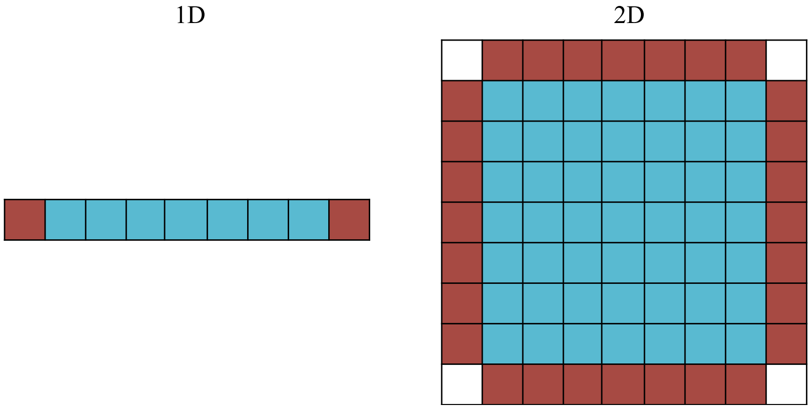


Figure 3.5 Boundary cells (red) in a 1D and 2D cellular automata

3.5 Relation to Brownian Motion

When describing the irregular movement of particles suspended in a liquid, Einstein assumed that each particle moves independently from the rest and that the set of path realizations after a time interval is mutually independent as long as the time increment is not too small [32].

If we select a time interval τ, then we can assume the displacement of particles along x-axis increases by Δx, where Δx may be positive or negative and is different for each particle. The displacement Δx has a domain on the real line and probability p(Δx), where p(Δx) is an even function, i.e., p(Δx) = p(-Δx), from the fact that displacement Δx is symmetrical about the y-axis. We know that

$$\int_{-\infty}^{\infty} p(\Delta x)\, d\Delta x = 1$$

for p(Δx) is a continuous probability distribution, now if we have a function that describes the number of particles per unit of volume φ(x,t), we can calculate the distribution of particles at time t + τ from the distribution at time t by calculating the particles in between two planes perpendicular to the x-axis, namely x and x + dx, then it is obtained

$$\varphi(x, t+\tau) \cdot dx = dx \cdot \int_{\Delta x\,=\,-\infty}^{\Delta x\,=\,\infty} \varphi(x+\Delta x, t)\, p(\Delta x)\, d\Delta x \qquad (3.9)$$

which can be simplify and rewritten it in terms of the expected value

$$\begin{aligned} \varphi(x, t+\tau) &= \int_{\Delta x\,=\,-\infty}^{\Delta x\,=\,\infty} \varphi(x+\Delta x, t)\, p(\Delta x)\, d\Delta x \\ &= E_{\Delta x}[\varphi(x+\Delta x, t)] \\ &= E_{\Delta x}[\varphi(x, t)] \qquad (3.10) \end{aligned}$$

Equation (3.10) derived in [32] is the same as (3.5), they are different in that the former is 1D over a continuous domain whereas the latter is 2D with a bounded discrete domain.

$$\varphi_{i\,j}^{t+1} = E\left[\varphi_{i\,j}^{t}\right]_{i,j \in N_{ij}}$$

$$\varphi(x, t+\tau) = E_{\Delta x}[\varphi(x, t)]$$

For those interested in how Einstein derived the diffusion equation from (3.9) the author highly recommends reading his original paper. Using the same rationale, the derivation of particle displacements for 2D is shown in Section 3.7.

3.6 Relation to Markov Chains

It was mentioned before that the solute particles ‘jump’ from a cell in time t to a neighbor cell in time t+Δt. Which is an indication that the future state of the system depends only on the current state and not on past states. This property is known as the Markov property.

For sake of simplicity in notation, let us consider diffusion in 1D (plus time) over a discretized domain. $\boldsymbol{\varphi}_t$ is the vector containing the concentration of solute along the x-axis at time t as follows

$$\boldsymbol{\varphi}_t = \begin{bmatrix} \varphi_1^t \\ \varphi_2^t \\ \vdots \\ \varphi_{L_x}^t \end{bmatrix}$$

the Markov property states that $P(\boldsymbol{\varphi}_t | \boldsymbol{\varphi}_{t-1}, \boldsymbol{\varphi}_{t-2}, \dots, \boldsymbol{\varphi}_0) = P(\boldsymbol{\varphi}_t | \boldsymbol{\varphi}_{t-1})$, let $\mathbf{P}_T$ be the transition probability matrix

$$\boldsymbol{P}_T = \begin{bmatrix} p_{ij} & \cdots & p_{iN} \\ \vdots & \ddots & \vdots \\ p_{Mj} & \cdots & p_{MN} \end{bmatrix}$$

where the entries are the same probabilities as those computed in equation (3.5), then it is possible to compute the concentration for the next time step using the formula

$$\boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t * \boldsymbol{P}_T. \tag{3.11}$$

The advantage of using a Markov chain is that it need not compute the concentration vector at every time. For example, $\boldsymbol{\varphi}_{t+3}$ may be computed directly from $\boldsymbol{\varphi}_t$:

$$\begin{aligned} \boldsymbol{\varphi}_{t+3} &= \boldsymbol{\varphi}_{t+2} * \boldsymbol{P}_T \\ &= (\boldsymbol{\varphi}_{t+1} * \boldsymbol{P}_T) * \boldsymbol{P}_T \\ &= \boldsymbol{\varphi}_{t+1} * \boldsymbol{P}_T{}^2 \\ &= \left(\boldsymbol{\varphi}_t * \boldsymbol{P}_T{}^2\right) * \boldsymbol{P}_T \\ &= \boldsymbol{\varphi}_t * \boldsymbol{P}_T{}^3. \end{aligned}$$

The general formula to get the concentration after n periods from time t is

$$\boldsymbol{\varphi}_{t+n} = \boldsymbol{\varphi}_t * \boldsymbol{P}_T{}^n \tag{3.12}$$

Initial conditions are introduced by specifying the initial vector **φ_0** with initial values. Boundary conditions can be introduced using the equation (3.6) and modifying the **P_T** matrix accordingly. For example, to fix the boundaries to a constant (Dirichlet) we need to introduce a column in **P_T** with zeros except at the boundary and place a one instead. An example is shown below. For Neumann boundaries the procedure is similar, except that the entries of the column depend on the form obtained from (3.6), below an example is shown as well.

Let us assume a 1D system with the following characteristics: $D_x = 0.1$ [cm^2/day], $\Delta x = 1$ [cm], $\Delta x^2 = 1$ [cm^2], $\Delta t = 1$ [day]. The corresponding probabilities are computed from equation 3.2 but for 1D. Probabilities of displacement are $p(\Delta x = -1) = p(\Delta x = 1) = D_x*(\Delta t/\Delta x^2) = 0.1$ and $p(\Delta x = 0) = 1- 2*D_x*(\Delta t/\Delta x^2) = 0.8$

The initial condition is **φ_0** = $[10, 0, 0, 0, 0]^T$. We introduced a Dirichlet left boundary equal to ten and a Dirichlet right boundary equal to zero by including two columns with zeros everywhere except at the boundaries:

$$\boldsymbol{P}_T = \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 \\ 0 & 0.8 & 0.1 & 0 & 0 \\ 0 & 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0 & 0.1 & 0.8 & 0 \\ 0 & 0 & 0 & 0.1 & 1 \end{bmatrix}.$$

The calculation of the concentration profile after 5, 10, and 100 days is

$$\boldsymbol{\varphi}_5 = \boldsymbol{\varphi}_0 * \boldsymbol{P}_T{}^5 = [10 \quad 3.43 \quad 0.66 \quad 0.07 \quad 0]^T$$

$$\boldsymbol{\varphi}_{10} = \boldsymbol{\varphi}_0 * \boldsymbol{P}_T{}^{10} = [10 \quad 4.88 \quad 1.71 \quad 0.42 \quad 0]^T$$

$$\boldsymbol{\varphi}_{100} = \boldsymbol{\varphi}_0 * \boldsymbol{P}_T{}^{100} = [10 \quad 7.49 \quad 4.98 \quad 2.48 \quad 0]^T.$$

After 100 days, the system is near steady steate, **$\varphi_{\text{SteadyState}}$** = $[10, 7.5, 5, 2.5, 0]^T$.

Now, for solving the same system but with a Neumann right boundary instead, $\partial\varphi(L_x)/\partial x = 0$, which yields $\varphi_{Lx} = \varphi_{Lx-1}$ as seen in eq. (3.7). The matrix changes to

$$\boldsymbol{P}_T = \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 \\ 0 & 0.8 & 0.1 & 0 & 0 \\ 0 & 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0 & 0.1 & 0.8 & 1 \\ 0 & 0 & 0 & 0.1 & 0 \end{bmatrix}.$$

The concentration distribution after 5, 10, and 100 days is

$$\boldsymbol{\varphi}_5 = \boldsymbol{\varphi}_0 * \boldsymbol{P}_T{}^5 = [10 \quad 3.43 \quad 0.66 \quad 0.07 \quad 0.03]^T$$

$$\boldsymbol{\varphi}_{10} = \boldsymbol{\varphi}_0 * \boldsymbol{P}_T{}^{10} = [10 \quad 4.88 \quad 1.72 \quad 0.48 \quad 0.37]^T$$

$$\boldsymbol{\varphi}_{100} = \boldsymbol{\varphi}_0 * \boldsymbol{P}_T{}^{100} = [10 \quad 9.20 \quad 8.55 \quad 8.17 \quad 8.14]^T.$$

After 100 days the system is becoming homogeneous as expected since the right boundary is a ‘wall’ preventing any concentration from leaving the system.

3.7 Particle Displacement

The relation between diffusion and displacement in 2D diffusion is given by the following equation

$$E[\Delta u^2] = 2t\big(D_x + D_y\big) \tag{3.13}$$

which is in agreement with the 1D result found in [32] and the result showed in [8] when both diffusivity coefficients are the same. To derive (3.13) assume a function φ(x,y,t) such that its value at t+Δt equals the expected value of the function at time t but with space increments Δx and Δy. Each pair (Δx, Δy) has a probability p(Δx, Δy), then we have

$$\varphi(x, y, t + \Delta t) = \iint \varphi(x + \Delta x, y + \Delta y, t)\, p(\Delta x, \Delta y)\; d\Delta x\; d\Delta y$$

assuming independence in the displacement along x and y, then the equation can be rewritten as

$$\varphi(x, y, t + \Delta t) = \int\int \varphi(x + \Delta x, y + \Delta y, t)\, p(x)\; d\Delta x\;\; p(y)\; d\Delta y$$

$$= \int E_{\Delta x}[\varphi(x + \Delta x, y + \Delta y, t)]\;\; p(y)\; d\Delta y$$

$$= \; E_{\Delta y}\big[E_{\Delta x}[\varphi(x + \Delta x, y + \Delta y, t)]\big] \tag{3.14}$$

applying Taylor series expansions on both functions at base point φ(x,y,t)

$$\varphi(x, y, t + \Delta t) = \varphi(x, y, t) + \dot{\varphi}(x, y, t)\Delta t \; + \frac{1}{2}\ddot{\varphi}(x, y, t)\Delta t^2 + \cdots$$

$$\varphi(x + \Delta x, y + \Delta y, t) = \varphi(x, y, t) + \varphi_x(x, y, t)\Delta x \; + \varphi_y(x, y, t)\Delta y +$$

$$\frac{1}{2}\varphi_{xx}(x,y,t)\Delta x^2 + \varphi_{xy}(x,y,t)\Delta x\Delta y + \frac{1}{2}\varphi_{yy}(x,y,t)\Delta y^2 + \cdots$$

taking the double expectation on the RHS and simplifying notation

$$\varphi + \dot{\varphi}\Delta t + \cdots = \varphi + \frac{1}{2}E[\Delta x^2]\varphi_{xx} + \frac{1}{2}E[\Delta y^2]\varphi_{yy} + \cdots$$

dropping higher order terms because they become insignificantly small and solving for φ_t

$$\dot{\varphi} = \frac{E[\Delta x^2]}{2\Delta t}\varphi_{xx} + \frac{E[\Delta y^2]}{2\Delta t}\varphi_{yy}$$

comparing this equation to eq. (3.1) we get that Dx = $E[\Delta x^2]/(2\Delta t)$ and Dy = $E[\Delta y^2]/(2\Delta t)$; setting $\Delta u^2 = \Delta x^2 + \Delta y^2$ and taking the expectation over Δs, and it is finally obtained

$$E[\Delta u^2] = E[\Delta x^2] + E[\Delta y^2]$$

$$= 2\Delta t D_x + 2\Delta t D_y$$

$$= 2t(D_x + D_y)$$

the same procedure can be applied to a 3D system.

Displacement of a particle can be computed by solving for Δs in eq. (3.13)

$$E[\Delta u^2] = 2t(D_x + D_y)$$

$$\sum_{i,j\,\in N_{cell}} \Delta u_{ij}{}^2 p_{ij} = 2t(D_x + D_y)$$

Assuming Δx = Δy and Dx = Dy = D produces equal probabilities for the neighbor cells, i.e., $p_{i+1\,j} = p_{i-1\,j} = p_{i\,j+1} = p_{i\,j-1} = p_{out}$, then we have

$$\Delta u^2 4p_{out} = 4Dt$$

$$\Delta u = \sqrt{\frac{Dt}{p_{out}}}. \qquad (3.15)$$

The displacement of a particle with D = 0.1 [cm^2/day], and p_{out} = ¼ after 10 days is 2 cm. If p_{out} = 1/8, the displacement changes to 2.83 cm; and for p_{out} = 1/16, the displacement becomes 4 cm for the same period. The displacement increases as p_{out} decreases; this is because the displacement must be larger since it is happening with less frequency to be in consistency with the value of the second raw moment of displacement.

3.8 Diffusion in Mixed Mediums

When a system has an interface of two mediums with different diffusivities, the effective diffusivity between them is different from the average of their respective diffusivities. This is due to the reciprocal interaction between the two adjacent cells. This work followed [33] to determine the effective diffusivity between any two different mediums. The author explains that the temperature infinitely close to the interface must converge to the same value and the flux into the interface equals the flux coming out of it, which equals the flux between the two cells.

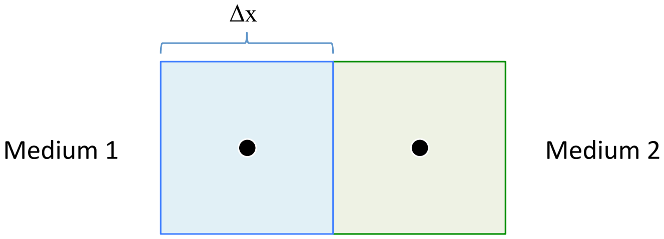


Figure 3.6 Adjacent cells with different diffusivities

Applying a conservation of flow restriction to the interface: incoming concentration is equal to the concentration coming out:

$$p_L\varphi_L + p_R\varphi_R + (1 - p_L - p_R)\varphi_I = p_L\varphi_I + p_R\varphi_I + (1 - p_L - p_R)\varphi_I$$

where φ_L, φ_R, φ_I are the concentration of the left and right cell and the interface, $p_L = D_L*(\Delta t/\Delta x^2)$, $p_R = D_R*(\Delta t/\Delta x^2)$. Simplifying and solving for φ_I, the concentration at the interface is

$$\varphi_I = \frac{p_L\varphi_L + p_R\varphi_R}{p_L + p_R}. \tag{3.16}$$

Factoring out the term ($\Delta t/\Delta x^2$), then it is obtained the same result as in [33]:

$$\varphi_I = \frac{D_L\varphi_L + D_R\varphi_R}{D_L + D_R}. \tag{3.17}$$

Applying the restriction of concentration from left to interface must equal that from the interface to the right, and must be equal that from left to right, we get

$$p_{1/2}\varphi_L - p_{1/2}\varphi_I = q_{1/2}\varphi_I - q_{1/2}\varphi_R = \mathrm{p}_I\varphi_L - \mathrm{p}_I\varphi_R$$

where $p_{1/2} = 2p$, $q_{1/2} = 2q$ due to

$$p_{1/2} = D_L \frac{\left(\frac{\Delta t}{2}\right)}{\left(\frac{\Delta x}{2}\right)^2} = \frac{4}{2} D_L \frac{\Delta t}{\Delta x^2} = 2D_L \frac{\Delta t}{\Delta x^2} = 2p$$

and similarly, for q, p_I is the probability of moving from left to right cell and vice versa is

$$p_I = \frac{2p_L p_R}{p_L + p_R}. \tag{3.18}$$

Factoring out the term $(\Delta t/\Delta x^2)$ and the probability of moving between mediums can be also expressed as

$$p_I = \left(\frac{2D_L D_R}{D_L + D_R}\right) \frac{\Delta t}{\Delta x^2}. \tag{3.19}$$

Eq. (3.18) or (3.19) will be used to model diffusion between different clay materials in Chapter 4.

3.9 Reaction Probability

Spatial transition is modeled by diffusion probabilities; however, when the system has also reaction terms, we must find a way to express species transition in a probabilistic framework. For that, we have defined **interspecies transition probabilities**, i.e., the probability that a species A will remain A, turn into B, or otherwise for the next time step.

Consider a simple chemical reaction $A \overset{k_R}{\rightarrow} B$, and $p(j_{t+1}|i_t);\ i,j \in \{A,B\}$ is the probability that species i will change to j for the next time step. Finite differences can be used to obtain the interspecies transition probabilities **from the numerical solution approximation** as follows

$$[\dot{A}] = -k_R[A]$$

$$\frac{[A]_{t+1} - [A]_t}{\Delta t} = -k_R[A]_{t+1}$$

$$[A]_{t+1} = \left(\frac{1}{1 + \Delta t k_R}\right)[A]_t$$

the transition probability from A to A is

$$P(A_{t+1}|A_t) = \left(\frac{1}{1 + \Delta t k_R}\right) \tag{3.20}$$

that from A to B (assuming 100% efficiency) is

$$P(B_{t+1}|A_t) = \left(\frac{\Delta t k_R}{1 + \Delta t k_R}\right)$$

$p(A_{t+1}|A_t) + p(B_{t+1}|A_t) = 1$ must be satisfied. The transition probabilities for B are $p(B_{t+1}|B_t) = 1$ and $p(A_{t+1}|B_t) = 0$, since once B is produced there is no change to A or otherwise.

Let us assume an efficiency of $\chi \in [0,1]$ in the production of B, then the transition probability from A to B changes to

$$P(B_{t+1}|A_t) = \chi\left(\frac{\Delta t k_R}{1 + \Delta t k_R}\right) \tag{3.21}$$

and the probability of losing chemical A, i.e., becomes unavailable in the system is

$$P(lost_{t+1}|A_t) = (1 - \chi)\left(\frac{\Delta t k_R}{1 + \Delta t k_R}\right)$$

$p(A_{t+1}|A_t) + p(B_{t+1}|A_t) + p(lost_{t+1}|A_t) = 1$ holds true. The transition probability from A to B remains the same and likewise for the transitions from B.

Example, consider a system with k_R=1/9 [1/s], χ=0.6, and Δt=1 [s], the interspecies transition probability matrix after one time step is

	A	B	lost	∑
A	0.9	0.06	0.04	1
B	0	1	0	1

Alternatively, the interspecies transition probabilities can be derived **from the analytical solution** as follows. Consider the same reaction $A \overset{k_R}{\rightarrow} B$ with analytical solution for A:

$$[A] = a_0 exp(-k_R t)$$

normalizing it such that the area under the curve is equal to one

$$\frac{a_0 exp(-k_R t)}{\int_0^\infty a_0 exp(-k_R t)dt} = \frac{a_0 exp(-k_R t)}{\left(\frac{a_0}{k_R}\right)} = k_R exp(-k_R t). \tag{3.22}$$

For this example, the normalized function is the exponential probability distribution, but if the integral does not converge, then we may use the numerical derivation instead. We can regard the area under eq. (3.22) as the fraction of chemical that has reacted from time zero until time Δt; then the probability of not reacting (i.e., the fraction of remaining chemical) during the same period is

$$P(A_{\Delta t}|A_0) = 1 - \int_0^{\Delta t} k_R exp(-k_R \tau)d\tau$$

$$= 1 - \left(-\frac{k_R}{k_R}[exp(-k_R \Delta t) - exp(0)]\right)$$

$$= 1 - \big(1 - exp(-k_R \Delta t)\big)$$

$$= exp(-k_R \Delta t).$$

Since the exponential probability distribution has the Markov property, the result can be reexpressed as a recurrent function

$$P(A_{t+\Delta t}|A_t) = exp(-k_R \Delta t) \tag{3.23}$$

then the transition probability from A to B is

$$P(B_{t+\Delta t}|A_t) = \chi\big(1 - exp(-k_R \Delta t)\big) \tag{3.24}$$

where χ is the efficiency coefficient.

Example, say we want to know [A] at time t=3. Taking steps of Δt=1, we have $a_0 exp(-k_R) exp(-k_R) exp(-k_R) = a_0 exp(-3k_R)$, which yields the same result as taking one step of Δt=3, which is the same as the value we get from the analytical solution.

3.10 Reaction-Diffusion Probability

Reaction and diffusion are described by independent distributions; since the diffusion probability is based on the expected concentration, we may use the law of total expectation and the law of total variance to compute the expected concentration and variance for each particular cell and time.

For simplicity, let us formulate the expected concentrations in terms of two chemicals {A, B} to show how they are computed:

$$A_{i\,j}^{t+1} = E\left[A_{i\,j}^{t+1}|A^t\right] P(A^{t+1}|A^t) + E\left[A_{i\,j}^{t+1}|B^t\right] P(A^{t+1}|B^t)$$

$$B_{i\,j}^{t+1} = E\left[B_{i\,j}^{t+1}|A^t\right] P(B^{t+1}|A^t) + E\left[B_{i\,j}^{t+1}|B^t\right] P(B^{t+1}|B^t)$$

where $E\left[A_{i\,j}^{t+1}|A^t\right]$ is the concentration of A coming from neighbor cells (by diffusion) conditional to them being A at time t, whereas $E\left[A_{i\,j}^{t+1}|B^t\right]$ is the concentration of A coming from neighbors conditional of them being B at time t. Likewise for B. Transition probabilities $p(A^{t+1}|A^t)$, $p(A^{t+1}|B^t)$, $p(B^{t+1}|A^t)$, and $p(B^{t+1}|B^t)$ have the same meaning as in 3.9.

There is some freedom to choose when the transition between species takes place: before, halfway or after the particle has moved. For simplicity in computation, in this work it was decided the interspecies change occurs after molecules 'jump' from one position to another.

The general formulation for concentration and variance is

$$\varphi_{i\,j}^{t+1} = E\left\{E\left[\varphi_{i\,j}^{t+1}|k^t\right]\right\} \tag{3.25}$$

$$V\left[\varphi_{i\,j}^{t+1}\right] = E\left\{V\left[\varphi_{i\,j}^{t+1}|k^t\right]\right\} + V\left\{E\left[\varphi_{i\,j}^{t+1}|k^t\right]\right\} \tag{3.26}$$

where $\varphi, k \in \{'chemical\,species'\}$; the conditional probabilities come from diffusion distribution, and the expectation and variance operators applied to them come from the reaction distribution.

3.11 Advection Term

It is possible to include an advection term using the procedure described in eq. (3.9-10) such that

$$p_{i+1} = D_x \frac{\Delta t}{\Delta x^2} + V_x \frac{\Delta t}{2\Delta x}$$

$$p_{i-1} = D_x \frac{\Delta t}{\Delta x^2} - V_x \frac{\Delta t}{2\Delta x}$$

$$p_0 = 1 - 2D_x \frac{\Delta t}{\Delta x^2}$$

where V_x is the x component of advection although it imposes the following restriction $2D_x \geq V_x\Delta x$. Using the method in eq. (3.9-10) is possible to show that $V_x = E[\Delta x]/\Delta t$. The above probability distribution meets this requirement ($E[\Delta x] = \Delta t V_x$) and that of diffusion ($E[\Delta x^2] = 2D_x\Delta t$). See Appendix B.

3.12 Consistency, Order, Stability, and Convergence

In conformance with standard engineering practices in using numerical methods as laid out by Hoffman [34], it is important to ensure that the discrete system has certain properties to yield a reliable and accurate approximation of the original PDE solution. These properties are

1. Consistency
2. Order
3. Stability
4. Convergence

Consistency refers to the truncation error between the finite difference and the PDF solution. If the error vanishes as space and time increments approach to zero, then the finite difference is consistent with the PDE. The *order* refers to the rate at which the global error decreases as the space and time increments approach to zero. For a stable PDE, a finite difference method is *stable* if it produces a bounded solution and *unstable* if such solution is unbounded. *Convergence* refers to the solution of the finite difference, if it approaches the exact solution of the PDE as space and time increments approach to zero, then the finite difference converges to the PDE [34].

A 1D diffusion problem is used to test these properties for simplicity in notation, noting that these results can be also be proven for 2D and 3D simply by introducing more terms. To test consistency, each term of the finite difference must be expressed using a Taylor series, then simplify the expression and approach the space and time increment to zero. The original PDE must be obtained from the finite difference.

The finite difference is

$$\frac{\varphi_i^{t+1} - \varphi_i^t}{\Delta t} = D_x \left(\frac{\varphi_{i+1}^t - 2\varphi_i^t + \varphi_{i-1}^t}{\Delta x^2} \right)$$

rearranging

$$\varphi_i^{t+1} = \varphi_i^t + D_x \frac{\Delta t}{\Delta x^2} (\varphi_{i+1}^t - 2\varphi_i^t + \varphi_{i-1}^t). \tag{3.27}$$

The base point for the Taylor series expansion is x = i and t. The expansions are

$$\varphi_i^{t+1} = \varphi_i^t + \dot{\varphi}|_i^t \,\Delta t \ + \frac{1}{2}\ddot{\varphi}|_i^t \,\Delta t^2 + \cdots$$

$$\varphi_{i+1}^t = \varphi_i^t + \varphi_x|_i^t \,\Delta x \ + \frac{1}{2}\varphi_{xx}|_i^t \,\Delta x^2 + \cdots$$

$$\varphi_{i-1}^t = \varphi_i^t - \varphi_x|_i^t \,\Delta x \ + \frac{1}{2}\varphi_{xx}|_i^t \,\Delta x^2 - \cdots$$

plugging in those series into eq. (3.27)

$$\varphi_i^t + \dot{\varphi}|_i^t \,\Delta t + \frac{1}{2}\ddot{\varphi}|_i^t \,\Delta t^2 + \cdots \ = \varphi_i^t + D_x \frac{\Delta t}{\Delta x^2} \,(\varphi_i^t + \varphi_x|_i^t \,\Delta x \ + \frac{1}{2}\varphi_{xx}|_i^t \,\Delta x^2 + \cdots$$

$$-2\varphi_i^t + \varphi_i^t - \varphi_x|_i^t \,\Delta x \ + \frac{1}{2}\varphi_{xx}|_i^t \,\Delta x^2 - \cdots)$$

cancelling zero-order terms and dividing by Δt

$$\dot{\varphi}|_i^t \ + \frac{1}{2}\ddot{\varphi}|_i^t \,\Delta t + \cdots$$
$$= \frac{D_x}{\Delta x^2} \,(\varphi_x|_i^t \,\Delta x \ + \frac{1}{2}\varphi_{xx}|_i^t \,\Delta x^2 + \cdots - \varphi_x|_i^t \,\Delta x \ + \frac{1}{2}\varphi_{xx}|_i^t \,\Delta x^2 - \cdots)$$

simplifying right-hand side terms and distributing the division by Δx^2

$$\dot{\varphi}|_i^t + \frac{1}{2}\ddot{\varphi}|_i^t \, \Delta t + \cdots = D_x \left(\varphi_{xx}|_i^t + \frac{1}{12}\varphi_{xxxx}|_i^t \, \Delta x^2 + \cdots \right) \qquad (3.28)$$

approaching Δt to zero and Δx to zero, the eq. (3.28) yields

$$\dot{\varphi} = D_x \, \varphi_{xx}$$

the original PDE. The order of the finite difference is obtained from eq. (3.28), which is $O(\Delta t) + O(\Delta x^2)$.

We performed a Neumann stability analysis on the finite difference using the following steps

1. Replace the terms φ_{i+1} and φ_{i-1} by Fourier complex components and simplify:

$$\varphi_i^{t+1} = \varphi_i^t + D_x \frac{\Delta t}{\Delta x^2}(\varphi_{i+1}^t - 2\varphi_i^t + \varphi_{i-1}^t)$$

$$\varphi_i^{t+1} = \varphi_i^t + D_x \frac{\Delta t}{\Delta x^2}(\varphi_i^t \, exp(i\theta) - 2\varphi_i^t + \varphi_i^t \, exp(-i\theta))$$

$$\varphi_i^{t+1} = \varphi_i^t \left[1 + D_x \frac{\Delta t}{\Delta x^2} \; (exp(i\theta) - 2 + exp(-i\theta)) \right]$$

$$\varphi_i^{t+1} = \varphi_i^t \left[1 + 2 \, D_x \frac{\Delta t}{\Delta x^2} \left(\frac{exp(i\theta) + exp(-i\theta)}{2} - 1 \right) \right]$$

2. Express the equation in terms of sin θ and cos θ, and determine the amplification factor G_A:

$$\varphi_i^{t+1} = \varphi_i^t \left[1 + 2 \, D_x \frac{\Delta t}{\Delta x^2} (cos(\theta) - 1) \right]$$

3. Analyze if $| G_A| \leq 1$ to determine stability:

$$G_A = 1 + 2 \, D_x \frac{\Delta t}{\Delta x^2} (cos(\theta) - 1)$$

$$-1 \leq 1 + 2 \, D_x \frac{\Delta t}{\Delta x^2} (cos(\theta) - 1) \leq 1$$

$$-1 \leq D_x \frac{\Delta t}{\Delta x^2} (cos(\theta) - 1) \leq 0$$

The upper limit holds always since D_x, Δx, and Δt are positive and max(cos θ - 1) = 0 and min(cos θ - 1) = -2.

The lower limit yields

$$D_x \frac{\Delta t}{\Delta x^2} \leq \frac{1}{1 - cos(\theta)}$$

which is undetermined for max(1-cos θ) = 0 and ½ for max(1-cos θ) = 2. Therefore, the minimum value for $D_x(\Delta t/\Delta x^2)$ is ½ and the finite difference is stable:

$$D_x \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}.$$

The *Lax equivalence theorem* is used to show convergence of the method, which states "Given a properly posed linear initial-value problem and a finite difference approximation to it that is consistent, stability is necessary and sufficient condition for convergence".

Therefore, our probability-based method is consistent, stable, and converges to the true solution of the PDE. Its order is O(Δt) + O(Δx^2) for 1D.

In the next chapter we apply this method to estimate the lifetimes of nuclear waste containers.

Chapter 4 Estimating Lifetimes of Nuclear Waste Containers

INTRODUCTION

This chapter describes the implementation of the cellular automata (CA) model to estimate lifetimes of used fuel containers (UFCs). First, this chapter explains the sulphate-reducing bacteria (SRB) threat and the steps followed to model their sulphide production inside the deep geological repository (DGR). Then the coupled CA model is presented to account for diffusion and reaction inside the DGR. Then the validation process is presented by replicating published results and solving equivalent finite differences problems. Some the scenarios of interest are presented as well as a sensitivity analysis. Finally, some conclusions are discussed about the advantages and disadvantages of CA models.

4.0 Assumptions and limitations

The basic assumptions and limitations behind the modeling and calculation of lifetimes presented in this chapter are the following

- Models do not account for any copper corrosion during the aerobic period of the DGR. Although it is possible to include an estimate of corrosion during that period into lifetime calculation.
- The activity of Sulphate reducers is well approximated by a first-order reaction rate.
- No other chemical produces any relevant interaction in terms of affecting diffusion, sulphide production, or copper corrosion.
- 1D and 2D models describe the endcaps of the containers because that is likely the region with the highest corrodent concentration.
- All the input values were drawn from literature and represent best estimates or experimental results. Therefore, the results of the models presented here depend on the accuracy of such values.

4.1 Sulphate-Reducing Bacteria

Sulphate-reducing bacteria (SRB) refers to a diverse group of anaerobic microorganisms that degrade organic compounds by reducing sulphate to sulphide. They are ubiquitous in anoxic habitats, whether natural or human-made, e.g., marine sediments, hydrothermal vents, mud volcanoes, oil fields, deep sub-surface, and waste-water treatment plants [35]. They have been found in habitats with extreme pressure and temperatures making them a potential threat to the barrier system protecting UFCs.

The chemical reaction describing the sulphate-reducing mechanism carried out by SRB is [35]

$$4H_2 + SO_4^{2-} + H^+ \longrightarrow HS^- + 4H_2O. \tag{4.1}$$

If the reaction rate is known, sulphate reduction to sulphide can be determined. Assuming (4.1) is an elementary reaction, the following ODEs describe the reduction of SO_4^{2-} and the production of HS^-:

$$\dot{\varphi}^{(SO_4)} = -k_R\left[\varphi^{(H_2)}\right]^4 \varphi^{(SO_4)}\varphi^{(H^+)} \tag{4.2a}$$

$$\dot{\varphi}^{(HS^-)} = k_R\left[\varphi^{(H_2)}\right]^4 \varphi^{(SO_4)}\varphi^{(H^+)} \qquad (4.2b)$$

where k_R is the reaction coefficient and $\varphi^{(\cdot)}$ is the concentration of the chemical species in [mol/L]. The approach followed in this chapter was to use experimental results about SRB activity under similar conditions and obtain the reduction rate of sulphate. Then use that rate to describe sulphide production but accounting for efficiency on the reaction. Finally compare the reaction rate to similar estimations.

4.1.1 Reaction Rate

Data from [36] and [37] was used, who performed an experiment to study copper corrosion on the presence and absence of SRB. Copper specimens were exposed to biotic and abiotic artificial anoxic groundwaters for periods of 4 and 10 months and then examined their chemical and electrochemical changes. The microorganisms were extracted from the deep bedrock of Okiluoto, Finland. Relevant results to our study are summarized on Table 4.1.

Table 4.1 Composition of groundwater before and after tests [mg/L]

Chemical	Initial	4 months	10 months
pH	7.78	7.18	7.09
SO_4	582	580	483
S^{2-}		<0.01	8.08
HS^-			8.08

A detail to notice is that pH decreased after the experiment, meaning that the concentration of hydrogen ions increased, since $[H^+]=10^{-pH}$ from $pH = -\log_{10}[H^+]$. It would be expected to decrease by reaction (4.1), it suggests that other reactions took place increasing the hydrogen ion concentration.

Since hydrogen ions were not limiting the reaction, they were assumed to be in abundance for groundwater with similar characteristics, this is a conservative, bounding scenario as the real rate of sulphide production will be significantly less than this. Additionally, the electrolysis of water into hydrogen gas (H_2) and oxygen (O_2) was assumed to be faster than sulphate reduction to represent a conservative scenario; therefore, hydrogen gas was also considered in abundance inside the DGR. However, the use of HCB would prevent the formation of a biofilm on the surface of the UFCs as it did in the experiments, therefore, this sulphide production rate is an extreme boundary condition representing the worst-case scenario in which every engineered barrier system fails [67].

Therefore, a conservative, bounding upper limit of the reaction rate is described by

$$\dot{\varphi}^{(SO_4)} = -k_R \varphi^{(SO_4)} \tag{4.3a}$$

$$\dot{\varphi}^{(HS^-)} = k_R \chi \varphi^{(SO_4)} \tag{4.3b}$$

where χ is the efficiency of the reaction, $\varphi^{(.)}$ is in [mol/L]. If $\varphi^{(.)}$ is expressed in [mg/L], then the right-hand side of eq. (4.3b) must be multiplied by κ_{HS} = 33.07/96.06 to account for the mole-to-mole conversion.

4.1.2 Optimal Reduction Rate Coefficient

The analytical solution of (4.3a) describes an exponential decay:

$$\varphi^{(SO_4)} = a_0 exp(-k_R t) \tag{4.4}$$

a_0 can be determined from the initial conditions. For k_R however, there are two data points. Optimization was used to find the k_R* that minimizes the error with respect to the observed data. Two objective functions were tested: the sum of squared differences and the sum of squared log differences.

The first optimization model is

$$Min\ \epsilon = \frac{1}{2}\left(d_1^{(SO_4)} - \hat{\varphi}_1^{(SO_4)}\right)^2 + \frac{1}{2}\left(d_2^{(SO_4)} - \hat{\varphi}_2^{(SO_4)}\right)^2 \tag{4.5}$$

subject to

$$\hat{\varphi}_i^{(SO_4)} = a_0 exp(-k_R t_i), \qquad \forall i = 1,2$$

$$k_R \geq 0$$

$d_1^{(SO4)}$ and $d_2^{(SO4)}$ are the two data points, k_R is the decision variable. After differentiating (4.5) with respect to k_R and using the bisection algorithm to find $\frac{\partial \epsilon}{\partial k_R} = 0$, we obtained k_R*= 0.015671/month = 5.96309×10^{-9}/s after 15 iterations. Figure 4.1 shows a plot of the iterative approximation to zero and the error function.

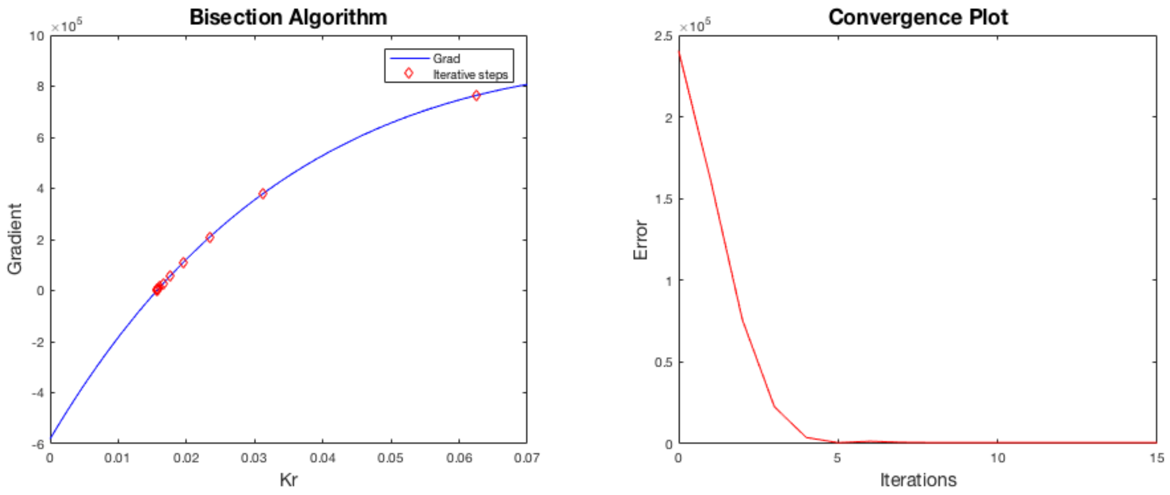


Figure 4.1 Bisection iterative approximation to zero and convergence plot

The second optimization model is

$$Min\ \epsilon = \frac{1}{2}\left\{ln\left[d_1^{(SO_4)}\right] - ln\left[\hat{\varphi}_1^{(SO_4)}\right]\right\}^2 + \frac{1}{2}\left\{ln\left[d_2^{(SO_4)}\right] - ln\left[\hat{\varphi}_2^{(SO_4)}\right]\right\}^2 \qquad (4.6)$$

subject to

$$ln\left[\hat{\varphi}_i^{(SO_4)}\right] = -k_R t_i, \qquad \forall i = 1,2$$

$$k_R \geq 0$$

Differentiating (4.6) with respect to k_R, setting $\frac{\partial \epsilon}{\partial k_R} = 0$ and solving for k_R, we get

$$k_R^* = \frac{t_1 ln\left(\frac{a_0}{d_1^{(SO_4)}}\right) + t_2 ln\left(\frac{a_0}{d_2^{(SO_4)}}\right)}{{t_1}^2 + {t_2}^2}$$

The optimal value is k_R* = 0.016192/month = 6.16146×10^{-9}/s. Both values are compared in Table 4.2. The highest value was chosen **k_R* = 0.016192/month = 6.16146×10^{-9}/s** because it relates to the most unsafe situation in addition to have the lowest error in two out of three error measures.

Table 4.2 Comparison of optimal values

Model	k_R*	SSE[1]	SSLE[2]	SAE[3]
1	5.96309×10^{-9}/s	662.8	2.19×10^{-3}	47.9
2	6.16146×10^{-9}/s	667.0	2.18×10^{-3}	46.5

[1] Sum of squared errors. [2] Sum of squared log errors. [3] Sum of absolute errors

The **reduction of sulphate** (4.3a) becomes

$$\dot{\varphi}^{(SO_4)} = -6.16146 \times 10^{-9}\varphi^{(SO_4)} \tag{4.7}$$

with units [mg*L^{-1}*s^{-1}] if $\varphi^{(SO4)}$ is in [mg/L] or [mol*L^{-1}*s^{-1}] if $\varphi^{(SO4)}$ is in [mol/L]. Its solution is

$$\varphi^{(SO_4)} = 582exp(-6.16146 \times 10^{-9}t) \tag{4.8}$$

with units [mg*L^{-1}] if $\varphi^{(SO4)}$ is in [mg/L] or

$$\varphi^{(SO_4)} = 6.06 \times 10^{-3}exp(-6.16146 \times 10^{-9}t) \tag{4.9}$$

with units [mol*L^{-1}] if $\varphi^{(SO4)}$ is in [mol/L].

4.1.3 Sulphide Production

To determine the efficiency parameter χ the total amount of sulphide produced must be known. From stoichiometry we know that about one millimoles of sulphate reacted (1.03[mmol/L] ≈ 99 [mg/L]) which must have produced the same millimoles of sulphide (1.03[mmol/L] ≈ 34.6 [mg/L]) had it not been reduced by another reaction. Therefore, the following relation must hold

$$\varphi_G^{(HS^-)} + \varphi_C^{(HS^-)} + \varphi_N^{(HS^-)} = 34.6\ [mg/L]$$

$\varphi_G^{(HS)}$, $\varphi_C^{(HS)}$, and $\varphi_R^{(HS)}$ are the sulphide measured in groundwater, that reacted with copper, and that not reacted/reacted in a different reaction. The total amount of sulphide produced is the sum of the first two terms. The first is given in the table; the second can be estimated from the amount of copper corroded. The weight lost was 0.8 mg/cm^2 of a copper coupon with area of 17.5 cm^2 (7cm×2.5cm) giving a total mass loss of 140 mg of Cu. That amount is more than 2 millimoles (2.2 mmol ≈ 140 mg), from stoichiometry the sulphide reacted with copper must have been 1.1 mmol ≈ 36.377 mg (or 2.2 mmol using the path reaction mentioned in [36, 37]), which divided by the 12 L where the experiment was performed, we get 3.0314 [mg/L] (or 6.0628 [mg/L]) of sulphide reduced. The highest number was chosen because it is the most unsafe case:

$$\varphi_G^{(HS^-)} + \varphi_C^{(HS^-)} = 8.08\,[mg/L] \,+\, 6.0628\,[mg/L] = 14.1428\,[mg/L].$$

The efficiency parameter becomes

$$\chi = \frac{14.1428\,[mg/L]}{34.6\,[mg/L]} = 0.408751$$

which is unitless. The **production of sulphide** (4.3b) after multiplying by κ_{HS} to account for the mole-to-mole production becomes

$$\dot{\varphi}^{(HS^-)} = 8.670301 \times 10^{-10} \varphi^{(SO_4)} \tag{4.10}$$

with units [mg*L^{-1}*s^{-1}] if $\varphi^{(SO4)}$ is in [mg/L] or

$$\dot{\varphi}^{(HS^-)} = 2.518503 \times 10^{-9} \varphi^{(SO_4)} \tag{4.11}$$

with units [mol*L^{-1}*s^{-1}] if $\varphi^{(SO4)}$ is in [mol/L].

Figure 4.2 shows a picture of the chemical reaction in milligrams and moles.

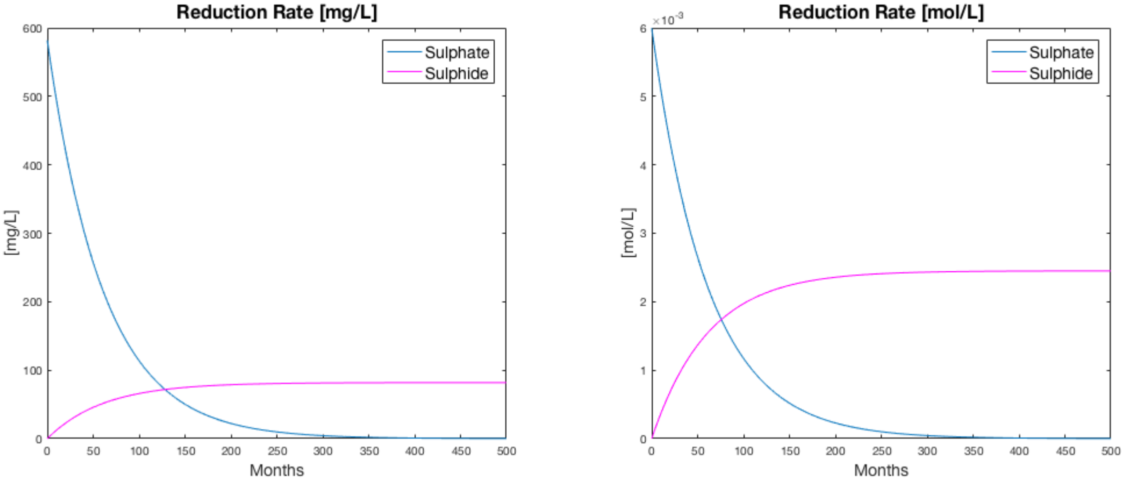


Figure 4.2 SRB-driven chemical reaction

4.1.4 Density-Dependent Reaction Rate

After considering that the DGR conditions are going to be different from the conditions of the experiments, it was decided to adjust the reaction rate using dry density. Dry density was chosen because it determines the pore size of the clay and the swelling pressure, which are the two major factors limiting SRB growth.

The **linear decay** obtained is

$$k_A(\rho_D) = k_R \frac{\rho_S - \rho_D}{\rho_S - 1}, \qquad \rho_D \in [1, \rho_S] \tag{4.12}$$

ρ_D is the dry density of the clay in [g/m^3] and ρ_S is the specific density ($\rho_S = 2.76$ [g/cm^3]), which is the density of bentonite with no voids in it, i.e., its maximum density possible.

For an **exponential decay**, it is known that the exponential function is never zero, meaning that density could be arbitrarily large and still have a positive reaction rate coefficient, which is unrealistic; therefore, a constant (C_E) was included to represent a value at which we will consider the rate to be zero:

$$k_A(\rho_D) = k_0 exp[-b_E(\rho_D - 1)] - C_E$$

$k_0 = k_R + C_E$, after some algebra and eliminating b_E from $k_A(\rho_S)=0$ and replacing back into the function (See Appendix C), the **exponential decay** is obtained

$$k_A(\rho_D) = \left[\left(\frac{k_R}{C_E} + 1\right)^{\frac{\rho_S - \rho_D}{\rho_S - 1}} - 1\right] C_E, \qquad \rho_D \in [1, \rho_S] \tag{4.13}$$

C_E must be selected such that $0 < C_E << k_R$, e.g. $C_E = k_R \times 10^{-4}$. Dividing eq. (4.13) by k_R gives the probability of having active SRB (See appendix C), therefore the equation is just an approximation to model local density heterogeneity that may allow SRB to remain active. However, it should be noted that at uniform density of ~1.6 [g/cm^3] SRB activity is suppressed [68].

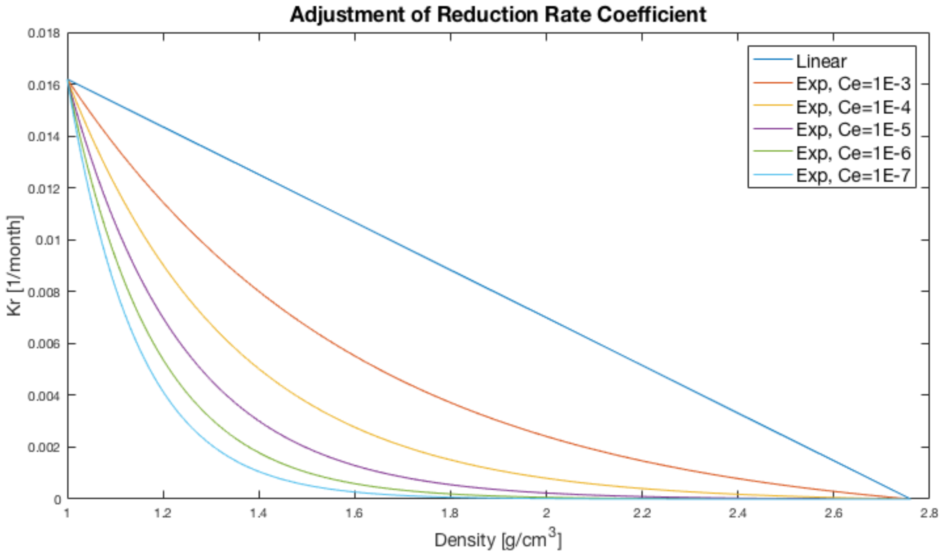


Figure 4.3 Linear and exponential adjustment of reaction rate

Sulphate to sulphide reduction rates estimated by Bengtsson and Pedersen in [38] were used to determine C_E. They did a series of experiments to study SRB activity inside different types of clays for two densities and three incubation days. Table 4.3 shows their estimations and our correspondent values. It is worth mentioning that their values are lower bounds. The exponential adjustment (k_A) was obtained by minimizing the error with respect to five of the six experimental results. The result for density 1,562 [kg/m^3] and 77 incubation days was excluded for it being inconsistent with the rest of the data.

Table 4.3 Comparison of reduction rates of SRB in bentonite MX-80

Dry Density [kg/m^3]	Wet Density [kg/m^3]	Incubation [days]	Sulphate to sulphide reduction rate [mol s^{-1} m^{-3}]	Source
1,000	1,000	Not adjusted	2.5×10^{-6}	Ours
1,171	1,750	47	8.2×10^{-7}	[38]
1,171	1,750	77	3.2×10^{-7}	[38]
1,171	1,750	123	5.0×10^{-8}	[38]
1,171	1,750	Linearly adjusted	2.2×10^{-6}	Ours
1,171	1,750	Exp. adjusted*	3.97×10^{-7}	Ours
1,562	2,000	47	6.1×10^{-9}	[38]
1,562	2,000	77	9.2×10^{-11}	[38]
1,562	2,000	123	3.2×10^{-10}	[38]
1,562	2,000	Linearly adjusted	1.7×10^{-6}	Ours
1,562	2,000	Exp. adjusted*	5.79×10^{-9}	Ours

Calculated with C_E* = 1.38×10^{-14}

The optimization model is

$$Min\,\epsilon = \frac{1}{2}\sum_{i=1}^{5}\left(k_D^{(i)} - k_A^{(i)}\right)^2$$

subject to

$$k_A^{(i)} = \left[\left(\frac{k_R}{C_E} + 1\right)^{\frac{\rho_S - \rho_D^{(i)}}{\rho_S - 1}} - 1\right] C_E, \qquad \forall i = 1, \ldots, 5;\ C_E > 0$$

where C_E is the decision variable, $k_D^{(i)}$ are the values from the experiment, k_R is our unadjusted estimation. Gradient descent was used to find **C_E*= 1.375159 × 10^{-14}**.

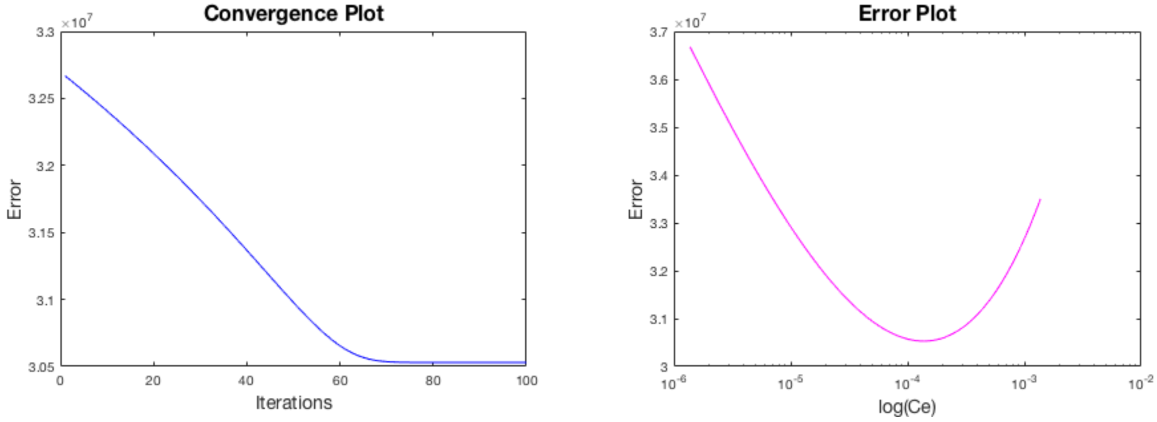


Figure 4.4 Convergence and error plot for parameter C_E

4.2 Reaction – Diffusion Model

The system of interest consists of two chemical species: sulphate and sulphide. The host rock of the emplacement acts as the source of sulphate. In between the rock and the UFC are two layers of clay with different densities, diffusivities and different probabilities of hosting and thriving SRB. Sulphate travels by diffusion from the rock to the clays and the UFC; if at any point SRB is present and active, sulphide will be produced. When sulphide interacts with the copper surface of the UFC, a corrosion reaction takes place. This reaction is assumed to be instantaneous.

The following problem is described in 1D by

$$\dot{\varphi}^{(SO_4)} = D_e(x, SO_4)\, \varphi_{xx}^{(SO_4)} + R^{(SO_4)}(x) \tag{4.14a}$$

$$\dot{\varphi}^{(HS^-)} = D_e(x, HS^-)\, \varphi_{xx}^{(HS^-)} + R^{(HS^-)}(x) \tag{4.14b}$$

with initial conditions

$\varphi^{(SO_4)}(x, 0) = h_0^{(SO_4)}(x)$

$\varphi^{(HS^-)}(x, 0) = h_0^{(HS^-)}(x)$

and boundary conditions

$\varphi^{(SO_4)}(0, t) = h_1^{(SO_4)}(t)$

$\varphi^{(SO_4)}(L_x, t) = h_2^{(SO_4)}(t)$

$\varphi^{(HS^-)}(0, t) = h_1^{(HS^-)}(t)$

$\varphi^{(HS^-)}(L_x, t) = h_2^{(HS^-)}(t)$

where $\varphi^{(SO4)}$ and $\varphi^{(HS-)}$ are the concentration of the chemical species in [mg/L], L_x is the right boundary dividing HCB and the surface of the UFC. $D_e(x, SO_4)$ and $D_e(x, HS^-)$ are the diffusivity coefficients for sulphate and sulphide depending on x. $R^{(SO4)}(x)$ and $R^{(HS-)}(x)$ are the reaction terms as a function of x. Solving such system of PDEs is challenging because of the space dependent diffusivity and reaction terms.

4.2.1 Cellular Automata Model

The equivalent CA representation of the same problem is a grid of cells with attributes or internal variables representing chemical species in [mg/L] and parameters. A 2D model was created to solve the deterministic case and another 1D model to solve stochastic scenarios and validation.

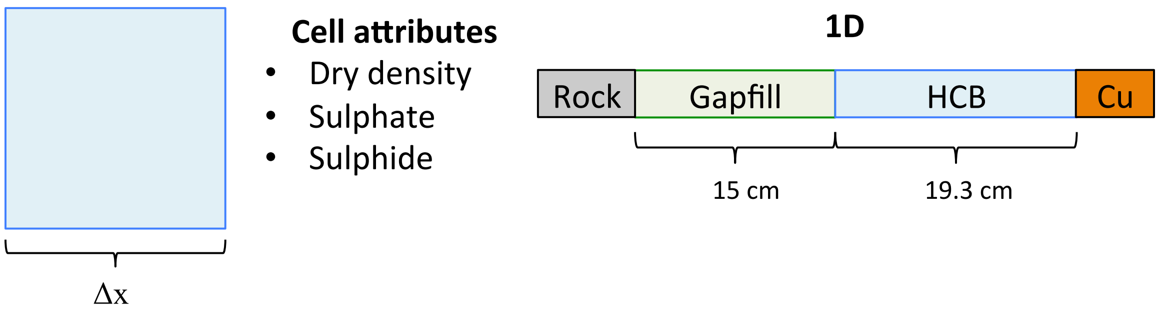


Figure 4.5 1D Cellular automata

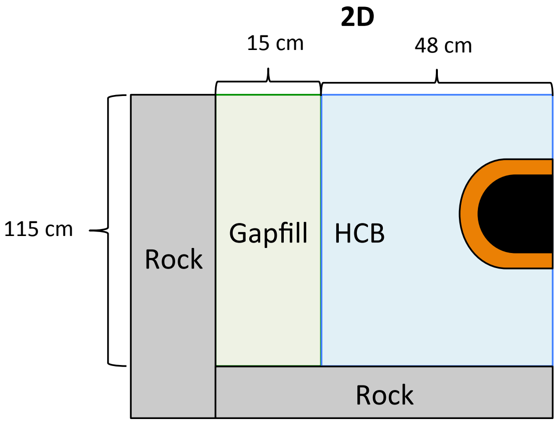


Figure 4.6 2D Cellular automata (image not to scale)

4.2.2 Diffusion

The probabilities of movement were calculated using the procedure described in Chapter 3. For 1D is

$$p_{i\pm1} = D_e \frac{\Delta t}{\Delta x^2}$$

$$p_0 = 1 - 2p_{i\pm1}$$

for 2D is

$$p_{j\pm1} = D_e \frac{\Delta t}{\Delta y^2}$$

$$p_{i\pm1} = D_e \frac{\Delta t}{\Delta x^2}$$

$$p_0 = 1 - 2p_{i\pm1} - 2p_{j\pm1}$$

where D$_e$ is the diffusivity coefficient obtained by eq. (2.4), p$_0$ is the probability to remain on the same cell, p±i is that of moving one cell ahead or backwards x-wise, and p±j is that on y-wise. For the cells located at the interface of GF and HCB, the probability of movement in between them p$_I$ is

$$p_I = \frac{2p_L p_R}{p_L + p_R} \tag{4.15}$$

where p$_L$ is p±i of the left cell and p$_R$ is p±i of the right cell.

4.2.3 Reaction

The interspecies transition probabilities derived in section 3.9 were used. The reaction probabilities from the analytical solution are

$$p\left({SO_4}^{t+1}|{SO_4}^{t}\right) = exp(-k_A \Delta t)$$

$$p\left(HS^{t+1}|{SO_4}^{t}\right) = \chi\left(1 - exp(-k_A \Delta t)\right)$$

$$p\left({SO_4}^{t+1}|HS^{t}\right) = 0$$

$$p(HS^{t+1}|HS^{t}) = 1$$

with k$_A$ dependent on the dry density of each patch *i, j* and computed according to eq. (4.13).

4.2.4 Reaction-Diffusion

Each cell of the grid computes the concentration for the next time step according to eq. (3.25)

$$\varphi^{(SO_4)}_{t+1,\ i,j} = E\left[\varphi^{(SO_4)}_{t+1,\ i,j}|{SO_4}^{t}\right]\ p\left({SO_4}^{t+1}|{SO_4}^{t}\right) + E\left[\varphi^{(SO_4)}_{t+1,\ i,j}|HS^{t}\right]\ p\left({SO_4}^{t+1}|HS^{t}\right)\ (4.16a)$$

$$\varphi^{(HS^-)}_{t+1,\ i,j} = E\left[\varphi^{(HS^-)}_{t+1,\ i,j}|{SO_4}^{t}\right]\ p\left(HS^{t+1}|{SO_4}^{t}\right) + E\left[\varphi^{(HS^-)}_{t+1,\ i,j}|HS^{t}\right]\ p(HS^{t+1}|HS^{t})\ (4.16b)$$

Variance is computed according to eq. (3.26).

4.2.5 Initial Conditions

Initial conditions depend on the amount of groundwater contained in the clays at the beginning of the simulation. Estimate values were used for dry and wet density given in

Table 2.1 of Chapter 2; the determination of water was made using two methods, both yielded the same result.

1) Multiplying porosity by saturation and by the number of liters of water in 1 [m^3]:

$$Water = Porosity * Saturation * 1000\ [L/m^3]$$

2) The second method is using wet and dry densities:

$$Water = \frac{(\rho_W - \rho_D)}{\rho_{water}} * 1000\ [L/m^3]$$

Table 4.4 Water content in 1 [m^3] of clay

Clay	Porosity	ρ_W [kg/m^3]	ρ_D [kg/m^3]	Water [L/m^3]
Gapfill	0.486	1439	1410	29
HCB	0.382	1955	1700	255

Deterministic case

For the crystalline rock, the initial amount of sulphate in GF is 1000 [mg/L] * 29 [L/m^3] = 29,000 [mg/m^3] = 29 [mg/L]. For HCB: 1000 [mg/L] * 255 [L/m^3] = 255,000 [mg/m^3] = 255 [mg/L]. The initial amount of sulphide is zero.

Table 4.5 Initial amounts per cell/patch [mg/L]

Clay	Sulphate	Sulphide
Gapfill	29	0
HCB	255	0

For sedimentary rock, the amout of sulphate is the groundwater is 310 [mg/L]. For GF: 310 [mg/L] * 29 [L/m^3] = 8,990 [mg/m^3] = 8.99 [mg/L]. For HCB: 310 [mg/L] * 255 [L/m^3] = 79,050 [mg/m^3] = 79.05 [mg/L]. The initial amount of sulphide is zero.

Stochastic case for randomness in groundwater

The concentration of sulphate was assumed to be log-normally distributed and have a standard deviation such that $\mu_{[so4]} + 3*\sigma_{[SO4]} = \ln(2*[SO_4])$

$$[SO_4]_{Gw} \sim Lognormal\left\{\mu_{[SO4]} = ln\,[SO_4], \sigma_{[SO4]} = \frac{ln(2)}{3}\right\}$$

where $[SO_4]_{Gw}$ is the concentration of sulphate in groundwater and $[SO_4]$ is the concentration from according to the estimates in Table 2.1.

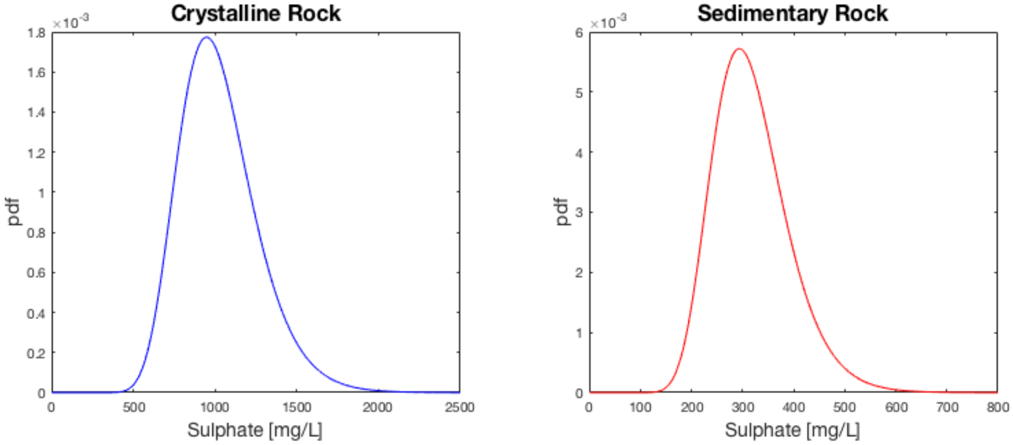


Figure 4.7 Assumed sulphate content in groundwater

4.2.6 Boundary Conditions

The rock interface was assumed to have a constant concentration of sulphate equal to the concentration contained in the groundwater or, for the stochastic case, a constant concentration drawn randomly from the lognormal distribution. The rock also acts as a barrier for sulphide, preventing any molecule from leaving the system.

Left boundary conditions

$$\varphi^{(SO_4)}_{t,\ i=0,\ j} = [SO_4]_{Gw}, \qquad \forall t, j$$

$$\frac{\partial}{\partial x}\varphi^{(HS^-)}_{t,\ i=0,\ j} = 0. \qquad \forall t, j$$

As mentioned previously, the reaction of sulphide and copper is assumed to occur infinitively fast, therefore the right boundary for sulphide is set to zero at any time. For sulphate however, it acts as a barrier.

Right boundary conditions

$$\frac{\partial}{\partial x}\varphi_{t,\ i=Lx,j}^{(SO_4)} = 0, \qquad \forall t, j$$

$$\varphi_{t,\ i=Lx,j}^{(HS^-)} = 0. \qquad \forall t, j$$

For the 2D model, the upper boundary consists of a block of clay with the same properties as the clay immediately below. They are also equally distant from the left and upper rock interfaces as the clays below are from the left and lower rock boundaries. Therefore, the upper clay should have the same concentration in both sulphate and sulphide as the clay below. This boundary was modeled by setting the gradient with respect to the y direction equal to zero.

Upper boundary conditions

$$\frac{\partial}{\partial y}\varphi_{t,\ i,j=Ly}^{(SO_4)} = 0, \qquad \forall t, j$$

$$\frac{\partial}{\partial y}\varphi_{t,\ i,j=Ly}^{(HS^-)} = 0. \qquad \forall t, j$$

The lower boundary is the rock interface; therefore, we used the same constant concentration for sulphate and a physical barrier for sulphide as in the left boundary.

Lower boundary conditions

$$\varphi_{t,\ i,j=0}^{(SO_4)} = [SO_4]_{Gw}, \qquad \forall t, i$$

$$\frac{\partial}{\partial y}\varphi_{t,\ i,j=0}^{(HS^-)} = 0. \qquad \forall t, i$$

4.3 Lifetime Calculation

4.3.1 Sulphide Flux

To determine the lifetime of the canisters, the flux of sulphide to the UFCs must be obtained first. Although Fick's first law describes diffusion flux under steady state, since our model has a discretized domain with discontinuous concentrations, to compute the flux between any two cells we used a time-varying flux:

$$J^t(L_x) = -D_e \frac{\partial \varphi^{(HS^-)}}{\partial x}, \qquad \forall t$$

using finite differences

$$J^t(L_x) = -D_e \frac{\varphi_{t,\ L_x}^{(HS^-)} - \varphi_{t,\ L_x-1}^{(HS^-)}}{\Delta x}, \qquad \forall t$$

from the boundary conditions, we know $\varphi_{t,\ L_x}^{(HS^-)} = 0$ for all t, therefore

$$J^t(L_x) = D_e \frac{\varphi_{t,\ L_x-1}^{(HS^-)}}{\Delta x}, \qquad \forall t$$

The amount of sulphide flowing to the copper layer of the UFC is

$$N_{HS^-}^t = \frac{J^t(L_x) \times A_{Corr} \times \Delta t}{M_{HS}}, \qquad \forall t$$

N^t_{HS} is in [mol/t], A_{Corr} is the area where corrosion will take place [m^2] and M_{HS} is the molar mass of sulphide [g/mol].

4.3.2 Corrosion depth

This work used a time-dependent formulation of eq. (2.3) to determine the depth of corrosion per unit of time [m/t]

$$d_{Corr}^t = \frac{N_{HS}^t\, f_{HS}\, M_{Cu}}{A_{Corr}\, \rho_{Cu}}. \qquad \forall t$$

The total depth corrosion from time one to T is then obtained by

$$\tau_{Corr} = \sum_{t=1}^{T} d_{Corr}^t \times \Delta t$$

τ_{Corr} is the accumulated from t = 1 to t = T in units [m], T is the number of time steps simulated in the model [t].

4.3.3 Determination of the lifetime of canisters

The lifetime of the canister is then computed as follows

$$Lifetime = \frac{0.003 - \tau_{Corr}}{d_{Corr}^S}$$

where $d_{Corr}{}^{S}$ is the depth of corrosion at steady-state [m/t], which is usually reached after 1,200 years.

4.4 Verification and Validation

The validation of the CA models in this thesis was made by replicating recent published results. This process proved to be useful also for troubleshooting and ensuring the right conversion of units. Verification of the models came from finding the source of discrepancies between the published results and the model predictions. This does not mean that we tuned the CA models to produce the published results, but rather those results were used to detect coding or unit conversion errors. Validation scenarios are described in Table 4.6, it is worth noting that none of them model SRB activity; rather, authors acknowledged that SRB may be present and used assumed values for sulphide.

Table 4.6 Validation scenarios for CA model

Validation scenario	Author	Sulphide at rock interface	CA model	Comments
I	Järvine [21]	30 ppb	1D	No reaction, only diffusion of sulphide.
II	Briggs [22]	3 ppm	1D and 2D	No reaction, only diffusion of sulphide.

4.4.1 Validation Scenario I

Results reported in [21] assumed a constant concentration of 30 ppb of sulphide, at an average distance of 0.165 [m] from the copper surface and diffusivity $D_e = 1\times10^{-11}$ [m^2/s]. We solved the system using a 1D CA with $\Delta x = 0.5$ [cm] and $\Delta t = 1/40$ [year], and 1D implicit finite differences (IFD) scheme with same space increment and $\Delta t = 1$ [year]. Steady state is reached after ~250 years.

Table 4.7 Lifetimes for Validation Scenario I [year]

Statistic	Results [21]	CA (1D)	IFD (1D)
Mean	9.6×10^7	11.2×10^7	10.7×10^7
Minimum	1.3×10^7	2.0×10^7	2.0×10^7
Maximum	5.8×10^8	4.8×10^8	4.7×10^8
Mode	8.0×10^7	9.0×10^7	8.0×10^7
Skewness		1.12	1.14
Kurtosis		5.20	5.27
SD		4.2×10^7	4.0×10^7
Realizations	95,737	32,000	95,000

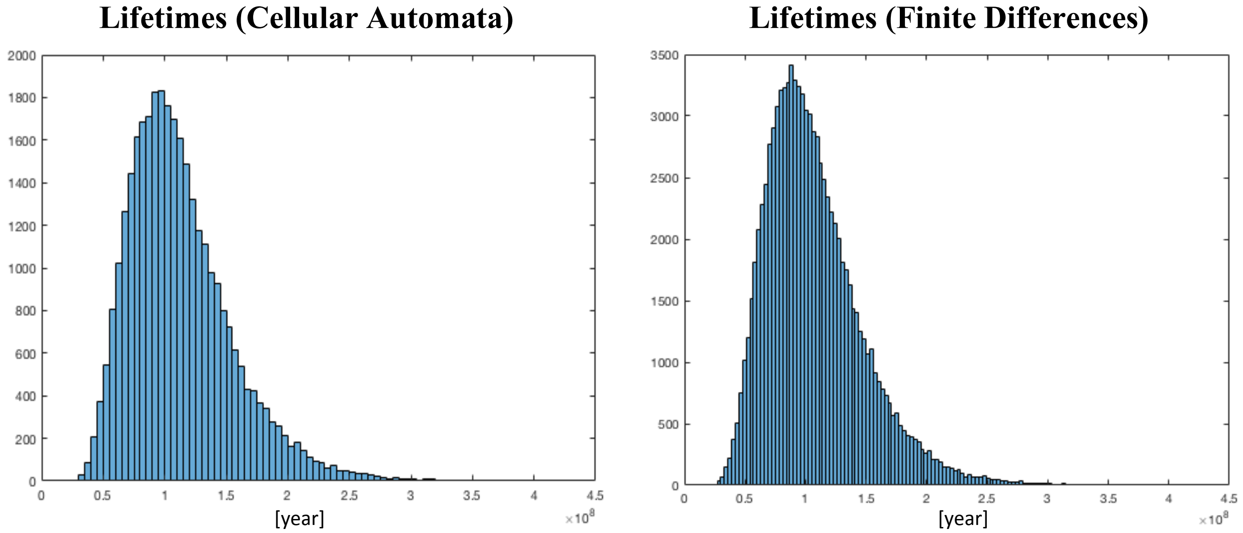


Figure 4.8 Histogram of UFC lifetimes for Validation Scenario I

4.4.2 Validation Scenario II

Results reported in [22] assumed a constant concentration of 3 ppm of sulphide solving a deterministic 3D model with $D_e = 1\times10^{-11}$ [m^2/s]. This work solved the same system using a 1D CA with $\Delta x = 1$ [cm] and $\Delta t = 1/13$ [year], and 1D implicit finite differences (IFD) scheme with same space increment and $\Delta t = 1$ [year]. Steady state is reached after ~500 years in the 1D models and after ~2000 years.

Table 4.8 Lifetimes for Validation Scenario II [year]

Statistic	Result	CA (1D)	CA (2D)	IFD (1D)
Mean	1.2×10^6*	2.5×10^6	2.2×10^6	2.4×10^6
Minimum		6.1×10^5	6.2×10^5	5.8×10^5
Maximum		9.3×10^6	6.6×10^6	8.6×10^6
Mode		2.3×10^6	2.0×10^6	2.2×10^6
Skewness		1.15	0.95	0.85
Kurtosis		5.19	4.51	4.6
SD		1.0×10^6	8.2×10^5	7.2×10^5
Realizations	1	10,000	1,000	10,000

* Computed from 0.8 nm/yr/ppm reported in [22]

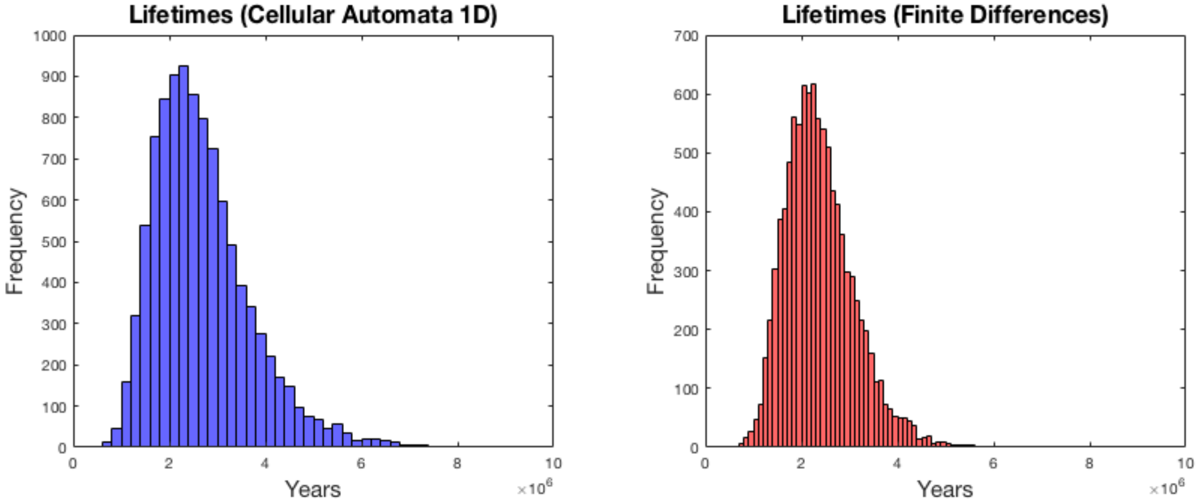


Figure 4.9 Histogram of UFC lifetimes for Validation Scenario II (1D models)

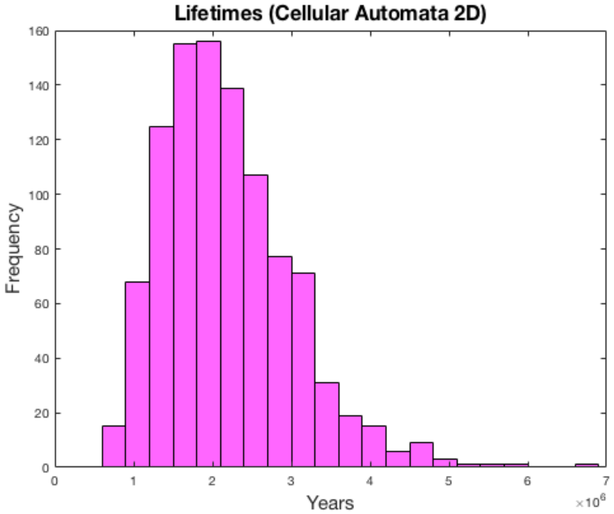


Figure 4.10 Histogram of UFC lifetimes for Validation Scenario II (2D CA)

4.5 Scenarios

The main source of uncertainty is the degree at which SRB will remain active; therefore, three scenarios were created to explore different situations that might happen inside the DGR: Scenario I represents the worst condition, however we decided to explore this case and compute lifetimes on highly adverse conditions; GF and HCB have different densities according to Table 2.2 (see section 4.1.1 and 4.1.4 for the assumptions on this scenario); Scenario II is the base case, it assumes SRB is active at the rock interface only, meaning that no bacterial activity is present inside the bentonite clay. Scenario III assumes a high homogeneous dry density of 1.7 [g/cm^3] and assumes that SRB activity is expected to be inside the bentonite clay (see section 4.1.1 and 4.1.4 for the assumptions on this scenario). All three scenarios have uncertainty in the groundwater composition, i.e., the amount of sulphate is assigned stochastically from run to run.

1. **Scenario I. SRB active in clays and rock interface**
   - SRB active in rock interface, GF, and HCB
   - Different dry densities for GF and HCB.

2. **Scenario II. SRB active in rock interface**
   - SRB active in rock interface
   - Different dry densities for GF and HCB

3. **Scenario III. SRB active, homogeneous densities**
   - SRB active in rock interface, GF, and HCB
   - Same dry densities for GF and HCB

4.6 Results

Results labeled ‘deterministic’ refer to the deterministic value in the groundwater sulphate content but they are the expectation value with respect to the movement and reaction probabilities. Results labeled ‘stochastic’ refer to a Monte Carlo sampling on the distribution of sulphate concentration, but they are also expectations with respect to movement and reaction. The size of the sample (10,000 runs) was chosen under the following criteria: 1) it should be large enough such that its shape and statistics become sample-independent, and 2) the sample is not large enough so as to require large computation times and limit the exploration of scenarios.

4.6.1 Scenario I. SRB active in clays and rock interface

Deterministic

Lifetimes of 2D models are about 2/3 of those of 1D. Concentration profiles show an abrupt change at the interface between GF and HCB, this is due to the change in density and its corresponding effect on diffusion and reaction. From looking at the concentration and flux plots at the canister boundary, it is noticeable the steady state was obtained after ~1,500 years.

Table 4.9 Lifetimes for Scenario I [year]

Rock type	1D	2D
Crystalline	1.34×10^5	8.79×10^4
Sedimentary	4.40×10^5	2.91×10^5

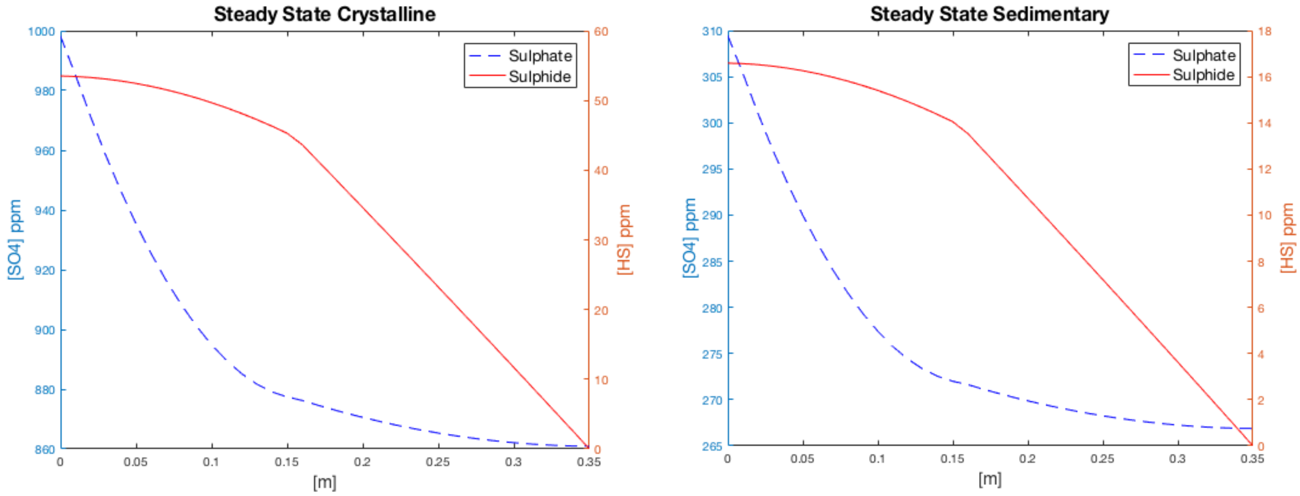


Figure 4.11. Profile of steady state concentration (distance from boundary)

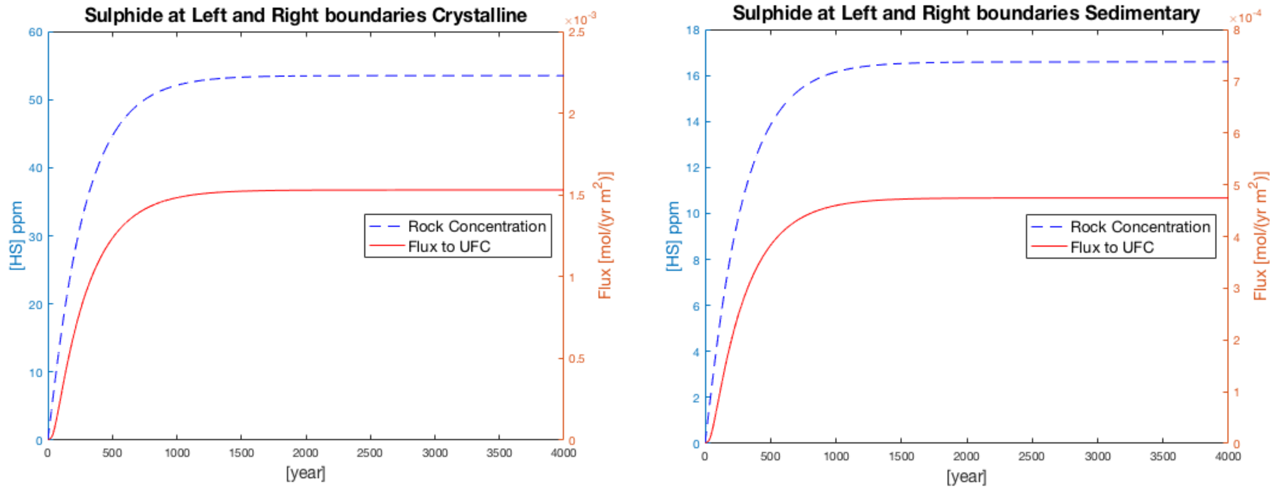


Figure 4.12. Sulphide concentration at the left boundary and flux for at the right boundary for Scenario I

Stochastic results

Mean lifetime values are close to those found using deterministic sulphate values. Lifetimes of sedimentary rock are about three times those of crystalline rock.

Table 4.10 Lifetimes for Scenario I, groundwater uncertainty [year]

Statistic	Crystalline rock (1D)	Sedimentary rock (1D)
Mean	1.38×10^5	4.53×10^5
SD	3.29×10^4	1.07×10^5
Minimum	5.65×10^4	1.64×10^5
Maximum	3.22×10^5	1.24×10^6
Mode	1.4×10^5	4.1×10^5
Skewness	0.72	0.76
Kurtosis	3.91	4.24
Realizations	10,000	10,000

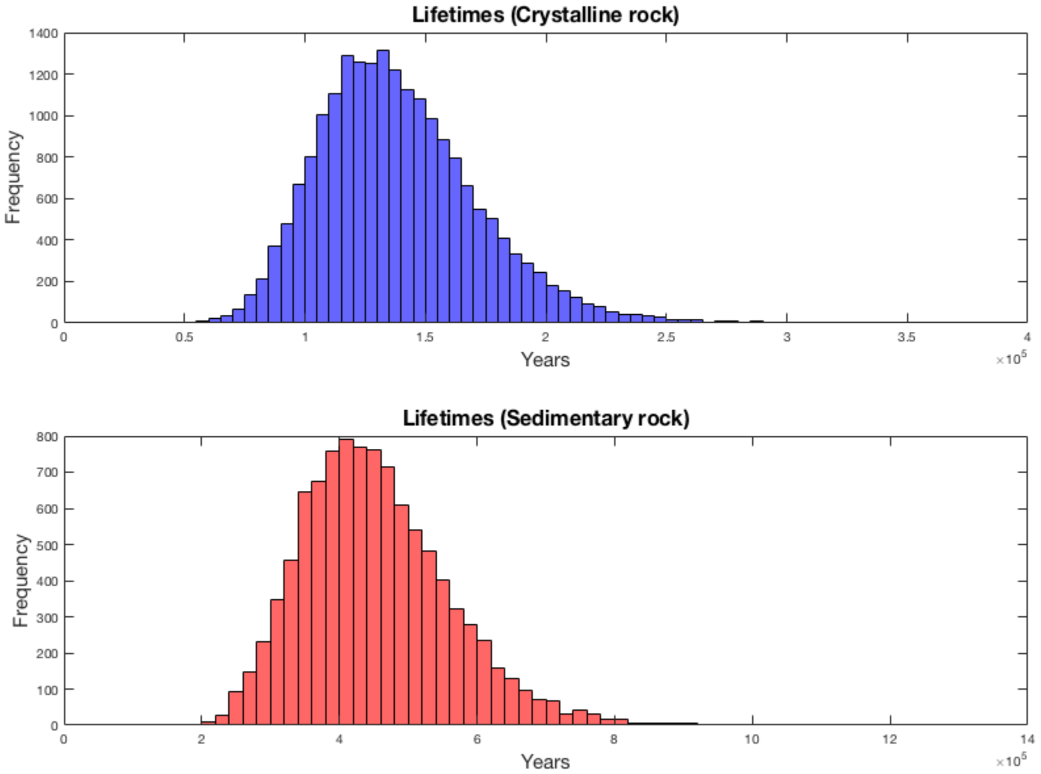


Figure 4.13 Scenario I lifetimes for 1D

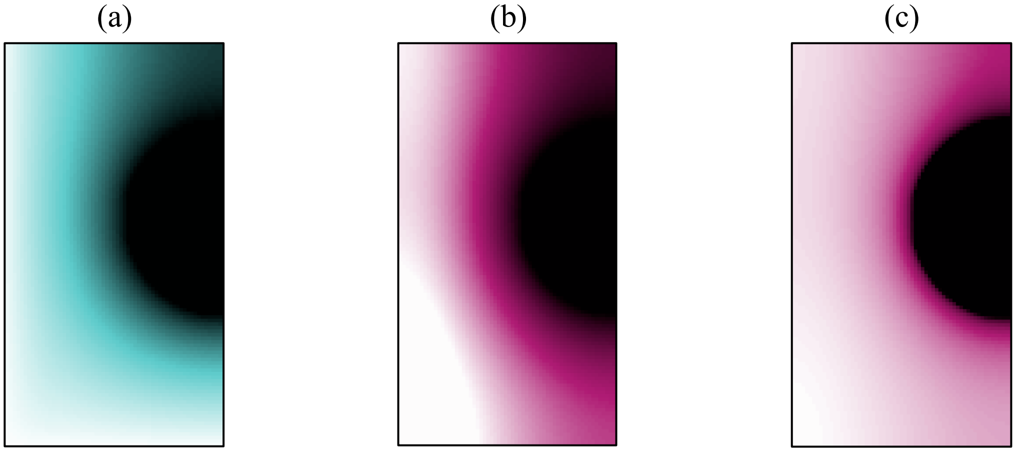


Figure 4.14 Frame of concentrations at steady state: (a) sulphate, (b) sulphide, (c) ln(sulphide)

4.6.2 Results Scenario II. SRB active in rock interface

Deterministic results

All the lifetimes are in the order of a million years for the deterministic results. Here again it is possible to see a change in concentration profile at the interface of GF and HCB. Sulphate concentration is practically saturated along the system; this is because the reaction takes place only at the rock boundary. Steady states were achieved ~1,000 years.

Table 4.11 Lifetimes for Scenario II [year]

Rock type	1D model	2D model
Crystalline	2.13×10^6	1.31×10^6
Sedimentary	6.87×10^6	4.25×10^6

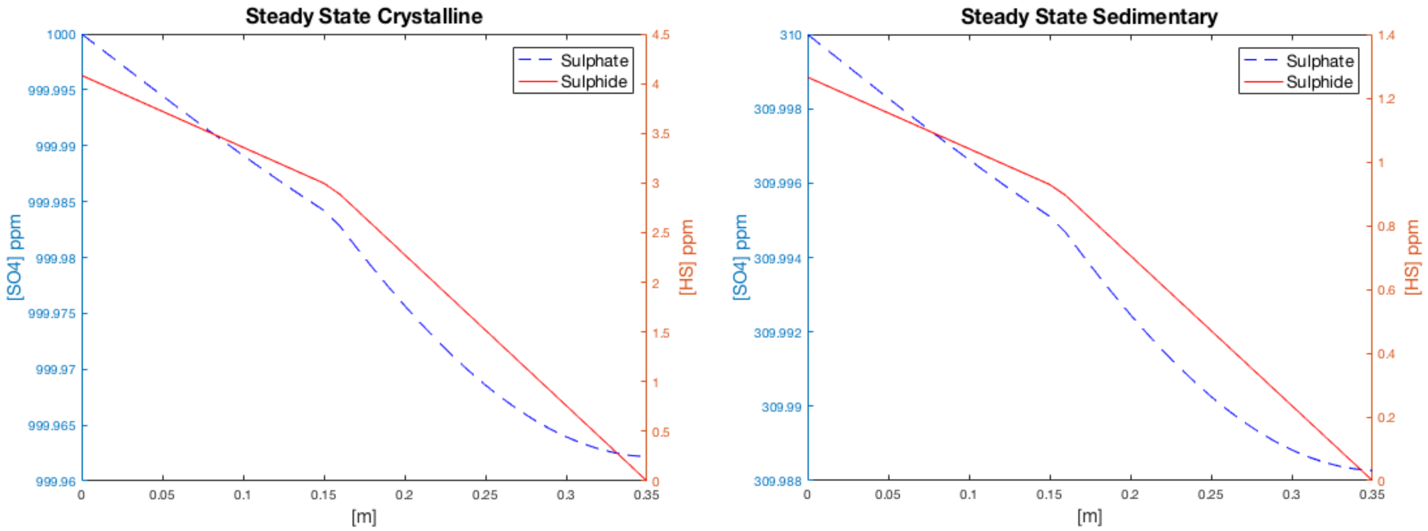


Figure 4.15 Profile of steady state concentration (distance from boundary)

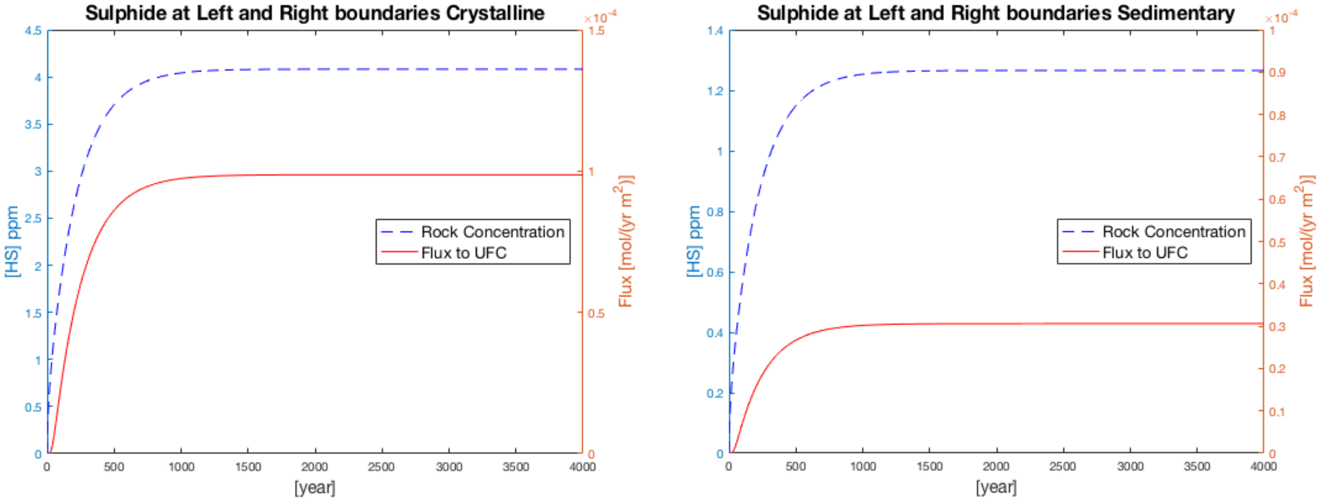


Figure 4.16 Sulphide concentration at the left boundary and flux for at the right boundary for Scenario II

Stochastic results

Sedimentary lifetimes are higher than three times those from crystalline rock. Mean values are close to those obtained using deterministic sulphate values.

Table 4.12 Lifetimes for Scenario II, groundwater uncertainty [year]

Statistic	Crystalline rock (1D)	Sedimentary rock (1D)
Mean	2.18×10^6	7.06×10^6
SD	5.1×10^5	1.6×10^6
Minimum	8.55×10^5	2.82×10^6
Maximum	4.9×10^6	14.7×10^6
Mode	2×10^6	6.6×10^6
Skewness	0.66	0.69
Kurtosis	3.74	3.79
Realizations	10,000	10,000

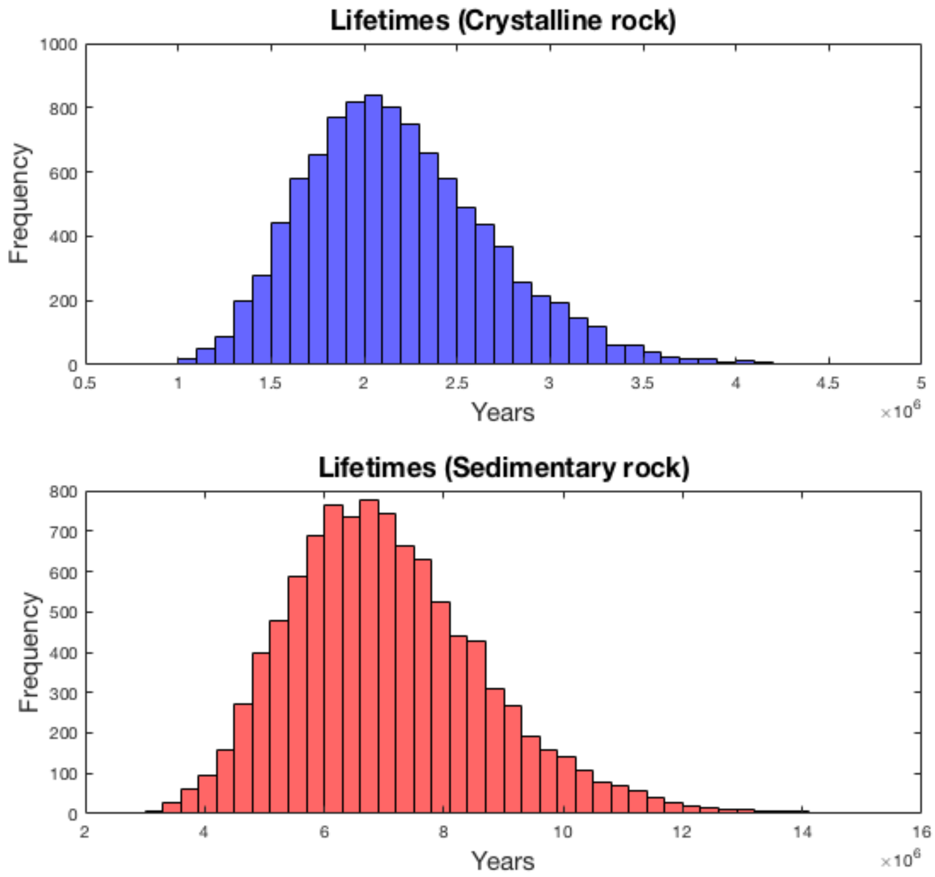


Figure 4.17 Scenario II lifetimes for 1D

4.6.3 Results Scenario III. SRB active, homogeneous densities

Deterministic

This scenario presented more consistent results in that the 2D lifetimes are about 4/5 of those from the 1D model. Concentration profiles are smooth due to the assumption of having a homogeneous density. The steady state was achieved after 2,000 years.

Table 4.13 Lifetimes for Scenario III [year]

Rock type	1D model	2D model
Crystalline	1.42×10^6	1.21×10^6
Sedimentary	4.58×10^6	3.92×10^6

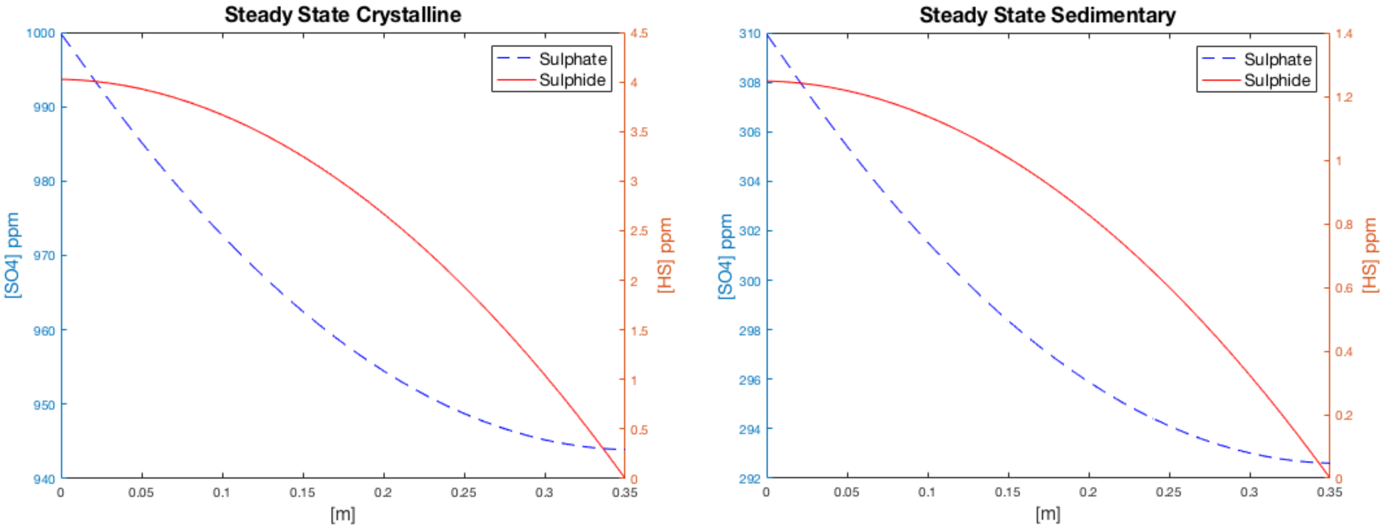


Figure 4.18 Profile of steady state concentration (distance from boundary)

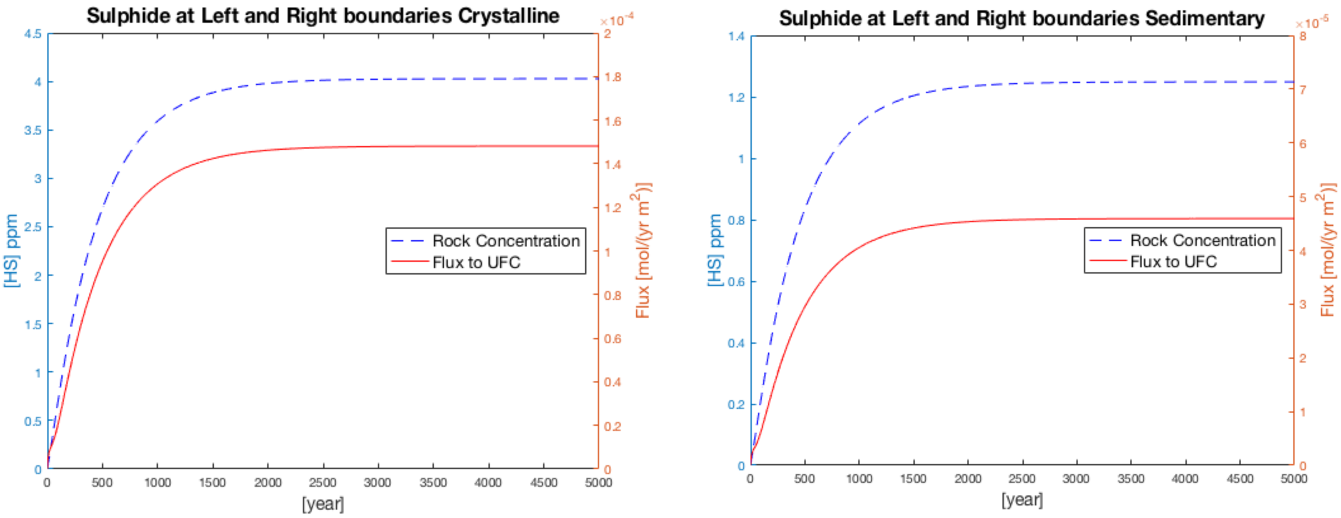


Figure 4.19 Sulphide concentration at the left boundary and flux for at the right boundary for Scenario III

Stochastic results

Here again the mean values were close to those obtained using a deterministic sulphate input, and lifetimes of sedimentary rock are about three times the lifetimes of crystalline rock.

Table 4.14 Lifetimes for Scenario II, groundwater uncertainty [year]

Statistic	Crystalline rock (1D)	Sedimentary rock (1D)
Mean	1.45×10^6	4.71×10^6
SD	3.4×10^5	1.1×10^6
Minimum	6.41×10^5	1.65×10^6
Maximum	3.62×10^6	10.7×10^6
Mode	1.3×10^6	4.0×10^6
Skewness	0.75	0.67
Kurtosis	4.0	3.72
Realizations	10,000	10,000

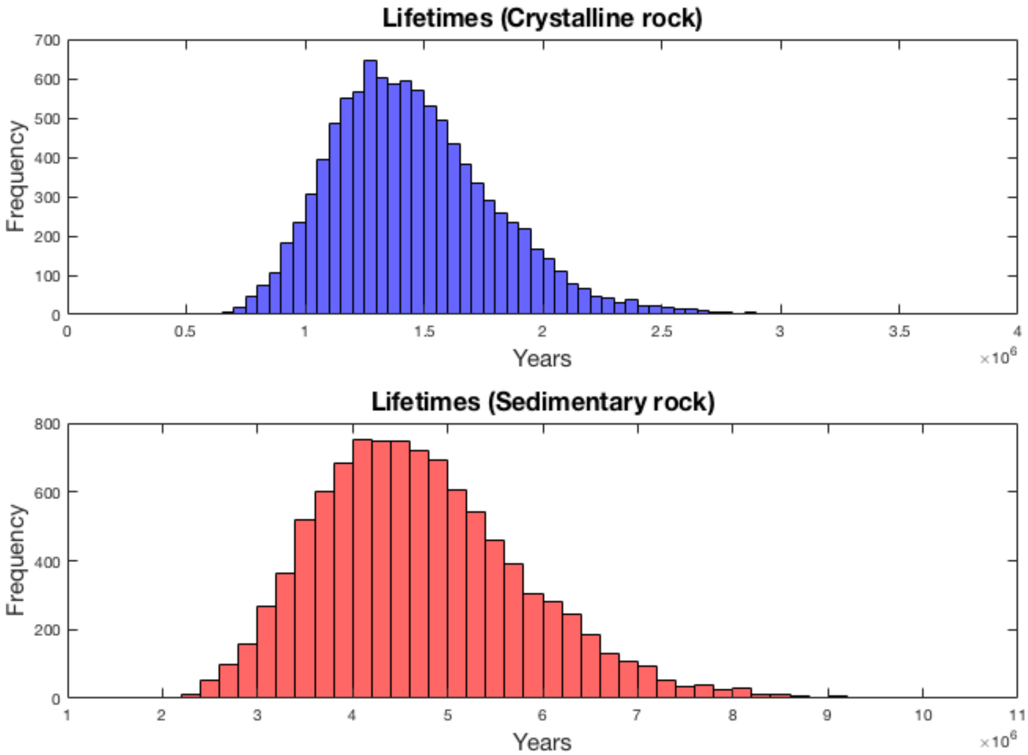


Figure 4.20 Scenario III lifetimes for 1D

4.7. Sensitivity Analysis

Since the density of the gapfill is another main source of uncertainty, lifetimes under different gapfill densities were computed, ranging from 1,300 to 1,700 [kg/m^3] (dry density). It's worth noting that the density of the gapfill determines the reduction rate at the rock interface and inside the gapfill, as well as the diffusivity of molecules through gapfill.

The density of the HCB remained constant at 1,700 [kg/m^3] for scenarios I and III. For scenario II, since it assumes homogeneous density, GF were HCB assumed to have the same density, which was changed according to the range mentioned above.

4.7.1 Scenario I. SRB active in clays and rock interface

Lifetimes exhibit an exponential response to gapfill density, for crystalline rock the range of lifetimes is ~[2 × 10^4 – 1.4 × 10^5] years and for sedimentary is ~[1.2 × 10^5 – 4.3 × 10^6] years.

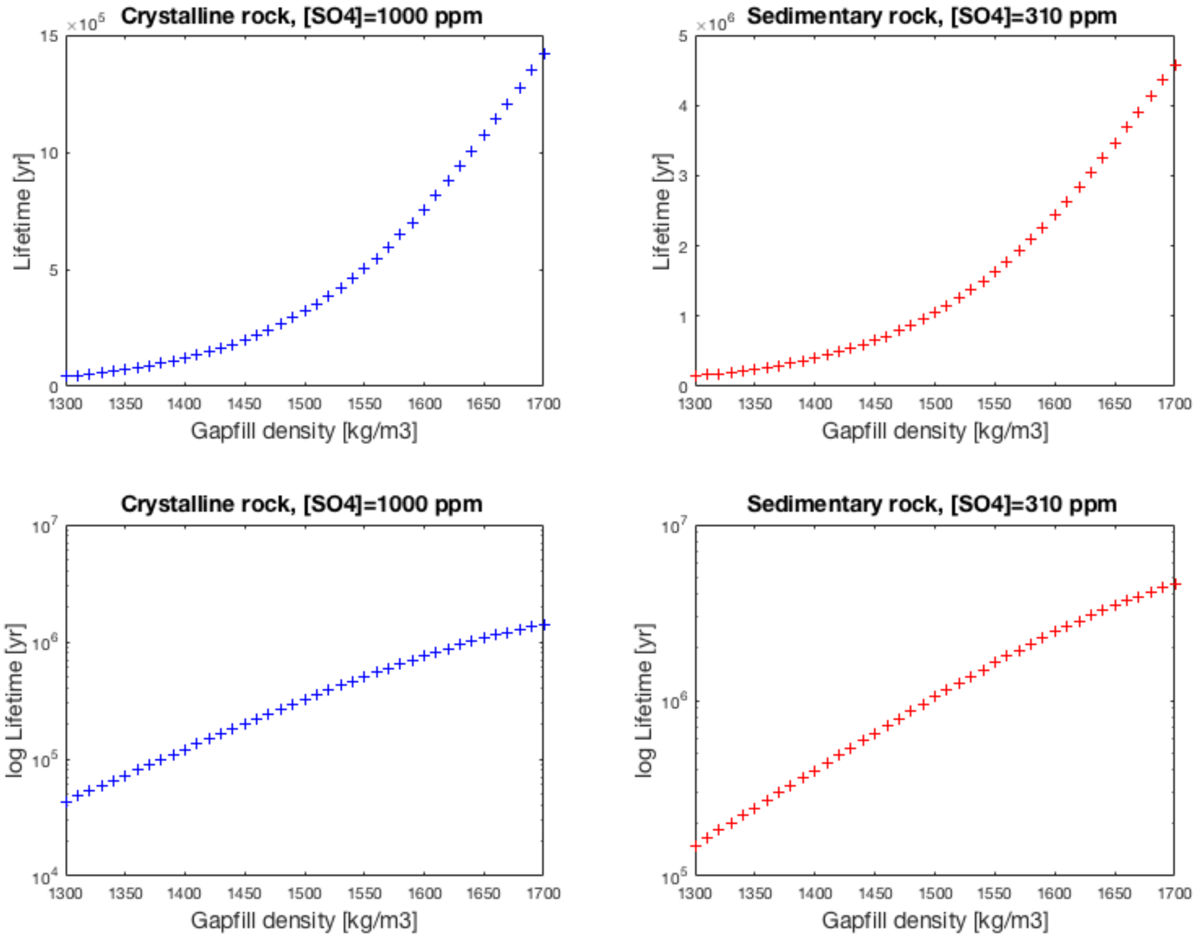
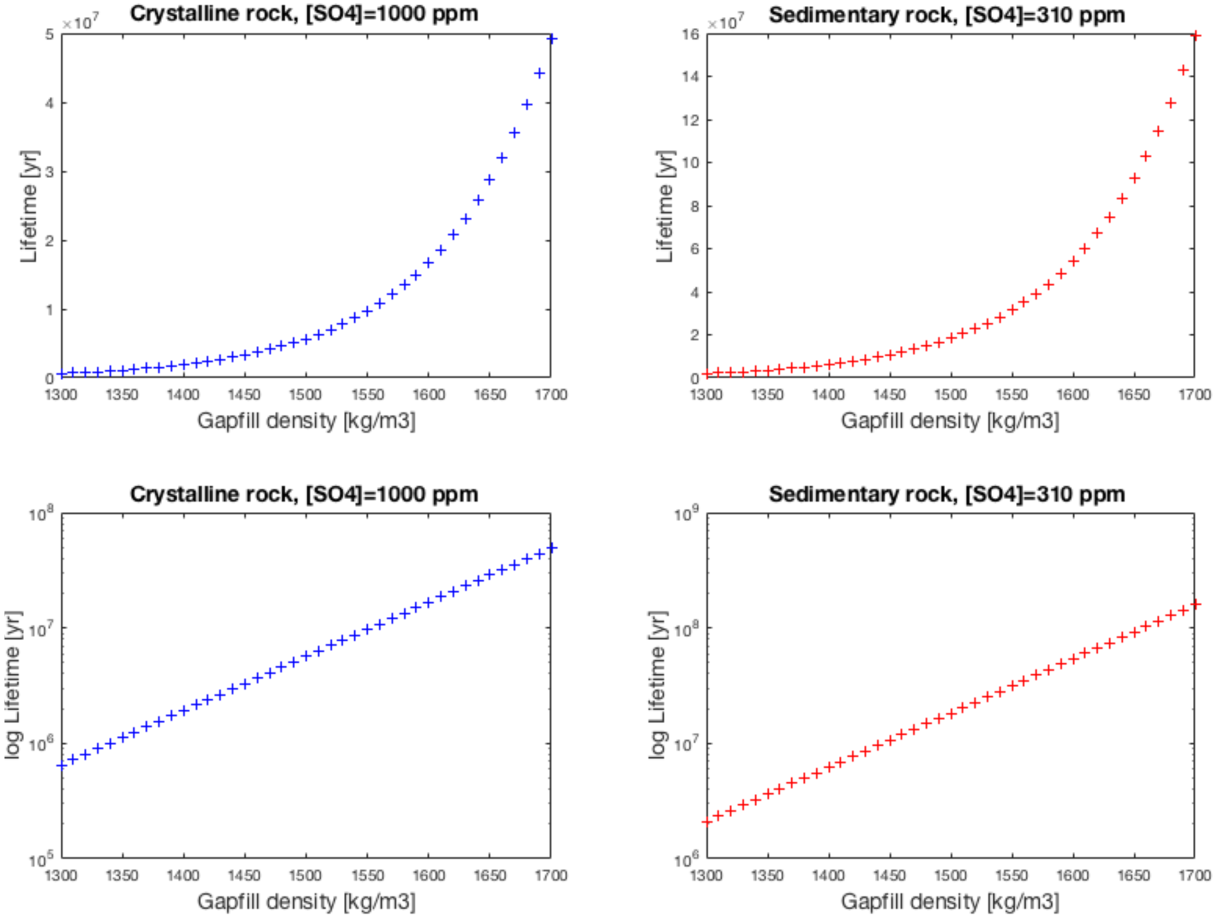


Figure 4.21 Lifetimes for Scenario I

4.7.2 Scenario II. SRB active only at rock interface

Here again lifetimes are exponential with respect to gapfill density, for crystalline rock the range of lifetimes is ~[3.9×10^5 – 5×10^7] years and for sedimentary is ~[1.2×10^6 – 1.6×10^8] years.


Crystalline rock, [SO4]=1000 ppm

Sedimentary rock, [SO4]=310 ppm

Crystalline rock, [SO4]=1000 ppm

Sedimentary rock, [SO4]=310 ppm

Figure 4.22 Lifetimes for Scenario II

4.7.3 Scenario III. SRB active, homogeneous densities

Lifetimes exhibit an exponential response to gapfill density, for crystalline rock the range of lifetimes is ~[1.5 × 10^4 – 1.4 × 10^5] years and for sedimentary is ~[1.0 × 10^5 – 4.8 × 10^6] years.

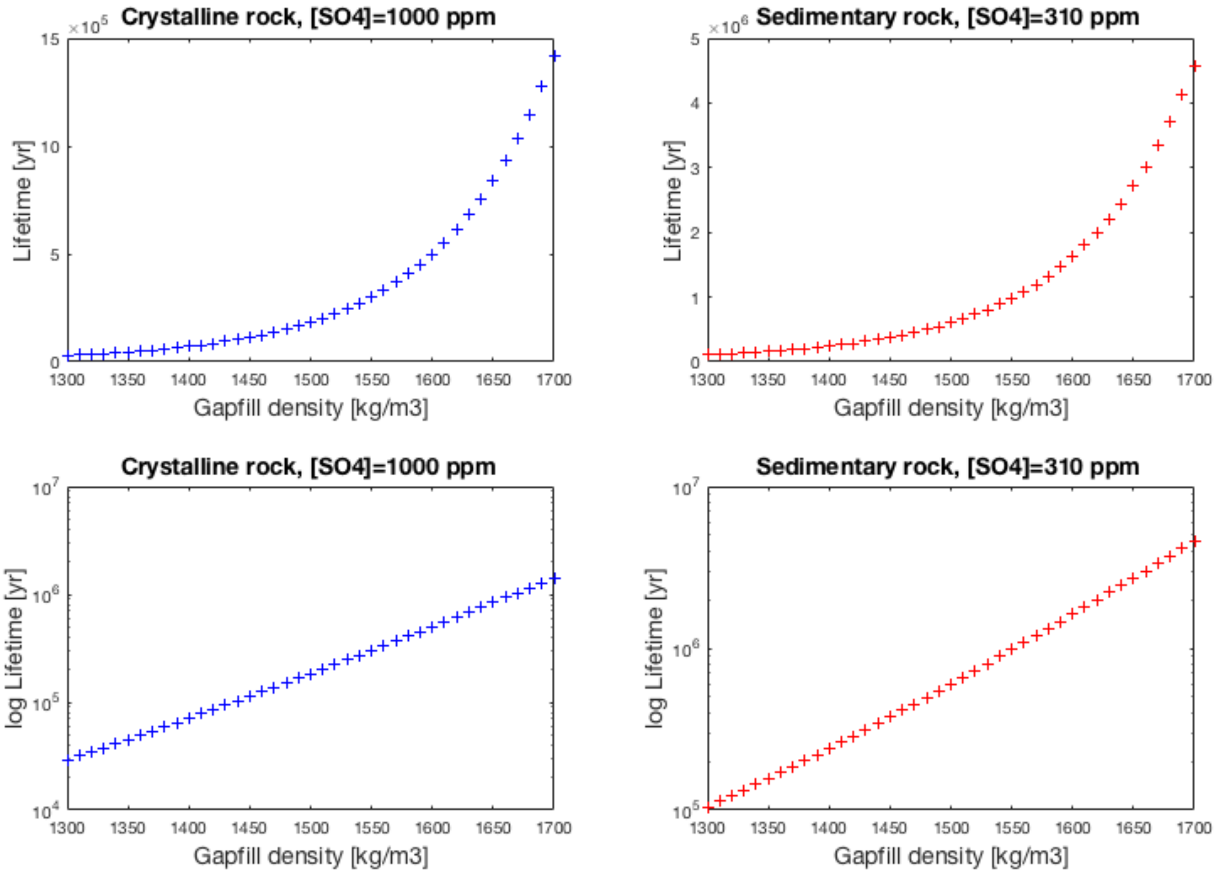


Figure 4.23 Lifetimes for Scenario III

4.8 Part I Conclusions

The use of cellular automata to describe diffusion has advantages as well as limitations. The following list outlines the most salient ones that were noticed from applying CA to the nuclear waste problem.

Advantages:

1. Uncertainty and space-dependency in parameters is easier to implement and treat than in analytical or numerical methods.
2. It makes the diffusion process more intuitive to understand, in particular when relating diffusivity (in [m^2/s]) and spreading of a solute.
3. Reaction terms are coupled easier than in a finite difference scheme.
4. Introduction of more chemical species may also be simpler since each species can be managed as an independent process, interacting only at discrete points in time in each cell.

Limitations:

1. Nonlinear shapes may require a finer grid to get an appropriate approximation.
2. The model requires decreasing the space increment to increase the accuracy of the results. If we decrease the size increment by a factor of 1/a, the number of cells grows by a factor of a^{dim}, dim being the number of dimensions.

Overall, using CA is recommended when the system to be modeled has some level of complexity, e.g. two or more chemicals or space-dependent parameters, reaction terms. If the system is only diffusion, the analytical solution or a finite difference method is recommended instead. If the system has complicated nonlinear shapes and is important to model them adequately, then the use of *ad hoc* software is recommended such as Comsol®.

With respect to the nuclear waste problem, the following conclusions were inferred from looking at the results of sections 4.6 and 4.7:

1. Bentonite dry density has a drastic effect on the lifetime of the container. If SRB is active only at the rock, lifetimes will be around 10^6 years. However, if SRB is also active inside the bentonite clay, having a homogeneous high dry density of at least 1,600 [kg/m^3] becomes essential to achieve lifetimes of 10^6 years or higher. In particular for the gapfill dry density, which is the clay that will be placed in between the host rock and the buffer boxes, may have a lower value due to in situ placement challenges.
2. We may remind the reader that the lifetimes computed here are for the canisters closest to the rock and that failure was defined as a breach in the copper layer, but canisters will be placed at different distances from the rock and they are also made of 30 mm steel. Therefore, our lifetimes are lower bounds of the lifetimes of the canisters under current assumptions.
3. Lifetimes seem to have a linear response with respect to the amount of sulphate and an exponential response with respect to the gapfill dry density. It may be

possible to construct a simpler surrogate model that inputs sulphate and dry density and outputs the lifetime of the canister.

PART II

Chapter 5 Agent-based Modeling

INTRODUCTION

In this chapter provides literature review on the use of agent-based modeling in road networks and route planning, as well as the use of information to improve agent's decision-making. Then it discusses the advantages of using ABMs over EBMs and show an example where an ABM is a more flexible tool to use and another where an EBM is a better option. It also introduces the functions of travel time and road density that will be use on the Chapter 7.

Models were solved using Matlab®, Netlogo® or Python®.

5.0 Literature Review

The use of information to enhance decision-making of agents has been studied widely in different contexts. Below we describe relevant literature regarding road networks and agent decision-making.

A thorough overview about the historical development of transportation models can be found on [39]. A summary is provided next. In the most general case, the transportation problem is a multi-scale mobility problem that deals with 1) the demand of trips; the 2) the allocation of travelling units to paths from origins to destinations such that their requirements in terms of timing, mode, and other preferences are met within some satisfaction interval; and 3) the simulation of the dynamic interaction of such agents as they move along the road network.

The selection of the aggregation is important in as much as there is a tradeoff between micro-scale accuracy (realism) and computational complexity. Three approaches have been studied: models of the macro-state using aggregate variables (generally mean values); microsimulations that aim at modeling road segments in intra urban roads, great detail is put in the modeling of drivers' behavior, lane crossing and intersections; and mesoscopic models that incorporate a sophisticated driver's decision model but with a medium level aggregation on road segment dynamics.

An example of the last type of model is the Dynamic Traffic Assignment (DTA), which has the same aggregation level as those used in this thesis. Vehicles are assumed to travel at the same speed on each road segment; speeds are computed from a travel-delay function with parameters that are segment-specific. Drivers based their path selection on travel times at the time of the decision, not on predicted times. Although there are models that use experienced travel times. Mesoscopic models are used mainly for design and real-time control purposes.

The use of microsimulation for traffic models is discussed in [40]. Microsimulation is the disaggregated or micro level replication of a system that is dynamic, has some probabilistic components, has different type of actors and those actors behave according to some rule. Some problems are suitable to study using EBMs; however, microsimulation becomes essential in situation/systems that cannot be simplified without incurring in significant inaccuracies or when the actual local behavior will determine the success of a given policy, design or control effectiveness.

However, the use of microsimulation requires disaggregate data, which is not always available. It is possible to use a disaggregation method called synthesis, which is a procedure to generate a population set that is statistically consistent with the aggregate data available. However, synthesis methods may fail to capture significant heterogeneity in the population.

Another feature microsimulations should include is the updating of the population as the simulation time runs. For short timespans it may not be necessary, but if the simulation time runs over a considerable timespan, then it becomes necessary the updating of population features to adequately replicate the real system. Therefore, the simulation must incorporate a model for each dynamic feature.

Major obstacles of microsimulation are 1) the accuracy of its results depend on the accuracy of the input data; therefore, the disaggregation of data is essential to obtain representative population and ensure the validity of the outputs of the model; 2) the determination of an appropriate level of aggregation in terms of space, time, behavior; 3) the statistical treatment of the results; and 4) the computational complexity associated with solving a detailed object-based model.

Diffusion of information has been studied in [41], where authors studied how news are transmitted throughout a network that describes a particular social structure. One way of transmission is when a given node reaches a threshold of neighbors with a given ‘position’, then that node also assumes the same position. Another way is using ‘infection’ spreading, meaning that when a node acquires certain news, it may ‘infect’ (transmit the news to) neighbors with a probability.

An agent-based approach to the study of road networks has been studied in [42] where current travel information is used to determine shortest paths and agents learn from previous experiences; user-equilibrium is achieved, and under certain circumstances, the system’s optimum. However, the study does not account for the estimation of the state of the roads nor the deterioration of information as time goes on.

Another example of an ABM of road networks with information sharing is [43] where the authors studied a double layer network, where one layer describes the connections among drivers and the spreading of information and the other layer describes the physical road network. Agents are given the capacity to learn from previous experiences and eventually the network converges to steady probabilities of route selection.

Wei et al. studied the social interaction of drivers in terms of choice behavior [44]. They found that the interaction (information sharing) of drivers affects their decision about route selection; however, it has no effect on the overall performance. They also discovered that user equilibrium in a network does not imply fixed route selection for drivers.

The implementation of a credit scheme to charge drivers for using determinate links and how it affects the overall network is studied in [45]. Drivers are given a certain amount of credits for free and they have to pay if that amount is exceeded. Fares vary depending on each road link.

The use of advanced traveler information systems (ATIS) that provides current or predictive information about traffic flows to drivers is studied in [46]. The authors used a road network with two roads to study the effects of ATIS and they also explored different information collection methods.

Lehrer and Smorodinsky [47] study how agents facing a sequential decision problem, starting from an initial belief of the payoffs distribution, learn to adjust its belief to match the true underlying distribution. They show that an optimal decision occurs when the relative entropy between the true and the believed distribution is zero, which in turn is the asymptotical result of the continuous adjustment of posterior beliefs as more realizations are observed.

In a very interesting paper [48] the authors show how brainless organisms are capable of complex decision-making simply by the use of a Bayesian rule of selection. Amoeboid organisms are stimulated to grow two arms (left or right) by providing different amounts of food at each side. They start by exploring equally both sides, and after finding or not food on that side, they adjust the growth of each arm to respond to the different food supply. The most accurate model that describes the organism's behavior is a Bayesian model. Following a simple conditional model yields an optimal strategy for any of the tested scenarios. This paper is relevant to our research in that it is an example of how a brainless agent, i.e., an agent with no prior information and with no information processor, is capable of making good decisions by 'sampling' the environment and adjusting its prior belief. Eventually, the organism matches the underlying true distribution behind the environment.

As an example of the use of entropy as a measure of uncertainty, in [49] the authors characterized two attributes of asset pricing: dispersion and horizon dependence. For dispersion they used information entropy of the pricing kernel; whereas for horizon dependence they measured the relative entropy between several periods and that from one. They used examples to show the goodness of their approach.

Van Nieuwerburgh and Veldkamp [50] studied how information acquisition affects posterior investment decisions on financial investors. They showed that there is positive feedback between assets chosen to acquire information from and assets expected to hold. It increases the chances of holding the assets the investor knows more about, which in many cases lead the investor to hold an undiversified portfolio. This paper provides some understanding about the information-dependence decision-making and the capacity of information acquisition.

C. Turkay et al. used information entropy and relative entropy [51] to improve the realism of agent-based crowd simulation models. They constructed a grid to represent the spatial location of a facility that pedestrians use to walk from one point to another. They mapped users' velocity (speed and direction) onto a grid and computed the frequency of each direction users are heading to. With that probability they computed the current information entropy of each grid. They kept track of previous time steps and compared them against the current calculation using the relative entropy. A linear combination is used to combine both entropies and then the final estimation is provided to agents, so they can choose the path with less 'chaotic' crowds, i.e., the path with minimal entropy. For the validation of the model they did a comparative analysis of the results of their model with previously known models. They also used real-world room evacuation videos and visually compared them with the simulation obtained using his model.

5.1 Agent-Based Modeling

Agent-Based Modeling is an approach to study any social or nature phenomena or system by focusing on the interactions of agents composing the system. An agent is an autonomous unit with simple rules of behavior, whose actions have an effect on other agents and it is affected by their actions. ABMs have proved to be useful in imitating or reproducing the emergent properties of complex systems; "large scale patterns in the world are usually the result of the interactions of large numbers of smaller pieces" [52].

Agents are discrete objects with well-defined properties and rules of interaction. Oftentimes, there are different types of agents within a simulation analysis, whose characteristics will depend on the experiment. For instance, modeling human disease transmission would require two types of agents: 'person' and 'environment'. A 'person' would represent an individual with certain attributes (health, age, spatial movement) whereas 'environment' would represent a 2D spatial location where people move and interact.

5.1.1 Properties of agents

Despite the fact that agent attributes are defined according to the purpose of the experiment or the goal of the simulation, some common attributes are general and distinctive of ABMs [53]:

- Autonomy. Agents are entities with internal goals that behave independently. They make decisions based on the state of the environment and its individual state.
- Self-contained. Agents are discrete objects whose internal description is not dependent on the outside world. They are fully defined entities with a known behavior spectrum.
- Interaction. A major advantage of ABMs is the fact that agents are able to interact and affect each other. This feature triggers new and unexpected global behavior.
- Environment. As mentioned before, the environment is the spatial location where agents interact. It can be a network, a Cartesian plane, cellular automata, etc. The environment will dictate the ways of interaction.
- Goals. Agents have internal goals that are the basis of the decisions they make. They may be represented by a single or multiple objective functions. Agents may want to minimize or maximize or reach a threshold of satisfaction on one or many attributes.
- Learning. Agents may have memory and the capacity of modify its behavior upon experience by comparing predictions against outcomes.

5.1.2 ABM implementation

The implementation of an ABM simulation experiment follows these steps [3]:

1. Problem. Definition of a problem to be solved or studied.
2. Design of agents. Definition of types of agents as well as the fixed and dynamic attributes.
3. Design of environment. The selection of a proper topology where agents are going to interact.
4. Design of behavior. The definition of the rules of decision in discretized time steps.
5. Design of interaction. This stage can be performed in the previous step. It is the extent in which an agent is going to be affected by others. Typically, this influence is reflected in the decision-making process.
6. Data. The collection of historical data from where to derive parameters from.
7. Validation. To ensure that the outcomes of the model are sufficiently predictive of the real phenomena.

5.2 Agent-Based Models vs. Equation-Based Models

ABMs have the following advantages over equation-based models (EBM) [1]:

- ABM can model heterogeneous populations
- Its results are discrete, as many of the components of real social, economic or ecological systems
- Including spatial effects in EBMs is difficult
- Modelling in ABM is easier since agents have a match with real entities
- ABM does not require full knowledge or description of the aggregate system
- ABM provides results at the aggregate and individual level

Next an example is shown where the ABM representation offers more advantages in terms of modeling compared to the EBM.

5.2.1 ABM vs. ODEs: A Microeconomics Example

To illustrate how ABMs allow for introducing more realism into the modeling of a system, the classic **microeconomics** problem of finding the price and quantity of equilibrium of a market was solved using agents and ODEs.

Let us assume a market with a single product under trade, composed of a fixed number of buyers and sellers, where

- m = 'number of buyers'
- n = 'number of suppliers'
- p = 'unitary price of the good'
- d(p) = 'individual demand'
- s(p) = 'individual supply'
- $D = \sum_{\forall i} d_i(p) =$ 'total demand'
- $S = \sum_{\forall j} s_j(p) =$'total supply'

Assuming the change in unitary price over time to be proportional to the difference between D and S, the following is obtained

$$\frac{dp}{dt} = \omega(D - S) \tag{5.1}$$

where ω controls the speed of the price adjustment, it was set to $\omega = \frac{1}{100(m+n)}$ [\$/unit]. Two cases were considered: linear and nonlinear demand and supply. For a linear demand and supply we have

$$d(p) = d_{max}\left(1 - \frac{p}{p_{max}}\right) \tag{5.2a}$$

$$s(p) = s_{max}\frac{p}{p_{max}}. \qquad (5.2b)$$

For a nonlinear demand and supply we have

$$d(p) = (d_{max} + 1)\ exp(-\eta_d p) - 1 \qquad (5.3a)$$

$$s(p) = exp(\eta_s p) - 1 \qquad (5.3b)$$

where p_{max} is the maximum price; d_{max}, s_{max} are the maximum unitary demand and supply, $\eta_d = \frac{ln(d_{max}+1)}{p_{max}}$and $\eta_s = \frac{ln(s_{max}+1)}{p_{max}}$. Functions were created such that d(0)=d_{max}, d(p_{max})=0 and s(0)=0, s(p_{max})=s_{max}. We followed the convention of plotting price in the y-axis and quantity in the x-axis.

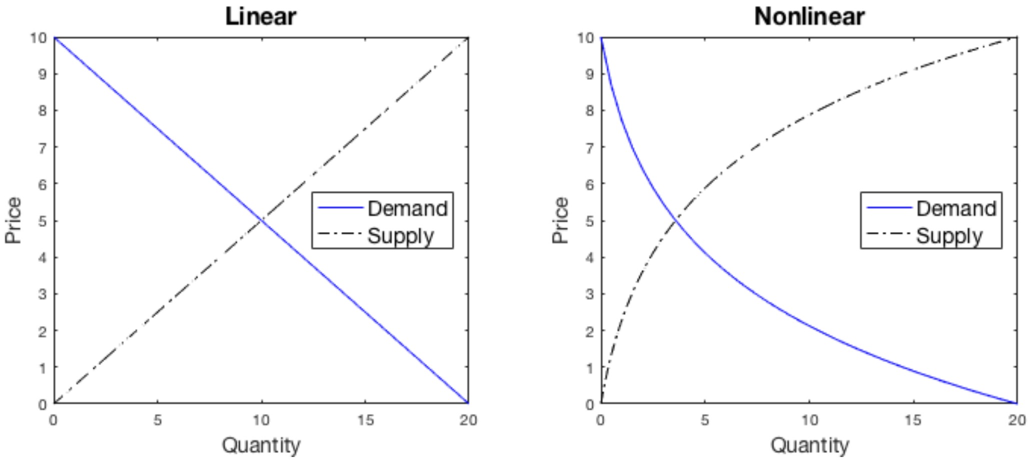


Figure 5.1 Single *buyer*'s demand and single *seller*'s supply

Solving the system using ABM

Two types of agents were created: *buyers* and *sellers*; and a variable to represent the unitary price of the product. *Buyers* have an internal variable called "demand" and that for *sellers* is called "supply". Their rule of behavior is described by equations (5.2a – 5.3b) for both linear and nonlinear cases, i.e., at each time step both agents input the current price p_t and output either the amount demanded or the amount supplied during that period. The unitary price is then adjusted according to the difference between total demand and total supply. The simulation starts with a random initial price p_o. A sketch of the algorithm is

- Step 0. Agents are created: $\{Buyers, Sellers\}$. *Buyers* are assigned the rule d(p), *Sellers* the rule s(p). The initial price is determined by

$$p_0 \sim U(a,b)$$

- Step 1. *Buyers* determine their individual demand:

$$d(p_t)$$

- Step 2. *Sellers* determine their individual supply:

$$s(p_t)$$

- Step 3. Set $D = \sum_{\forall i} d_i(p)$ and $S = \sum_{\forall j} s_j(p)$. Adjust unitary price according to (derived from eq. (5.1)):

$$p_{t+1} = p_t + \Delta t \alpha (D_t - S_t)$$

- Step 4. Stop if

$$p_t = p_{t-1},$$

  else, set t = t+1 and repeat from Step 1.

For the purpose of comparing results an example was solved using the following inputs:

Table 5.1 Input values for the microeconomics problem

p_0	p_{max}	d_{max}	s_{max}	m	n
10	10	20	20	20	20

For the **linear** response, the market reaches equilibrium at price **$5/unit and quantity 200 units**.

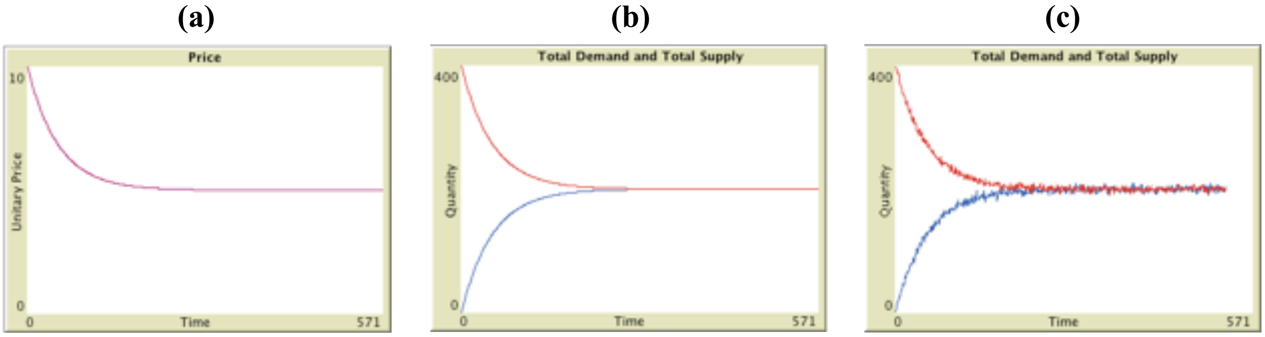


Figure 5.2 Linear model: (a) Evolution of price, (b) Evolution of total demand and supply (c) Evolution of demand and supply with random terms

For the **nonlinear** response, the market reaches equilibrium at price **$5/unit and quantity 71.65 units**.

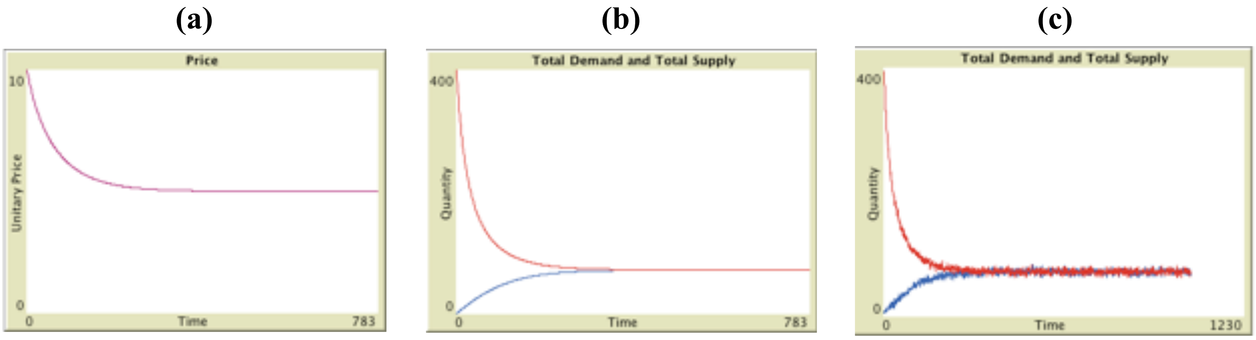


Figure 5.3 Nonlinear model: (a) Evolution of price, (b) Evolution of total demand and supply (c) Evolution of demand and supply with random terms

Solving the system using ODEs

Demand and supply curves are price-dependent, i.e., D(p) and S(p), to find their time derivative the chain rule was used as follows

$$\frac{dD}{dt} = \frac{dD}{dp}\frac{dp}{dt} \tag{5.4a}$$

$$\frac{dS}{dt} = \frac{dS}{dp}\frac{dp}{dt} \tag{5.4b}$$

where $\frac{dD}{dp} = \frac{d}{dp}[\sum d_i(p)]$ and $\frac{dS}{dp} = \frac{d}{dp}[\sum s_i(p)]$. The above system describes the evolution of total demand and total supply over time.

For a **linear** demand and supply and having **homogeneous agents**, the system (5.4a-b) becomes

$$\frac{dD}{dt} = -m\frac{d_{max}}{p_{max}}\omega(D-S)$$

$$\frac{dS}{dt} = n\frac{s_{max}}{p_{max}}\omega(D-S).$$

The initial condition was computed from the initial price p_0 as $D_0 = D(p_0)$ and $S_0 = S(p_0)$; Figure 5.4 describes the evolution of the market reaching a steady state at **price $5/unit and quantity 200 units**.

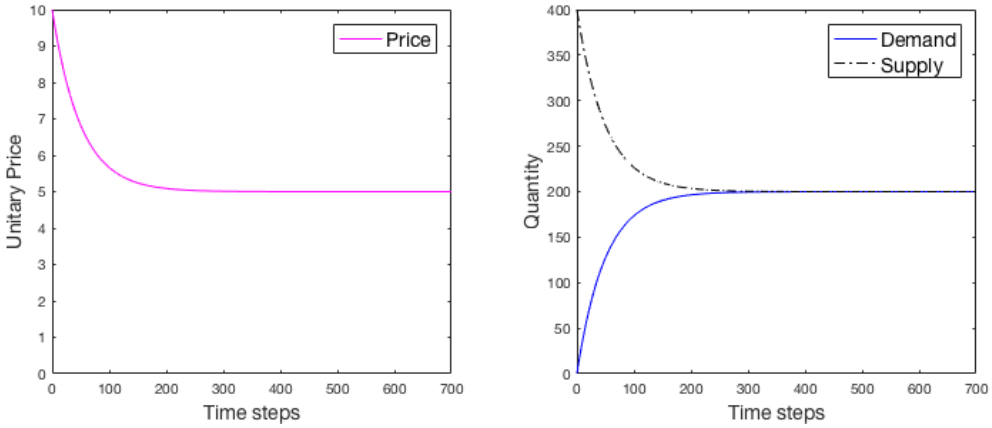


Figure 5.4 Linear model: Solution to the ODE system

For the **nonlinear** demand and supply and having **homogeneous agents**, the system (5.4a-b) becomes

$$\frac{dD}{dt} = -\eta_d(D + m)\,\omega(D - S)$$

$$\frac{dS}{dt} = \eta_s(S + n)\,\omega(D - S)$$

The initial condition was computed from the initial price p_0 as $D_0 = D(p_0)$ and $S_0 = S(p_0)$; Figure 5.5 describes the evolution of the market reaching a steady state at **price $5/unit and quantity 71.65 units**.

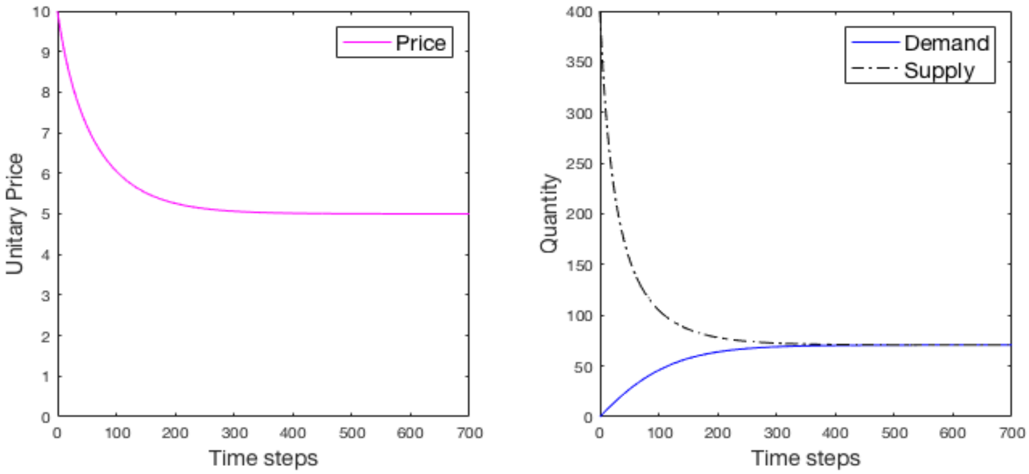


Figure 5.5 Nonlinear model: Solution to the ODE system

For the ODEs we did not attempt to include random terms into the equations and solve them since it requires a careful examination using stochastic differential equations, which falls out of the scope of this example. This seems however, a compelling area of study.

Comparison remarks

ABM is by far more flexible for this problem: the domain can be restricted to integer products; any of the parameters can be replaced by a random variable; allows for introducing heterogeneous agents, not only with different parameter values but with different functions as well; it also allows for a variable population. All that is achievable with little increase in computational effort. Overall, ABMs are easier to interact with; however, they require some coding skills.

Formulating and solving the ODE system requires a deeper understanding of the problem. Introducing random terms requires knowledge of stochastic differential equations, which may involve the treatment of multiplicative noise. However, ODEs are concise and elegant formulations. Perhaps full knowledge comes from understanding how a system behaves at different scales. For an example where using simulation is inefficient and inaccurate, please see Appendix D.

5.3 Road Networks

5.3.1 Traveling Time

For the calculation of traveling times along road segments we chose the BPR (Bureau of Public Road) travel time function [54] for kth segment.

$$T_k = t_f \left\{ 1 + \beta_0 \left(\frac{w_k}{cap_k} \right)^{\beta_1} \right\} \tag{5.5}$$

where $t_f > 0$ is the free flow time for that segment in time units [t/segment] or [t/km], w_k is the volume of vehicles per time or flow [vehicle/t], cap_k the capacity of the segment [vehicle/t], β_0 and β_1 are shape parameters. Although it may underestimate delays for low volumes flows or overestimating delays for high volumes, it satisfies some interesting properties and is widely used in the field [54], [55]

- Second order continuously differentiable
- Positive over the domain of f $\quad \forall f \geq 0,\ T(f) \geq t_f$
- Monotone increasing $\quad f_1 \geq f_2,\ T(f_1) \geq T(f_2)$
- Strictly monotonic slope $\quad T'' > 0$
- Bounded slope $\quad \forall T', \exists\, h : T' < h$

The selection of β_0 should be made considering that when the volume of vehicles is equal to its capacity then its travel time is $\{1+\beta_0\}$ times its free-flow time:

$$T_k(cap_k) = t_f\{1 + \beta_0\}.$$

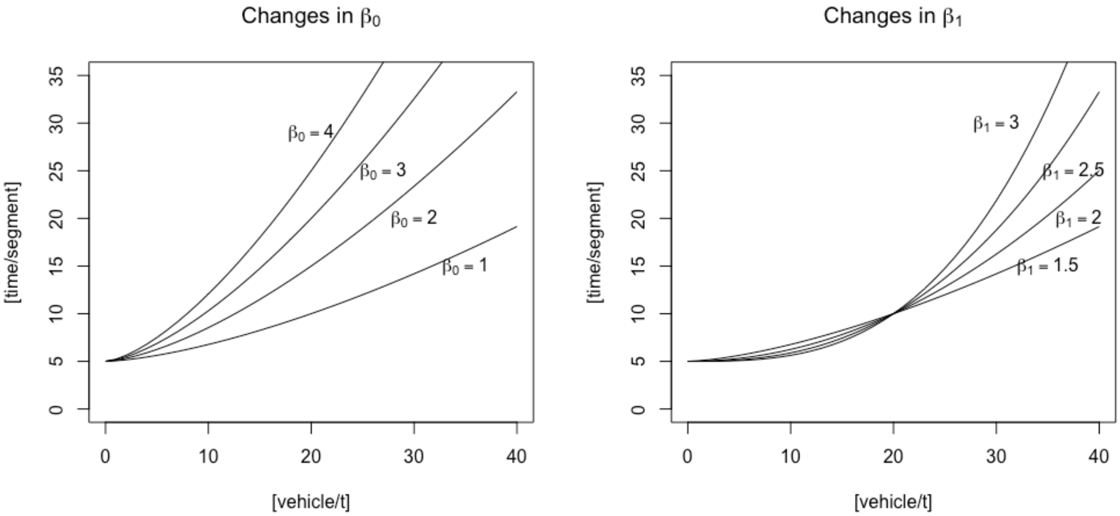


Figure 5.6 BRP travel time function with *cap = 20* [vehicle]

This thesis followed [55] choosing $\beta_0 = 1$, meaning that at full capacity the driver's time duplicates. For the selection of β_1, ranges used in literature vary significantly from 1 to 12 [54]-[56]. We tried different values and chose $\beta_1 = 3$ because it allowed to have a higher flow of cars before producing a significantly increase in travel time (at 50% capacity, travel time increases 12%).

5.3.2 Road Segment Speed

Current speed on road segment *k* is computed using

$$v_k = \frac{d_k}{T_k(w_k)}$$

where v_k is the speed in units [segment/t] or [km/t], d_k distance of the segment and $T_k(.)$ the travel time function based on the current volume of vehicles w_k. Using the BPR travel time function and expressing the free-flow time term, t_f, in terms of the free-flow speed: $t_f = d_k/v_f$, the following is obtained

$$v_k = \frac{d_k}{\frac{d_k}{v_f}\left\{1 + \beta_0\left(\frac{w_k}{cap_k}\right)^{\beta_1}\right\}}$$

$$v_k = \frac{v_f}{\left\{1 + \beta_0 \left(\frac{w_k}{cap_k}\right)^{\beta_1}\right\}} \tag{5.6}$$

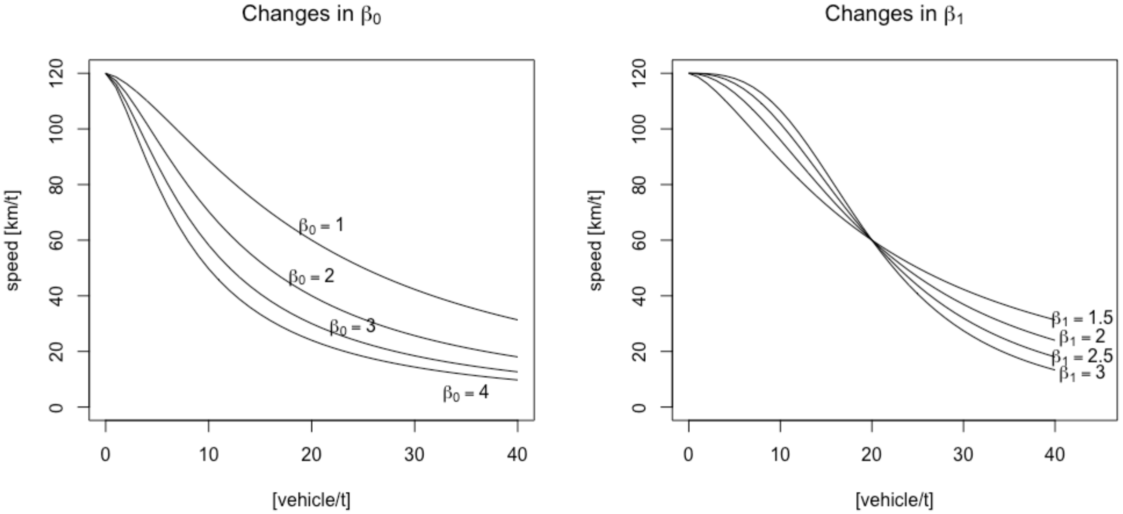


Figure 5.7 Road segment speed function with *cap = 20* [vehicles]

5.3.3 Road density

Density refers to the number of vehicles on the road segment [54], [56]; its relation to the volume of vehicles and capacity is the following

$$w_k = v_f \rho_k \left(1 - \frac{\rho_k}{\rho_{jam}}\right)$$

$$cap_k = \frac{v_f \rho_{jam}}{4}$$

where ρ_k is the density of the link k in units [vehicle/segment] and ρ_{jam} is the maximum density achieved under congestion in units [vehicle/segment]. Here v_f is expressed in units [segment/t].

In the next chapter we develop an algorithm to estimate the state of a system incorporating observations from agents.

Chapter 6 Bayesian State Estimation using Collective Information

Introduction

This chapter describes the development of an algorithm to estimate system states based on current information observed by agents. First the problem of estimation is presented as well as the definition of agent perception. Then the Bayesian mechanism to update observations is described, and some examples are presented. The process of diffusion of probability is explained; its relevance to real-time systems; and some examples are also solved. The algorithm with Bayesian and diffusion mechanism in then presented as well as its numerical implementation. Finally, agent learning is described which is the individual procedure to correct any systematic bias on estimations.

6.1 Definitions and Assumptions

The system of interest is defined as the dynamic interaction between a sequence of subsystem states $\{\mathcal{S}_t\}$ (e.g. a road segment, attendance in a beach) and the finite set of *autonomous* agents $\mathcal{Z}$ whose actions are based on and affect $\{\mathcal{S}_t\}$.

Autonomy refers to the capacity of each agent to make decisions on its own and act accordingly based on its *goals* and current *estimation* of $\mathcal{S}_t$. *Goals* are described by a utility function $\mathcal{U}$ which agents optimize every time they make a decision (e.g. time, satisfaction).

Agents also have two main properties: *perception* and *learning*. *Perception* is the mapping from the true state of the system to a *perceived* state and *learning* is the adjustment made to the *perceived* state to eliminate any bias. The latter requires historical information while the former does not.

It is assumed there is a subset of agents, $\mathcal{A}$: $\mathcal{A} \subseteq \mathcal{Z}$, who share their *perceived* states and have access to a collectively *perceived* state to improve their decisions. Therefore, a mechanism by which agents in $\mathcal{A}$ are able to update and extract current information before making a decision is assumed to exist (e.g. an app, an online site). $\mathcal{B} = \mathcal{Z} \setminus \mathcal{A}$ is the set of agents who do not share information nor have access to any collective information.

Some definitions

1) Discrete time index $t \in \mathbb{R}, \forall t$
2) Continuous state domain $\mathcal{S} \in \mathbb{R}$
3) Set of autonomous agents $\mathcal{Z} = \mathcal{A} \cup \mathcal{B}, \mathcal{A} \cap \mathcal{B} = \emptyset$
4) $\mathcal{A}$ is the partition who share and have access to collective information and $\mathcal{B}$ does not share nor have access to collective information
5) Agent perception mapping $\mathcal{P}: \mathcal{S} \rightarrow \Lambda$
6) Collective estimation $\mathcal{E}: \{\Lambda\} \rightarrow \hat{\mathcal{S}}$
7) Agent learning mapping $\mathcal{L}: \hat{\mathcal{S}} \rightarrow \check{\mathcal{S}}$

$P(\mathcal{S}_t{}^k)$ denotes the probability density function of the subsystem $\mathcal{S}_t$ at time t and after receiving observations from k agents. For simplicity, $P(\mathcal{S}_t{}^0)$ may be simply referred to as $P(\mathcal{S}_t)$. The following sections discuss the method to gather *perceptions* from $\mathcal{A}$, and section 6.7 studies the process of adjusting that information so as to remove any bias.

6.2 Agent Perception

Agent perception deals with the problem of estimating a true state $\mathcal{S}_t$ from a state perceived Λ_t at time t. When we have access to more than one observation, then a Bayesian procedure to incorporate new information can be used to obtain a better estimate. This procedure is aimed only at agents in $\mathcal{A}$ who are able to share and retrieve information.

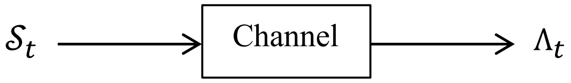


Figure 6.1 Diagram of an agent's perception

We want to know the probability of a true state given a vector of observations collected during a time interval Δt = t – t0. The noise or error introduced by the channel is assumed to be known and described by the prior pdf $P(\Lambda_t|\mathcal{S}_t)$ and that observations are conditionally independent, i.e., $P(\lambda_t{}^1, \lambda_t{}^2|\mathcal{S}_t) = P(\lambda_t{}^1|\mathcal{S}_t)\, P(\lambda_t{}^2|\mathcal{S}_t)$. The rationale behind conditional independence is that noise is agent-independent; therefore, the joint probability of having two or more agents perceiving the same true state is described by the multiplication of the priors given the true state.

To properly estimate the true state $\mathcal{S}_t$, two processes must be taken into consideration: the addition of new information and the process of forgetting past information. At times these two may oppose or reinforce each other. The reason why we need a procedure to 'forget' information is because the Bayesian scheme fails to adapt to a changing state, which is what agents are interested in knowing. For example, say that during a Δt we received *n* observations stating $\mathcal{S}_t = 0$, now say that for the next time step the state has changed to $\mathcal{S}_{t+\Delta t} = 1$, we would need *n* additional observations just to even out the probabilities of both states, which is inefficient.

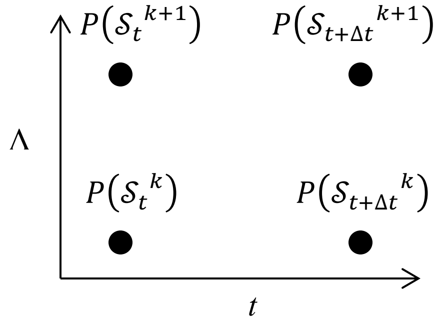

$P(\mathcal{S}_t{}^{k+1})$
$P(\mathcal{S}_{t+\Delta t}{}^{k+1})$
Λ
$P(\mathcal{S}_t{}^{k})$
$P(\mathcal{S}_{t+\Delta t}{}^{k})$
t

Figure 6.2 Diagram of the evolution of $P(\mathcal{S}_t{}^k)$

The aim of the algorithm is to produce the most accurate function $P(\mathcal{S}_t)$, based on the information received and the time when it was received.

We would like to have a function to update $P(\mathcal{S}_t)$ when we received an observation $\{{\lambda_{\Delta t}}^{\Delta k}\}$ at time Δt. To compute $P({\mathcal{S}_{t+\Delta t}}^{k+\Delta k})$ we used a first-order Taylor series approximation at point [t, k]

$$P({\mathcal{S}_{t+\Delta t}}^{k+\Delta k}) = P({\mathcal{S}_t}^k) + \nabla P({\mathcal{S}_t}^k) * [\Delta t, \Delta k]^T$$

$$= P({\mathcal{S}_t}^k) + \Delta \mathrm{t}\frac{\partial}{\partial t}P({\mathcal{S}_t}^k) + \Delta k\frac{\partial}{\partial k}P({\mathcal{S}_t}^k) \qquad (6.1)$$

The incorporation of new observations is described by the term $\frac{\partial}{\partial k}P({\mathcal{S}_t}^k)$ and a Bayesian updating step was used to determine its form. The term $\frac{\partial}{\partial t}P({\mathcal{S}_t}^k)$ describes the diffusion of probability, i.e., the spreading of probability over its domain as time goes on. This term could also be thought of as a process of forgetting past information because as the pdf becomes flatter its uncertainty increases and previous information is lost.

6.3 Addition of New Information

This process is the assimilation of new perceptions $\Lambda_t = \{{\lambda_t}^1, {\lambda_t}^2, {\lambda_t}^3, \dots, {\lambda_t}^k\}$ made by *k* agents about the current state $\mathcal{S}_t$ during Δt. The Bayesian framework was used to update the pdf of the system's state, i.e., $P(\mathcal{S}_t)$. We chose the updated function to be a linear combination of its previous value and its posterior pdf. After the first perception is received, $P(\mathcal{S}_t)$ changes to

$$P({\mathcal{S}_t}^1) = \pi_A P({\mathcal{S}_t}^0 | {\lambda_t}^1) + (1 - \pi_A) P({\mathcal{S}_t}^0)$$

where π_A is the speed of assimilation of new information, the general formula is

$$P({\mathcal{S}_t}^k) = \pi_A P({\mathcal{S}_t}^{k-1} | {\lambda_t}^k) + (1 - \pi_A) P({\mathcal{S}_t}^{k-1}), \quad \pi_A \in [0,1] \qquad (6.2)$$

where $P({\mathcal{S}_t}^k)$ is the pdf of the system after receiving *k* perceptions from the set of agents.

By setting π_A=1 eq. (6.2) becomes commutative, meaning that the order in which we receive and compute perceptions during Δt does not alter the form of $P({\mathcal{S}_t}^k)$. Consider the case of $P({\mathcal{S}_t}^{2,1})$, which is the pdf after incorporating $\{{\lambda_t}^1, {\lambda_t}^2\}$ in that order. It can be shown that it is equivalent to $P({\mathcal{S}_t}^{1,2})$, i.e., incorporating $\{{\lambda_t}^2, {\lambda_t}^1\}$ as follows

$$P\left(\mathcal{S}_t^{\,2,1}\right) = \frac{P\left(\lambda_t^{\,2}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,2}|\lambda_t^{\,1}\right)} P\left(\mathcal{S}_t^{\,1}\right)$$

replacing $P\left(\mathcal{S}_t^{\,1}\right) = \frac{P\left(\lambda_t^{\,1}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,1}\right)} P(\mathcal{S}_t)$ using Bayes theorem

$$= \frac{P\left(\lambda_t^{\,2}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,2}|\lambda_t^{\,1}\right)} \left[\frac{P\left(\lambda_t^{\,1}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,1}\right)} P(\mathcal{S}_t)\right]$$

replacing $P\left(\lambda_t^{\,2}|\lambda_t^{\,1}\right)P\left(\lambda_t^{\,1}\right) = P\left(\lambda_t^{\,1},\lambda_t^{\,2}\right)$

$$= \frac{P\left(\lambda_t^{\,2}|\mathcal{S}_t\right)P\left(\lambda_t^{\,1}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,1},\lambda_t^{\,2}\right)} P(\mathcal{S}_t)$$

replacing $P\left(\lambda_t^{\,1},\lambda_t^{\,2}\right) = P\left(\lambda_t^{\,2}\right)P\left(\lambda_t^{\,1}|\lambda_t^{\,2}\right)$ and regrouping

$$= \left[\frac{P\left(\lambda_t^{\,2}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,2}\right)} P(\mathcal{S}_t)\right] \frac{P\left(\lambda_t^{\,1}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,1}|\lambda_t^{\,2}\right)}$$

replacing $\frac{P\left(\lambda_t^{\,2}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,2}\right)} P(\mathcal{S}_t) = P\left(\mathcal{S}_t^{\,2}\right)$ using Bayes theorem

$$= P\left(\mathcal{S}_t^{\,2}\right) \frac{P\left(\lambda_t^{\,1}|\mathcal{S}_t\right)}{P\left(\lambda_t^{\,1}|\lambda_t^{\,2}\right)}$$

finally, using Bayes theorem and the conditionally independence assumption we get

$$= P\left(\mathcal{S}_t^{\,1,2}\right).$$

We would like to express eq. (6.2) in a finite difference scheme because we are more interested in the *change* in the pdf rather than its new value. This allow to combine the change of adding information and the change of forgetting past information as seen on eq. (6.1). Eq. (6.2) can be reexpressed in the following way

$$\frac{P\left(\mathcal{S}_t^{\,k+\Delta k}\right) - P\left(\mathcal{S}_t^{\,k}\right)}{\Delta k} = \pi_A P\left(\mathcal{S}_t^{\,k}|\lambda_t^{\,k}\right) + (1-\pi_A)P\left(\mathcal{S}_t^{\,k}\right) - P\left(\mathcal{S}_t^{\,k}\right)$$

$$= \pi_A\left[P\left(\mathcal{S}_t^{\,k}|\lambda_t^{\,k+\Delta k}\right) - P\left(\mathcal{S}_t^{\,k}\right)\right]$$

where Δk=1, therefore the difference equation is

$$\frac{\Delta P(\mathcal{S}_t)}{\Delta k} = \pi_A\left[P\left(\mathcal{S}_t|\lambda_t{}^k\right) - P(\mathcal{S}_t)\right]. \tag{6.3}$$

The posterior probability is computed according to

$$P(\mathcal{S}_t|\lambda_t) = P(\mathcal{S}_t = s|\lambda_t) = \frac{P(\lambda_t|\mathcal{S}_t = s)\, P(\mathcal{S}_t = s)}{P(\lambda_t)}, \qquad \forall s \in \mathcal{S}_t$$

where $P(\lambda_t) = \int P(\lambda_t|\mathcal{S}_t)\, P(\mathcal{S}_t)ds$ is a normalizing constant whose value can be obtained by precomputing $P(\lambda_t|\mathcal{S}_t)\, P(\mathcal{S}_t), \forall s$ and find the area under such curve. Eq. (6.3) can be reexpressed in an equivalent form

$$\frac{\Delta P(\mathcal{S}_t)}{\Delta k} = \pi_A\left[\frac{P\left(\lambda_t{}^k|\mathcal{S}_t\right)}{P\left(\lambda_t{}^k\right)} - 1\right] P(\mathcal{S}_t) \tag{6.4}$$

Example 6.1

With the purpose of showing how eq. (6.3) works, let us consider a system with a bounded continuous domain and a truncated Gaussian noise

$$\mathcal{S}_t = \Lambda_t + n_t, \qquad \mathcal{S}_t, \Lambda_t, n_t \in [a, b], \qquad n_t \sim N_{Trunc}(0, \sigma_n^2 = 1)$$

Therefore,

$$P(\lambda_t|\mathcal{S}_t = s) = \frac{exp\left(-\frac{(\lambda - s)^2}{2\sigma_N^2}\right)}{\sigma_n\sqrt{2\pi}\, C} \tag{6.5}$$

C is the normalizing constant and is equal to the area under the curve between *a* and *b*, the bounded region, i.e., C = F(b) – F(a), F(.) being the cumulative distribution function. Starting from an initial uniform distribution function

$$P(\mathcal{S}_0) = \begin{cases} 1/(b-a), & s \in [a, b] \\ 0, & elsewhere \end{cases}$$

the evolution of $P(\mathcal{S}_t)$ will be computed, having [a, b]=[0, 10], σ_n=1, a sequence of perceptions received from agents {5,5,5,5,5} and {3,4,5,6,7}, and π_A=1/2 (Figures 6.3 and 6.4).

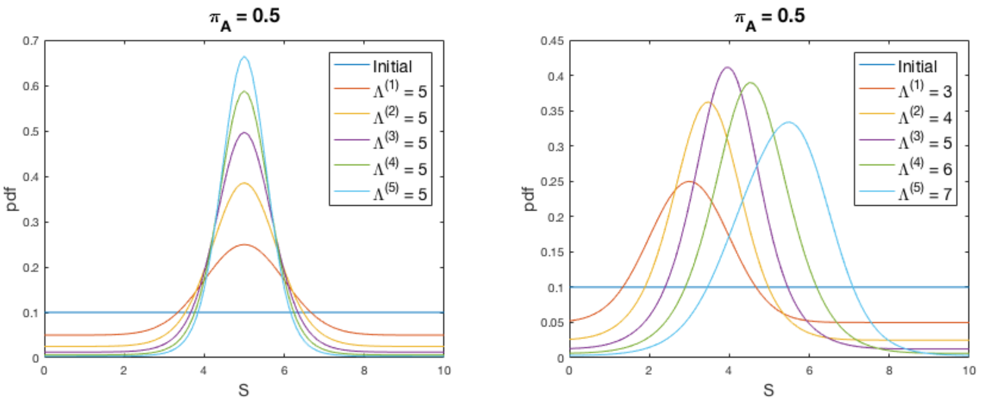


Figure 6.3 Evolution of P(S) as new information is incorporated π_A=1/2

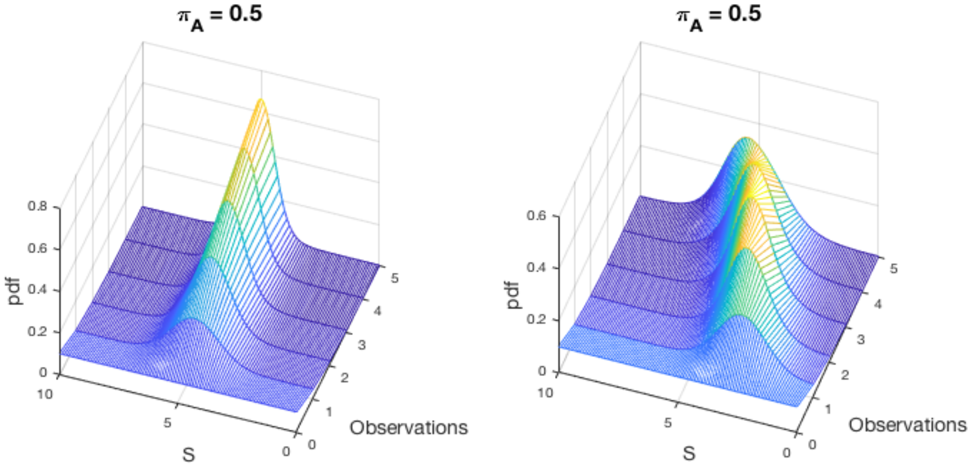


Figure 6.4 Mesh plot of P(S) as observations are incorporated Λ={5,5,5,5,5} (left) and Λ={3,4,5,6,7} (right) , π_A=1/2

Now, setting π_A=1/20 for the same sequences of perceptions, the updated function becomes flatter than before (Figures 6.5 and 6.6).

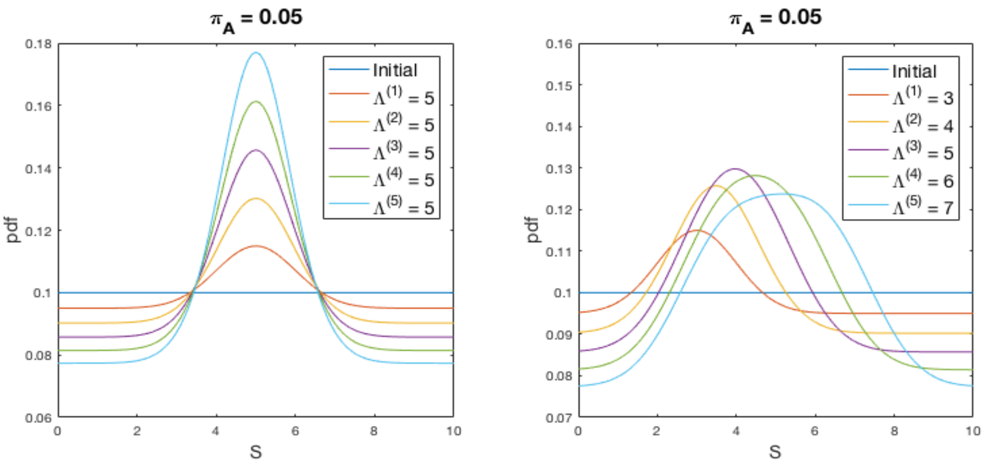


Figure 6.5 Evolution of P(S) as new information is incorporated π_A=1/20

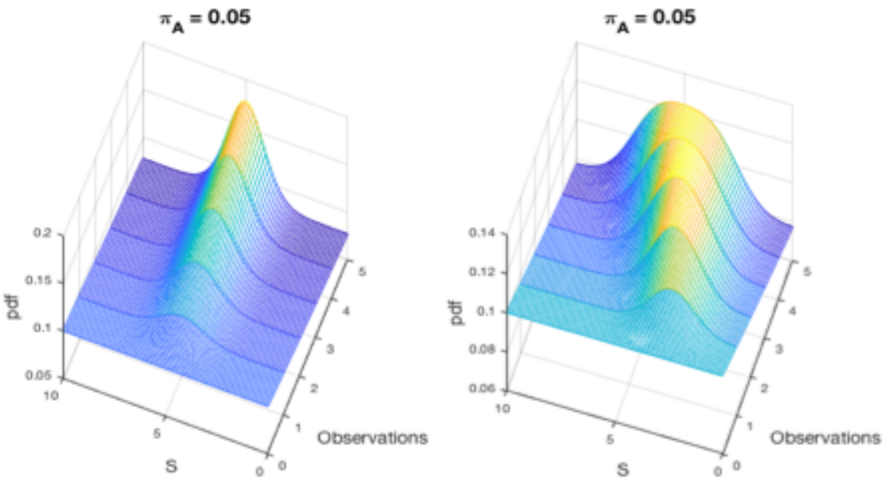


Figure 6.6 Mesh plot of P(S) as observations are incorporated Λ={5,5,5,5,5} (left) and Λ={3,4,5,6,7} (right) , π_A=1/20

Setting π_A=1 the commutative property is preserved; therefore, if we receive a sequence like {3,4,5,6,7} it yields the same $P(\mathcal{S}_t)$ as if we had received {7,6,5,4,3} or the same values in any other order.

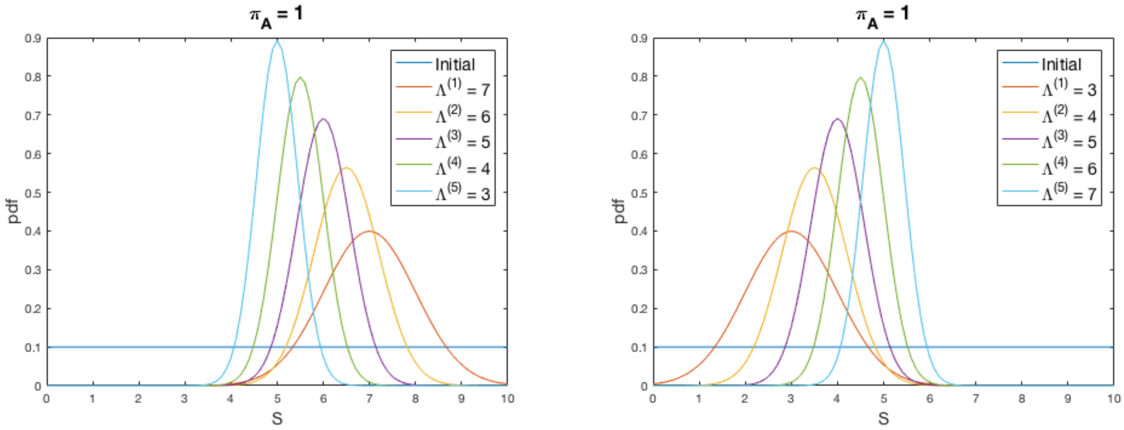


Figure 6.7 Commutative property in $P(\mathcal{S}_t)$ preserved when π_A=1

It is worth mentioning that the area under the curve $P(\mathcal{S}_t)$ is equal to one and that feature is preserved after applying the finite difference scheme. Therefore, we need not rescale it for it is always a probability distribution function (See Appendix E).

6.4 Process of Forgetting Information

Two approaches were explored to model the increase of uncertainty as a function of time, i.e., the process of forgetting past observations made by agents. The first approach is to derive a function similar to diffusion, dubbed 'diffusion-like', and the second is to use a diffusion term.

6.4.1 Diffusion-like process

For the first approach the rationale is the following: if we have a double bounded domain on $\mathcal{S}_t$, no matter which shape $P(\mathcal{S}_t)$ has, we know that over a long period of time and with no new information added, its final form should be uniform (equiprobable), and it should be steady state (not changing any more). Therefore, it was assumed that the change in probability is proportional to the difference between its steady state value and the current one:

$$\frac{\partial}{\partial t} P(\mathcal{S}_t) = \gamma \left[\frac{1}{b-a} - P(\mathcal{S}_t)\right] \tag{6.6}$$

where $\gamma \geq 0$ controls the speed of the change. The steady state is reached when $P(\mathcal{S}_t) = \frac{1}{b-a}$.

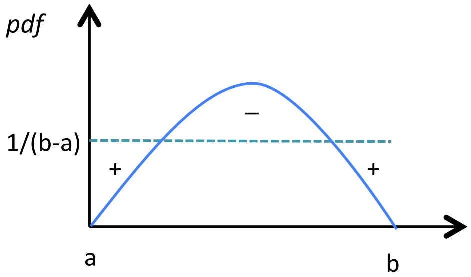


Figure 6.8 The diffusion-like process increases values lower than 1/(b-a) and decreases values higher than that

The next step is to find the analytical solution of eq. (6.6), which is directly integrable:

$$\int \left[\frac{1}{b-a} - P(\mathcal{S}_t)\right]^{-1} \partial P(\mathcal{S}_t) = \int \gamma \, \partial t$$

$$-ln\left[\frac{1}{b-a} - P(\mathcal{S}_t)\right] = \gamma t + c$$

$$ln\left[\frac{1}{b-a} - P(\mathcal{S}_t)\right] = -\gamma t - c$$

$$\frac{1}{b-a} - P(\mathcal{S}_t) = exp(-\gamma t)\, exp(-c)$$

setting $a_0 = exp(-c)$ and solving for $P(\mathcal{S}_t)$

$$P(\mathcal{S}_t) = \frac{1}{b-a} - a_0 exp(-\gamma t) \tag{6.7}$$

The initial condition of eq. (6.7) is $P(\mathcal{S}_{t_0}) = P_0$, therefore $a_0 = \frac{1}{b-a} - P_0$. If the function is computed at intervals Δt (not necessary of the same length), then it can reexpress eq. (6.7) in terms of its current value rather than its initial, this is because of the Markov property of the exponential function. We do so by setting $a_0 = \frac{1}{b-a} - P(\mathcal{S}_t)$, replacing this term into eq. (6.7) and simplifying; we obtained

$$P(\mathcal{S}_{t+\Delta t}) = [1 - exp(-\gamma\, \Delta t)]\frac{1}{b-a} + exp(-\gamma\, \Delta t)P(\mathcal{S}_t)$$

subtracting $P(\mathcal{S}_t)$ from both sides of this equation to obtain the increment in the pdf and arranging terms we get

$$\Delta P(\mathcal{S}_{t+\Delta t}) = [1 - exp(-\gamma\, \Delta t)] \left[\frac{1}{b-a} - P(\mathcal{S}_t)\right] \tag{6.8}$$

Example 6.2

An interesting initial condition was created to show how the probability density function converges to its steady state as time goes on, the initial condition is

$$P(\mathcal{S}_{t_0}) = \begin{cases} \dfrac{1+sin(s)}{2\pi}, & s \in [0, 2\pi] \\ 0, & elsewhere. \end{cases}$$

The evolution over time of the pdf was computed using eq. (6.8). The example converged to its steady state $P(\mathcal{S}_t) = \frac{1}{2\pi}$ as shown in Figure 6.9.

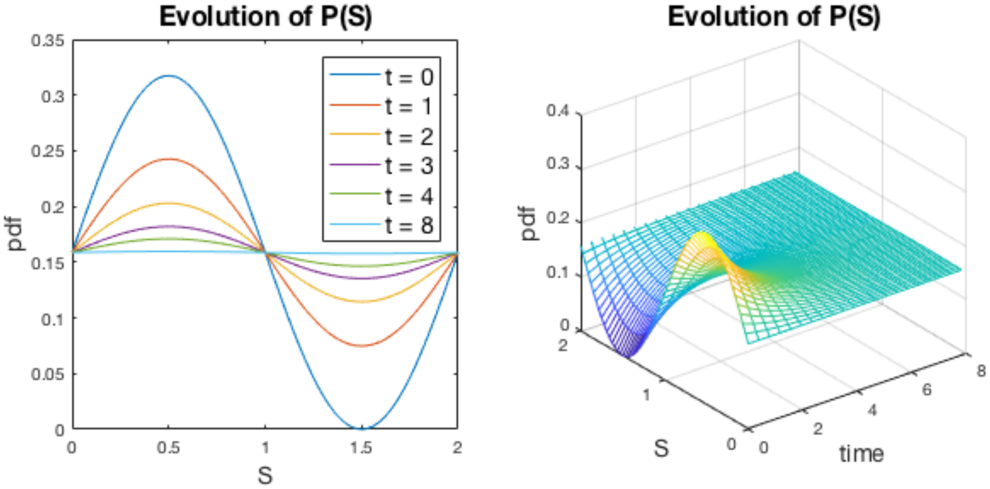


Figure 6.9 Evolution of $P(\mathcal{S}_t)$ using the diffusion-like process of eq. (6.7)

6.4.2 Diffusion process

The diffusion-like process is driven by the gradient between the current value of the pdf at every point of its domain and its steady state. Diffusion, in change, is driven by the gradient of every point with respect to its neighbors.

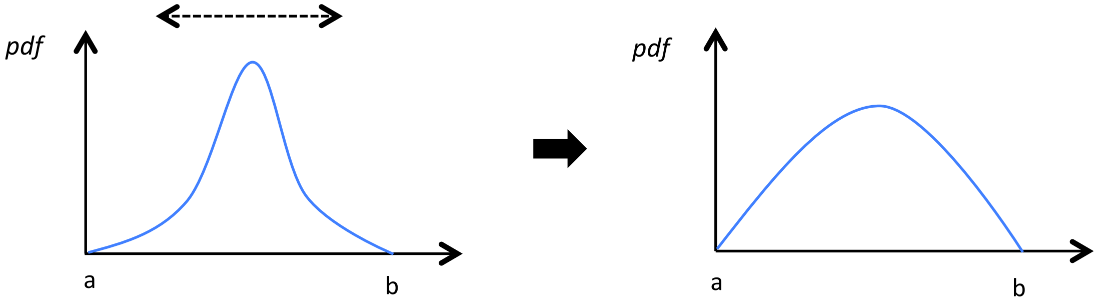


Figure 6.10 Diffusion spreads the function

The diffusion, as described previously on Chapter 3, is described by

$$\frac{\partial}{\partial t}P(\mathcal{S}_t) = D_p\frac{\partial^2}{\partial s^2}P(\mathcal{S}_t) \tag{6.9}$$

D_p is the diffusion coefficient of probability. It is also necessary to specify the left and right boundary conditions. In order to avoid 'losing' probability since we are using a double-bounded domain on $\mathcal{S}$, insulating boundaries were implemented, i.e., von Neumann conditions equal to zero.

$$\frac{\partial}{\partial s}P(\mathcal{S}_t = a) = 0, \qquad \forall t$$

$$\frac{\partial}{\partial s}P(\mathcal{S}_t = b) = 0, \qquad \forall t$$

Solving the same example 6.2 using implicit finite differences, the following evolution of $P(\mathcal{S}_t)$ towards steady state was obtained

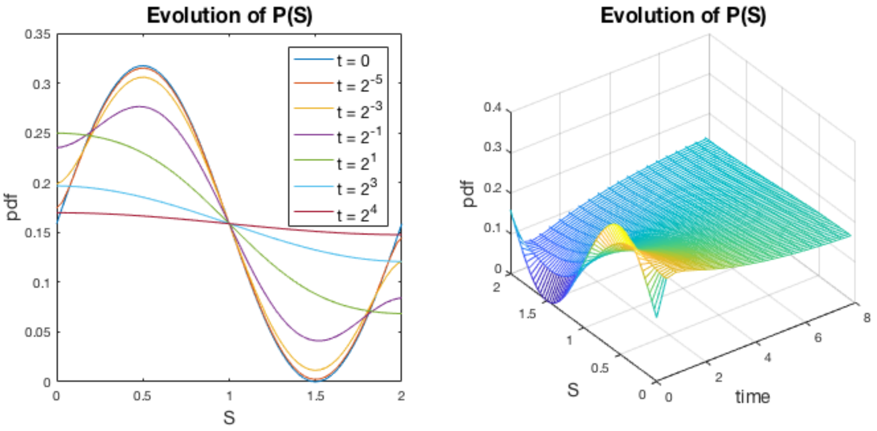


Figure 6.11 Evolution of $P(\mathcal{S}_t)$ using the diffusion process of eq. (6.9)

After comparing this solution to the one on Figure 6.9, we noticed that diffusion shifts the shape of the curve while the diffusion-like process rescales it.

6.5 Most Probable State

At any time, we would like to extract a value from the domain $\mathcal{S}$ that best represents the current state; the point in the domain with the maximum probability was chosen:

$$\hat{s}_t = arg \max_{s \in \mathcal{S}} P(\mathcal{S}_t)$$

In the unlikely case that $\hat{s}_t$ is more than one point, the average of the modes could be computed and set $\hat{s}_t$ equal to that value. When there is a flat or fairly flat $P(\mathcal{S}_t)$, then it could be set to a default value for $\hat{s}_t$. Such value will depend on the context of the problem. For the speed estimation problem of Chapter 7, it seems reasonable to set $\hat{s}_t$ equal to its maximum value, i.e., free-flow speed, because having no information may be due to the road being empty.

Therefore, the process is summarized as follows

If $max\ [P(\mathcal{S}_t)] - min\ [P(\mathcal{S}_t)] \leq \delta$, then $\hat{s}_t = s_{default}$

Otherwise $\hat{s}_t = arg \max_{s \in \mathcal{S}} P(\mathcal{S}_t)$

Where δ is a cutoff value, e.g., δ = E-2.

Example 6.3

The next step is to solve a case where observations are received as time goes on. Suppose there is an agent moving from along a road segment and at every minute it reports back its speed. Say its speed is bounded by [a, b] = [0, 2] [km/min]. We would like to have an idea of how crowded the road is from the information provided by the agent. The agent provides the following data [2, 2, 2, 0, 0, 0, 1, 1, 1, N/A, N/A, N/A], where N/A represents no communication received. Extreme values were chosen to show how both processes behave in upset conditions.

Using the diffusion-like process and the following input values, the evolution of the pdf describing the state of the road was computed, assuming a gaussian noise as described in eq. (6.5) with $\sigma_N = 1/6\ [km/min]$; diffusion-like parameters are γ = ½, $\pi_A = 0.1$; and the initial function is

$$P(\mathcal{S}_{t_0}) = \begin{cases} \frac{1}{2}, & s \in [0,2] \\ 0, & elsewhere. \end{cases}$$

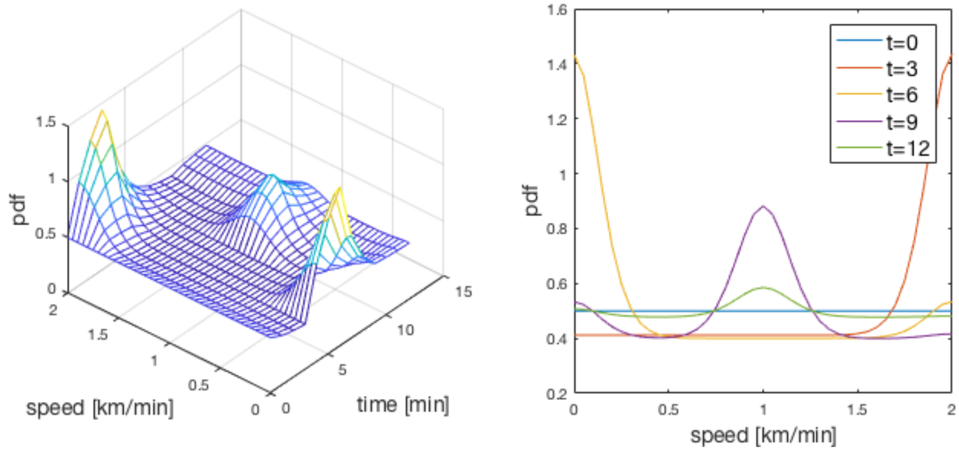


Figure 6.12 Evolution of pdf using the diffusion-like process

The diffusion-like process is quick to respond to changes of state as seen in Figure 6.12 and 6.13. By extracting the value of the domain with highest probability at every time step, the following sequence is obtained [2, 2, 2, 2, 0, 0, 0, 1, 1, 1, 1, 1], which fails on two out of nine numerical values. The contour plot shows how the probability concentrates around the observations received.

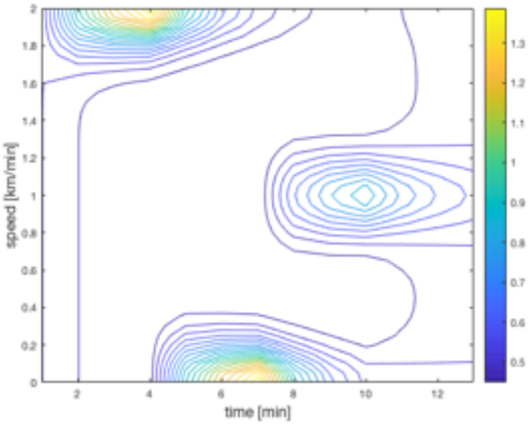


Figure 6.13 Contour plot of the solution using diffusion-like process

Now, solving the same problem using the diffusion process with the same inputs except for $D_p = 0.1$ [s^2/min] and $\pi_A = 0.5$; the evolution of the pdf over time is shown in Figure 6.14.

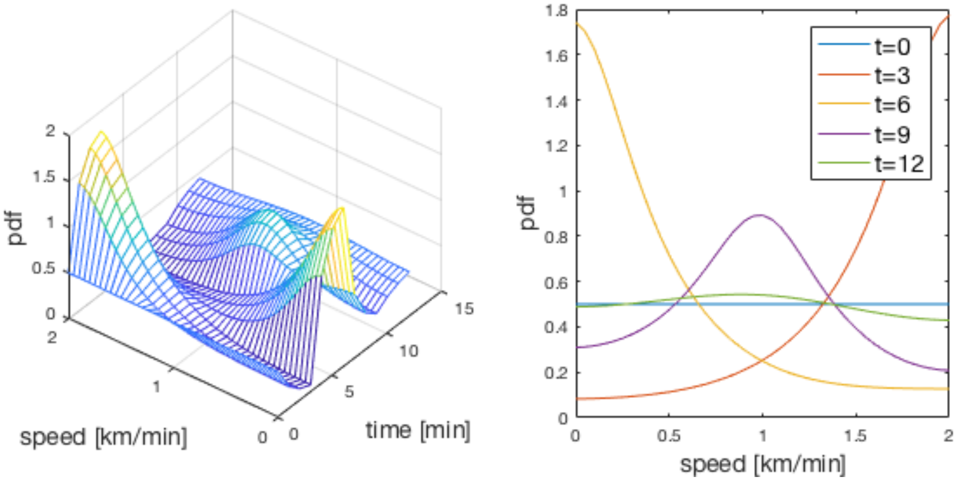


Figure 6.14 Evolution of the pdf using the diffusion process

The contour plot shows how the probability spreads around the neighborhood of the observation, which is a characteristic of the diffusion term. The sequence of most likely states using the diffusion process is [2, 2, 2, 0, 0, 0, 0.9, 0.95, 1, 0.95, 0.95, 0.9], this process also failed in two out of nine numerical values.

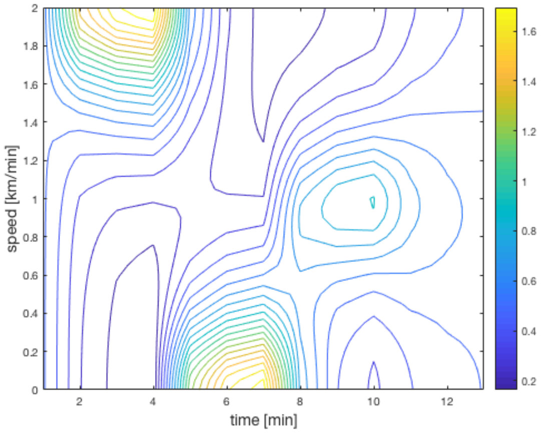


Figure 6.15 Contour plot of the solution using diffusion process

An important aspect to mention is that when the numerical diffusion is applied, the pdf has to be rescaled to ensure its area is equal to one, this is due to the numerical error of the numerical method.

6.6 Numerical Implementation

Finding an analytical solution to both models is not an option for two reasons: 1) the updating step using Bayesian probability is only defined for discrete changes in the observations, 2) since the pdf $P(\mathcal{S}_t)$ may take any shape, it complicates greatly any attempt to derive an analytical solution for the diffusion process. Therefore, a numerical method may provide us an accurate and fast solution. In particular, we would like to have an unconditionally stable scheme; so that its implementation is unconditionally stable regardless of the values chosen for the diffusion parameters.

The observations received during time t are

$$\boldsymbol{\Lambda_t} = \{\lambda_t{}^1, \lambda_t{}^2, \lambda_t{}^3, \dots, \lambda_t{}^{k_t}\}$$

meaning that k_t agents sent their *perception* at time t.

6.6.1 Diffusion-like Process Implementation

From eq. (6.1), the governing equation is

$$P(\mathcal{S}_{t+\Delta t}{}^{i+1}) = P(\mathcal{S}_t{}^i) + \Delta k \frac{\partial}{\partial k} P(\mathcal{S}_t{}^i) + \Delta \mathrm{t} \frac{\partial}{\partial t} P(\mathcal{S}_t{}^i)$$

and its finite difference representation is

$$P(\mathcal{S}_{t+\Delta t}{}^{i+1}) = P(\mathcal{S}_t{}^i) + \Delta k \frac{\Delta P(\mathcal{S}_t{}^i)}{\Delta k} + \Delta \mathrm{t} \frac{\Delta P(\mathcal{S}_t{}^i)}{\Delta t}$$

replacing the difference terms with those form eq. (6.3) and (6.8)

$$P(\mathcal{S}_{t+\Delta t}{}^{i+1}) = P(\mathcal{S}_t{}^i) + \pi_A[P(\mathcal{S}_t|\lambda_t{}^{i+1}) - P(\mathcal{S}_t{}^i)] + [1 - exp(-\gamma\, \Delta t)] \left[\frac{1}{b-a} - P(\mathcal{S}_t{}^i)\right].$$

Taking the first two terms of the RHS, rearranging and setting them equal to a new variable. Then reexpressing the third RHS term with respect to this new variable

$$P(\mathcal{S}_t{}^i) + \pi_A[P(\mathcal{S}_t|\lambda_t{}^{i+1}) - P(\mathcal{S}_t{}^i)] = (1 - \pi_A)P(\mathcal{S}_t{}^i) + \pi_A P(\mathcal{S}_t|\lambda_t{}^{i+1}) = P_{cond}{}^{i+1}$$

$$P\left(\mathcal{S}_{t+\Delta t}{}^{i+1}\right) = \ P_{cond}{}^{i+1} + [1 - exp(-\gamma\ \Delta t)]\left[\frac{1}{b-a} - P_{cond}{}^{i+1}\right]$$

$$= \ exp(-\gamma\ \Delta t)\ P_{cond}{}^{i+1} + [1 - exp(-\gamma\ \Delta t)]\ \frac{1}{b-a} \qquad (6.10)$$

This equation must be applied on each point in the domain of $\mathcal{S}$.

Summary

Therefore, the algorithm of the diffusion-like process for each time step containing $\boldsymbol{\Lambda_t} = \{\lambda_t{}^1, \lambda_t{}^2, \lambda_t{}^3, \dots, \lambda_t{}^{k_t}\}$ observations can be written as follows

1. If $k_t = 0, i.e., \boldsymbol{\Lambda_t} = \emptyset$, go to step 3b.

2. Apply the Bayesian updating scheme.

For $i \in \{1, 2, \dots, k_t - 1\}$

$$P_{cond}{}^{i+1} = (1 - \pi_A) P_{cond}{}^{i} + \pi_A P\left(\mathcal{S}_t | \lambda_t{}^{i+1}\right)$$

where

$$P\left(\mathcal{S}_t | \lambda_t{}^{i+1}\right) = \frac{P\left(\lambda_t{}^{i+1} | \mathcal{S}_t\right) .* \ P_{cond}{}^{i}}{P\left(\lambda_t{}^{i+1}\right)}$$

.* means element-wise multiplication. Go to step 3a.

3a. Apply the diffusion-like term

$$P\left(\mathcal{S}_{t+\Delta t}{}^{k}\right) = \ exp(-\gamma\ \Delta t)\ P_{cond}{}^{k_t} + [1 - exp(-\gamma\ \Delta t)]\ \frac{1}{b-a}$$

3b. Apply the diffusion-like term

$$P\left(\mathcal{S}_{t+\Delta t}{}^{k}\right) = \ exp(-\gamma\ \Delta t)\ P\left(\mathcal{S}_t{}^0\right) + [1 - exp(-\gamma\ \Delta t)]\ \frac{1}{b-a}$$

The parameter domains are

$$\pi_A \in [0,1], \qquad \gamma \in [0, \infty).$$

As mentioned before, this algorithm is unconditionally stable for the diffusion-term, setting a γ = 0 deactivates the diffusion term, while γ >> 0 sets $P(\mathcal{S}_t)$ close to $\frac{1}{b-a}$.

6.6.2 Diffusion Process Implementation

In order to have an unconditionally stable scheme, we would have to use backward differentiation with respect to the diffusion term. Again, we start with eq. (6.1)

$$P(\mathcal{S}_{t+\Delta t}{}^{i+1}) = P(\mathcal{S}_t{}^{i}) + \Delta k \frac{\partial}{\partial k} P(\mathcal{S}_t{}^{i}) + \Delta \mathrm{t} \frac{\partial}{\partial t} P(\mathcal{S}_t{}^{i}).$$

Moving the derivative term to the LHS

$$P(\mathcal{S}_{t+\Delta t}{}^{i+1}) - \Delta \mathrm{t} \frac{\partial}{\partial t} P(\mathcal{S}_t{}^{i}) = P(\mathcal{S}_t{}^{i}) + \Delta k \frac{\partial}{\partial k} P(\mathcal{S}_t{}^{i}). \qquad (6.11)$$

Replacing the time derivative by the diffusion term on the LHS and appling backward differentiation

$$P(\mathcal{S}_{t+\Delta t}{}^{i+1}) - \Delta \mathrm{t} \frac{\partial}{\partial t} P(\mathcal{S}_t{}^{i}) = P(\mathcal{S}_{t+\Delta t}{}^{i+1}) - D_p t \frac{\partial^2}{\partial s^2} P(\mathcal{S}_t{}^{i})$$

$$= P^{i+1}_{t+\Delta t,\ j} - D_p \frac{\Delta t}{\Delta s^2} \left(P^{i+1}_{t+\Delta t,\ j+1} - 2P^{i+1}_{t+\Delta t,\ j+1} + P^{i+1}_{t+\Delta t,\ j-1} \right)$$

$$= -D_p \frac{\Delta t}{\Delta s^2} P^{i+1}_{t+\Delta t,\ j+1} + \left(1 + 2D_p \frac{\Delta t}{\Delta s^2} \right) P^{i+1}_{t+\Delta t,\ j} - D_p \frac{\Delta t}{\Delta s^2} P^{i+1}_{t+\Delta t,\ j+1}$$

$$= -\theta\, P^{i+1}_{t+\Delta t,\ j+1} + (1 + 2\theta) P^{i+1}_{t+\Delta t,\ j} - \theta\, P^{i+1}_{t+\Delta t,\ j+1}$$

the index $j \in [a, b]$ refers to the partition on the domain $\mathcal{S}$ and $\theta = D_p \frac{\Delta t}{\Delta s^2}$.

The RHS of eq. (6.11) is

$$P(\mathcal{S}_t{}^{i}) + \pi_A [P(\mathcal{S}_t | \lambda_t{}^{i+1}) - P(\mathcal{S}_t{}^{i})] = (1 - \pi_A) P(\mathcal{S}_t{}^{i}) + \pi_A P\left(\mathcal{S}_t | \lambda_t{}^{i+1}\right) = P_{cond}{}^{i+1}$$

which in this case is expressed in terms of one observation but it can be based on multiple observations. The vector representation becomes

$$M * \boldsymbol{P}^{i+1}_{t+\Delta t} = \boldsymbol{P}_{cond}{}^{i+1}$$

where

$$M = \begin{bmatrix} 1+\theta & -\theta & 0 & & & \\ -\theta & 1+2\theta & -\theta & & & \\ 0 & -\theta & 1+2\theta & & & \\ & & & 1+2\theta & -\theta & 0 \\ & & & -\theta & 1+2\theta & -\theta \\ & & & 0 & -\theta & 1+\theta \end{bmatrix}$$

M is a square matrix of size = $\left(\frac{b-a}{\Delta s}+1\right)$.

Summary

The diffusion process algorithm for each time step containing $\boldsymbol{\Lambda_t} = \{\lambda_t{}^1, \lambda_t{}^2, \lambda_t{}^3, \ldots, \lambda_t{}^{k_t}\}$ observations as follows

1. If $k_t = 0, i.e., \boldsymbol{\Lambda_t} = \emptyset$, go to step 3b.

2. Apply the Bayesian updating scheme.

For $i \in \{1, 2, \ldots, k_t - 1\}$

$$P_{cond}{}^{i+1} = (1-\pi_A)P_{cond}{}^{i} + \pi_A P\left(\mathcal{S}_t | \lambda_t{}^{i+1}\right)$$

where

$$P\left(\mathcal{S}_t | \lambda_t{}^{i+1}\right) = \frac{P\left(\lambda_t{}^{i+1} | \mathcal{S}_t\right) .* P_{cond}{}^{i}}{P\left(\lambda_t{}^{i+1}\right)}$$

.* means element-wise multiplication. Go to step 3a.

3a. Solve the linear system

$$P\left(\mathcal{S}_{t+\Delta t}{}^{k}\right) = \boldsymbol{M}^{-1}\boldsymbol{P}_{cond}{}^{i+1}$$

3b. Solve the linear system

$$P\left(\mathcal{S}_{t+\Delta t}{}^{k}\right) = \boldsymbol{M}^{-1}\, P\left(\mathcal{S}_t{}^{0}\right)$$

4. Compute $\hat{s}$ for the

$$\hat{s}_t = arg \max_{s \in \mathcal{S}} P(\mathcal{S}_t)$$

The parameter domains are

$$\pi_A \in [0,1], \qquad \gamma \in [0, \infty)$$

This algorithm is also unconditionally stable for the diffusion-term, setting a $\gamma = 0$ deactivates the diffusion term and D_p controls the diffusivity.

It is possible to extend the both algorithm to **discrete** states or **fuzzy** input observations.

6.6.3 Normalizing a Function

It is worth mentioning how to normalize a function since this step is used during the Diffusion process. If the integration of a function, e.g. $P(\mathcal{S}_t)$, between bounds [a, b] is equal to I

$$\int_a^b P(\mathcal{S}_t)\, ds = I$$

it can be rescaled by I as follows

$$P(\mathcal{S}_t) \coloneqq \frac{P(\mathcal{S}_t)}{I}.$$

Then, its integration will be equal to one

$$\int_a^b P(\mathcal{S}_t)\, ds = \frac{I}{I} = 1$$

6.7 Agent Learning

Learning is an internal process carried out by each agent $\mathcal{A}$, therefore there is no sharing among agents in this process. Its purpose is to increase the predictive power of the collective information algorithm by removing any linear bias. Since this process is going to be carried out by each agent, we would like to have a simple and computationally economic procedure. Therefore, agents $\mathcal{A}$ will perform an online adjustment using gradient descent to minimize an error function. The linear adjustment is

$$\breve{\mathcal{S}} = \alpha_0 + \alpha_1\, \hat{s}$$

where α₀ and α₁ are the two parameters to be estimated. A linear mapping was chosen because its been shown that we humans are capable to do at best linear relations when it comes to decision-making. For the error function we have chosen the sum of square differences for simplicity but other functions such as cross entropy could be used

$$\epsilon = \frac{1}{2}\sum_{j=1}^{n}\left(\breve{\mathcal{S}}_j - \mathcal{S}_j\right)^2$$

where n refers to the size of the sample and $\mathcal{S}_j$ to the actual value experienced by the agent. After differentiating the error function and applying gradient descent, the online adjustment becomes

$$\alpha_0 \coloneqq \alpha_0 - \xi\frac{\partial\epsilon}{\partial\alpha_0} = \alpha_0 - \xi\left(\breve{\mathcal{S}}_j - \mathcal{S}_j\right) \qquad (6.12a)$$

$$\alpha_1 \coloneqq \alpha_1 - \xi\frac{\partial\epsilon}{\partial\alpha_1} = \alpha_1 - \xi\left(\breve{\mathcal{S}}_j - \mathcal{S}_j\right)\hat{s} \qquad (6.12b)$$

the index j refers to the current values and ξ is the learning rate. Initially, agents start with α₀ = 0 and α₁ = 1. It is worth notice that since there is only one feature for agents to learn from (the system state), there is no need to incorporate more sophisticated machinery like machine learning or any of its tools such as variance, bias-correction, and activation function. However, the 'intelligence' of the agent may be expanded to incorporate such techniques.

Example 6.4

Suppose a bank customer that upon arriving at the bank, glances at the queue and gets an estimate of the people waiting to be attended: $\hat{s}$ (true value is $\mathcal{S}$). Using that value, she gets a linear estimate of the time she will have to wait in line $\breve{\mathcal{S}} = \alpha_0 + \alpha_1\,\hat{s}$. After leaving the bank she computes the time difference $\breve{\mathcal{S}} - \mathcal{S}$ and adjust her beliefs α₀ and α₁. She would like to have an online procedure with only previous experience memory to improve her estimation.

Say that true waiting time is

$$\breve{\mathcal{S}} = 10 + 3\,\mathcal{S}, \qquad \mathcal{S}\sim U[1{,}20] \qquad (6.13)$$

and the noise in perception is described by

$$\hat{s} = \mathcal{S} + randN, \qquad randN\sim N(0{,}1)$$

and a learning rate of ξ = 1E-2. The evolution of the parameter values is shown below.

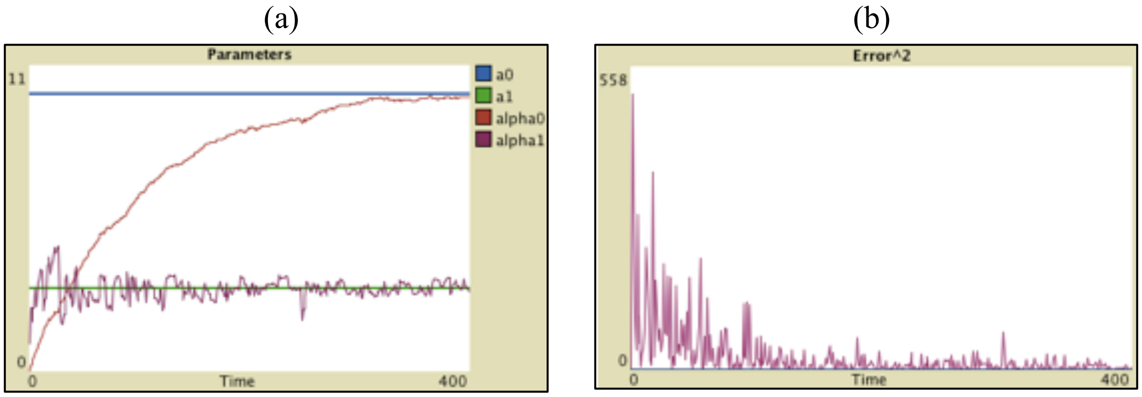


Figure 6.16 (a) Plot the parameter values (b) Error^2 plot

Table 6.1 Results of Example 6.4

Parameters	Initial	Visits to Bank			True values
		200	400	800	Eq. (6.13)
α_0	0	8.54	9.82	10.1	10
α_1	1	2.87	2.69	2.7	3
Error^2	100	8.15	4.06	2.25	

Estimates α_0 and α_1 will keep moving around its true values due to the noise introduced in $\hat{s}$, but this is a bearable consequence of having an online short memory procedure. Parameters α_0 and α_1 will converge to a local minimum as long as the learning rate is low enough.

In the next chapter we apply this algorithm to estimate travel times in road networks.

Chapter 7 Traveling Time Estimation in Road Networks

Introduction

This chapter presents the application of the Bayesian state estimation algorithm to estimate speed and travel time of road segments and improve drivers' trip planning. First the roads networks to be studied are presented and their formulation as an optimization model and ABM. Then agent rules of behavior are derived from the optimization model. The purpose of the simulation is to test how real-time information about the current state of the roads, collected from a sample of the vehicles traveling on the network, may help future agents into adjusting their traveling plan so as to minimize their traveling times and how that impacts the overall performance of the network.

Formulation and solution of the models were made in R®, Netlogo®, or Lingo®.

7.0 Assumptions and limitations

The assumptions behind the creation and simulation of the road network models are the following

- Road network models have two sets of state variables, one describes the actual movement of cars and travel times, and the other describes the estimations from the Bayesian algorithm. Unless stated otherwise, drivers will always make decisions based on the estimates.
- Cars moving along the same road segment move at the same speed. However, the 'perceived' speed that is reported back to the Bayesian algorithm most likely is different from car to car because of the noise. Therefore, the Bayesian algorithm has no access to the true state.
- Travel times along each road segment are computing using a time-delay function that inputs current cars on the road.
- Drivers decide the path to follow when they arrive at the origin node and do not make any change of path afterwards.
- Drivers decide randomly when there is a tie in two or more paths with the same time cost.

7.1 Road Networks

Road networks are represented by a directed graph G=(V,E) where V is the set of vertices, E the set of edges, source vertices $s \in V$, sink vertices $z \in V$, $f_i^{s\,z}$ is the flow on the ith path from s to z, and w_k is the flow on edge $k \in E$. Auxiliary variables are $\delta_{k\,i}^{s\,z} = 1$ if k is in path i between s and z, otherwise 0; and $d_{i\,j}^{s} = 1$ if path i and j come from source s, otherwise 0.

7.1.1 Road network I (RN-I)

This network is used primarily to test and compare results from the optimization model and the agent-based model. It has one source and one destination vertex; all of the edges have the same distance and capacity. We only labeled the source and destination vertices.

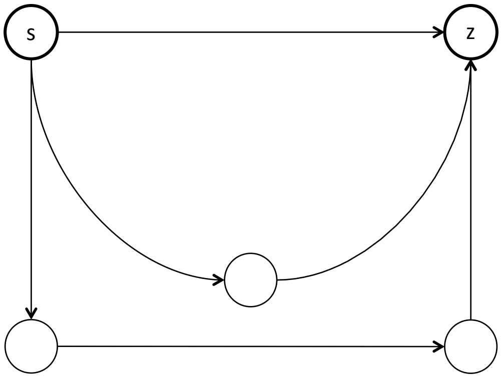


Figure 7.1 Road network I

7.1.2 Road network II (RN-II)

This network is used to exemplify the Braess' Paradox that was first studied by Braess [57], in which the addition of a link may produce a decrease on the system's performance. This network has been extensively studied from different perspectives such as robustness [58], information implications [59], nonlinear dynamics and control theory [60], game theory [61], and agent-based models [62] – [64]. However, we wanted to determine if it is possible to derive agent rules that avoid falling in a suboptimal state.

It is important to mention that the time-delay functions of this network are defined differently than our definitions. For this network (and only for this network) those functions are $T_1 = 2w_1$, $T_2 = 25$, T_3=0, $T_4 = 25$, and $T_5 = 2w_5$. All links have a distance of 15 units.

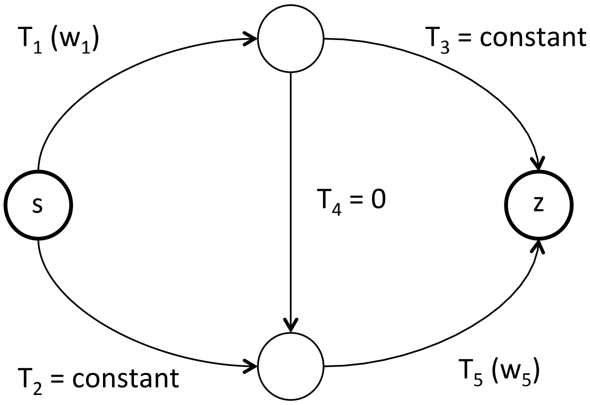


Figure 7.2 Road network II

7.1.3 Road network III (RN-III)

This is a larger network intended to represent a small real road network. Here again, each edge has the same distance and capacity.

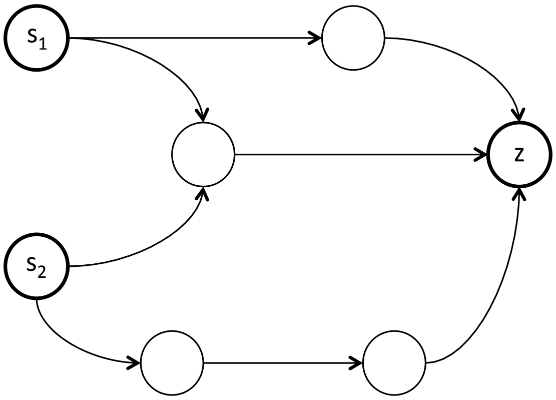


Figure 7.3 Road network II (RN-II)

7.2 Road Network Optimization Model

The mathematical formulation for user-equilibrium presented in [54], [65] it has been shown that it yields a solution consistent with Nash equilibrium; it minimizes the delay accumulation per edge. We divided by F, i.e., the total flow, to obtain the average time per vehicle. It has been shown this objective function produces a solution for which alternative paths have equal delay times while unused paths have higher delay times. This formulation is classified as a mixed-integer nonlinear program (MINLP) when the car flow is restricted to integer values, but it may be relaxed to continuous values.

$$Min\, Z_1 = \frac{1}{F}\sum_{k\in E}\int_0^{w_k} T_k(\omega)\, d\omega \qquad (7.3)$$

s.t

$$\sum_i f_i^{s\,z} = F^{s\,z}, \forall s, z$$

$$w_k = \sum_s \sum_z \sum_i f_i^{s\,z} \delta_{k\,i}^{s\,z}$$

$$F = \sum_{s} \sum_{Z} F^{s\,z}$$

$$f_i^{s\,z} \geq 0, \forall s, z, i$$

$$f_i^{s\,z} \in \mathbb{Z} \quad (optional)$$

Another objective function used in the literature is to minimize the average car-delay/travel time per edge [54], [65]. This is called the system optimal solution because minimizes the total time for all travellers. The solution it yields may or may not be a Nash. The formulation is also a MINLP.

$$Min\, Z_2 = \frac{1}{F} \sum_{k} w_k T_k \tag{7.4}$$

with the same set of restrictions as eq. (7.2).

The third objective function we will examine is to minimize the average car-delay/travel time per vehicle. This program is also MINLP.

$$Min\, Z_3 = \frac{1}{F} \sum_{k} T_k \tag{7.5}$$

with the same set of restrictions as eq. (7.3). Section 7.4 discusses the derivation of agent rules from these objective functions.

7.3 Agent-Based Model

The agent-based model created representing the same system has three types of agents: nodes, cars and directed links. Nodes and links are used to create the road network and cars are the vehicles traveling from one node to another. Their attributes are as follows

1) Nodes
   i) Node number
   ii) (x, y) location in plane

2) Links
   i) Length of the link
   ii) Initial and final node
   iii) Number of cars (current)

iv) Speed (current)
v) Traveling time (current)
vi) Value of Gradient (current)

3) Cars
i) A?: binary variable where A=1 iff $car \in \mathcal{A}$, i.e., is a car that shares and uses collective information, otherwise A=0.
ii) Path: list of nodes to follow to arrive destination
iii) Goal: next node to move to
iv) Road: current link the car is travelling

In addition, procedures were created to find shortest paths or minimal gradient paths (according to the objective function we want agents to minimize) and the computation of the probability density function describing each edge (when using the Bayesian state estimation algorithm). Several global variables and arrays were created to keep track of the objective function values, mean values, and other model features.

In order to have results comparable to those from the optimization model, we computed the flow or volume of cars [vehicle/t] by counting the number of cars on a given link per unit of time and compute their speed using eq. (7.2).

7.3.1 Pseudo code

The general pseudo code of the models is written next; however, changes were made according to the different scenarios that were tested.

```
Create GlobalVariables
Create AgentType: {Cars, Nodes, Links}
Create IntervalVariables

Execute InitializeProcedure:
      Execute CreateNetworkProcedure()
      Assign InitialValues to GlobalVariables and InternalVariables
      Reset time-counter;
end

Execute SimulationProcedure:
      While (time-counter ≤ T) do
      Execute Speed_and_Time():
            Ask each link [
                  Count Cars on Myself
                  Compute TravelTime
                  Compute Speed
                  ]
      end
      Execute Bayesian_Algorithm():
            Foreach car [
                  Read location and speed
                  ]
            Foreach link [
                  Compute BayesianAlgorithm():
                        Estimate speed, travel time
                  ]
      end
      Create-car: x [
            Set initialNode InitialNode()
            Set finalNode FinalNode()
            Set path MinimalPath()
            ]
end
```

7.4 Derivation of Agent Rules

Agents arrive at source nodes and must choose a path to arrive to their destination. An agent in $\mathcal{A}$ will determine its path according to the gradient of the objective function by selecting the path with minimum gradient cost. The gradient is computed according to current information, which is supplied to the agents.

$$\min_i \left\{ \frac{\partial Z}{\partial f_i^{s\,z}} \right\}$$

Z is one of the objective functions described previously in eq. (7.3-7.5) and i is the index to represent the set of paths available to that agent to reach its destination.

For Z_1:

$$\frac{\partial Z_1}{\partial f_i^{s\,z}} = \sum_k \frac{\partial Z_1}{\partial w_k} * \frac{\partial w_k}{\partial f_i^{s\,z}} = \frac{1}{F} \sum_k T_k * \frac{\partial w_k}{\partial f_i^{s\,z}}$$

where $\frac{\partial w_k}{\partial f_i^{s\,z}} = 1$ if w_k is an edge along path i, otherwise 0. This rule implies selecting the path with minimal time.

For Z_2:

$$\frac{\partial Z_2}{\partial f_i^{s\,z}} = \sum_k \frac{\partial Z_2}{\partial w_k} * \frac{\partial w_k}{\partial f_i^{s\,z}} = \sum_k \left\{ t_f \left[1 + \beta_0(\beta_1 + 1) \left(\frac{w}{cap_k} \right)^{\beta_1} \right] \right\} * \frac{\partial w_k}{\partial f_i^{s\,z}}.$$

The first term of the summation resembles the time function except for one factor. We may say this rule of selection implies selecting the minimal time path using a modified time function. $\frac{\partial w_k}{\partial f_i^{s\,z}}$ has the same meaning as described above.

For Z3:

$$\frac{\partial Z_3}{\partial f_i^{s\,z}} = \sum_k \frac{\partial Z_3}{\partial w_k} * \frac{\partial w_k}{\partial f_i^{s\,z}} = \frac{1}{F} \sum_k t_f\, \beta_0\, \beta_1 \frac{{w_k}^{\beta_1 - 1}}{{cap_k}^{\beta_1}} * \frac{\partial w_k}{\partial f_i^{s\,z}}.$$

If all road segments have the same capacity and same free-flow time, then this equation can be expressed as

$$\frac{\partial Z_3}{\partial f_i^{s\,z}} = \frac{t_f\, \beta_0\, \beta_1}{F\, cap^{\beta_1}} \sum_k {w_k}^{\beta_1 - 1} \frac{\partial w_k}{\partial f_i^{s\,z}}$$

Agents in $\mathcal{B}$, agents who do not have access to collective information, choose their path according to the following rule: the probability of selecting the i^{th} path is inversely proportional to the exponential of the normalized distance:

$$P(f_i) = \frac{exp(-ð_i)}{\sum_i exp(-ð_i)}, \qquad \forall i \tag{7.6}$$

where $ð_i$ is the normalized distance of path *i*:

$$ð_i = \frac{d_i}{\min\limits_i \{d_i\}}, \qquad \forall i$$

Longer paths are less likely to be selected by agents in $\mathcal{B}$. For example, a road network with three paths with distances [1, 2, 3], the probability of selecting of each path is ~[67 24 9]%.

7.4.1 Example I. Solving RN-I

The optimal flow for RN-I was found using the optimization and agent-based models. For the agent-based model we assumed all the agents to be from the set $\mathcal{A}$ to remove any noise from random choice. Agents arrive at *arrival rate* to the source vertex and decide which path to follow to reach their destination vertex *z*. Input values are shown on table 7.1. All edges have the same distance and capacity.

Table 7.1 Input values for Example I

Parameter	**Value**
β_0	1
β_1	3
cap_w	20 [vehicle]
d_i	15 [distance unit]
t_f	7.5 [t]
v_f	2 [distance unit/t]
Constant Arrival rate	2 [vehicle/t]

First we solved the ABM and when it reached steady state flow, the number of vehicles on the network was counted, $F^{s\,z}$, this value was entered into the optimization model as a input parameter then we solved that model and compared both solutions. Solutions are shown in table 7.2. It is worth noting that the ABM not always converge to steady state, when the arriving rate of cars is higher than the output rate of the network, the system accumulates cars increasingly over time in the same ways as queue systems do. Therefore, to prevent oversaturation, values of capacity and arrival rate were carefully tested.

Table 7.2 Value of Objective functions for Example I

	Min Z_1		**Min Z_2**		**Min Z_3**	
Obj. func.	Opt	ABM	Opt	ABM	Opt	ABM
Z_1	11.3	8.7	10.3	7.7	13.8	7.5
Z_2	16.3	12.5	11.6	8.5	14.7	7.6
Z_3	1.8	1.7	2.4	2.3	1.8	1.7
Error*	2.7		3.1		0.1	
F [vehicle]	30		20		26	

*Error = absolute difference of minimized values

Table 7.3 Optimal flow for Example 7.1 $[f_I, f_{II}, f_{III}]$ in vehicles

	Z_1	**Z_2**	**Z_3**
Optimization	[21 9 0]	[14 6 0]	[11 8 7]
ABM	[20 10 0]	[12 8 0]	[4 10 12]
MSE	2/3	2.6	26

Table 7.4 Optimal path times for Example I $[T(f_I), T(f_{II}), T(f_{III})]$ in time units

	Z_1	Z_2	Z_3
Optimization	[16.1 16.3 22.5]	[10 15.4 22.5]	[8.7 15.9 23.4]
ABM	[15 15.2 22.5]	[9.1 15.2 22.5]	[7.5 15.2 22.6]
MSE	0.8	0.3	0.8

Solutions and objective function values are relatively close for objective functions Z_1 and Z_2. Mean square errors for vehicle allocation and times are small except for Z_3, this is because of the dynamic feature of the ABM that the optimization model does not have. The optimization model 'assumes' all vehicles in a path travel all together and since the delay function is not linear, i.e., T(a) + T(b) ≠ T(a+b), it may fail to capture a better allocation. For the same reason, the ABM performs better in all three objective functions. It is interesting to note how agents minimizing Z_3 were able to find a different flow allocation with lower cost. It is also worth noting Z_2 maximizes the throughput of vehicles since it keeps the minimal number of vehicles traveling throughout the network.

7.4.2 Example II Solving RN-II

Now we proceed to determine optimal flows for network RN-II. When formulating and solving this network using the optimization model we were able to produce the paradox or avoid it depending on which objective function we minimized; however, when using the ABM, it became somewhat problematic, explained next, because of the dynamic nature of the ABM. If the arriving rate of agents is low, then no paradox arises because the speed on the links is always high. If the arriving rate is high, then speed on the links becomes low enough that the paradox does not arise.

In order to produce the paradox in the ABM we had to fix the number of cars in the network to a number F such that the network would have ~F cars at any time; this will make the flow per link and per path to converge to an average value. Those values were compared to the solution from the optimization model. To have similar flows per link, the optimization model has to be solved for a flow rate ~0.5F.

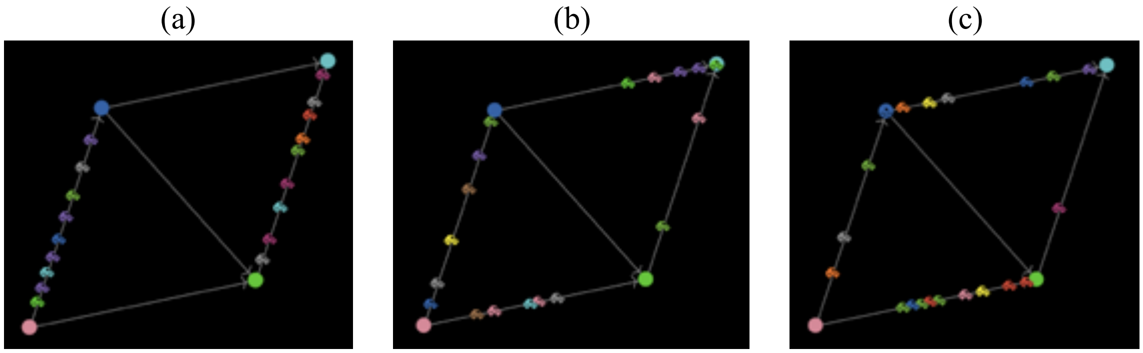


Figure 7.4 ABM following rules: (a) Min Grad(Z_1), (b) Min Grad(Z_2), (c) Min Grad(Z_3)

Table 7.5 Value of Objective functions for Example II

	Min Z_1		**Min Z_2**		**Min Z_3**	
Obj. func.	Opt	ABM	Opt	ABM	Opt	ABM
Z_1	20	10.2	26.5	15.1	30	23.5
Z_2	40	20.5	34.3	16.9	35	26.6
Z_3	9	4.5	7.5	3.5	7	3.31
F [vehicle]	10	18.9	10	18.7	10	19.1

Table 7.6 Optimal flow for Example II $[f_I, f_{II}, f_{III}]$ in % of F

	Z_1	**Z_2**	**Z_3**
Optimization	[0 100 0]	[37.5 25 37.5]	[50 0 50]
ABM	[12.5 68.5 19]	[26.8 37.3 35.9]	[51.1 0 48.9]
MSE	503	89	0.8

Table 7.7 Optimal path times for Example II $[T(f_I), T(f_{II}), T(f_{III})]$ in time units

	Z_1	**Z_2**	**Z_3**
Optimization	[45 40 45]	[37.5 25 37.5]	[35 20 35]
ABM	[39 28.9 39.9]	[33 20.2 37.1]	[26.6 3.3 26.6]
MSE	61.7	14.5	140

The paradox arises from having two different performance measures/objective functions. If drivers minimize its travel time (columns Min Z_1 of Table 7.5) then the other objective function increase, the paradox appear as having a system optimal (Z_2) higher than its minimal achievable. However, when drivers minimize the average travel time of all cars in the network (columns Min Z_2 of Table 7.5) then paradox does not appear.

If we look at this behavior from an optimization framework, we notice it is perfectly possible that minimizing one variable increases others. From a social perspective, we may say that agents pursuing exclusively individual goals may harm the collectivity if they are given a wider possibility space to gain from. As seen when drivers minimized their individual travel time but increased the total average time of the network.

Optimal flows solutions were not significantly different for Z_2 and Z_3, the highest error occurred on Z_1. This discrepancy came from the tendency of cars to cluster inside the road segments with travel times proportional to its flow. Therefore, subsequent cars opt for faster alternatives. This phenomenon only appears in dynamic models. The equation-based Braess paradox formulation does not account for this because it implicitly assumes uniformity. It is interesting to note that minimization of Z_3 yielded no flow through the middle path on both models. Optimal path times were more consistent on both models, although the error of the third result was higher than the rest due to the dynamic nature of the ABM commented above.

7.5 Agent Decision-Making

As seen in Example 7.1, objective function Z_1 may be called the **agent optimal rule** of selection since it chooses the path with minimal travelling time, whereas Z_2 may be called the **system optimal rule** because it minimizes travelling time for all agents. We would like to give agents a utility function that takes into account both objectives and balance individual gain vs. social gain. In addition to do that, agents will use estimates of current times instead of true current times; therefore, some measure of uncertainty about those estimates can be computed and supplied along with the estimates. The agent decision-making model then becomes

$$\operatorname*{Max}_{i} \mathcal{U}\left(\frac{\partial}{\partial f_i} Z_1(\widehat{w}), \frac{\partial}{\partial f_i} Z_2(\widehat{w})\right) \tag{7.7}$$

s.t

$$\bar{H}(f_i) \leq Target$$

$$\overline{H}(f_i) = \frac{1}{\ell_i}\sum_k H(w_k)\delta_{k\,i}$$

Since agents have a fixed initial and final destination node, we have removed indices s and z from $f_i^{s\,z}$ and use f_i instead. H(.) is the differential entropy, $\delta_{k\,i} = 1$ if edge k is part of path *i*, otherwise 0. ℓ_i is the length of path *i*, i.e., the number of edges it contains.

Target is the upper bound desired of uncertainty on the path measured either in [bit/edge] or [nat/edge] depending on the base of the logarithm and $\overline{H}(f_i)$ is the average entropy of path i. Since the speed of vehicles is restricted below or above certain values, say (a,b), the maximum entropy achievable on any edge is *log (b-a)*. However, introducing a restriction may produce an infeasible problem, in order to avoid that, we included the restriction in the utility function as a penalizing term in the same way as the mean – variance rule [66]: $\theta\ *\overline{H}(f_i)$, however if we wanted to penalize only values above the target, then we would use $\theta\ *max\{\overline{H}(f_i) - Target, 0\}$.

The utility function of eq. (7.7) including the uncertainty penalization term is

$$\underset{i}{\text{Max}}\,\mathcal{U}_i = -\mu\frac{\partial}{\partial f_i}Z_1(\widehat{w})\ - (1-\mu)\frac{\partial}{\partial f_i}Z_2(\widehat{w}) - \theta\ \overline{H}(f_i)$$

which can be expressed in terms of a loss function as

$$\underset{i}{\text{Min}}\,\mathcal{L}_i = \mu\frac{\partial}{\partial f_i}Z_1(\widehat{w})\ + (1-\mu)\frac{\partial}{\partial f_i}Z_2(\widehat{w}) + \theta\ \overline{H}(f_i) \tag{7.8}$$

where θ is a penalizing factor in [cost/(bit*vehicle)] and μ ∈ $[0,1]$ determines the level of 'selfishness'. μ=1 is an agent pursuing only individual gain, μ=0 is a socially driven agent.

7.6 Bayesian State Estimation Implementation

We proceeded to solve RN-I and RN-III with agents acting based on collective estimations instead of true values. Therefore, we implemented the Bayesian State Estimation algorithm, which inputs current speeds of agents (with noise), computes a probability density function per road, and obtains the most likely speed and entropy per road. From these estimates, it is straightforward to obtain the most likely time and vehicles per road. This information is supplied to agents $\mathcal{A}$ for them to make a decision that is best for their interest according to eq. (7.8). Values of objective function and performance measures were computed using true values and not estimates; estimates only affect the agents' decision making.

7.6.1 Input Values

The following are the default parameter values for all the experiments, unless mention otherwise. Agents do not execute the learning algorithm described in Section 6.7 of Chapter 6.

Table 7.8 Input values for network RN-I

Parameter	Value
β_0	1
β_1	3
cap_w	5 [vehicle]
d_i	10 [distance unit]
t_f	5 [t]
v_f	2 [distance unit/t]
Arrival rate	1 [vehicle/t]

Table 7.9 Input values the Bayesian State Estimation

Parameter	Value
Δt	1 [t]
γ	½
D_p	0.1
π_A	½
v_{max}	2 [km/t]
v_{min}	0 [km/t]

Initial speed probability density function for all links is

$$P(v) = \begin{cases} 1/2, & v \in [0,2] \\ 0, & elsewhere \end{cases}$$

Noise in speed measurements by agents is described by

$$\hat{v} = v + \frac{1}{20} v\, n, \qquad v \in [0,2], \qquad n \sim N_{random}(0, \sigma_N^2 = 1)$$

The Bayesian algorithm assumes that conditional probabilities are described by a truncated normal distribution

$$P(v|\hat{v}) = \frac{exp\left(-\frac{(v-\hat{v})^2}{2\sigma_N^2}\right)}{\sigma_n\sqrt{2\pi}\,C}, \qquad v \in [0,2]$$

and C = F(2) – F(0), F(.) being the cumulative distribution function.

7.6.2 Comparison Bayesian vs. True values for RN-I

In this simulation we ran the ABM using true values (TV), Bayesian with diffusion (BD), and Bayesian with diffusion-like process (BDL) to compare how close to true values the Bayesian estimates are. When agents set decision parameters [μ, θ] = [1, 0] they act minimizing Z_1, and when they set [μ, θ] = [0, 0] they minimize Z_2, which are the two setting we tested.

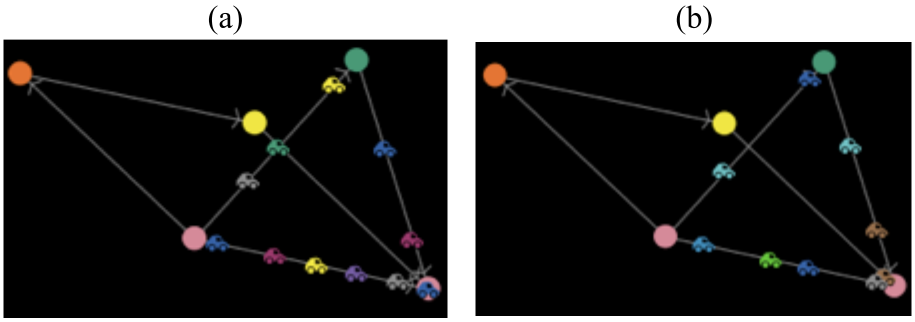


Figure 7.5 Road Network I using Bayesian State Estimation

Table 7.10 Objective functions for RN-I with Bayesian State Estimation

	Min Z_1	**Min Z_2**
TV	7.4	5.5
BD	7.4	5.3
BDL	8.4	5.7
F [vehicle]	12	10

Table 7.11 Average flows after 500 time steps for RN-I $[f_I, f_{II}, f_{III}]$ in vehicles

	Min Z₁	**Min Z₂**
TV	[5.4 4.5 0]	[3.4 3.6 1]
BD	[5.7 5.7 0.5]	[3.4 4.2 0.8]
BDL	[5.9 5.9 1]	[3.7 3.7 2]

Table 7.12 Error between true values and estimates for RN-I

	MSE(speed)	MSE(travelTime)	MSE(vehicles)
BD	2.6×10^{-3}	2.1×10^{-2}	1
BDL	5.8×10^{-4}	7×10^{-3}	7.8×10^{-2}

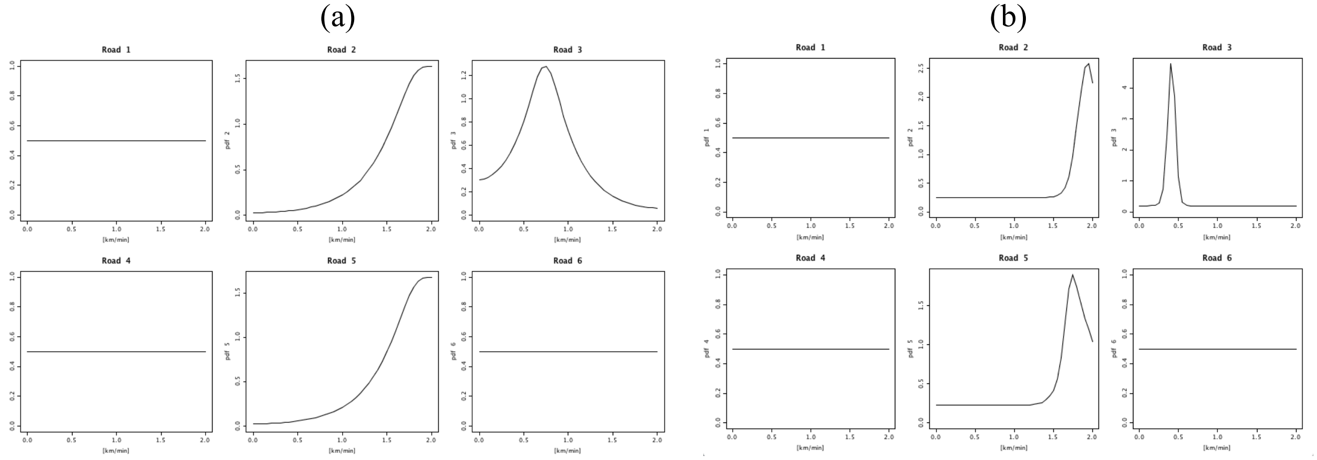


Figure 7.6 Probability density functions of roads (a) Bayesian diffusion (b) Bayesian diffusion-like

After looking at the average path flows and the error of the estimates we may conclude the Bayesian state estimation provided accurate results about the state of the road. In particular, the diffusion-like process seems to provide a more accurate estimation.

7.6.3 Comparison between different types of agents

In this simulation we compared performance of agents $\mathcal{A}$, agents who share and have access to collective information; against agents $\mathcal{B}$, agents who do not share nor have access to information and decide based on eq. (7.6). Agents are created randomly according to a given proportion.

We computed the average path time per agent for agent sets $\mathcal{A}$ and $\mathcal{B}$. We set $\mu = 1$ and $\theta = 0$ for $\mathcal{A}$ and used the diffusion-like procedure. The results are shown below.

Table 7.13 Average path time per agent for RN-I

Proportion of $\mathcal{A}$	$\mathcal{A}$ [t/agent]	$\mathcal{B}$ [t/agent]
0	0	137.2
0.1	5.9	165.6
0.3	7.5	31.9
0.5	10.4	15.2
0.7	13.5	25.7
0.9	11.8	6.8
1	11.3	0

Agents $\mathcal{A}$ have a lower average path time per agent compare to agents $\mathcal{B}$, except when its proportion exceeds nine to one. This last result may have been produced because agents $\mathcal{B}$ were few enough to benefit from having a majority of efficient drivers.

We may also look at these results as an overall performance measure of the road network. When agents have access to information, its travel time significantly decreases; whereas when they decide based on the shortest path, their travel time increases nonlinearly because they all concentrate on the same road (the shortest one).

7.6.4 Implementing Learning

For this simulation we included the learning feature described in eq. (6.12a –b) applied to the travel time to determine how inaccurate was the estimation provided by the Bayesian procedure.

Table 7.14 Learning parameters after 150 trips

	BD	BDL
α_0	0.0410	0.0407
α_1	0.9785	1.0089

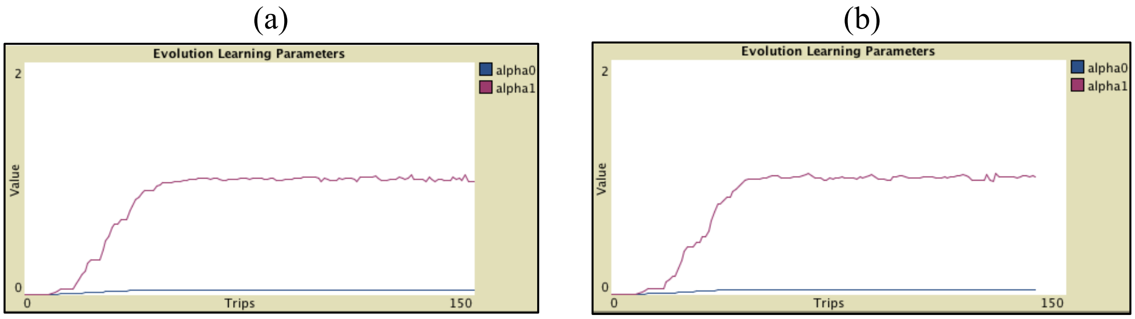


Figure 7.7 Evolution of parameter values for the learning online procedure from estimates (a) BD, (b) BDL

Parameter values have values $\alpha_0 \approx 0$ and $\alpha_1 \approx 1$ in both procedures, which indicates that there is no systematic linear bias introduced by the Bayesian State Estimation; therefore, there is no need to implement the learning online adjustment on time estimations since it would barely have any effect into the agents' path selection.

7.6.5 Implementing RN-III

In this experiment we used network RN-III with two source nodes and one sink. The arrival rate on each source is described by a Poisson-distributed random variable with arrivalRate = 2/5.

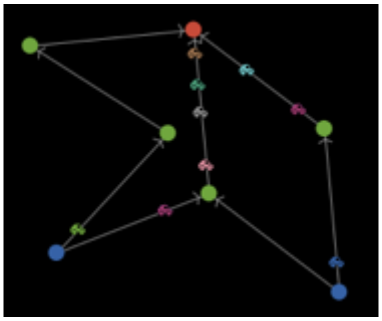


Figure 7.8 Simulation of RN-III (source nodes in blue; sink in red)

Table 7.15 Average minutes on network per driver for RN-III

Parameters [μ, θ]	[1, 0]	[0, 0]	[1/2 0]
BD	10.81	9.91	6.40
BDL	10.69	7.89	7.75

Table 7.16 Average vehicles on network for RN-III

Parameters [μ, θ]	[1, 0]	[0, 0]	[1/2 0]
BD	8.84	10.69	8.17
BDL	9.76	8.18	10.45

Table 7.17 Average flows per path in $[f_I, f_{II}, f_{III}, f_{IV}]$ in % (rounded) for RN-III

Parameters [μ, θ]	[1, 0]	[0, 0]	[1/2 0]
BD	[29, 18, 42, 11]	[35, 21, 22, 22]	[37 17 34 12]
BDL	[31, 21, 38, 10]	[33, 24, 24, 19]	[24, 16, 35, 25]

Table 7.18 Error between true values and estimates for RN-III

	MSE(speed)	MSE(travelTime)	MSE(vehicles)
BD	5.9×10^{-3}	4.6×10^{-2}	1.63
BDL	5.72×10^{-4}	3.8×10^{-3}	0.55

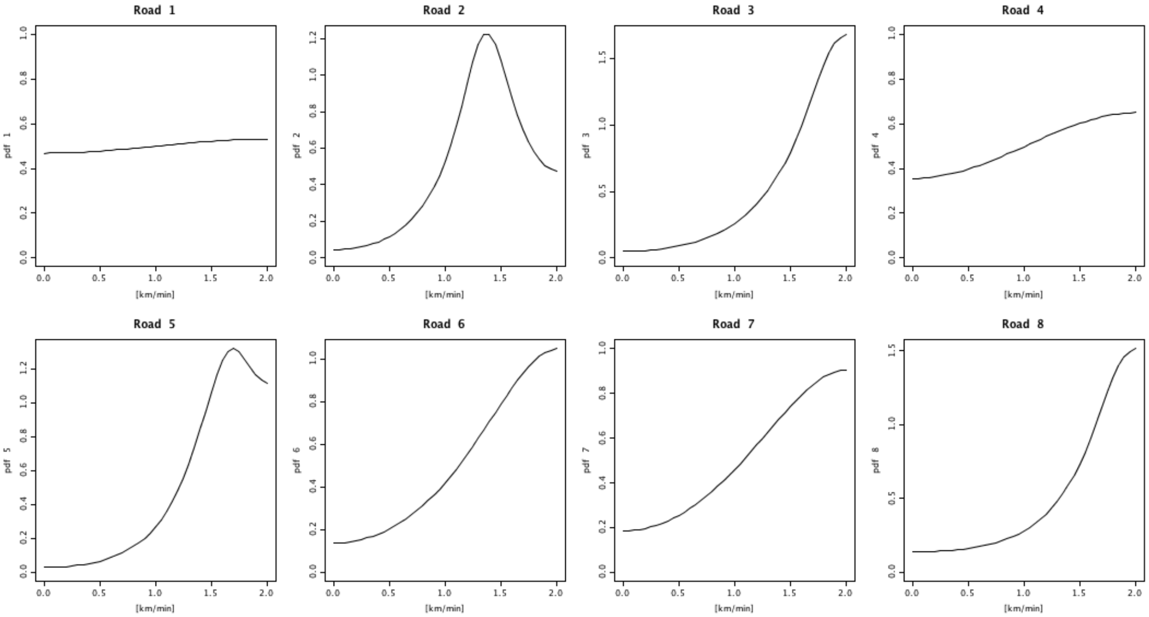


Figure 7.9 Probability density functions of roads using Bayesian with diffusion

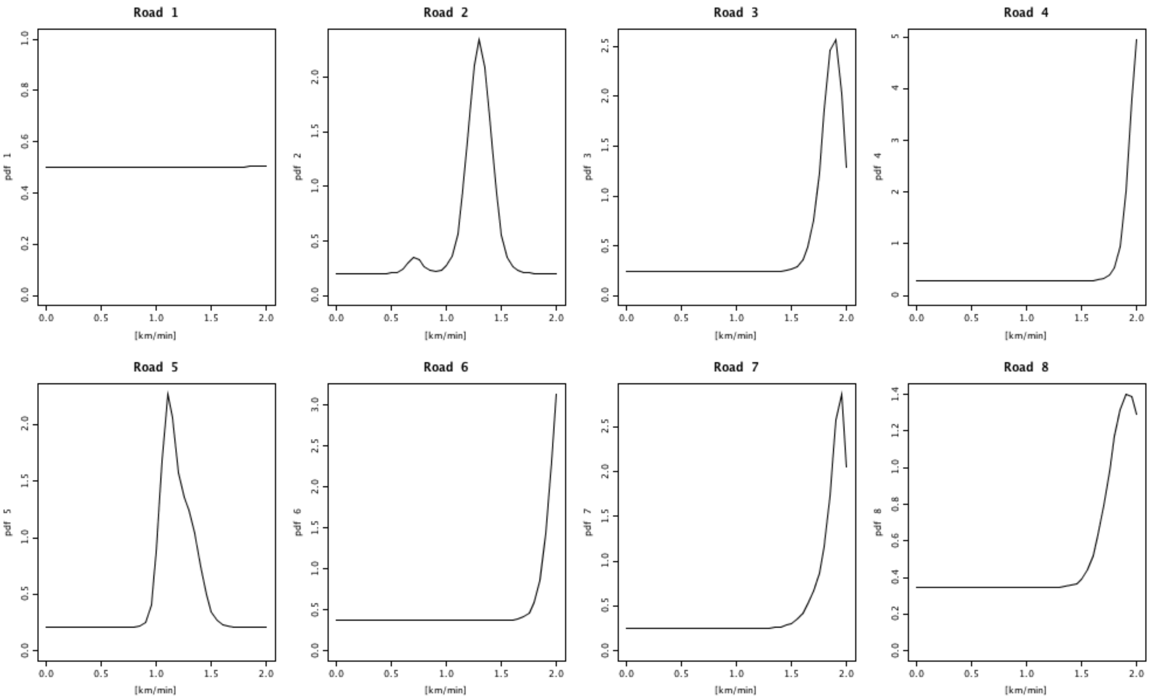


Figure 7.10 Probability density functions of roads using Bayesian with diffusion-like

In this network again we see that when agents behave ‘socially responsible’ the average time on network per agent decreases compared to that of the user-equilibrium. When agents assign equal weight to the user-equilibrium and system optimum (μ= ½) a better performance is achieved in terms of average minutes on network.

There was no significant difference in terms of path flows between the BD and the BDL procedures, as both assigned approximately the same proportion of vehicles to the paths. Error measures where low again with respect to true speeds, times, and cars on each road. The Bayesian diffusion-like procedure produced more accurate results.

7.7 Part II Conclusions

This chapter showed it is possible to derive agent rules of behavior from a global performance measures (or objective functions) and depending on that measure, agents may behave greedy or ‘socially responsible’, which establishes a connection between classical optimization theory and agent-based methodologies. We showed that it is possible to combine many performance measures into the agent decision-making so as to represent different agent goals and risk aversion.

It was shown that it is possible to obtain somewhat similar results when solving the route-planning problem using an optimization model as well as using an ABM. We found that the optimization model does not capture dynamic interactions that are present in the ABM; however, it is possible to use a more sophisticated optimization model to incorporate some dynamic, e.g., multi-stage models.

The derivation of agent rules from an objective function allows for the inverse process, i.e., having an individual decision-making model, obtain the aggregate function that describes the collective behavior. We did not explore this feature, but we would have to double integrate the individual objective function with respect to the path and with respect to the link flow to obtain the overall objective function.

The Bayesian state estimation algorithm proved to be a reliable, unbiased, and computationally not expensive algorithm. In particular, the BDL showed lower error in estimations. We believe the algorithm has the potential to be applied on other situations where information is collected from different sources or individuals.

Chapter 8. Summary and Conclusions

8.1 Summary

This thesis explored the connections between agent-based models and equation-based models in some domains, namely, the reaction-diffusion equation and the assignment of traffic flows to road networks using optimization. Agent rules were derived such that their collective behavior produced similar results than those from the equation-based models. The main tasks involved in the realization of this thesis were as follows

- Derivation of displacement and reaction probabilities to model diffusion using cellular automata.
- Application of those probabilities in the problem of estimating lifetimes of nuclear waste containers.
- Comparison of results from cellular automata models, those from literature, and those from numerical methods.
- Sensitivity analysis on the density of gapfill to determine the degree to which it affects the lifetime of the used fuel containers.
- Derivation of a Bayesian state estimation algorithm that includes diffusion to estimate the current state of a system of interest.
- Application of such algorithm into the problem of estimating speed of road segments in a road network.
- Derivation of driver decision rules from system's performance measures/objective functions, which creates a bridge between optimization and ABMs.
- Modification of the driver's decision-making to incorporate different objectives (e.g. individual vs. social) and the inclusion of a risk aversion term.
- Derivation of an online learning procedure for agents to remove any possible linear bias in their estimations received about the system payoff.

8.2 Contributions

The main contributions of the thesis are

Developing a cellular automata/ABM probability-based method to describe diffusion and reaction. Previous work treated diffusion and reaction as separate processes with probabilities for diffusion and Monte Carlo simulations for reaction. The work developed in this thesis derives probabilities for both processes using conditional expectation.

Developing a Bayesian state estimation algorithm for inferring current state of a system of interest. Conditional probabilities are widely used in many fields of science, engineering, etc. However, the method developed in this thesis is new, to the best of our knowledge, in combining conditional probabilities with diffusion to describe a changing process.

Derivation of agent rules to match results from the network flow optimization model. Although there have been many implementations of road network models using agents, they have not explicitly derived their behavior from system's performance measures. This thesis implemented a procedure to derive agent rules from the optimization model using the chain rule.

8.3 Conclusions

The general objective of the thesis about finding local agent behavior whose macrostate is the same or similar to the one described by its correspondent equation model (either PDE or optimization) was achieved.

In this work methods to derive probabilities from analytical or numerical solutions to ODEs/PDEs were found. Those methods can potentially be extended to other typical models such as the Lotka-Volterra equations and it may offer a glimpse of how to proceed with the inverse problem, namely, from a consistent agent behavior find the ODE/PDE that describes it. In the case of the optimization model, the limitations of the equation model restricted the degree of agreement in the results of both models (optimization vs. ABM). However, it was noted how the user-equilibrium objective function did in fact correspond to a decision rule where agents select the minimal-time path, therefore the ABM validated that assumption.

Finding a way to relate optimization and ABMs could be useful in other fields of study with an ecological or environmental outlook such as waste management or pollution reduction. If social/economical are indeed valid global representations, then there must be a way to translate those models into single individual representations; this thesis is a step in that direction as shown in the microeconomics example of Chapter 5.

ABMs due to its nature of representing individuals as objects are more robust to sudden changes in input values of the model. For example, if the agent is a molecule, it either exists or not, but it can never be negative as it sometimes happens with numerical differentiation. Or in the case of CA, the probability distribution condition in itself corresponds to the stability condition of numerical differentiation. However, attention should be paid to how the ABM evolves over time because it may be the case that agents are being accumulated, resulting in an exponential increase of memory consumption. This situation occurred few times during the simulation of the road networks when the system oversaturated its capacity, but problems like this can be resolved by adding a monitoring variable, e.g., car counting variable or a travel time threshold. Overall, agents allow for a more controlled behavior because its attributes or themselves can be bounded by the analyst.

Another feature that could be exploited from having a way to relate equation models to ABMs is that the latter are easier to sell to public or policy makers because of its more explicit mapping to the real-world problem. In this sense, ABMs could be presented as an approximation to the actual model.

Another potential benefit from using equation and agent representations is that the former may be used as the true or more accurate solution for deterministic values and latter be used as a tool to explore the parameter space, behavior under uncertainty, or the sensitivity of the system.

ABMs may be easier to formulate due to its discretized nature that maps well onto real-world problems; however, the process of refining the model to increase its realism may increase the number of variables or procedures, which in turn requires more advanced coding skills and makes the model resource consuming. Therefore, it is important to determine the degree of aggregation in the model so as to capture the dynamics of the process without introducing unnecessary information.

The treatment of stochastic variables in equations may not always be an easy task, particularly for ordinary or partial differential equations. However, for agent-based models it is easier to incorporate and study stochastic components. This feature proves to be a huge advantage because it allows the introduction of heterogeneity into the system.

A proper understanding of any phenomena must come from knowledge of how the system behaves at different aggregation levels. Equations are concerned with modeling emergent properties while agent models describe the interactions that produce such properties.

8.4 Future work

There are parts of this thesis susceptible to be expanded further, the most salient of which are the following

- The use of a more accurate numerical scheme to derive probabilities of displacement.
- Modeling of a 3D model of nuclear waste containers and the calculation of sulphide flux intervals.
- Explicit modeling of sulphate-reducing bacteria using the monod equation or Michaelis-Menten kinetics.
- Implementation of the Bayesian state estimation algorithm for discrete states or fuzzy inputs from agents.
- The use of reinforcement learning in driver’s behavior.
- Solve a larger and more realistic road network using the methods of this thesis.

References

[1] U. Wilensky and W. Rand, *An Introduction to Agent-Based Modeling. Modeling Natural, Social and Engineered Complex Systems with Netlogo*. Cambridge: MIT Press, 2015.

[2] M. Mitchell *Complexity: A Guided Tour.* New York: Oxford University Press, 2009.

[3] H. Sayama *Introduction to the Modeling and Analysis of Complex Systems.* Geneseo, NY: Open SUNY Textbooks, 2015. [E-book] Available: https://textbooks.opensuny.org/

[4] X.S. Yang and Y. Young "Cellular Automata, PDEs, and Pattern Formation", in *Handbook of Bioinspired Algorithms and Applications,* S. Olariu and A. Zomaya, Eds. Boca Raton, FL: Chapman & Hall/CRC, 2006, pp 18-273 – 18-284.

[5] J. Kroc, P. Sloot, and A. Hoekstra, *Simulating Complex Systems by Cellular Automata.* Berlin: Springer, 2010.

[6] D. Scalise and R. Schulman, "Emulating cellular automata in chemical reaction-diffusion networks", *Natural Computing*, vol. 15(2), p. 197(18), 2016.

[7] K. Odagiri and K. Takatsuka, "Threshold effect with stochastic fluctuation in bacteria-colony-like proliferation dynamics as analyzed through a comparative study of reaction-diffusion equations and cellular automata", *Physical Review E,* Issue 79, 026202, 2009.

[8] M. Azimil, Y. Jamali and M. Mofrad, "Accounting for Diffusion in Agent Based Models of Reaction-Diffusion Systems with Application to Cytoskeletal Diffusion", *PLoS ONE* 6(9): e25306, DOI: 10.1371/journal.pone.0025306, 2011.

[9] I. Kawamata *et. al*, "Discrete DNA Reaction-Diffusion Model for Implementing Simple Cellular Automaton", In Proc. International Conference on Unconventional Computation, *Unconventional Computation and Natural Computation 2016*, LNCS vol. 9726, pp. 168–181, DOI: 10.1007/978-3-319-41312-9 14, 2016.

[10] W. Tang and D. Benett, "Parallel agent-based modeling of spatial opinion diffusion accelerated using graphics processing units", *Ecological Modelling,* Vol. 222, Issue 19, October, Pages 3605-3615, 2011.

[11] A. Faber, M. Valente and P. Janssen, "Exploring domestic micro-cogeneration in the Netherlands: An agent-based demand model for technology diffusion", *Energy Policy* Vol. 38, Issue 6, June, Pages 2761-2775, 2010.

[12] O. Bandman, "Parallelization efficiency versus stochasticity in simulation reaction–diffusion by cellular automata", *The Journal of Supercomputing*, Vol. 73, Issue 2, pp 687–699, 2017.

[13] Nuclear Waste Management Organization, *Postclosure Safety Assessment: Data*, Toronto, Canada, 2011.

[14] Posiva Oy, *Safety Case Plan for the Operating License Application*, Olkiluotou, Finland, 2017.

[15] Svensk Kärnbränslehantering AB (SKB), *Environmental Impact Statement*, Stockholm, Sweden, 2011.

[16] Nuclear Waste Management Organization, *Deep Geological Repository Conceptual Design Report Crystalline / Sedimentary Rock Environment*, Toronto, Canada, 2016.

[17] Nuclear Waste Management Organization, *Ensuring Safety: Multiple-Barrier System*, Toronto, Canada, Retrieved from [https://www.nwmo.ca/], 2015

[18] Nuclear Waste Management Organization, *Choosing a Way Forward The Future of Canada's Used Nuclear Fuel (Final Study - Summary)*, Toronto, Canada, 2005.

[19] F. Garisto, *Fourth Case Study: Features, Events and Processes*, NWMO TR-2012-14, Toronto, Canada, 2012.

[20] F. Garisto, *Fifth Case Study: Features, Events and Processes*, NWMO TR-2013-06, Toronto, Canada, 2013.

[21] A. Jarvine et al. "A Probabilistic Approach to Predicting Used Fuel Container Lifetimes in a Deep Geological Repository", Technical Report, Department of Civil and Environmental Engineering, University of Waterloo, Waterloo, Canada, 2016.

[22] S. Briggs et al. "Multi-dimensional Transport Modelling of Corrosive Agents Through a Bentonite Buffer in a Canadian Deep Geological Repository", *Science of Total Environment*, 599-600 (2017) 348-354.

[23] Svensk Kärnbränslehantering AB (SKB), *Corrosion Calculations Report for the Safety Assessment SR-Site*, Technical Report TR-10-66, Stockholm, Sweden, 2010.

[24] Svensk Kärnbränslehantering AB (SKB), *Data and Uncertainty Assessment, Migration parameters for the bentonite buffer in the KBS-3 concept*, Technical Report TR-04-18, Stockholm, Sweden, 2004.

[25] Svensk Kärnbränslehantering AB (SKB), *Microbial Surphate-producing Activity in Water Saturated MX-80, Asha and Calcigel Bentonite at Wet Densities From 1500 to 2000 kg* m^{-3}, Technical Report TR-16-09, Stockholm, Sweden, 2017.

[26] J. von Neumann "Theory of Self-reproducing Automata (edited by Arthur W. Burks)", University of Illinois Press, 1966. Available: http://cba.mit.edu/events/03.11.ASE/docs/VonNeumann.pdf [Accessed Nov. 2017].

[27] M. Gardner, "Mathematical Games – The fantastic combinations of John Conway's new solitaire game of "life"", *Scientific American*. October 223: 120–123, 1970.

[28] S. Wolfram, "Statistical Mechanics of Cellular Automata", *Review of Modern Physics*, 55 (1983) 601.

[29] S. Wolfram, *A New Kind of Science*, Wolfram Media Inc, 2002.

[30] S. Omohundro, S. "Modelling cellular automata with partial differential equations", *Physica D* 10 128-134, 1984.

[31] K. Ponnambalam, A.W. Heemink, S.G. Fletcher, and P.E. Kloeden, *Models for Water and Environmental Analysis and Design: An Interactive Web-Book*, 2010. Available: http://epoch.uwaterloo.ca:8008/software/ [Accessed Nov. 2017].

[32] A. Einstein, "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen", *Annalen der Physik* 322: 549–560, 1905.

[33] T.M. Roberts, "Numerical Study for Diffusion in Mixed Material Mixures: The Treatment of Mixed Material Cells", *Computational Physics Summer Workshop*, 2013-06-10 - 2013-08-16, Los Alamos, New Mexico, United States, 2013. Available: http://www.columbia.edu/~tmr2122/files/roberts_final_report.pdf [Accessed: Nov. 2017].

[34] J.D. Hoffman, *Numerical Methods for Engineers and Scientists,* Second ed. Marcel Dekker, Inc., 1992.

[35] G. Muyzer and J.M. Stams, "The Ecology and Biotechnology of Sulphate-Reducing Bacteria", *Nature Reviews Microbiology*, May 7, 2008.

[36] E. Huttunen-Saarivirta et al., “Corrosion of Copper in Oxygen-deficient Groundwater with and without Deep Bedrock Micro-organisms: Characterisation of Microbial Communities and Surface Processes”, *Applied Surface Science* 396 (2017) 1044-1057.

[37] E. Huttunen-Saarivirta et al., “Corrosion Behaviour of Copper Under Biotic and Abiotic Conditions in Anoxic Groundwater: Electrochemical Study”, *Electrochimica Acta* 203 (2016) 350-365.

[38] A. Bengtsson and K. Pedersen “Microbial Sulphide-Producing Activity in Water Saturated Wyoming MX-80, Asha and Calcigel Bentonites at Wet Densities From 1500 to 2000 kg m^{-3}”, *Applied Clay Science* 137 (2017) 203-212.

[39] E.J. Miller, “Transportation models”, in *Handbook of Microsimulation Modelling*, C. O’Donoghue, Ed., Bingley, UK: Emeral, 2014, pp. 385-420.

[40] E.J. Miller, “Microsimulation”, in Transportation Systems Planning Methods and Applications, K.G. Goulias, Ed., Boca Raton, Florida: Taylor & Francis Group, 2003, pp. 12-1 – 12-22.

[41] J. Flamino et al., *Characterizing Information Importance and the Effect on the Spread in Various Graph Topologies*, arXiv: 1706.07405v1, 2017.

[42] I. Klein et al., “An Agent-Based Model of the Emergence of Cooperation and a Fair and Stable System Optimum using ATIS on a Simple Road Network”, *Transportation Research Part C*, 86: 183-201, 2018.

[43] W. Shang et al., “Agent-Based day-to-day Traffic Network Model with Information Percolation”, *Transportmetrica A: Transport Science*, Vol. 13, No. 38-66, 2017.

[44] F. Wei et al., “Day-to-day Dynamics Considering Social Interaction: From Individual Route Choice Behavior to a Network Flow Model”, *Transportation Research Part B*, 94: 335-354, 2016.

[45] F. He et al., “Tradable Credit Schemes on Networks with Mixed Equilibrium Behaviors”, *Transportation Research Part B*, 57: 47-65, 2013.

[46] J. Wahle et al., “The Impact of Real-time Information in a Two-Route Scenario using Agent-Based Simulation”, *Transportation Research Part C*, 10:399-417, 2002.

[47] E. Lehrer and R. Smorodinsky “Relative Entropy in Sequential Decision Problems”, *Journal of Mathematical Economics* 33: 425-439, 2000.

[48] C.R. Reid et al., “Decision-making without a brain: how an amoeboid organism solves the two-armed bandit”, *The Royal Society Publishing*, Vol. 13 issue 119, June, 2016.

[49] D Backus, M. Chernov and S. Zin Stanley “Sources of Entropy in Representative Agent Models”*, The Journal of Finance*, Vol. LXIX No 1 February, 2014.

[50] S. Van Nieuwerburgh and L. Veldkamp “Information Acquisition and Under-Diversification”, EFA 2005 Moscow Meetings, January, 2008.

[51] Turkay et al., “Integrating Information Theory In Agent-Based Crowd Simulation Behavior Models”, *The Computer Journal*, Febuary 23th, 2011.

[52] U. Wilensky, “Modeling Nature’s Emergent Patterns with Multi-agent Languages”, *Proceeding from the 8th European Logo Conference*, Austria, 2011.

[53] C. Macal and M. North, “Agent-Based Modeling and Simulation”, Proceedings of the 2009 Winter Simulation Conference, 2009.

[54] K. Pushkin and S. Shankar, “Traffic Assignment using a Density-Based Travel-Time Function for Intelligent Transportation Systems”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 5, May 2016.

[55] H. Spiess, (1990) Technical Note-Conical Volume-Delay Functions, *Transportation Science* 24(2): 153-158 https://doi.org/10.1287/trsc.24.2.153, 1990.

[56] Transportation Research Board, *Highway capacity manual*, Technical Report 204, Washington, DC, USA, 1985.

[57] D. Braess, “On a Paradox of Traffic Planning”, *Transportation Science*, Vol. 39, No. 4, November 2005, pp. 446-450.

[58] C. Zhao et al., “Braess Paradox and Robustness of Traffic Networks under Stochastic User Equilibrium”, *Transportation Research Part E*, 61(2014) 135-141.

[59] D. Acemoglu et al., “Informational Braess’ Paradox: The Effect of Information on Traffic Congestion”, arXiv: 1601.02039v1, 2016.

[60] R. Colombo and H. Holden “On the Braess Paradox with nonlinear Dynamics and Control Theory”, *Journal of Optimization Theory and Applications*, 168: 216-230, Springer, 2016.

[61] D. Fotakis et al., “Resolving Braess’s Paradox in Random Networks”, *Algorithmica*, July 2017, Vol. 78, issue 3, pp. 788-818.

[62] S. Bittihn and A. Schadscheider, “Braess Paradox in a Network of Totally Asymmetric Exclusion Processes”, arXiv: 1808.03753v2, 2016.

[63] T. Thunig and K. Nagel, “Braess’s Paradox in an Agent-Based Transport Model”, *Procedia Computer Science* 83 (2016) 946-951, 5Th International Workshop on Agent-Based Mobility Traffic and Transportation Models, Methodologies and Applications 2016.

[64] A. Bazzan and F. Klügl “Case Studies on the Braess Paradox: Simulating Route Recommendation and Learning in Abstract and Microscopic Models”, *Transportation Reseach Part C*, 13(2005) 299-319.

[65] Y. Sheffi, *Urban Transportation Networks: equilibrium Analysis with Mathematical Programming Methods*, Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[66] H. Markowitz, “Portfolio Selection”, *The Journal of Finance*, Vol. 7, No. 1, pp 77-91, 1952.

[67] J. Binns, Nuclear Waste Management Organization, Personal communication, [April 4th, 2018].

[68] S. Stroes-Gascoyne, C.J. Hamon, P. Maak, S.Russell, “The effects of the physical properties of highly compacted smectitic clay (bentonite) on the culturability of indigenous microorganisms”, *Applied Clay Science*, Vol. 47, Issues 1-2, January, 2010, pp. 155-162

Appendix A

Derivation of a realistic right boundary condition with reaction.

Let us assume a 1D (plus time) diffusion problem with a constant concentration on the left BC, a chemical reaction on the right BC, and initial condition zero everywhere (but the left BC):

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2}$$

Initial condition

$$C(x, t = 0) = 0, \qquad \forall x$$

Left BC

$$C(x = 0, t) = C_0, \qquad \forall t$$

For the right BC, let us describe the relation between accumulation, incomings and outcomings at the boundary as follows

$$\begin{bmatrix} Rate\ of \\ Accumulation \end{bmatrix} = \begin{bmatrix} Incoming\ rate \\ by\ diffusion \end{bmatrix} - \begin{bmatrix} Outcoming\ rate \\ by\ reaction \end{bmatrix}$$

the incoming concentration arrives at a speed $V_x = \frac{D_x}{\Delta x}$ and the outcoming rate is described by the reaction term $R\left(C_{L_x}\right)$, therefore we have at the right BC

$$\frac{\partial C(L_x)}{\partial t} = -V_x \frac{\partial C(L_x)}{\partial x} + R\left(C_{L_x}\right), \qquad \forall t \qquad (A.1)$$

However, we know that the incoming flux should be the same as the outcoming flux, therefore, eq. A.1 should have $\frac{\partial C(L_x)}{\partial t} = 0$, then the **right BC becomes**

$$V_x \frac{\partial C(L_x)}{\partial x} = R\left(C_{L_x}\right) \qquad (A.2)$$

where $V_x = \frac{D_x}{\Delta x}$

Dimensional analysis

Rearranging eq. A.2 we have

$$V_x \frac{\partial C(L_x)}{\partial x} = R\left(C_{L_x}\right)$$

$$\frac{D_x}{\Delta x}\frac{\partial C(L_x)}{\partial x} = R\left(C_{L_x}\right)$$

$$D_x \frac{\partial C(L_x)}{\partial x} = \Delta x\, R\left(C_{L_x}\right)$$

Left hand side units

$$\left[\frac{m^2}{s}\right]\left[\frac{ppm}{m}\right] = \left[\frac{m\, ppm}{s}\right]$$

Right hand side units

$$[m]\left[\frac{ppm}{s}\right] = \left[\frac{m\, ppm}{s}\right]$$

Example

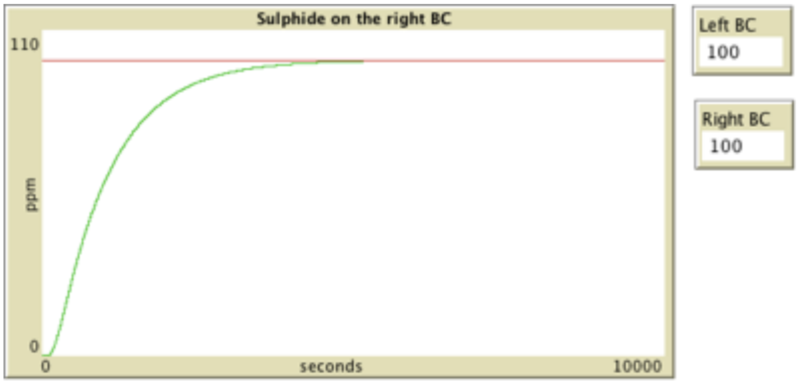
Consider the case where $R\left(C_{L_x}\right) = -kC_{L_x}$, after applying finite differences the right BC becomes

$$C_{L_x}^{t+1} = \frac{D_x C_{L_x-1}^{t}}{D_x + \Delta x^2 k}$$

we solved the equation using cellular automata with the following values:

C_0	D_x	Δt	Δx	L_x
100 ppm	0.4 [m^2/s]	1 [s]	1 [m]	30 [m]

For k=0 (no reaction at the boundary)



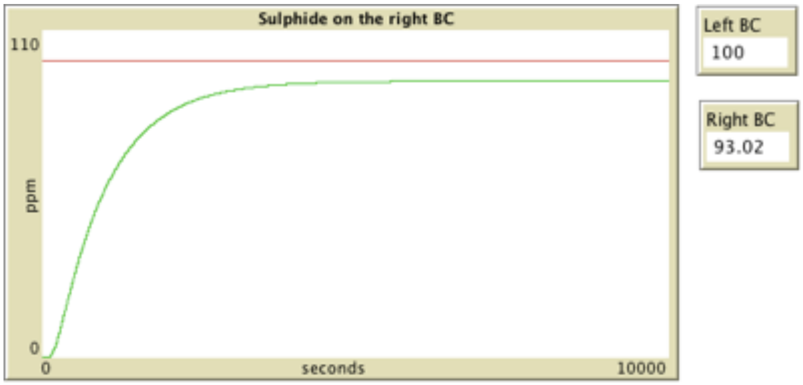
Left BC
100

Right BC
100

After 10,000 [s]	
Left BC	Right BC
100 ppm	100 ppm

Figure A.1

For k=0.001 (slow reaction)



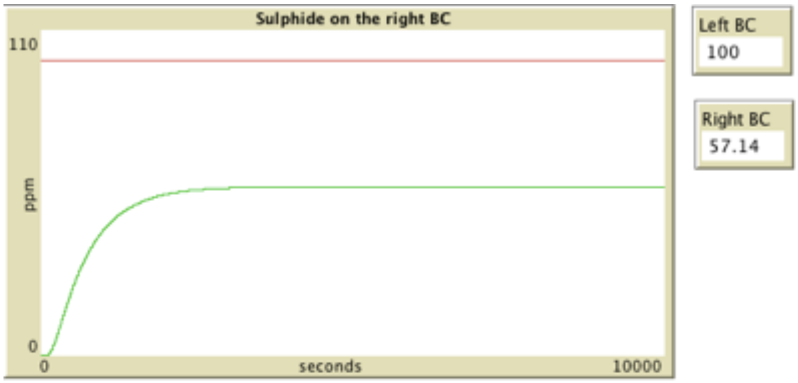
Left BC
100

Right BC
93.02

After 10,000 [s]	
Left BC	Right BC
100 ppm	93 ppm

Figure A.2

For k=0.01 (medium slow reaction)



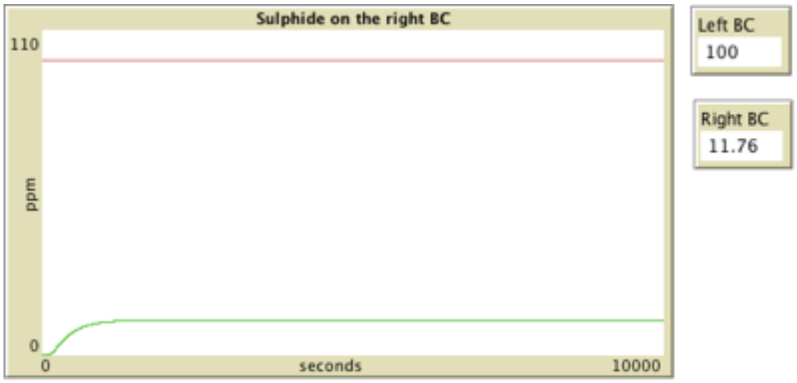
Left BC
100

Right BC
57.14

After 10,000 [s]	
Left BC	Right BC
100 ppm	57 ppm

Figure A.3

For k=0.1 (fast reaction)



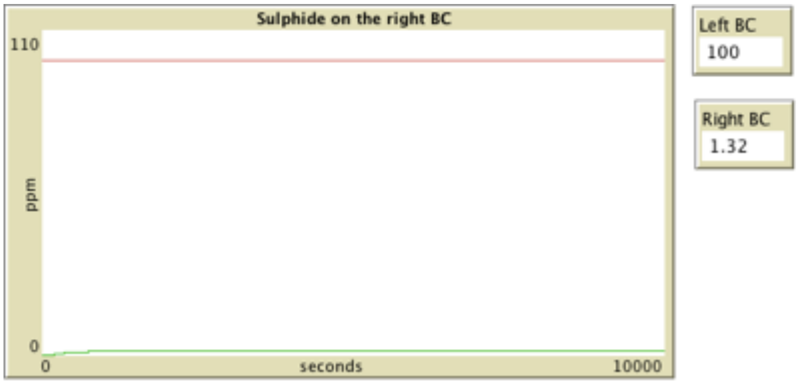
Left BC
100

Right BC
11.76

After 10,000 [s]	
Left BC	Right BC
100 ppm	11.7 ppm

Figure A.4

For k=1 (faster reaction)



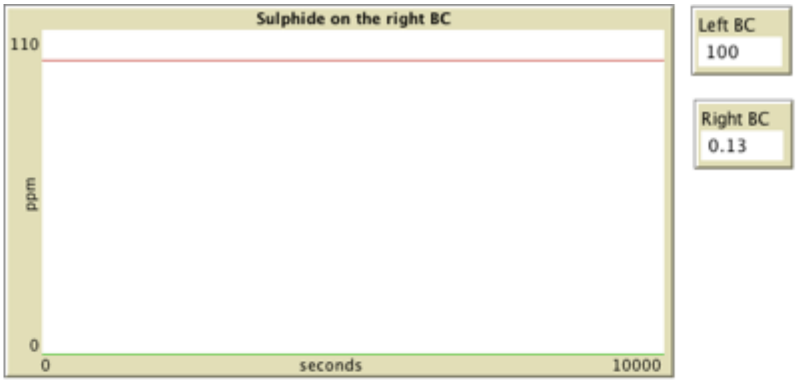
Left BC
100

Right BC
1.32

After 10,000 [s]	
Left BC	Right BC
100 ppm	1.3 ppm

Figure A.5

For k=10 (~ instantaneous reaction)



Left BC
100

Right BC
0.13

After 10,000 [s]	
Left BC	Right BC
100 ppm	0.13 ppm

Figure A.6

Conclusion

Assuming instantaneous reaction is the worst case, since any other rate produces some accumulation at the boundary, i.e., chemical that does not react. Having concentration at the boundary slows down the diffusion process.

A realistic right boundary condition must have the form of eq. A.2 that balances mass transport and reaction rate.

The faster the reaction rate, the closer it resembles the SKB & NWMO assumption of fixing $C(L_x) = 0$.

Appendix B

For the derivation of the advection diffusion equation we start again using a function whose value at (x, t+Δt) is equal to the expectation at (x+Δx, t). Then we proceed as follows

$$f(x,t+\Delta t) = E[f(x+\Delta x,t)]$$

$$= \int f(x+\Delta x,t)P(\Delta x)\,d\Delta x$$

Expanding the LHS using Taylor series

$$f(x,t+\Delta t) = f(x,t) + f_t(x,t)\Delta t + R_f(\Delta t^2)$$

Expanding the RHS using Taylor series

$$\int f(x+\Delta x,t)P(\Delta x)\,d\Delta x$$

$$= \int \left\{ f(x,t) + f_x(x,t)\Delta x + f_{xx}(x,t)\frac{\Delta x^2}{2!} + R_f(\Delta x^3) \right\} P(\Delta x)\,d\Delta x$$

Taking the expectation of the RHS

$$= f(x,t) + E[\Delta x]\,f_x(x,t) + \frac{E[\Delta x^2]}{2} f_{xx}(x,t) + E\left[R_f(\Delta x^3)\right]$$

Setting equal and simplifying LHS and RHS

$$f_t(x,t)\Delta t + R_f(\Delta t^2) = E[\Delta x]\,f_x(x,t) + \frac{E[\Delta x^2]}{2} f_{xx}(x,t) + E\left[R_f(\Delta x^3)\right]$$

Solving for the derivative with respect to time

$$f_t(x,t) = \frac{E[\Delta x]}{\Delta t}\,f_x(x,t) + \frac{E[\Delta x^2]}{2\Delta t} f_{xx}(x,t) + \frac{E\left[R_f(\Delta x^3)\right] - R_f(\Delta t^2)}{\Delta t}$$

In the limit when Δt→0 the residual terms become zero, therefore we can remove them from the equation. Then we have

$$f_t(x,t) = \frac{E[\Delta x]}{\Delta t}\, f_x(x,t) + \frac{E[\Delta x^2]}{2\Delta t} f_{xx}(x,t)$$

which yields the following coefficients for advection and diffusion: $V_x = \frac{E[\Delta x]}{\Delta t}$ and $D_x = \frac{E[\Delta x^2]}{2\Delta t}$. Therefore

$$E[\Delta x] = V_x\, \Delta t$$

$$E[\Delta x^2] = 2D_x \Delta t.$$

Appendix C

The exponentially adjusted reduction rate is

$$k_A(\rho_D) = k_0 exp[-b_E(\rho_D - 1)] - C_E \qquad (C.1)$$

where $k_0 = k_R + C_E$, k_R being the original reduction rate from the experiment and C_E the value at which we will consider the rate to be zero.

For $\rho_D = 1$ [g/m^3] we must get the reduction rate from the experiment, i.e., $k_A(\rho_D) = k_R$, which holds true:

$$k_A(1) = k_0 - C_E = k_R + C_E - C_E = k_R$$

For $\rho_D = \rho_S = 2.76$ [g/m^3], i.e., the specific density, the rate must become zero:

$$k_A(\rho_S) = k_0 exp[-b_E(\rho_S - 1)] - C_E = 0$$

Solving for b_E we get

$$b_E = \frac{ln\left(\frac{k_0}{C_E}\right)}{\rho_S - 1} \qquad (C.2)$$

Substituting eq. C.2 into eq. C.1, we get

$$k_A(\rho_D) = k_0 exp\left[-\frac{ln\left(\frac{k_0}{C_E}\right)}{\rho_S - 1}(\rho_D - 1)\right] - C_E$$

Simplifying the above, we get

$$k_A(\rho_D) = k_0 \left(\frac{k_0}{C_E}\right)^{\frac{\rho_D - 1}{\rho_S - 1}(-1)} - C_E$$

$$= k_0 \left(\frac{k_0}{C_E}\right)^{\frac{1 - \rho_D}{\rho_S - 1}} - C_E$$

$$= \left[\frac{k_0}{C_E}\left(\frac{k_0}{C_E}\right)^{\frac{1-\rho_D}{\rho_S-1}} - 1\right] C_E$$

$$= \left[\left(\frac{k_0}{C_E}\right)^{\frac{\rho_S-\rho_D}{\rho_S-1}} - 1\right] C_E$$

$$= \left[\left(\frac{k_R + C_E}{C_E}\right)^{\frac{\rho_S-\rho_D}{\rho_S-1}} - 1\right] C_E$$

$$= \left[\left(\frac{k_R}{C_E} + 1\right)^{\frac{\rho_S-\rho_D}{\rho_S-1}} - 1\right] C_E$$

Therefore eq. C.1 becomes

$$k_A(\rho_D) = \left[\left(\frac{k_R}{C_E} + 1\right)^{\frac{\rho_S-\rho_D}{\rho_S-1}} - 1\right] C_E, \qquad \rho_D \in [1, \rho_S], \qquad 0 < C_E < k_R \quad (C.3)$$

Bacterial Activity Probability

Dividing eq. (C.3) by k_R then we can get the probability of having SRB active at a given dry density:

$$P(\rho_D) = \left[\left(\frac{k_R}{C_E} + 1\right)^{\frac{\rho_S-\rho_D}{\rho_S-1}} - 1\right] \frac{C_E}{k_R}, \qquad \rho_D \in [1, \rho_S] \qquad (C.4)$$

where ρ_D is the dry density of the clay, $k_R = 6.16146 \times 10^{-9}$ [1/s] is the reaction rate coefficient calculated from experiments and $C_E = 1.37516 \times 10^{-14}$ [1/s] represents the value at which we will consider the rate to be zero. Both values where found by optimization methods to match experiment results and the fact that at $\rho_S = 2.76$ [g/cm^3] no bacterial activity is possible. Replacing those values, we get

$$P(\rho_D) = \left[(a \times 10^5 + 1)^{\frac{2.76-\rho_D}{1.76}} - 1\right] * \frac{1}{a} \times 10^{-5}, \qquad \rho_D \in [1, 2.76]$$

$$P(\rho_D) = \frac{(a \times 10^5 + 1)^{\frac{2.76-\rho_D}{1.76}} - 1}{a \times 10^5}, \qquad \rho_D \in [1, 2.76]$$

where a = 4.4805.

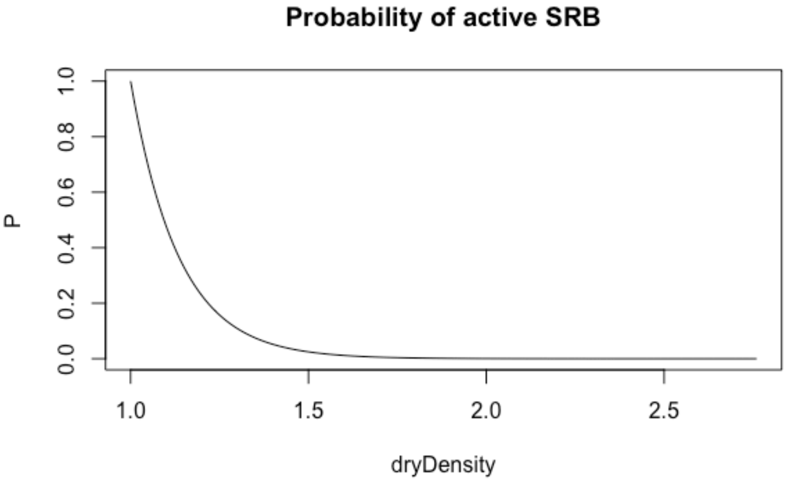


Figure C.1 Probability of having active SRB as a function of dry density

Table C.1 Selected values

Dry density [g/cm^3]	P(SRB)
1	1
1.1	0.477420198
1.2	0.227929436
1.3	0.108817216
1.4	0.051950497
1.5	0.024801111
1.6	0.011839414
1.7	0.005651223
1.8	0.002696848
1.9	0.001286367
2	0.000612973
2.1	0.00029148
2.2	0.000137992
2.3	6.47142E-05
2.4	2.97296E-05
2.5	1.30272E-05
2.6	5.05312E-06
2.7	1.24613E-06
2.76	0

Appendix D

Computation of π

As an example of a situation where simulation is not the ideal tool to use, we will compute an estimation of π using simulation and two convergent series: Leibniz and Ramanujan π series. We chose those series to represent two extremes, a sublinear convergent series (Leibniz's) and a superlinear convergent series (Ramanujan's).

For the simulation method we used a unitary circle centered at the origin, we created a set of points using Monte Carlo and evaluated if they fell inside the unit circle, then we proceeded to compute π according to

$$\frac{\pi}{4} = \frac{Points\ inside\ circle}{Total\,points}$$

The Leibniz's π series is

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$$

$$\frac{\pi}{4} = \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}$$

The Ramanujan's π series is

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{i=0}^{\infty} \frac{(4i)!\,(26390i + 1103)}{(i!)^2\, 396^{4i}}$$

The estimation of π using simulation is shown in Table D.1, we see its value gets closer to π as we increase the size of the sample and the number of runs; however, it requires great computational effort.

The estimation using Leibniz's series after 100,000 iterations is

- 3.1415826535897198

That using Ramanujan's series after 2 iterations is

- 3.141592653589793

Table D.1 Estimation of π using simulation

		Sample size			
		30	100	1,000	10,000
Runs	1	*3.3333*	*3.2450*	*3.1840*	*3.1288*
	10	*3.2666*	*3.2000*	*3.1040*	*3.1437*
	30	*3.1111*	*3.1213*	*3.1309*	*3.1413*
	100	*3.1578*	*3.1432*	*3.1423*	*3.1416*

For this example, we have seen that simulation behaves badly, not only consumes huge computational power but it provides an inaccurate estimation. Its most accurate estimation was obtained after creating 2×10^6 random numbers achieving a ~20% better estimation than that using Leibniz at 10^5 iterations. In contrast, two iterations of Ramanujan series and the default accuracy of the computer is not enough to carry on with the calculation. In addition to that, results from simulation should be treated statistically, adding another layer of time or resources.

Appendix E

Proof the area under the diffusion-like curve is always equal to one.

Suppose an initial value f0(s), such that $\int_a^b f_o(s)ds = 1$.

The analytical solution of the diffusion-like ODE is $f(t) = \frac{1}{b-a} - a_0 exp(-\gamma t)$.

At time t = 0, the solution should be equal to its initial condition:

$$f(t=0) = \frac{1}{b-a} - a_0 = f_o(s)$$

solving for a0 gives

$$a_0 = \frac{1}{b-a} - f_o(s).$$

Replacing a0 into the ODE solution and after some algebra it yields

$$f(s,t) = \frac{1}{b-a}[1 - exp(-\gamma t)] + f_o(s)\, exp(-\gamma t).$$

Since f(s,t) is always nonnegative, to ensure it is a pdf at any time, the following must be met

$$\int_a^b f(s,t)ds = 1, \qquad \forall t$$

$$\int_a^b f(s,t)ds = \frac{[1 - exp(-\gamma t)]}{b-a}\int_a^b ds + exp(-\gamma t)\int_a^b f_o(s)ds, \qquad \forall t$$

$$= \frac{[1 - exp(-\gamma t)]}{b-a} s|_a^b + exp(-\gamma t) * 1, \qquad \forall t$$

$$= 1 - exp(-\gamma t) + exp(-\gamma t), \qquad \forall t$$

$$= 1, \qquad \forall t$$

Therefore, at any time, the area under the curve of the diffusion-like process is equal to one.