

**A Novel Delay Differential Equation Model of the Germinal Center Reaction
and an Algorithm for Minimum Length Surveillance Paths**

A Dissertation
Presented to
The Academic Faculty

By

Benjamin C. Ide

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mathematics
College of Sciences

Georgia Institute of Technology

August 2022

© Benjamin C. Ide 2022

**A Novel Delay Differential Equation Model of the Germinal Center Reaction
and an Algorithm for Minimum Length Surveillance Paths**

Approved by:

Dr. Sung Ha Kang, Advisor
School of Mathematics
Georgia Institute of Technology

Dr. Haomin Zhou
School of Mathematics
Georgia Institute of Technology

Dr. Martin Short
School of Mathematics
Georgia Institute of Technology

Dr. Veronika Zarnitsyna
Microbiology and Immunology
Emory University School of Medicine

Dr. Molei Tao
School of Mathematics
Georgia Institute of Technology

Date approved: May 19, 2022

ACKNOWLEDGMENTS

I would like to thank my advisor Sung Ha Kang for not just putting up with me, but believing in me. Her unwaivering insistence that I had the ability to complete this program is the reason that it has proven true. This dissertation would not exist without her.

I would like to thank my committee member, research mentor, and collaborator Veronika Zarnitsyna. She welcomed me into her research group at Emory with open arms and guided me through learning a new research area from scratch.

Thank you to the rest of my committee, Haomin Zhou, Martin Short, and Molei Tao. Each one has been more than kind in my years at Tech. Their personal stories shared at ACM lunches were vital to my understanding of how maybe I actually can fit in in academia. Each have shaped my mathematical thinking in various ways as well.

Teaching has been one of my most valuable experiences at Tech. I would like to thank Klara Grodzinsky for providing me with so many teaching opportunities. From being a TA, a head TA, an instructor, and even a TA for the mandatory class for new TAs, I have learned far more than just how to teach.

Thank you to all of my friends who have helped me along the way. I'd like to specifically name Eric Sabo who has always been there to pull me out of a rut, Libby Taylor and Lea Marzo who have been exemplars of grit to (attempt to) emulate, and the guys in the men's group at the GT Counselling Center who always help me put things into perspective.

I'd like to thank my parents for their support, my uncle Gregg for always convincing me to reach further (geographically as well as academically), and to Mollie for always voicing a real interest in my progress and success.

Thank you to my wife, Liz. Your love and support are everything. "My" progress through this program has always been our progress, and I do not have the words to express my gratitude for that.

TABLE OF CONTENTS

Acknowledgments	iii
List of Tables	viii
List of Figures	ix
List of Acronyms	xi
Summary	xii
I Model of the Germinal Center Reaction	1
Chapter 1: Background and Introduction: The Germinal Center Reaction . . .	2
1.1 The Humoral Adaptive Immune System	2
1.2 The Germinal Center	3
1.3 The Role of Mathematical Modeling in Understanding the GC	5
1.3.1 Mutation	6
1.3.2 Number of Divisions	7
1.3.3 Antibody Feedback	7
1.3.4 Differential Equations vs. Agent-based	8
Chapter 2: Model of the Germinal Center	9
2.1 Main equations	10

2.1.1	Variables and parameters	10
2.1.2	Initial condition	10
2.1.3	Day 0 to day 3	11
2.1.4	Full equation from day 3	13
2.2	Epitope Masking via Antibody Feedback	13
2.3	T-cell Selection Strength	14
2.4	Antibody Production	16
2.5	Number of Divisions	16
2.6	Mutation	17
2.7	Flux of B-cells Completing Division, $E(x, t)$	18
2.8	Assumptions and Numerical Considerations	19
2.8.1	Single epitope model	19
2.8.2	FDCs remain stocked	20
2.8.3	Both primary and secondary GC reactions	20
2.8.4	Quadrature and DE solvers	20
Chapter 3: Results and Discussion		21
3.1	Validation	21
3.1.1	GC Size	22
3.1.2	Dividing cell ratio	23
3.1.3	Rate of movement between zones and LZ/DZ size ratio	24
3.2	Main results	25
3.3	Discussion	26

II	Surface Surveillance from Minimum Length Paths	29
Chapter 4:	Introduction and Background: Path Optimization	30
4.1	Previous Work and Literature on Path Planning	30
4.2	Literature on Surface Surveillance	32
4.3	Mathematical Details	34
4.3.1	Environment Definition and Surface Discretization	35
4.3.2	Paths	37
4.3.3	Visibility of Surfaces	38
Chapter 5:	Surface Surveillance from Minimum Length Path Algorithm	40
5.1	Constructions, Algorithms, and Optimization Techniques for SSMILE	41
5.1.1	The Environment and Surface to be Surveyed	41
5.1.2	Path Construction from Junction Points	42
5.1.3	Visibility Calculation	45
5.1.4	Projection to Full Surveillance	47
5.1.5	Global Optimization by Intermittent Diffusion	50
5.1.6	The Benefit of Intermittent Diffusion vs. Simulated Annealing	52
5.2	The SSMILE Algorithm	53
5.3	Computational Experiments	55
Chapter 6:	Sphere Surveillance	57
6.1	Background	57
6.1.1	History and Results in \mathbb{R}^2	57
6.1.2	The Inradius Problem in \mathbb{R}^3	60

6.2	SSMILE Implementation for Inradius	62
6.2.1	Computing the Inradius	62
6.2.2	Modified Local Optimization Strategy	64
6.2.3	Modified SSMILE Algorithm	66
6.3	Numerical Results, Validation, and Prediction	66
Chapter 7: Theoretical Explorations		71
7.1	Introduction	71
7.2	Generalized Setup and Observations	71
7.3	Surface Surveillance in Simple Polygonal Environments	73
7.3.1	Closed Paths	74
7.3.2	Open paths	76
Chapter 8: Conclusions		79
8.1	Model of the Germinal Center Reaction	79
8.2	Surface Surveillance from Minimal Length Paths	80
References		82

LIST OF TABLES

2.1	Model variables.	10
2.2	Model functions.	11
2.3	Model parameters and initial conditions.	12

LIST OF FIGURES

2.1	Model schematic.	9
2.2	The number of divisions resulting from T-cell signal strength p_s	17
3.1	The number of B-cells comprising the GC over time.	22
3.2	Dark zone to light zone ratio of S-phase B-cells.	24
3.3	Light zone/dark zone dynamics.	24
3.4	Inclusion of naive seeders and the inclusion of positive antibody feedback both lead to improved antigen neutralization.	25
3.5	Seeder cell distribution	26
3.6	BC output rate and LZ distribution with realistic seeding.	27
5.1	An example of G_{LOS}	44
5.2	Local modification to this path leaves visibility unchanged.	47
5.3	Surveillance of all walls in an art gallery floor plan.	56
5.4	Optimal path for torus surveillance.	56
6.1	Optimal path for the lost in a forest problem.	58
6.2	The shortest planar inradius 1 curve.	59
6.3	Zalgaller's conjectured optimal inradius 1 path in \mathbb{R}^3	61
6.4	The construction of the shortest closed path of inradius 1 in \mathbb{R}^3	62

6.5	Local reduction in path length followed by global rescaling does not work.	65
6.6	Relative error from true/conjectured solutions.	67
6.7	SSMILE results for the open path inradius problem with various numbers of junction points.	68
6.8	SSMILE results for the closed path inradius problem with various numbers of junction points.	69
7.1	Example of disentangling a path.	74
7.2	An example of a path of the form given in Theorem 7.3.1.	75
7.3	Open path in a simple polygon example where vertex surveillance does lead to boundary surveillance.	77
7.4	This construction gives a path that surveys the entire boundary.	77
7.5	Hallway example.	78

SUMMARY

The humoral adaptive immune system in vertebrates includes a process called the germinal center reaction in which B-cells rapidly increase their binding affinity to an antigen that is part of a pathogen. A fraction of germinal center B cells differentiates into plasma cells that secrete antibodies. Antibodies bind to the pathogen and neutralize it. In a secondary immune response to the same pathogen, memory B-cells and long-lived plasma cells generated during the primary immune response encode higher affinity antibodies and are able to fight the pathogen more efficiently.

We develop a delay differential equation model of the germinal center reaction incorporating known physical mechanisms. We find that secondary immune responses including low affinity seeder B-cells outperform those seeded only with higher affinity. This helps to explain a recent laboratory observation that a high fraction of seeder B-cells in a secondary immune response are naive. Further, two mechanisms of antibody feedback are explored, where antibodies produced in the reaction interact with the reaction itself. Negative feedback occurs via epitope masking, which is consistent with experimental data. Positive feedback occurs via improved antigen presentation on follicular dendritic cells, which is a mechanism we propose.

Additionally, we propose a novel path optimization algorithm. Given a path connected environment, our proposed algorithm finds the shortest paths from which surfaces in the environment are surveyed under a limited visibility constraint. We further explore how this is related to the inradius problem in classical geometry: What is the shortest curve whose convex hull includes the unit sphere? The solution is known for closed curves, but not for open curves. Our algorithm seems to converge numerically to the true solution for closed curves and to the best-known conjecture for open curves. This offers validation of our method and evidence for the open path conjecture.

Part I

Model of the Germinal Center Reaction

CHAPTER 1
BACKGROUND AND INTRODUCTION: THE GERMINAL CENTER
REACTION

1.1 The Humoral Adaptive Immune System

The immune system has two arms: The innate immune system that includes preprogrammed responses, and the adaptive immune system which includes processes that adapt to combat *a priori* unknown pathogens. This work is focused specifically on the humoral adaptive immune response which is mediated by B-cells which secrete 'Y'-shaped proteins called *antibodies* (Ab).

The root of the immune system begins in the bone marrow. *Stem cells*, cells that have not yet differentiated to a particular purpose, are found in the marrow. While exiting the marrow, they differentiate to a number of different cells, two of which are important in this work: *T-cells* (TCs) and *B-cells* (BCs).

T-cells have many roles, but here we focus on one: T-cells help in selection and activation of B-cells via a combination of membrane-bound and secreted signal proteins. One of the results of T-cell help is increasing B-cell cMyc expression which determines the number of divisions that the B-cell will undergo in the germinal center.

Each B-cell encodes for a single type of antibody which appears on its surface as a B-cell receptor (BCR). Through genetic recombination, a diverse repertoire of BCRs is generated to combat a wide range of pathogens. Additional diversification occurs via somatic hypermutation in the *germinal center reaction* during an immune response.

A variety of B-cells types are important in this work. Plasma cells secrete antibodies to bind and neutralize the pathogen. Memory B-cells have a long lifespan and are used to maintain immunity over years. Naive follicular B-cells encounter antigen in the secondary

lymphoid organs, and after receiving T-cell help, undergo proliferation and differentiation either into plasma cells or memory B-cells.

An antibody (or B-cell) is called *Ag-specific* if it is capable of binding to the antigen. The binding sites on the antigen are called *epitopes*. The strength of that binding is determined by the rate of binding, k_{on} , and the rate of unbinding, k_{off} . The ratio $K_a = k_{\text{on}}/k_{\text{off}}$ is the antibody's *affinity* to the epitope.

The topic of this work is the reaction through which follicular B-cells rapidly evolve to express antibodies of high affinity to epitopes found on antigen during an antigenic challenge such as infection. This reaction is called the *germinal center (GC) reaction* and the result of the reaction, the increase of affinity to the Ag in a B-cell population, is called *affinity maturation*.

For more information, see [1, 2]

1.2 The Germinal Center

Germinal Centers (GCs) are transient structures that appear during the immune response within B-cell follicles of lymph nodes and the spleen. GCs form around *follicular dendritic cells* (FDCs) that serve as a repository for pieces of antigen delivered by macrophages. They require a critical number of *T follicular helper cells* (T_{FH}) and follicular B-cells with Ag-specific BCRs for the process to begin.

During the germinal center reaction, two distinct functional regions appear, termed the *dark zone* (DZ) and the *light zone* (LZ) [3]. Within the dark zone, B-cells are found proliferating with an extreme rate of mutation affecting the antibody variable region with, on average, one point mutation per division [4]. Because the extreme mutation only occurs for the encoding of antibodies, not for the B-cell itself, this does not lead to an abnormal number of cell deaths and the B-cells do continue to function properly.

In the DZ, B-cell proliferation and mutation are the only mechanisms important for our model. Within the LZ, a more complicated interplay emerges involving B-cells and the

affinity of their BCRs, FDC-bound Ag, T_{FH}-cells, and free Ab. First, B-cells must collect Ag from the FDC. Once collected, they display peptide MHC (pMHC) on their surface. The B-cell interacts with T_{FH}-cells via the pMHC resulting in T-cell help. Higher BCR affinity to the Ag leads to more collection, thereby more pMHC presentation on the B-cell surface, more T_{FH} interaction, finally resulting in a stronger T-cell signal [2].

During this trek through the LZ, a B-cell must contend with external Ab entering the GC and masking the epitopes of FDC-bound Ag that the B-cell is attempting to collect. A major source of these Ab are plasma B-cells produced by the current immune response, i.e., there is negative feedback occurring [5]. Antigen collection, although modulated via Ab feedback, is not affected by direct competition with other B-cells of differing affinities, meaning that each B-cell has equal access to the FDC-presented antigen. Higher affinity leads to more Ag collection, but the affinity of neighboring B-cells does not affect this.

After collecting Ag, the B-cells compete with each other for access to T_{FH}-cells. B-cells that collect more Ag will bind longer and more frequently to T_{FH}-cells limiting the ability of other B-cells to receive a strong signal. Thus, unlike during Ag collection, competition for selection is directly affected by the distribution of affinities of the BCR repertoire of B-cells in the LZ.

If a strong enough T_{FH} signal is not received, the B-cell will undergo apoptosis. When a B-cell does receive a strong enough signal, we say that it is *selected*. Once selected, it is epigenetically altered to express higher levels of the onco-protein cMyc [6], with cMyc expression dependent on the strength of the signal [7]. The signal also changes the phenotype of the B-cell so that it expresses receptors for chemokines that cause it to migrate back to the DZ [8].

The expression of cMyc (and thus, the strength of the received signal) determines the number of divisions that the B-cell will undergo once back in the DZ, which is known to be between 1 and 6 [7]. After dividing, some portion of the daughter cells will migrate back to the LZ to continue participating in the reaction (a process termed *cyclic re-entry* [9, 10]),

while others differentiate to plasma and memory B-cells and enter the blood stream.

The competition in the LZ combined with cyclic re-entry induces Darwinian selection for B-cells encoding antibodies with greater affinity for the antigen, a process known as *affinity maturation* [2]. The goal of this work is to understand how the initial distribution of B-cells over affinity in a single GC affects affinity maturation.

In a *secondary* or *recall* immune response, the immune system has prior experience with the antigen it is responding to. B-cells resulting from affinity maturation during a prior immune response are available to seed GCs during a secondary response. Recent research has found that a major portion of the B-cells seeding the GC in a secondary response are naive [11].

It is interesting that more experienced B-cells are available, but they are not recruited more often than naive B-cells. The proposed model shows that a high proportion of naive seeder cells results more effective affinity maturation, implying that the experimental observation is predictable via the mechanisms included in the model.

1.3 The Role of Mathematical Modeling in Understanding the GC

Mathematical modeling has proven indispensable for the understanding of the GC reaction and affinity maturation. Experiments offer valuable data, but the interpretation of data in cellular biological processes is fraught with ambiguity due to complexity and the high variability from one experiment to the next. Modeling offers a way to pinpoint which mechanisms are important and give hints about causality within complicated systems.

The dynamics of the GC reaction are studied by systems of ordinary differential equations and agent based models. Successes of ODE models include predicting iterative maturation of GC B-cells via cyclic re-entry [9], the first large-scale understanding of the dynamics of the GC reaction and the role of T-cell help [12] that inspired many subsequent ODE models, a study predicting the recycling probabilities that includes the first use of shape-space for affinity [13] within GC models [10], and a study that proposed that recy-

clinging probability is time-varying and that it could be beneficial for high affinity B-cells to be preferentially output from the reaction [14].

Subsequently, as we saw increased speed and decreased price of computer hardware, agent based models became important in GC modeling [15]. Starting with a base model that included the vast majority of mechanisms that the community was aware of [16], many hypotheses were derived [8, 5, 17, 18] and experimentally verified [19, 5, 11].

We highlight here mechanisms for which our work makes significant contribution: Mutation, affinity dependent number of divisions, antibody feedback.

1.3.1 Mutation

A mutation has 3 outcomes for the daughter B-cell [4]:

1. The daughter B-cell is no longer capable of binding to the the Ag at all, which we call a *deleterious mutation*,
2. the daughter B-cell has exactly the same affinity as the parent (e.g., the mutation doesn't cause a phenotypic change), which we call a *silent mutation*, or
3. the mutation changes the affinity.

There is no predictive theory of the link between genotype and phenotype, so every extra presumption beyond this either implies a new hypothesis or constitutes a parameter of the model. Previous models that have included mutation have done this in one of two ways:

1. Split B-cells in to a small number of bins of affinity and let mutation describe flow between bins. E.g., B_{low} , B_{med} , B_{high} .
2. The shape space model [13, 10]. “Shape” refers to the phenotype of the antibody. Model a particular antibody's shape as a location in a high dimensional lattice, usually \mathbb{Z}^4 [10, 8, 18], with the origin representing the highest possible affinity, and other

locations having an affinity that decays exponentially with respect to the taxi-cab distance from the origin.

The first option is straightforward and makes no assumptions beyond acknowledging that there are different possible affinities. The latter includes additional assumptions. When using the shape space model, there are three major parameters introduced without predictive benefit: 1) The space itself, 2) a map from shape space to affinity, and 3) the choice of possible movements within the space to represent mutation.

We introduce a continuous affinity model analogous to the binning model above. Affinity is given by a variable $x \in [0, 1]$. Mutation in this space is informed by the literature [4] and is given by a discrete time, continuous space Markov process. This avoids the large parameter space that a shape space model introduces and makes no more assumptions than the binning model.

1.3.2 Number of Divisions

Some models have included more than one division for a single cycle through the dark zone. Many ODE models that have done this have used a fixed number of divisions, e.g., 10 [12]. Recent agent-based models [17] have implemented affinity and time dependent number of divisions in accordance with recent evidence [7], but this has not been included in a differential equations model.

Including this in a differential equation model is difficult because the number of divisions (and therefore the length of time that proliferation should occur) depends on the state of the light zone at the time when T-cell selection occurred. We implement this by using *delay* differential equations.

1.3.3 Antibody Feedback

It has been shown that exogeneous antibodies enter the GC and bind to antigen on the FDC [5]. However, it has been discovered that the antibody appears on the FDC-bound antigen

in *two* ways. 1) The traditionally understood masking, but also, 2) the antibodies are found participating in the connection holding the Ag to the FDC, meaning that more Ag-specific Ab available will allow the Ag to be presented more robustly in the GC reaction giving *positive* antibody feedback [20]. This mechanism has not been used in prior models and we investigate whether it produces results consistent with the literature.

1.3.4 Differential Equations vs. Agent-based

Prominent researchers in the field have argued that differential equations models and agent based models complement each other [15]. Agent based modeling has been highly successful recently in studying the physical dynamics of individual cells (the speed and nature of cell migration, e.g.). Meanwhile, there have been few differential equations based models in recent years and some mechanisms have not been explored in this framework.

Agent based models are resource intensive. There are hundreds of GCs in each lymph node, and thousands of lymph nodes in the human body. To get a full picture of the immune response, the interaction between GCs and lymph nodes must be understood. Differential equations solvers are fast enough to make this feasible with today's consumer laptops, while trying to run hundreds of thousands of interacting agent based models is not.

CHAPTER 2

MODEL OF THE GERMINAL CENTER

We model the germinal center reaction using a delay differential equation. There are two independent variables: Time t (days) and affinity x . All derivatives in the model are in time.

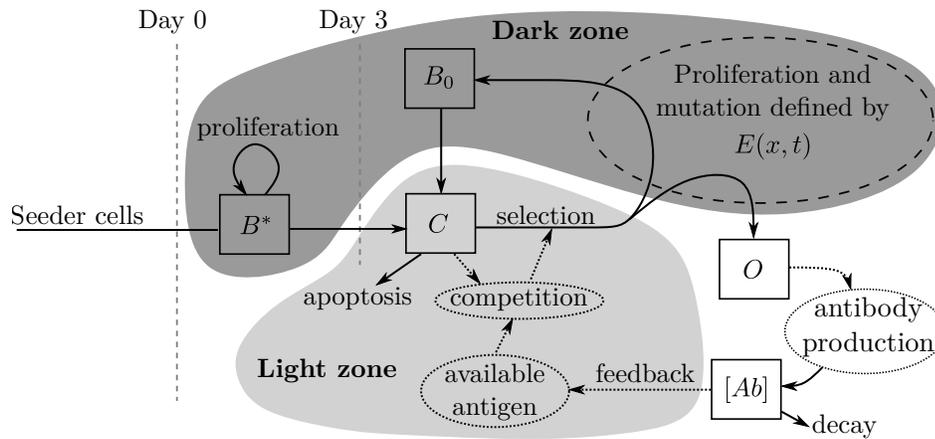


Figure 2.1: Model schematic.

Figure 2.1 presents an overview of the model dynamics. Boxes represent dependent variables, all of which are given as distributions over affinity, and solid arrows represent the flow of quantities. Dotted ovals represent mechanisms that are explicitly modeled with dotted arrows implying a direct effect. At day 0 an initial distribution of DZ B-cells start proliferating (B^*), and at day 3, these cells begin migrating to the LZ (C) where competition for available antigen determines selection. All selected B-cells migrate to the DZ to proliferate (for a length of time determined by selection strength) with mutation (a process included in $E(x, t)$). Some of these cells will be output (O) and go on to produce antibodies ($[Ab](x, t)$ is the *concentration* of antibodies), while some will be recycled (B_0).

The function $E(x, t)$ is the *flux of B-cells completing their final division* at time t of affinity x . Thus, this function contains the part of the model that requires a delay. This

function is of particular interest since it includes 1) a variable number of divisions 2) determined by the state of the system at a previous time and it is also 3) where the mutation model is incorporated. We start with the main equations of the model.

2.1 Main equations

Time t is measured in days starting at day 0. Affinity $x \in [0, 1]$ is a unitless quantity in the model that can be transformed to *real* affinity via Equation (2.1). The range of real affinities represented, $[10^{5.5}, 10^{9.5}]$, and Equation (2.1) itself have been used frequently in the literature [10, 16, 8, 5, 17].

$$K_a(x) = 10^{4x+5.5}. \quad (2.1)$$

2.1.1 Variables and parameters

The model variables are given in Table 2.1, functions are given in Table 2.2, and parameters are given in Table 2.3.

Table 2.1: Model variables.

Variable	Description
x	log of linearly transformed affinity, $0 \leq x \leq 1$
t	time (days)
$B^*(x, t)$	Distribution of seeder cells
$B_0(x, t)$	Distribution of recycled DZ B-cells after division
$C(x, t)$	Distribution of light zone B-cells
$O(x, t)$	Distribution of output B-cells
$[Ab](x, t)$	Distribution of Ab concentration produced by immune response

2.1.2 Initial condition

For $t \leq 3$, $C(x, t) = B_0(x, t) = O(x, t) = [Ab](x, t) = 0$. $B^*(x, 0)$ is nonnegative for all $x \in [0, 1]$. The variable $[Ab]$ only describes antibodies produced from this reaction, so it must start at 0. There could be Ag-specific antibody from previous immune responses. If

Table 2.2: Model functions.

Function	Description
$M(t)$	Proportion of epitopes that are bound by Ab
$f_i(x y)$	PDF for mutating from affinity y to x after i divisions
$p_s(x, t)$	Selection rate scale, $p_s \in [0, 1]$ (depends on C and M)
$D(p_s(x, t))$	Destined number of divisions for BCs of affinity x selected at time t
$E(x, t)$	Flux of BCs completing their final division
$K_a(x)$	Transformation from x to real affinity

the model describes a primary immune response then this is not an issue. For a secondary immune response, Ag-specific antibody from previous immune responses is present. We call this *background antibody*, $[Bg](x)$, and model it as a distribution of antibody concentration over affinity x . The background antibody does not change over time because we assume it is at or near equilibrium.

The initial condition $B^*(x, 0)$ is chosen so that $\int_0^1 B^*(x, 0) dx = 3$ which leads to the correct GC size at day $t = 3$ when the full reaction begins. It is not necessary for the initial distribution $B^*(x, 0)$ to be the sum of three delta distributions representing three real cells because the initial three days in this model approximate a more complicated recruitment of initial cells [11].

2.1.3 Day 0 to day 3

During the first three days of the GC reaction, recruited seeder B-cells proliferate in the dark zone with no mutation, so for $0 \leq t \leq 3$, the model follows Equations (2.2)–(2.3).

$$\frac{\partial B^*}{\partial t}(x, t) = \rho B^*(x, t), \quad (2.2)$$

$$\frac{\partial B_0}{\partial t}(x, t) = \frac{\partial C}{\partial t}(x, t) = \frac{\partial O}{\partial t}(x, t) = \frac{\partial [Ab]}{\partial t}(x, t) = 0. \quad (2.3)$$

Table 2.3: Model parameters and initial conditions.

Parameter	Description	Reference
$T_{\text{div}} = 1/4$	Dark zone BC division time (days)	[21]
$T_{\text{diff}} = 1$	Time (days) for output BCs to differentiate	[5]
$\rho = (\ln 2)/T_{\text{div}}$	Continuous rate of proliferation (days^{-1})	
$\delta = \frac{24}{16} \ln 2$	Centrocyte apoptosis rate (days^{-1})	[22]
$m_d = 9/32$	Probability of a deleterious mutation (used in $f_i(x y)$)	[4]
$m_s = 17/32$	Probability of a silent mutation (used in $f_i(x y)$)	[4]
$g = (\ln 2)/10$	Antibody decay rate (days^{-1})	[5]
$\sigma = 0.06$	Standard deviation of affinity change under mutation	
$\eta = 4.0$	Rate of BC migration, $B_0 \rightarrow C$ (days^{-1})	
$\lambda = 2.6$	Maximum rate of BC migration, $C \rightarrow \text{DZ}$ (days^{-1})	
$\mu = 7.0$	Rate of BC migration, $B^* \rightarrow C$ (days^{-1})	
$p_r = 0.7$	Probability of recycling given selection	
$h = 0.8, n = 3.0$	Terms defining the function $D(p_s(x, t))$	
$c = 2.5 \times 10^{-13}$	Antibody production coefficient	
$[Bg](x)$	Concentration of background Ag-specific antibody	
$s = 0.05, z = 0.2$	Parameters for positive antibody feedback	
$\int B^*(x, 0) dx = 3$	Initial number of seeder cells	[14]
$B^*(x, 0)$	Initial distribution of seeder cells	[23]
$B_0(x, 0) = 0$	Initial distribution of B-cells migrating $\text{DZ} \rightarrow \text{LZ}$	
$C(x, 0) = 0$	Initial proportion of light zone B-cells	
$O(x, 0) = 0$	Initial proportion of output B-cells	
$[Ab](x, 0) = 0$	Initial distribution of Ab concentration from this reaction	

2.1.4 Full equation from day 3

For $t \geq 3$, B-cells in $B^*(x, t)$ begin migrating to the light zone and the full model begins described by Equations (2.4)–(2.8).

$$\frac{\partial B^*}{\partial t}(x, t) = \rho B^*(x, t) - \mu B^*(x, t), \quad (2.4)$$

$$\frac{\partial B_0}{\partial t}(x, t) = p_r E(x, t) - \eta B_0(x, t), \quad (2.5)$$

$$\frac{\partial C}{\partial t}(x, t) = \mu B^*(x, t) + \eta B_0(x, t) - \lambda p_s(x, t) C(x, t) - \delta C(x, t), \quad (2.6)$$

$$\frac{\partial O}{\partial t}(x, t) = (1 - p_r) E(x, t), \quad (2.7)$$

$$\frac{\partial [Ab]}{\partial t}(x, t) = cO(x, t - T_{\text{diff}}) - g[Ab](x, t). \quad (2.8)$$

2.2 Epitope Masking via Antibody Feedback

We build $M(t)$ so that it gives the proportion of antigen binding sites that are bound by antibodies. If $[Ag]$ is the concentration of unbound antigen binding sites, $[Ab]$ is the concentration of antibodies, and $[IC]$ is the concentration of immune complexes (antibody-bound antigen), then the proportion that is bound or masked is given by

$$\theta = \frac{[IC]}{[Ag] + [IC]}. \quad (2.9)$$

Assuming equilibrium, basic chemical kinetics gives for a single affinity antibody:

$$[IC] = K_a [Ag][Ab]. \quad (2.10)$$

This is true for any affinity. Adding up all possibilities (including background antibody not

produced from this reaction, $[Bg](x)$ yields

$$[IC] = \int_0^1 K_a(x)[Ag] \left([Ab](x) + [Bg](x) \right) dx = [Ag] \int_0^1 K_a(x) \left([Ab](x) + [Bg](x) \right) dx. \quad (2.11)$$

Plugging (2.11) into (2.9) gives

$$M(t) = \frac{\int_0^1 K_a(x) \left([Bg](x) + [Ab](x, t) \right) dx}{1 + \int_0^1 K_a(x) \left([Bg](x) + [Ab](x, t) \right) dx}. \quad (2.12)$$

In addition to this describing masking, it is a general calculation that applies to the antigen elsewhere in the body as well. We use the same Equation (2.12) to describe what proportion of antigen in the body is being disabled by the immune response. This gives a natural and physically meaningful measure of the effectiveness of the immune response.

2.3 T-cell Selection Strength

We construct a function $p_s(x, t) \in [0, 1]$ that measures the strength of the T-cell signal a light zone B-cell of affinity x at time t (i.e., cells described by $C(x, t)$) receives. It is used directly as a scale on the rate of selection for such cells. Further, this value will determine the number of divisions that the cell is destined to undergo described in Section 2.5.

The determinates of T-cell signal strength that we consider in this construction are:

1. The proportion of FDC-bound Ag that is not masked.
2. The ability of a B-cell to bind with and capture Ag via its BCR.
3. Competition between B-cells attempting to interact with a limited supply of T-cells.
4. Optionally, we include *positive* antibody feedback.

Item 1 is given by $1 - M(t)$. Item 2 simply implies that higher affinity B-cells should receive a stronger T-cell signal, and for item 3 we will use the average affinity of light zone

B-cells defined by Equation (2.13).

$$x_{avg}(t) = \frac{\int_0^1 xC(x, t) dx}{\int_0^1 C(x, t) dx}. \quad (2.13)$$

We define a function $G(x, t)$ given in Equation (2.14) that for a fixed time t is made up of two line segments, each segment with one of their endpoint at $(x_{avg}(t), 1/2)$, and the other endpoints at $(0, 0)$ and $(1, 1)$.

$$G(x, t) = \begin{cases} \frac{x}{2x_{avg}(t)} & x \leq x_{avg}(t) \\ \frac{x-x_{avg}(t)}{2(1-x_{avg}(t))} + \frac{1}{2} & x > x_{avg}(t). \end{cases} \quad (2.14)$$

Finally, item 4: *Positive* antibody feedback. This can occur because the mechanism for presenting antigen on FDCs to make them available for collection by B-cells involves immune complexes (antigen bound to antibody). All numerical experiments are conducted both with and without this mechanism.

Positive feedback depends on the proportion of antigen found as immune complexes, i.e., $M(t)$ as defined in Section 2.2. Let z be the proportion of antigen binding sites on the FDC that we expect to be filled with immune complexes rather than antigen without antibody. One of the functions of macrophages is to transport immune complex in the body quickly to lymph node follicles and onto FDCs. Because of this, we expect that these FDC binding sites will be filled far more efficiently and with more robust immune complex presentation as $M(t)$ increases. We introduce a shape parameter s to control for how quickly this occurs giving Equation (2.15).

$$H(M) = \frac{zM}{s + M} + (1 - z). \quad (2.15)$$

Putting all of this together, we arrive at Equation (2.16) defining $p_s(x, t)$. The denominator is necessary so that $p_s \in [0, 1]$. For a light zone B-cell of affinity x at time t , this

function is a scale on the rate of T-cell selection.

$$p_s(x, t) = \frac{(1 - M(t))G(x, t)H(M(t))}{\max_{0 \leq y \leq 1} \{(1 - y)H(y)\}}. \quad (2.16)$$

$p_s(x, t)$ will also be used in determining the number of divisions a B-cell should undergo, covered in the following section.

2.4 Antibody Production

Some proportion of the output B-cells $O(x, t)$ differentiate to *plasma B-cells* which secrete antibody at a high rate. It takes time to differentiate and begin producing antibodies, so we incorporate a delay of $T_{\text{diff}} = 1$ day in Equation (2.8) from output to antibody production [16, 5, 17].

2.5 Number of Divisions

The number of divisions a B-cell is selected to undergo is determined by its cMyc expression. The cMyc expression of newly selected B-cells—and therefore the number of divisions—is directly proportional to the strength of the received T-cell signal. The number of divisions has been experimentally observed to be between 1 and 6. [7]

Equation (2.16) defining $p_s(x, t)$ is a measure of the strength of the T-cell signal. Following [17] we use a (shifted) Hill function to represent this. We map $p_s(x, t)$ from $[0, 1] \rightarrow [1, 7]$ using this Hill function, then take the floor (treating 7 as a special case) to obtain an integer number of divisions, 1–6, resulting in Equation (2.17):

$$D(p_s) := \begin{cases} \left\lfloor 1 + 6(1 + h^n) \frac{p_s^n}{p_s^n + h^n} \right\rfloor & 0 \leq p_s < 1 \\ 6 & p_s = 1. \end{cases} \quad (2.17)$$

Here h and n are parameters of the Hill function. A graph of $D(p_s)$ using the parameters from our model, $h = 0.8$ and $n = 3$, is shown in Figure 2.2.

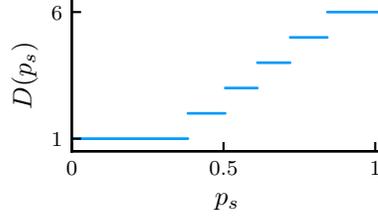


Figure 2.2: The number of divisions resulting from T-cell signal strength p_s .

2.6 Mutation

Recycled dark zone B-cells undergo somatic hypermutation (SHM) [3, 24] which greatly increases mutation rate in the antibody encoding variable region to roughly one point mutation per division [4]. Following [4, 25, 14], we estimate that the probability of a deleterious mutation is $m_d = 9/32$ and the probability of a silent mutation is $m_s = 17/32$. If the mutation is neither silent nor deleterious, we assume that the affinity x of daughter cells will be normally distributed around the parent's affinity y (truncated to $[0, 1]$) with a standard deviation σ that becomes a parameter of the model.

Given this, we define a continuous space, discrete time Markov process with kernel given in Equation (2.18). From this, we can compute the affinity distributions of further generations by Equation (2.19).

$$f_1(x|y) = \begin{cases} m_d \delta(x) + m_s \delta(x - y) + (1 - m_d - m_s) \mathcal{N}(\mu = y, \sigma; 0 \leq x \leq 1) & y \neq 0 \\ \delta(x) & y = 0, \end{cases} \quad (2.18)$$

$$f_{i+j}(x|y) = \int_0^1 f_i(x|z) f_j(z|y) dz. \quad (2.19)$$

The special case for $y = 0$ in Equation (2.18) treats affinity $x = 0$ as an absorbing boundary in the mutation process. Once a deleterious mutation is accumulated, it is astronomically unlikely that a subsequent point mutation in a single GC reaction is going to

grant Ag-specificity again.

Given a distribution of B-cells $B_{\text{before}}(x)$, if the entire distribution undergoes a single round of division (double the size) and mutation (apply the Markov process defined by the kernel in Equation (2.18)), the resulting distribution is given by

$$B_{\text{after 1 division}}(x) = 2 \int_0^1 B_{\text{before}}(y) f_1(x|y) dy. \quad (2.20)$$

Similarly, given k rounds of division and mutation applied to $B_{\text{before}}(x)$, the resulting distribution is given by

$$B_{\text{after } k \text{ divisions}}(x) = 2^k \int_0^1 B_{\text{before}}(y) f_k(x|y) dy. \quad (2.21)$$

2.7 Flux of B-cells Completing Division, $E(x, t)$

Since we know how many divisions B-cells will undergo when they enter the DZ from Equation (2.17) and how they will be distributed after a division from Equation (2.18), we also know exactly what distribution will be entering the LZ after the divisions are complete. Because of this, explicitly tracking the B-cells in the DZ is unnecessary. Instead of tracking B-cells through the DZ, we derive the flux $E(x, t)$ at time t of B-cells of affinity x moving into the LZ from the DZ.

This is where a delay is used in this model, and the length of the delay depends on the number of divisions. In Equation (2.22) we define subsets $\mathcal{X}_i(t) \subset [0, 1]$ of affinity at time t on which the B-cells selected at time t will undergo i divisions for $i = 1, \dots, 6$:

$$\mathcal{X}_i(t) = \left\{ x \in [0, 1] : D(p_s(x, t)) = i \right\} \quad i = 1, \dots, 6. \quad (2.22)$$

The flux of cells out of the light zone of affinity x at time t , via inspecting Equa-

tion (2.6), is given by

$$\lambda p_s(x, t)C(x, t),$$

therefore, applying Equation (2.22) to this distribution to only consider those cells that will divide i times, we arrive at Equation (2.23) defining $E_i(x, t)$ the flux of B-cells completing division that were selected to divide i times. Note that the time iT_{div} is precisely the amount of time the divisions take. This is where the most important delay occurs in our model.

$$E_i(x, t + iT_{div}) = 2^i \lambda \int_{\mathcal{X}_i(t)} f_i(x|y)p_s(y, t)C(y, t) dy. \quad (2.23)$$

And thus we arrive at Equation (2.24) for $E(x, t)$:

$$E(x, t) = \sum_{i=1}^6 E_i(x, t). \quad (2.24)$$

Note that the model does not explicitly represent the distribution of actively dividing B-cells in the model. We calculate the number of actively dividing B-cells via Equation (2.26):

$$g_i(t) = \int_{\mathcal{X}_i(t)} \lambda p_s(x, t)C(x, t) dx, \quad (2.25)$$

$$B_{div}(t) = \sum_{i=1}^6 \int_0^{iT_{div}} e^{\rho\tau} g_i(t - \tau) d\tau. \quad (2.26)$$

Note that Equation (2.26) approximates by assuming continuous proliferation rather than discrete division events.

2.8 Assumptions and Numerical Considerations

2.8.1 Single epitope model

Many antigen are highly complex and have multiple epitopes available for antibody binding. Here we use a single epitope model, highly common in the literature [9, 12, 10, 14].

2.8.2 FDCs remain stocked

We assume that there is enough antigen that the FDCs stay fully stocked throughout the reaction. The restocking of antigen on FDCs is supported by laboratory experiment [5].

2.8.3 Both primary and secondary GC reactions

A primary GC reaction is one in which the immune system has no B or T-cell memory of the antigen. While most models are restricted to studying either primary or secondary immune responses, this model reproduces both by choosing appropriate initial conditions. In a primary response, the B-cells that seed the reaction have relatively low affinity to the epitope, while in the secondary response, at least some recruited seeder cells have higher affinity to the epitope. Further, in a secondary response, background antibody from previous immune responses is included.

2.8.4 Quadrature and DE solvers

For numerical computation of integrals, a fixed quadrature is chosen for affinity $x \in [0, 1]$. Results remain identical with any reasonable choice, and here we use Gauss-Lobatto quadrature. Using a fixed quadrature allows the main equations (2.4)–(2.8) to be easily represented as a basic DDE. E.g., if we choose 100 quadrature nodes then the 5 equations can be easily represented as a system of $100 \times 5 = 500$ equations.

We use the Julia language [26] package *DifferentialEquations.jl* [27] with the Tsitouras 5/4 Runge-Kutta method [28]. Numerically for these equations the Euler method is also acceptable and leads to indistinguishable results.

CHAPTER 3

RESULTS AND DISCUSSION

Here we study the *secondary immune response*, that is, the immune system has prior experience with the antigen. Some memory B-cells specific to the antigen from past immune responses will be recruited to participate in the germinal center reaction. It is reasonable to expect that starting with the best B-cells will lead to neutralizing antigen faster. However, recent results suggest that a majority of the B-cells seeding the GC in a secondary response are naive [11]. We devise numerical experiments to study this.

In all experiments, parameters are as given in Table 2.3 unless otherwise noted. We study four variations of the model given by the combination of positive antibody feedback either being included or not, and two possible initial seeder cell scenarios to investigate the role of naive seeders.

Throughout, when positive antibody feedback is included, we use parameters $s = 0.05$ and $z = 0.2$ as in Table 2.3. When positive antibody feedback is not included, $z = 0$, in which case s drops out of Equation (2.15) entirely.

3.1 Validation

We verify that the model provides reasonable predictions for basic phenomena that is observed in laboratory experiments and established models. For the first experiments, the initial condition for high affinity seeder cells will be given by

$$B^*(x, 0) = 3\delta(x - 0.5),$$

where all seeder cells have affinity $x = 0.5$, and for a mixture of experienced, high affinity seeder cells and naive seeder cells, we use

$$B^*(x, 0) = 1.5\delta(x - 0.5) + 1.5\delta(x - 0.1),$$

where half of the seeder cells have affinity $x = 0.1$ and half have affinity $x = 0.5$. We will make a more sophisticated choice in a later experiment, but the simplicity here is beneficial.

Since these experiments are for secondary immune responses, we choose background antibody given by

$$[Bg](x) = 10^{-7}\delta(x - 0.1). \quad (3.1)$$

Throughout Section 3.1, results obtained without positive antibody feedback are in blue and results with positive antibody feedback are in red. Results for the high affinity seeder cell scenario are given by dashed curves and results for the mixed affinity scenario (naive seeder cells are also recruited) are given by solid curves.

3.1.1 GC Size

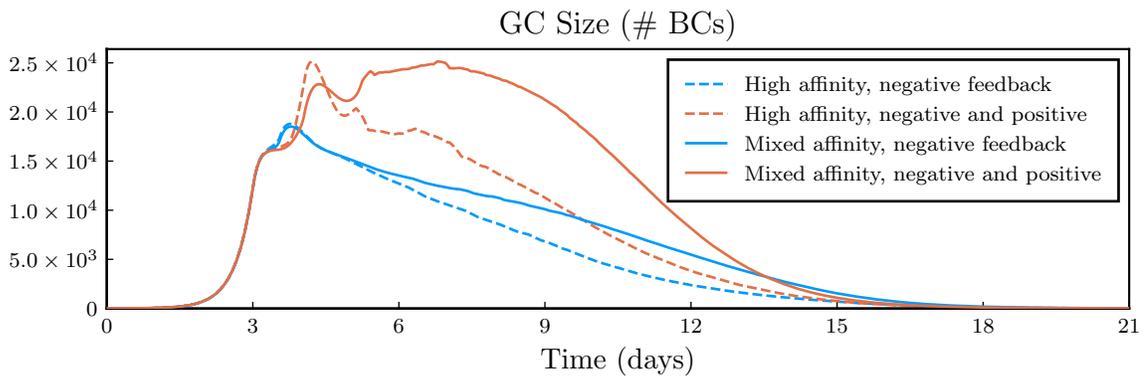


Figure 3.1: The number of B-cells comprising the GC over time.

A realistic GC size and profile over time [14, 21] is obtained for all four scenarios (Figure 3.1). This is an important first step towards validation of our model. For a given

choice of inclusion of positive antibody feedback, the inclusion of naive seeder cells in the mixed affinity scenario leads to a larger GC. Note that a larger GC does not necessarily imply more antigen neutralization.

3.1.2 Dividing cell ratio

When cells divide, they pass through a few distinct phases, one of which is S-phase. In the GC, a B-cell in the light zone that is selected to divide enters its first S-phase before reaching the dark zone. On average, the ratio of S-phase cells in the dark zone to those in the light zone is approximately 2 [7].

To calculate this ratio in our model, we make an approximation that assumes the makeup of selected cells is the same over a period of 1.25 days. Under this assumption, we compare cells that are instantaneously being selected (representing the light zone S-phase cells) with the number of cells that will be at exactly that the same point in the cell cycle in the dark zone.

The instantaneous rate of light zone cells being selected at time t is calculated from Equation (2.6), resulting in

$$\int_0^1 \lambda C(x, t) p_s(x, t) dx. \quad (3.2)$$

The dark zone calculation is more difficult. Cells that are selected to divide only once will never be found in S-phase in the dark zone. For cells selected to divide $i > 1$ times, assuming that all selection has been identical for the past 1.25 days, then the average factor increase in cell number is given by

$$\frac{\sum_{d=1}^{i-1} 2^d}{i-1}, \quad 2 \leq i \leq 6. \quad (3.3)$$

S-phase occurs before division, so this sum is stopped before the final division.

Finally, recall that $\mathcal{X}_i(t) \subset [0, 1]$ is the set of affinities $x \in [0, 1]$ such that $D(p_s(x, t)) = i$, i.e., it is the set describing which cells will divide i times. Using this, we have our

estimate of the ratio of dark zone S-phase cells to light zone S-phase cells:

$$\text{S-phase DZ:LZ ratio} = \frac{\sum_{i=2}^6 \frac{\sum_{d=1}^{i-1} 2^d}{i-1} \int x_i(t) \lambda C(x, t) p_s(x, t) dx}{\int_0^1 \lambda C(x, t) p_s(x, t) dx}. \quad (3.4)$$

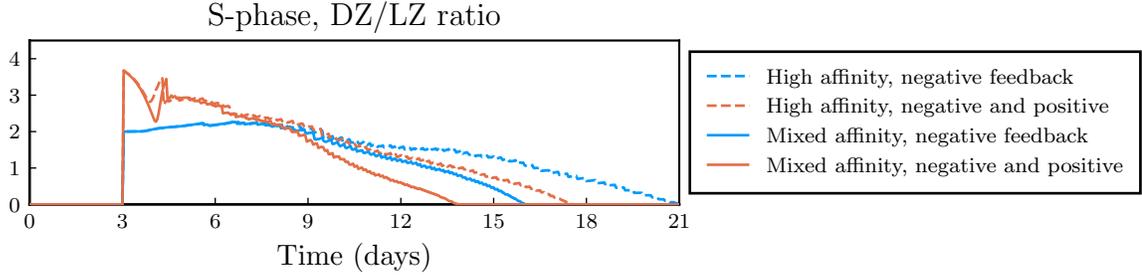


Figure 3.2: Dark zone to light zone ratio of S-phase B-cells.

For all scenarios, we reproduce a reasonable DZ/LZ ratio of S-phase B-cells [7] seen in Figure 3.2. All scenarios have a ratio of around 2 during their most productive times.

3.1.3 Rate of movement between zones and LZ/DZ size ratio

We also calculate the ratio of the number of B-cells in the light zone to the dark zone, and the intra-migration rates between in Figure 3.3. Again, all scenarios match experiment reasonably well [19].

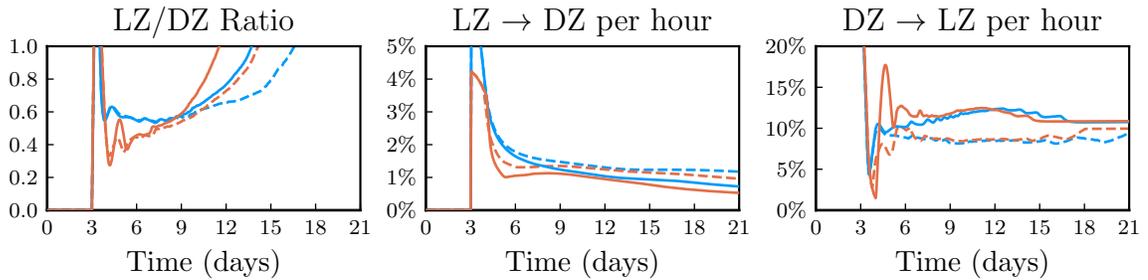


Figure 3.3: Light zone/dark zone dynamics.

3.2 Main results

The goal of the GC reaction is to manufacture a large supply of Ag-specific antibodies which bind to antigen throughout the body. The more binding that occurs, the more likely it is that antigen can be found and removed by other cells in the immune system. We've already established Equation (2.12) which gives $M(t)$ the proportion of epitopes (antigen binding sites for antibodies) are bound at any time during the reaction. Therefore, we use Equation (2.12) to measure the success of an immune response.

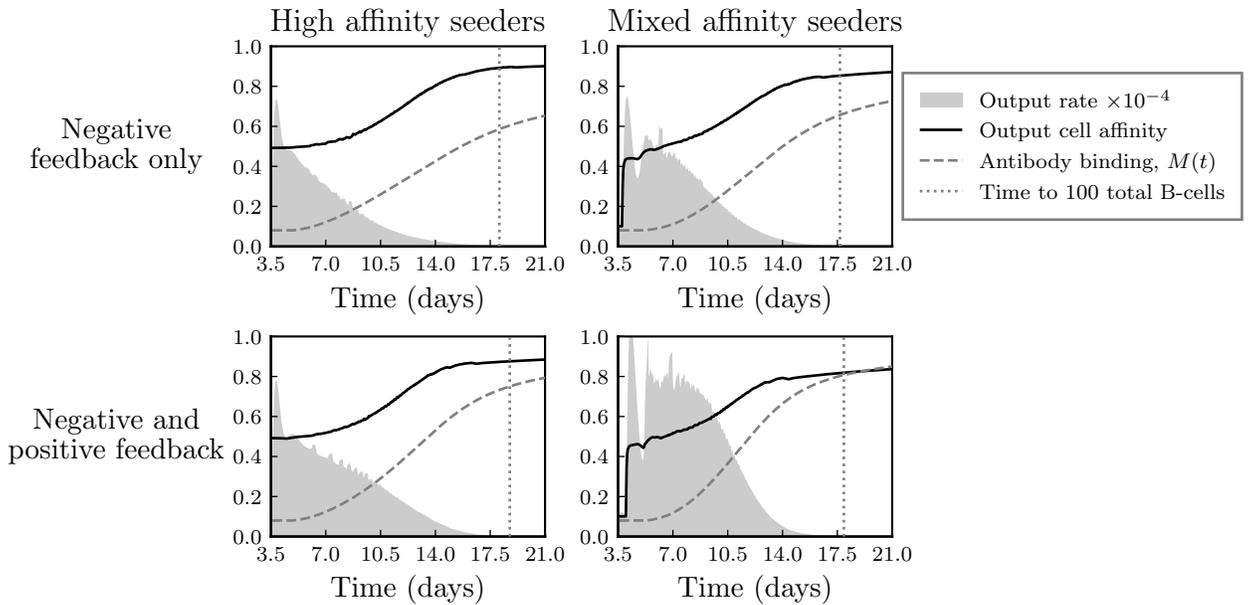


Figure 3.4: Inclusion of naive seeders and the inclusion of positive antibody feedback both lead to improved antigen neutralization.

In Figure 3.4 we find that naive seeders and positive antibody feedback have little effect on the length of time the reaction lasts (measured as the time it takes for the GC to drop to 100 B-cells). Average affinity of output cells is slightly lower when naive seeders are included. This is countered by the output rate being higher later in the reaction when there has been more time to reach higher affinities.

The result shows that $M(t)$ is higher when naive cells are included. Further, inclusion of positive antibody feedback is noticeably effective at producing a stronger response in

terms of increased $M(t)$.

The result continues to hold when more realistic initial seeding is used. There are many distinct clonal lineages found in a typical GC [11]. We draw 50 seeder cells from a distribution for the following experiments. While the current work does not attempt to create a full model of the seeding process itself, using a more diverse initial condition serves to confirm both the model’s generality and the results thereof.

High affinity seeder cells are given by a normal distribution with $\mu = 0.5$, $\sigma = 0.05$ truncated to $[0, 1]$. Naive seeder cells are given by a normal distribution with $\mu = 0$, $\sigma = 0.05$ truncated to $[0, 1]$. The mixed distribution weights its two components equally. See Figure 3.5.

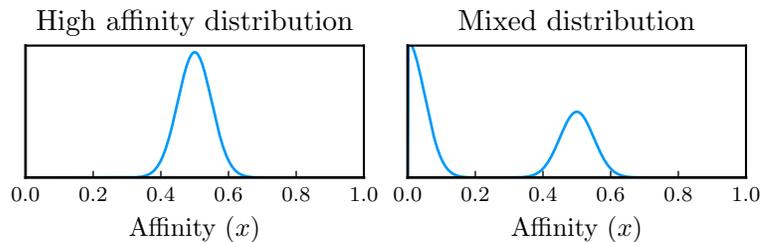


Figure 3.5: Seeder cell distribution

An immune response typically consists of many thousands of GC reactions. Since we are drawing seeder cells from a distribution, the initial condition varies from one run to the next. To see a more accurate picture of reality, we average over 50 runs.

The resulting output rate and light zone B-cell distribution can be seen in the contour plots of Figure 3.6. It is again clear that including naive seeder cells vastly improves the resulting affinity maturation and gives a more robust GC reaction.

3.3 Discussion

Secondary immune responses have the advantage of memory B and T-cells that have prior experience with the antigen, while recent research [11] has found that in fact many naive B-cells are recruited to participate in secondary immune responses. There are two (not nec-

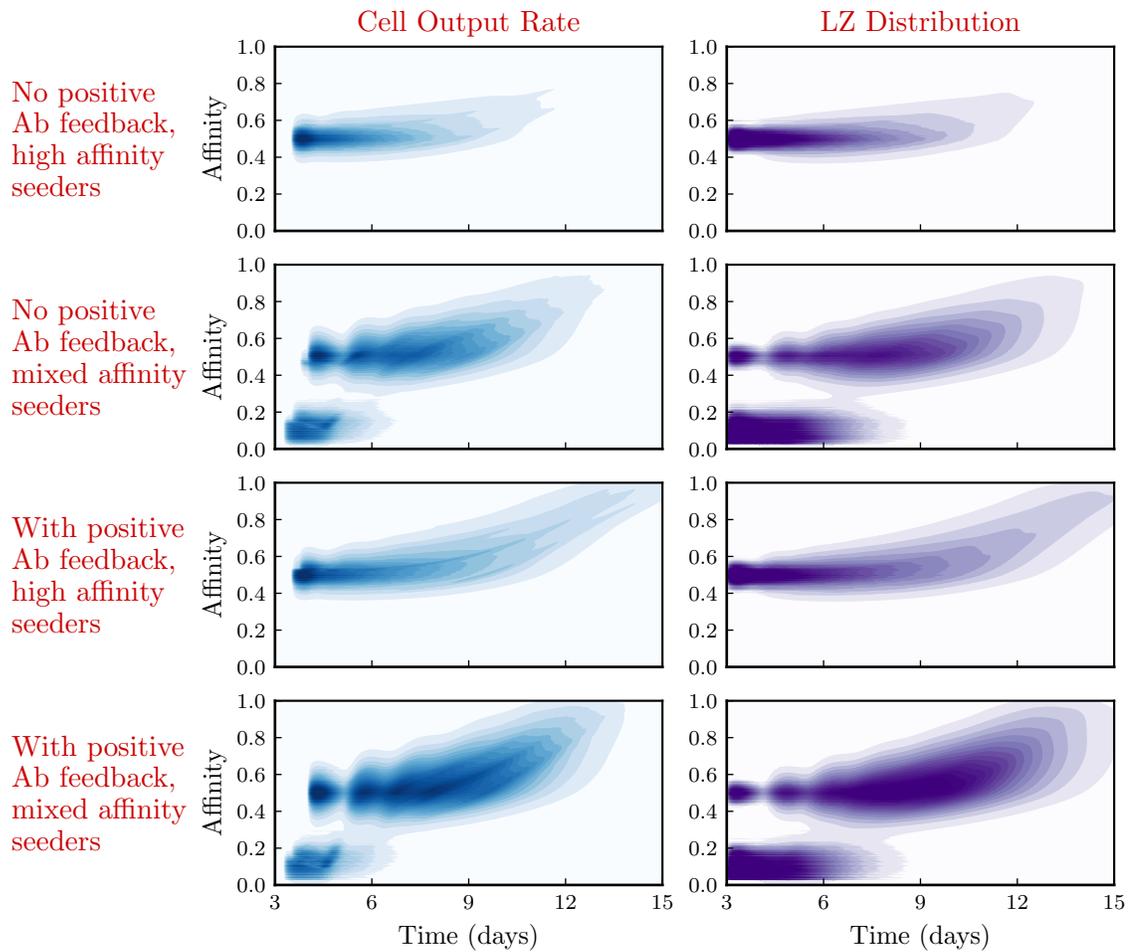


Figure 3.6: Output rate and LZ distribution with realistic seeding. Darker implies a higher concentration. The color scale is identical for each experiment so that the contour plots can be compared.

essarily mutually exclusive) possible explanations for this: 1) More efficient recruitment of experienced cells is not evolutionarily attainable, and 2) It is in fact advantageous to recruit naive cells.

We have shown that option 2 is true, though not to the exclusion of option 1. This appears to be due to a slower start when naive B-cells are recruited, and therefore later peak in GC size. The later peak allows more time for the previously experienced B-cells to mature to higher affinity (Figure 3.4).

A further contributor is that around days 4–7, experienced B-cells do not have as many other experienced cells to compete with for T-cell signal (Figure 3.6). Having so much of the available space taken up by B-cells that simply are not expressing enough pMHC to be competitive improves the chances of the experienced cells and gives an extended productive period where the GC has the opportunity to take advantage of the affinity maturation that has occurred.

We have also seen that positive antibody feedback helps the reaction, as would be expected. Biologically, we proposed positive antibody feedback because having higher affinity antibody available should lead to more stable immune-complexes available on FDCs for B-cells to collect. In our model, we see that this leads to a more efficacious immune response.

Mathematically, this work represents a new avenue to study affinity maturation at a lower computational cost. This allows for future work to model many interacting GCs to form a more complete picture of how the body mounts an immune response on a large scale.

Part II

Surface Surveillance from Minimum Length Paths

CHAPTER 4

INTRODUCTION AND BACKGROUND: PATH OPTIMIZATION

4.1 Previous Work and Literature on Path Planning

Path and route planning has been a topic of mathematical interest for centuries. In the 1750s, Euler studied the Knight’s Tour problem: Find a sequence of knight moves on a chess board such that every square is visited exactly once. The Knight’s Tour is an early example of the problem of finding a Hamiltonian path on a graph where the goal is to find a path on a graph that visits each vertex exactly once [29].

The most famous discrete path planning problem is the travelling salesman problem: Given a complete weighted graph, find the lowest weight cycle that visits every vertex exactly once [30]. Equivalently, the name “traveling salesman” derives from the following description: Given a list of cities and some known cost associated with traveling between the cities, find the lowest cost route to visit all cities once. The travelling salesman problem remains NP-hard even when restricted to general grids [31].

A significant amount of work has been done in continuous settings as well which we discuss for the remainder of this section [32, 33, 34, 35]. The art gallery problem [32] poses the problem of finding the fewest number of stationary guards or security cameras necessary to fully survey an art gallery. Viewing distance is assumed to be unlimited. The goal is to survey the entire environment, not just the walls or surfaces. Throughout the literature, environment surveillance is a much more common problem to address than surface surveillance.

The art gallery is NP-hard even for polygonal environments in \mathbb{R}^2 . In \mathbb{R}^3 there is an additional subtlety: In 2-dimensions guard locations can be chosen at vertices and full environment surveillance is still possible, however in 3-dimensions, simple (but non-convex)

polyhedra can be constructed such that even with a camera at every vertex, there will be interior regions that can not be seen [32, chapter 10].

The watchman route problem [33] studies paths that a watchman could traverse so that an entire environment in \mathbb{R}^2 is surveyed. As in the art gallery problem, the watchman is assumed to have unlimited viewing distance.

For polygonal environments, if the environment is simple (i.e., has no holes), then there is a polynomial time algorithm to find exact solutions [36]. For every convex corner, the path must enter the cone created by that corner and the two edges meeting it. If this is done for every corner, full surveillance will be achieved. In the majority of cases, the shortest path is unique, but in a few cases there are an infinite number of equally short paths [37]. For polygonal environments with holes an exact solution is NP-hard, but a solution can be approximated in polynomial time [38].

The literature discussed above assumes that the visibility distance is infinite. The d -sweeper problem is identical to the watchman route problem except that the watchman has a limited visibility distance d [34]. The goal is to find the shortest route that the entire polygonal environment is surveyed by the watchman with limited visibility. The name “sweeper” comes from thinking of the watchman surrounded by a disc of radius d that must sweep over the entire area.

Similarly, the d -watchman problem is identical to d -sweeper except that only the boundary of the environment needs to be surveyed. This is a subset of the problem space for which the proposed surface surveillance algorithm of this thesis applies. There are polynomial time approximate algorithms for both the d -sweeper and d -watchman problems in simple (i.e., there are no holes) polygonal environments in \mathbb{R}^2 [34].

Algorithms for finding safari routes [39] and zookeeper routes [40] are necessary for the d -watchman algorithm [34]. A safari route is one that intersects a given collection of subregions in an environment. A zookeeper route intersects the boundaries of a given collection of subregions, but does not enter the interior of those regions. Safari routes

in particular are important: A d -watchman or d -sweeper route must see all corners of a polygonal environment. These routes must visit regions within a distance d of each corner, which means these routes must be safari routes for those regions.

The path optimization with limited sensing ability (POLSA) algorithm uses a level-set framework to define the environment in \mathbb{R}^2 or \mathbb{R}^3 [35]. The POLSA algorithm finds the shortest full environment surveillance paths. A path is represented by a list of points that are connected via shortest paths, and paths are evolved by moving their representative points.

To determine the shortest path between two points, POLSA employs the evolving junctions over boundaries (E-JOB) algorithm [41] which uses an initial value problem of ODEs. The level-set framework is used to define a gradient flow that shortens the path while intermittent diffusion [41] provides random perturbations and allows for global optima to be found.

Visibility in the level-set framework is determined using PDE based visibility dynamics [42]. This combined with gradient flow applied to the list of points representing the path is used to evolve paths to full surveillance. To shorten the paths, a local optimum can be found through alternating gradient descent on the length and gradient ascent on visibility to maintain full surveillance. POLSA finds global optima via global optimization by intermittent diffusion [43].

4.2 Literature on Surface Surveillance

There are limited references studying surface surveillance with limited visibility. The only surface surveillance problem discussed above was the d -watchman route problem for which there exists an approximate algorithm that applies to simple polygons in \mathbb{R}^2 [34].

It should be noted when visibility distance is unlimited in \mathbb{R}^2 as in the original watchman route problem, surface and full environment surveillance routes are equivalent. A difference only arises under limited visibility distance [34]. Thus, in \mathbb{R}^2 with unlimited

visibility, there has implicitly been significant work [32, 33, 36, 37, 38].

Surface surveillance of the unit sphere (without a visibility distance restriction) in \mathbb{R}^3 has received substantial attention. This is known as the asteroid surveillance problem [44], the sphere inspection problem [45], or the inradius problem in \mathbb{R}^3 [46]. For closed curves, i.e., routes that start and end at the same point, this problem was recently solved analytically by a curve of length 4π [47]. For open curves, the shortest sphere surveillance path remains unsolved.

Definition 4.2.1 (Inradius). The *inradius* of a curve is the radius of the largest sphere contained in that curve's convex hull.

The inradius problem is to find the shortest curve of inradius 1. If it happens that this curve does not intersect the unit sphere contained in its convex hull, then this is equivalent to sphere surveillance [45]. It isn't *a priori* guaranteed that a solution to the inradius problem will not intersect the sphere, but the solution for closed curves [47] and the best conjecture for open curves [45] do not, and so they are also sphere surveillance paths.

The history of sphere surveillance begins in 1956 with a proposed problem often called the lost in a forest problem: If you are a distance 1 into a forest (modeled as a half-plane in \mathbb{R}^2) but don't know what direction leads out of the forest, find the route that minimizes the maximum possible time to exit the forest [48]. A solution with a proof outline was proposed the following year [49] and a rigorous proof was published in 1980 [50].

The forest problem is equivalent to a modification of the inradius problem in \mathbb{R}^2 . It can be restated as: Find the shortest path in \mathbb{R}^2 whose convex hull contains a unit disk such that the path intersects the center of that disk. Since the path must pass through the disk, it is not the same as a surface surveillance problem. Removing the section of path inside the disk leaves us with a path that does solve a constrained surface surveillance problem: Find the shortest surface surveillance path that starts on the surface of the unit disk but doesn't intersect the interior of the disk [45, 51]. The latter interpretation can be thought of as a drone that starts its surveillance from the ground before taking flight.

These problems and other variations have been solved in \mathbb{R}^2 [45]. Analytic solutions to the analogous sphere surveillance problems in \mathbb{R}^3 have been more elusive. Only sphere surveillance via a closed path has been solved [47]. There is a conjecture for the open path [45].

Section 4.3 provides necessary definitions and other details of the precise setup of the surface surveillance problem. In Chapter 5, the SSMILE algorithm is proposed as a general method for computing optimal surveillance paths in general environments. Chapter 6 uses SSMILE to study the closed path solution and open path conjecture to the inradius problem. Chapter 7 presents a potential starting point for analytical study of surface surveillance in a 2-dimensional environment.

4.3 Mathematical Details

In this work, our focus is the study of path optimization with full surface surveillance. A *surface* refers to a 1-dimensional subspace of \mathbb{R}^2 or a 2-dimensional subspace of \mathbb{R}^3 . For example, the environment could be a 2-dimensional floorplan with surfaces being the 1-dimensional walls in that floorplan. The objective is to find the shortest path such that the entirety of the walls can be seen when the path is traversed. An example in \mathbb{R}^3 that we will discuss in detail is the surveillance of the surface of a sphere.

A surface could be surveyed from the inside or outside determined by the definition of the environment, e.g., consider a security guard patrolling either the interior of a building or the exterior. Broadly, a surface is surveyed by a path if each point on the surface can be seen from some point along the path. In addition to the security guard analogy which works well when considering 2d maps, in 3d we can think of the flight path of a drone with a mounted camera.

This work explores various constraints on visibility. In all cases, visibility requires that the line of sight must be unobstructed. Other limitations on visibility that we consider are limited visibility distance and a requirement that surfaces be seen from angles that are

not too extreme.

4.3.1 Environment Definition and Surface Discretization

We begin by defining the possible environments which can be considered for surface surveillance from paths.

Definition 4.3.1 (environment). An *environment* is a subset $X \subseteq \mathbb{R}^n$ ($n = 2, 3$) such that:

1. X is path-connected,
2. X is equal to the closure of its interior, and
3. the boundary of X is oriented and has finite area when $n = 3$ and finite length when $n = 2$.

Definition 4.3.2 (path-connected). X is called *path-connected* if for any $x, y \in X$, there exists a continuous function $\gamma : [0, 1] \rightarrow X$ such that $\gamma(0) = x$ and $\gamma(1) = y$. In other words, there exists a path connecting any two points.

The first requirement of Definition 4.3.1 is necessary because there is no way for a single path to survey more than one path-connected component. The second requirement enforces that all boundary points are accessible from the interior of X . The final requirement removes some pathological and physically unrealistic possibilities.

We denote the boundary of X by ∂X and call this the surface. The subset to be surveyed will be denoted by $S \subseteq \partial X$ throughout and referred to as the *target surveillance surface*. Frequently we set $S = \partial X$ to survey all surfaces, but the proposed algorithm in Chapter 5 applies regardless.

We discretize the boundary of the environment ∂X by *triangulation* [52] (see Definition 4.3.3). When S is a strict subset of ∂X , we require that S be a sub-triangulation in the sense that all elements of the triangulation of S are also elements of the triangulation of ∂X .

Definition 4.3.3 (triangulation). A *triangulation* $\mathcal{T}_{\partial X}$ of the boundary of an environment ∂X is a collection of elements which are oriented line segments in \mathbb{R}^2 or oriented triangles in \mathbb{R}^3 with normal vector pointing out of X such that

$$\bigcup_{e \in \mathcal{T}_{\partial X}} e = \partial X,$$

and the intersection of two elements is either empty, a single vertex, or in \mathbb{R}^3 a complete edge.

For polyhedral environments, this allows our computational description of ∂X to be exact. The restrictions on environments in Definition 4.3.1 allow for triangulation to provide a good approximation. Conversely, any triangulation with a finite number of elements and consistent outward pointing normals defines an environment.

Using this discretization framework provides many benefits:

- Since we require elements in surface descriptions to be oriented, it is easy to test if a point is in X . Follow a ray in any direction until it intersects with an element, and then check if the dot product of this ray with the normal of the element is positive or negative.
- It's easy to refine triangulations. In \mathbb{R}^2 , one may break any line segment into as many pieces as desired. In \mathbb{R}^3 , one strategy is to add vertices at the midpoint of each edge and create four triangles similar to the original using the 6 vertices, a method which has the advantage of maintaining Delaunay triangulations [53].
- Using triangulations allows the use of a variety of common tools used in animated film and video game development such as Blender [54].

The algorithm we propose in Chapter 5 applies to environments defined by triangulation. In Chapter 6 we present a modification to apply the algorithm to a true sphere, not a triangulation.

4.3.2 Paths

We define precisely what we mean by a path in X :

Definition 4.3.4. The *space of all (rectifiable) paths* in X is given by

$$\Omega = \{\text{Image of } \gamma \mid \gamma : [0, 1] \rightarrow X \text{ is continuous}\}.$$

The Euclidean length of a path $\Gamma \in \Omega$ is denoted by $L(\Gamma)$.

The space of all rectifiable paths Ω in an environment X is infinite dimensional and computationally infeasible to search within directly. We use a discretization developed for the POLSA algorithm [35] to approximate the space of all paths. In this framework, a discretized path is given by an ordered list of points in X which are connected by the shortest path from one point to the next.

Definition 4.3.5. Given a list of points $(x_1, \dots, x_N) \in X^N$, construct a *discretized path* starting at x_1 that connects x_i to x_{i+1} for $i = 1, \dots, N - 1$ via the shortest path in X . We denote this path $\Gamma[x_1, \dots, x_N]$. The set of all paths defined by N points is denoted $\Omega_N \subset \Omega$ and we call the defining points of such a path *junction points*.

From Definition 4.3.5 we see that a path $\Gamma \in \Omega_N$ is identified by a point in $X^N \subset \mathbb{R}^{N \times n}$ where $n = 2, 3$ is the dimension of the environment. Therefore this discretization of paths provides a finite dimensional search space.

Recall that we required an environment X to be path connected, so the construction in Definition 4.3.5 is possible. The construction of a path from N junction points is not necessarily made up of $N - 1$ line segments because there could be an obstacle between two points. In general, the shortest path between two points will consist of a union of line segments and geodesic curves along ∂X . Within the environment framework defined in Section 4.3.1, since we use a triangulation, these geodesics will be a union of (possibly very many) line segments.

It's possible that there are multiple shortest paths connecting two junction points. When there are multiple shortest paths between x_i and x_{i+1} , a random small perturbation of either point results almost surely (in a probabilistic sense) in a unique shortest path between the two points. Because of this, the ambiguity causes no issues.

It is worth noting that $\Omega_N \subset \Omega_{N+1}$. Indeed, for points $x_1, \dots, x_N \in X$, we have that $\Gamma[x_1, \dots, x_N] \in \Omega_N$ defines the same path as $\Gamma[x_1, x_1, \dots, x_N] \in \Omega_{N+1}$.

4.3.3 Visibility of Surfaces

Common to all visibility problems discussed in this chapter, visibility requires a line of sight. Further restrictions we place on visibility are given by a maximum visibility distance $d \in (0, \infty]$ and a maximum visibility angle $\alpha \in (0, \pi/2]$. Choosing $d = \infty$ and $\alpha = \pi/2$ removes the distance and angle restrictions leaving only only line of sight for determining visibility.

Definition 4.3.6. A point $x \in X$ in the environment has a *line of sight* to a point $s \in S$ if the line segment connecting x and s is contained entirely in X .

Definition 4.3.7. A point $s \in S$ on the target surveillance surface is *visible* from a point $x \in X$ with maximum visibility distance d and maximum visibility angle α if:

- $\|s - x\| \leq d$,
- $\frac{(s - x) \cdot n}{\|s - x\|} \geq \cos \alpha$ where n is the outward pointing normal of ∂X at s ,
- and x has a line of sight to s .

We say a point $s \in S$ is visible from a path Γ if it is visible from any point $x \in \Gamma$, and S is visible from Γ if every $s \in S$ is visible from Γ .

An illuminating example is the unit square. Let the environment be $X = [0, 1] \times [0, 1]$ with target surveillance surface the entire boundary $S = \partial X$. If $d = \infty$ and $\alpha = \pi/2$ (i.e.,

only consider line of sight), then every path will survey all of S . In fact, this is the case for any convex X with only a line of sight constraint on visibility.

If $d = 1/4$, then there is no point in X from which all surfaces are visible, so the optimal paths will necessarily have length greater than 0 and the solution is not obvious. Alternatively, if $d = \infty$ and $\alpha = \pi/6$, again the optimal paths are not obvious. So even in simple environments, these added restrictions to visibility provide opportunities for interesting problems.

CHAPTER 5

SURFACE SURVEILLANCE FROM MINIMUM LENGTH PATH ALGORITHM

We propose a method to find globally optimal paths that survey surfaces in an environment that accounts for constraints on visibility distance and visibility angle. We refer to this as the *Surface Surveillance from Minimum Length Paths* (SSMILE) algorithm. SSMILE is developed to apply to the following problem: Given an environment $X \subset \mathbb{R}^n$, $n = 2, 3$ (see Definition 4.3.1) with a target surveillance surface $S \subseteq \partial X$, constraints for visibility distance $d \in (0, \infty]$ and angle $\alpha \in (0, \pi/2]$ (see Definition 4.3.7), find the shortest paths in X that survey all of S .

For computational feasibility we consider a discretization of both paths and environments. The choice we make for the proposed algorithm gives the following problem setup:

Given

- a triangulation $\mathcal{T}_{\partial X}$ defining the boundary ∂X of an environment X in \mathbb{R}^n , $n = 2, 3$ (see Definition 4.3.3),
- a target surveillance surface $S \subseteq \partial X$ defined by a sub-triangulation $\mathcal{T}_S \subseteq \mathcal{T}_{\partial X}$,
- constraints for visibility distance $d \in (0, \infty]$ and angle $\alpha \in (0, \pi/2]$, and
- a number of junction points $N \in \mathbb{N}$ to define the discretized path space Ω_N (see Definition 4.3.5),

find the shortest paths in Ω_N that fully survey S .

Section 5.1 details constructions, algorithms, and optimization techniques relevant to the main algorithm starting with Section 5.1.1 which discusses the environment triangulation. In Section 5.1.2 we show how to construct full paths from a list of junction points. Section 5.1.3 introduces a test for determining visibility of elements of a triangulation. Following this, in Section 5.1.4 we develop a gradient based algorithm to smoothly transform

a path that doesn't survey all of S to one that does, which we call projection to full surveillance. Section 5.1.5 discusses global optimization via intermittent diffusion [43] which we use in the SSMILE algorithm. The SSMILE algorithm is presented in Section 5.2. In Section 5.3 we show the results of numerical experiments applying the SSMILE algorithm to various examples.

5.1 Constructions, Algorithms, and Optimization Techniques for SSMILE

5.1.1 The Environment and Surface to be Surveyed

The input that defines the environment for the SSMILE algorithm is a triangulation $\mathcal{T}_{\partial X}$ (see Definition 4.3.3) of the boundary of the environment. The elements of $\mathcal{T}_{\partial X}$ are oriented with consistent, outward pointing normal vectors, meaning that the surface ∂X defined by the triangulation is itself oriented. The volume or area enclosed by the triangulation, including the boundary, is what we call the environment X (see Definition 4.3.1).

Note that changing the sign on the normal vectors results in the same surface, but with the environment on the other side of the surface. If the triangulation given is the sides of the unit square in \mathbb{R}^2 , the direction of the normal vectors determines whether the environment is the interior or exterior of the unit square.

We distinguish between convex, concave, and saddle vertices of the triangulation $\mathcal{T}_{\partial X}$ in upcoming algorithms. The intuition for the naming we choose is that these corners are seen from inside the environment. Note that if the signs are switched on the normal vectors changing the environment, then convex vertices become concave and vice versa.

Definition 5.1.1. Given a triangulation $\mathcal{T}_{\partial X}$ that defines an environment X and v a vertex of $\mathcal{T}_{\partial X}$, in \mathbb{R}^2 , if the interior angle at v is, greater than π then v is called *convex*, and if it is less than π , v is called *concave*.

In \mathbb{R}^3 , consider all edges in $\mathcal{T}_{\partial X}$ that include v . If the interior angle of each edge is greater than or equal to π with at least one inequality strict, we call v *convex*. If all interior

angles are less than or equal to π with at least one inequality strict, we call v *concave*. If there is at least one interior angle around v strictly greater than π and one strictly less than π , we call v a *saddle vertex*.

The surface to be surveyed is given as a sub-triangulation $\mathcal{T}_S \subseteq \mathcal{T}_{\partial X}$ which defines a subset $S \subseteq \partial X$ of the boundary of the environment. In many applications it is reasonable for the surveillance surface to be the entire boundary, $\mathcal{T}_S = \mathcal{T}_{\partial X}$, such as in sphere surveillance. The flexibility to choose a subset of the boundary for surveillance allows there to be obstacles in the environment that are not of interest to the surveyor, but still must be taken into account for path planning and visibility.

The refinement of \mathcal{T}_S directly affects the accuracy of SSMILE. This is discussed in Section 5.1.3. If $S \neq \partial X$, then the refinement of the elements in $\mathcal{T}_{\partial X} \setminus \mathcal{T}_S$ does not affect accuracy. For computational efficiency it is advantageous for this part of the triangulation to be as coarse as possible.

5.1.2 Path Construction from Junction Points

In this section we discuss how to construct paths from junction points. Recall from Definition 4.3.5 that a path $\Gamma \in \Omega_N$ defined by N junction points $x_1, \dots, x_n \in X$ is created by connecting x_i to x_{i+1} via the shortest path in X . The focus of this section is on connecting two points in an environment X defined by a triangulation $\mathcal{T}_{\partial X}$ of its boundary ∂X .

In any environment, not limited to those defined by triangulation, the shortest path between two points is a union of zero or more geodesics along surfaces and line segments in the environment [41]. Since the environment is given by triangulation, surfaces are polyhedral and therefore the geodesics are also unions of line segments. This fact informs our method for determining the shortest path.

This will be reformulated as optimization on a weighted graph where we apply Dijkstra's algorithm. There is a subgraph that is needed both for finding the shortest path and for computations in Section 5.1.4. We call it the *line of sight graph* and denote it G_{LOS} . The

vertices of G_{LOS} are selected from points on the surface ∂X as described in the upcoming Definitions 5.1.2 and 5.1.3. Graph edges are included between any two vertices that have a line of sight in X with weight given by the distance between them.

The choice of vertices to include in G_{LOS} affects the accuracy of the shortest path algorithm that we introduce in this section. For environments in \mathbb{R}^2 we choose the vertices so that the calculation is analytically exact. The graph that accomplishes this is the following:

Definition 5.1.2. In \mathbb{R}^2 , the *line of sight graph* G_{LOS} is constructed with graph vertices given by all convex vertices of $\mathcal{T}_{\partial X}$, an edge connecting two vertices if the connecting line segment is in X , and edge weight given by the distance between vertices.

The choice of vertices to define G_{LOS} for environments in \mathbb{R}^3 is more complex, and a general analytic solution to the shortest path is NP-hard [41]. Frequently, geodesics on polyhedral surfaces will intersect the interior of edges and not pass through vertices of the triangulation. Because of this, to approximate the shortest path with higher accuracy we include points sampled from convex edges of $\mathcal{T}_{\partial X}$ as graph vertices, not just vertices of $\mathcal{T}_{\partial X}$. The density of sampled points on edges determines the accuracy of the approximated geodesics where paths are on surfaces.

Definition 5.1.3. In \mathbb{R}^3 , with sampling density parameter $\delta > 0$, the *line of sight graph* G_{LOS} is constructed with graph vertices given by:

- Every convex and saddle point vertex of $\mathcal{T}_{\partial X}$ and
- sample points at a density of at least 1 per length δ for each convex edge of $\mathcal{T}_{\partial X}$.

A graph edge connects two vertices if the connecting line segment is in X and the two vertices are not sampled from the same convex edge of the triangulation. The edge weight given by the distance between vertices.

In the preceding Definition 5.1.3, the reason that points sampled from the same edge of $\mathcal{T}_{\partial X}$ are never connected by a graph edge is that such an edge can never be in the optimal

path. These edges would not harm accuracy if included, but they cause the path construction to take longer with no benefit.

Using the line of sight graph G_{LOS} , we propose Algorithm 5.1 to find the shortest path between two points.

Algorithm 5.1 Shortest path between two points

- 1: Given an environment X defined by triangulation and two points $x, y \in X$
 - 2: If the line segment \overline{xy} is contained in X , return \overline{xy} immediately. Otherwise:
 - 3: Construct a graph G by adding to the line of sight graph G_{LOS} :
 - a. Include x and y as vertices
 - b. For each vertex v of G_{LOS} , if $\overline{xv} \subset X$ then include the edge $\{x, v\}$ with weight $\|x - v\|$, and if $\overline{yv} \subset X$ then include the edge $\{y, v\}$ with weight $\|y - v\|$
 - 4: Apply Dijkstra's algorithm to G and return the shortest path between x and y .
-

An example of the construction through step 3 of Algorithm 5.1 is seen in Figure 5.1.

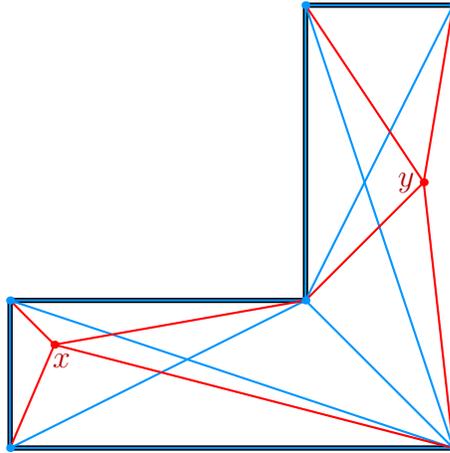


Figure 5.1: An example of G_{LOS} . The boundary of the environment is in black. G_{LOS} is in blue. The added edges are in red.

The graph G_{LOS} that was used in Algorithm 5.1 will be used again in Section 5.1.4 to define a gradient based method to project a path to full surveillance. In both uses, it will be needed repeatedly and should be precomputed.

5.1.3 Visibility Calculation

Recall from Definition 4.3.7 that for a point $x \in X$ in the environment to survey a point $s \in S$ on the surface, x must have a line of sight to s (the line segment \overline{xs} is entirely contained in X), it must be within the maximum viewing distance $d > 0$, and the angle between \overline{xs} and the normal vector n of the surface S at s must be less than the maximum viewing angle $\alpha \leq \pi/2$. A path Γ surveys the surface S if for every $s \in S$ there exists a point $x \in \Gamma$ that surveys s .

Using the triangulation, we discretize the definition of visibility of a point $s \in S$ from a point $x \in X$ by checking if an element $e \in \mathcal{T}_S$ of the triangulation is visible from x . This is accomplished by checking if each vertex of e is visible using the outward pointing normal vector n of the element. Note that these vertices are each a part of more than one element, but the normal vectors for these adjacent elements may differ. This visibility calculation is summarized in the following definition.

Definition 5.1.4. An element $e \in \mathcal{T}_S$ of a the triangulation of S with outward pointing normal vector n is *visible* from a point $x \in X$ with maximum visibility distance d and maximum visibility angle α if the following conditions hold:

1. $\frac{(x - v) \cdot n}{\|x - v\|} \geq \cos \alpha$ for each vertex v of e .
2. $\|x - v\| \leq d$ for each vertex v of e .
3. For each vertex v of e , the line segment \overline{xv} does not intersect any other element of the triangulation.

Condition 1 prevents the viewing angle from being too extreme. In experiments when this restriction is not required, set $\alpha = \pi/2$ which gives the simpler condition $(x - v) \cdot n \geq 0$. When $\alpha = \pi/2$, the calculation is called backface culling in computer graphics. Condition 2 imposes a limited visibility distance. When a limited distance is not required, set $d = \infty$. Condition 3 is satisfied when there are no obstructions to the line of sight.

When conditions 1 and 2 are met for an element, *every* point in the element is within the angle and distance requirements for visibility. This is because the element is a convex set and the vertices are the extreme points of this convex set. The maximum angle and distance must occur at an extreme point. Thus, conditions 1 and 2 are analytically sufficient.

When $X \subset \mathbb{R}^2$, condition 3 can also be guaranteed to imply the entire element has an unobstructed line of sight when we impose an extra requirement on the triangulation \mathcal{T}_S : The largest element length in \mathcal{T}_S should not exceed the smallest width of all holes in the environment. In other words, no element of \mathcal{T}_S should be so large that an obstacle can obstruct vision to the interior while both end points are still visible.

When $X \subset \mathbb{R}^3$, analytically there is no guarantee that condition 3 implies total visibility to the element. This is because, even when all vertices are seen, the corner of some obstruction can protrude into the line of sight to the interior of the element via the element's edge. This didn't occur in \mathbb{R}^2 because the element's vertices were also its entire boundary. In \mathbb{R}^3 , we accept that condition 3 is an approximation.

Using Definition 5.1.4, we propose Algorithm 5.2 to find the proportion of S that is visible from a path Γ in X . For $X \subset \mathbb{R}^2$, visibility from this algorithm implies that analytically the surface is visible from the path as in Definition 4.3.7. As discussed above, in \mathbb{R}^3 , this is an approximation and no such guarantee can be made.

Note that a path could survey a surface analytically as in Definition 4.3.7 and this algorithm could find less than complete surveillance. This underestimation of surveillance has two sources: First, for an element of the triangulation to be surveyed, then entire element must be surveyed from a single point on the path. This source of error is proportional to the maximum element size. Second, we sample points along the path to test visibility, which produces error proportional to the inverse of the sampling density.

Algorithm 5.2 Visible Surface Proportion from a Path

Require: A path Γ in X , a triangulation $\{e_1, \dots, e_n\}$ of S , and a number $k \gg 0$ of points to be sampled from Γ

- 1: Sample points x_1, \dots, x_k evenly by length from Γ and initialize $V \leftarrow 0$
 - 2: For each element e_i , if e_i is visible from any sample point x_j , then update:
 - 3:
$$V \leftarrow V + \frac{|e_i|}{|S|}$$
 - 4: **return** V
-

5.1.4 Projection to Full Surveillance

In this section we develop a method to modify paths so that they achieve full surveillance. This is used in the SSMILE algorithm to maintain full surveillance as paths are modified.

Traditional application of gradient ascent to the visibility calculation given in Algorithm 5.2 is not sufficient because frequently there is no local modification to a path that increases visibility. For example, see Figure 5.2 where any local modification to the path will still only see the target surveillance surface in the left arm. Instead we construct an algorithm by replacing conditions for visibility with penalty functions that have nonzero gradient for paths that don't fully survey S .

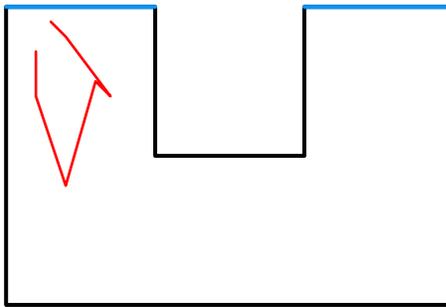


Figure 5.2: With no limits on visibility distance or angle, half of the target surveillance region (blue) is visible from the path (red). All local modifications to the path leave visibility unchanged.

We start by creating penalty functions for the visibility of a single point $s \in S$ on the surveillance surface from a single point $x \in X$ in the environment. Recall the three

conditions for visibility of $s \in S$ from $x \in X$ given in Definition 4.3.7:

1. $\|s - x\| \leq d$,
2. $\frac{(s-x) \cdot n}{\|s-x\|} \geq \cos \alpha$, and
3. x has a line of sight to s .

Here d is the maximum visibility distance and α is the maximum visibility angle from the normal n of ∂X at s . Conditions 1 and 2 will be replaced with penalty functions. The last condition will be satisfied by using the line of sight graph G_{LOS} (Definitions 5.1.2 and 5.1.3) to define the penalties.

Denote the penalty function for distance from $x \in X$ to $s \in S$ by $P_{\text{dist}}^*(x, s)$. Using Algorithm 5.1, find the shortest path from x to s . This path will be constructed of line segments with a sequence of end points we denote by $x = y_0, y_1, \dots, y_k = s$. Then we have

$$P_{\text{dist}}^*(x, s) = \sum_{i=1}^{k-1} \|y_i - y_{i-1}\| + \max\{0, \|y_{k-1} - y_k\| - d\}. \quad (5.1)$$

If the line of sight condition is met, the first sum is vacuously 0 because $k = 1$. Otherwise, the first sum evaluates to the distance x would need to travel to obtain a line of sight to s . The latter summand gives the remaining distance necessary to meet the limited visibility distance requirement.

For the angle penalty function, there are three angles to consider: The maximum visibility angle α , the angle between the normal vector n at $s \in S$ and the final leg $\overline{sy_{k-1}}$ of the path derived from G_{LOS} above given by

$$\cos \theta_1 = \frac{(s - y_{k-1}) \cdot n}{\|s - y_{k-1}\|}, \quad (5.2)$$

and the angle between n and \overline{sx} given by

$$\cos \theta_2 = \frac{(s - x) \cdot n}{\|s - x\|}. \quad (5.3)$$

For s to be visible, modification of x must eventually result in θ_2 being less than both α and θ_1 . I.e., we need $\max\{\cos \alpha, \cos \theta_1\} \leq \cos \theta_2$. This suggests the following penalty function for the angle:

$$P_{\text{angle}}^*(x, s) = \max\{0, \max\{\cos \alpha, \cos \theta_1\} - \cos \theta_2\}. \quad (5.4)$$

Adding (5.1) and (5.4), we have a penalty function that is almost everywhere smooth with a nonzero gradient when s is not visible from x :

$$P^*(x, s) = P_{\text{dist}}^*(x, s) + P_{\text{angle}}^*(x, s). \quad (5.5)$$

Recall that an element of the triangulation of the surface is defined to be surveyed when all vertices are visible (Definition 5.1.4). Given $e \in \mathcal{T}_S$, we extend Equation (5.5) to e by summing over the vertices of e :

$$P^*(x, e) = \sum_{s \in e} P^*(x, s), \quad (5.6)$$

where the normal vector n in Equations (5.2) and (5.3) is taken to be the normal of e .

To complete the construction of the penalty function, we extend Equation (5.6) to paths and full triangulations. To extend to triangulations, we take the sum of penalties for each triangulation element scaled by the size of the element. To extend to paths, for each element, the point along the path that provides the minimum penalty is used.

Definition 5.1.5. Given a path Γ and a triangulation \mathcal{T}_S of the surveillance region of the surface, the *visibility penalty function* is given by

$$P(\Gamma, \mathcal{T}_S) = \sum_{e \in \mathcal{T}_S} |e| \min_{x \in \Gamma} P^*(x, e). \quad (5.7)$$

With equation (5.7), we now present the path projection algorithm:

Algorithm 5.3 Project Path to Full Surveillance

Require: A path $\Gamma[\mathbf{x}]$ in X where $\mathbf{x} \in X^m$ is a list of junction points, a triangulation \mathcal{T}_S of S , and a number $k \gg 0$ of points to be sampled from $\Gamma[\mathbf{x}]$

- 1: Sample points y_1, \dots, y_k evenly by length from $\Gamma[\mathbf{x}]$
- 2: Use gradient descent on \mathbf{x} applied to Equation (5.7) via:

$$\frac{d\mathbf{x}}{dt} = -\nabla_{\mathbf{x}}P(\Gamma[\mathbf{x}], \mathcal{T}_S), \quad (5.8)$$

with the approximation $P(\Gamma[\mathbf{x}], \mathcal{T}_S) \approx \sum_{e \in \mathcal{T}_S} |e| \min_{i=1, \dots, m} P(y_i, e)$

- 3: **return** $\Gamma[\mathbf{x}]$
-

Note that this method is heuristic. Algorithm 5.3 can not return a full surveillance path if there aren't enough junction points to obtain full surveillance. In an extreme case, e.g., if a path Γ is only given by two junction points, full surveillance will not be possible for most possible surveillance targets $S \subseteq \partial X$. Even when there are enough junction points for some full surveillance path to exist, since gradient based methods are local, it is possible that this path would not be reachable.

Intuitively, using more junction points improves the heuristic. In practice, when choosing enough junction points for paths to look reasonable, the heuristic already has more than enough junction points. Where this might fail is if an experiment is specifically looking for a minimum number of junctions. We do not explore this because in this work the junction point construction of paths is meant to approximate the space of all paths.

5.1.5 Global Optimization by Intermittent Diffusion

The problem our algorithm is intended to solve requires global optimization in a highly non-convex space. We cannot hope to have a deterministic algorithm that guarantees a solution within any pre-specified error bound in finite time. Instead, we employ the strategy of *global optimization by intermittent diffusion* [43]. This Section 5.1.5 is a summary of

the ideas in [43] as they apply to our problem.

At its core, the intermittent diffusion strategy is a stochastic extension of gradient descent as in Equation (5.9). The diffusion term defined in Equation (5.10) turns on and off over time (i.e., the diffusion is intermittent). The idea is that when there is no diffusion, trajectories of the SDE will find local optima. When diffusion is turned on, trajectories have the opportunity to break out of local optima traps to search for global optima.

$$d\mathbf{x}(t, \omega) = -\nabla_{\mathbf{x}}L(\Gamma[\mathbf{x}]) dt + \sigma(t) dW(t), \quad t \geq 0, \quad \mathbf{x} \in X^n \quad (5.9)$$

$$\sigma(t) = \sum_{i=1}^N \sigma_i \mathbf{1}_{[S_i, T_i]}(t). \quad (5.10)$$

In Equation (5.10), $\mathbf{1}_{[S_i, T_i]}(t)$ is the characteristic function of the interval $[S_i, T_i]$. The intervals are chosen such that $0 = S_1 < T_1 < S_2 < T_2 < \dots < S_N < T_N$ and the length of the each interval $[S_i, T_i]$ for $i = 1, \dots, N$ is drawn from a uniform distribution on $[0, T_{max}]$. Similarly, each σ_i for $i = 1, \dots, N$ is drawn from a uniform distribution on $[0, \sigma_{max}]$.

The values $\sigma_{max} > 0$ and T_{max} can theoretically be chosen arbitrarily by the experimenter. Regardless of the choice, we have that as $N \rightarrow \infty$, the probability of finding all minima (including the global minima) of $L(\cdot)$ approaches 1. In practice, these values can dramatically change the run time and it is reasonable to run smaller experiments in the problem domain to see how they effect behaviour before running more expensive experiments.

The lengths of the intervals $[T_i, S_{i+1}]$ where $\sigma(t) = 0$ are unimportant. Indeed, when $\sigma(t) = 0$, Equation (5.9) becomes a local gradient descent:

$$\frac{d\mathbf{x}}{dt} = -\nabla_{\mathbf{x}}L(\Gamma[\mathbf{x}]), \quad t \in [T_i, S_{i+1}], \quad i = 1, \dots, N \quad \mathbf{x} \in X^n. \quad (5.11)$$

Because of this, all trajectories of the SDE in Equation (5.9) will converge exponentially quickly to a local minimum when $\sigma(t) = 0$. In fact, this implies that we can use any

valid local optimization strategy in our algorithm when $\sigma(t) = 0$. Thus, this SDE visits N local minima at times S_1, \dots, S_N , with increasing probability that global minima have been found as N increases.

5.1.6 The Benefit of Intermittent Diffusion vs. Simulated Annealing

Simulated annealing is a popular global optimization strategy that, in a continuous SDE setting, also uses Equation (5.9) but has a different diffusion term $\sigma(t)$. In simulated annealing, the diffusion $\sigma(t) = c/\sqrt{\log t}$ (where $c > 0$ is large) starts high and decays gradually. This will converge weakly to a distribution concentrated at the global optima of $L(\Gamma[\mathbf{x}])$ [43].

Numerically, this requires an SDE algorithm to be run to completion to find a single minimum, as each trajectory will only converge to a single solution at very large t . This can be expensive. Intermittent diffusion, for a single trajectory, converges to a local optimum every time the diffusion is turned off.

The intermittent diffusion strategy allows for any local optimization technique to be used, so this can be tuned to the problem domain. Further, there is a bounded amount of time for diffusion to be non-zero during intermittent diffusion and there is no need check for convergence during this time. Thus, the SDE solver part of intermittent diffusion is very cheap, and the local optimization is as cheap as the problem domain allows.

Both global optimization strategies are very useful in a variety of domains. Though the equations have the same form—both are a stochastic extension of gradient descent—they work quite differently in practice.

For SSMILE, intermittent diffusion is the best choice. This is because the problem requires the paths $\Gamma[\mathbf{x}]$ to stay in the highly non-convex set of full surveillance paths. This is computationally expensive, and being able to choose any local optimization technique expands the possibilities for improving computational efficiency.

5.2 The SSMILE Algorithm

We can now present the main result: The Surface Surveillance via Minimum Length Paths (SSMILE) Algorithm. SSMILE is the application of global optimization via intermittent diffusion to the structure and algorithms presented in Section 5.1.

The structural idea of the algorithm is: Given a starting path, jiggle the path to break out of local gradient traps, find a local optimum with respect to length and save it, repeat. Throughout, maintain full surveillance. This results in a list of local optima. As discussed in Section 5.1.5, since we are using global optimization by intermittent diffusion, probability approaches 1 that global optima are included in that list as time increases.

During intermittent diffusion, while the diffusion term is positive the path is perturbed (i.e., this is the “jiggling” step). Path perturbation is presented in Algorithm 5.4 explicitly showing the use of the Euler-Maruyama scheme [55] for solving SDEs. While other schemes can be used, we recommend this one because the purpose of the perturbation is explicitly to break out of local optima traps. This does not require high accuracy or an expensive algorithm.

When the diffusion term is 0 in intermittent diffusion, trajectories of the SDE converge to a local optimum exponentially quickly. Thus, we are free to use any local optimization technique. We use an explicit 2nd order PDE accelerated gradient descent scheme [56] which is a momentum based technique.

Algorithm 5.5 shows how any step-wise local optimization technique can be used. Full surveillance must be maintained while minimizing length, so we alternate between taking steps towards the local optimum and projection to full surveillance.

Putting the perturbation step and local optimization step together, we arrive at Algorithm 5.6.

Algorithm 5.4 Perturb Path via Euler-Maruyama

Require: A path $\Gamma[\mathbf{x}]$ given by junction points $\mathbf{x} \in X^n$

Require: A maximum diffusion strength $\sigma_{max} > 0$ and time $T_{max} > 0$

Require: A small time step $\Delta t > 0$

- 1: Draw from uniform distributions: $\sigma \in [0, \sigma_{max}]$ and $T \in [0, T_{max}]$
 - 2: $t \leftarrow 0$
 - 3: **repeat**
 - 4: Draw ΔW from a multivariate normal distribution $\mathcal{N}(\vec{0}, \text{diag}(\Delta t))$
 - 5: $\mathbf{x} \leftarrow \mathbf{x} - \nabla_{\mathbf{x}} L(\Gamma[\mathbf{x}]) \cdot \Delta t + \sigma \cdot \Delta W$
 - 6: $t \leftarrow t + \Delta t$
 - 7: Project \mathbf{x} to full surveillance using Algorithm 5.3
 - 8: **until** $t \geq T$
 - 9: **return** \mathbf{x}
-

Algorithm 5.5 Local Optimization for SSMILE

Require: An initial path given by junction points $\mathbf{x} \in X^n$

Require: A choice of step-wise local optimization scheme

- 1: **repeat**
 - 2: Update \mathbf{x} by taking a step towards the local optimum with the chosen scheme
 - 3: Project \mathbf{x} to full surveillance using Algorithm 5.3
 - 4: **until** convergence is reached
 - 5: **return** \mathbf{x}
-

Algorithm 5.6 Surface Surveillance via Minimum Length Path (SSMILE)

Require: An initial path given by junction points $\mathbf{x}_0 \in X^n$

Require: A maximum diffusion strength $\sigma_{max} > 0$ and time $T_{max} > 0$ for perturbation

- 1: **for** $i = 1, \dots, N$ **do**
 - 2: Perturb the junction points \mathbf{x}_{i-1} using Algorithm 5.4 to obtain \mathbf{x}_i^*
 - 3: Find a local minimum \mathbf{x}_i of $L(\Gamma[\mathbf{x}])$ with initial value \mathbf{x}_i^* using Algorithm 5.5
 - 4: **end for**
 - 5: Return \mathbf{x}_k such that $L(\Gamma[\mathbf{x}_k]) \leq L(\Gamma[\mathbf{x}_j])$ for $j = 1, \dots, N$
-

5.3 Computational Experiments

We present numerical experiments for various environments to validate the SSMILE algorithm.

First we have an example of an art gallery floor plan seen in Figure 5.3. The length of the top wall is 2.4, the maximum viewing radius is $r_{max} = 1$ and the minimum viewing radius is $r_{min} = 0.1$. The maximum viewing angle from the normal vectors off of the walls is $\theta_{max} = \pi/3$. From the given initial path (a), a local minimum found by SSMILE is shown in (b). With intermittent diffusion, a possible global minimum is found shown in (c). Notice the configuration of the three paths are quite different, demonstrating that as the algorithm continues, it searches through distant regions of the the space of all paths.

Figure 5.4 presents a 3-dimensional example of surveying a torus. The torus has an interior radius of 0.5 and outer radius of 1.5. We discretize this torus with a Delaunay triangulation containing 812 triangles. The longest edge in the triangulation has length $h = 0.3115$.

The viewing radius of the sensor is $r = 1.0$. We impose no minimum viewing distance or maximum angle. The path has 91 junction points to give a smoother approximation. As can be seen, the path passes through the hole in the torus, demonstrating that the algorithm can handle surfaces with non-trivial genus.

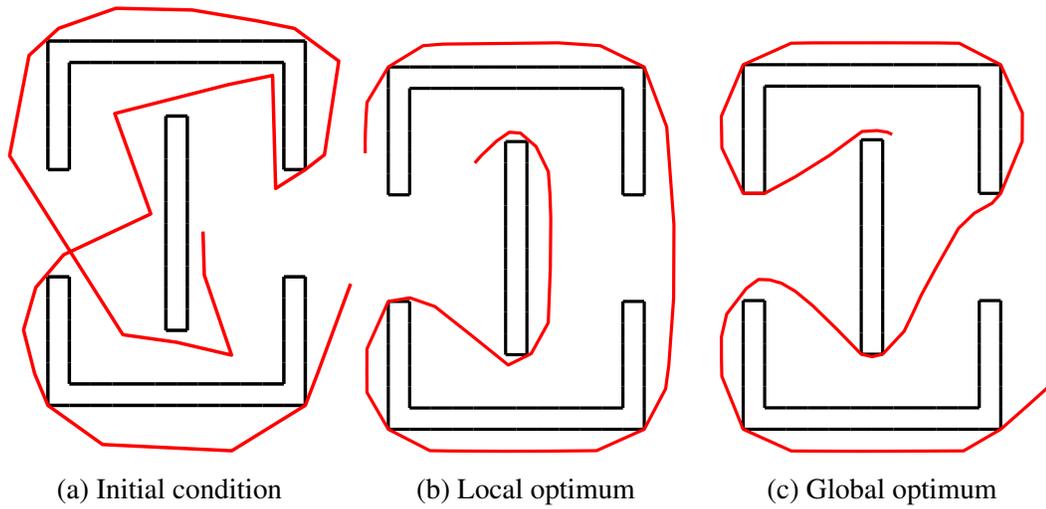


Figure 5.3: Surveillance of all walls in an art gallery floor plan. Parameters for this experiment were a minimum visibility distance of $r_{\min} = 0.1$ (surfaces within this distance are not considered surveyed), a maximum visibility distance of $r_{\max} = 1.0$, and a maximum visibility angle of $\theta_{\max} = \pi/3$.

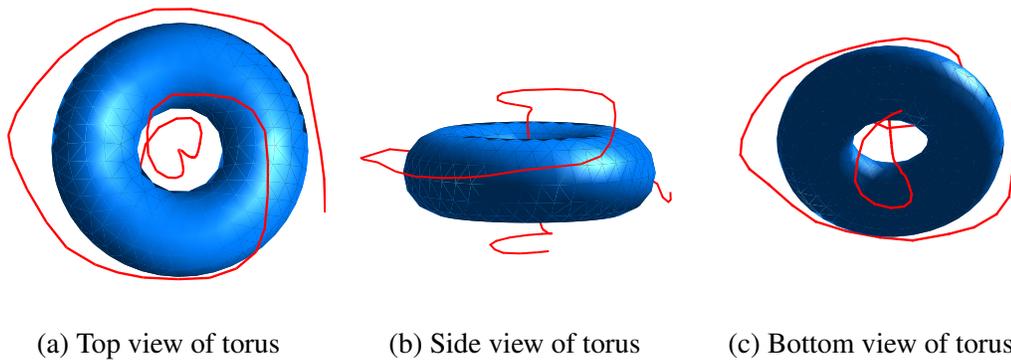


Figure 5.4: Optimal path for torus surveillance. Distance from center of torus to center of tube = 1.0. Radius of tube is = 0.5. The parameters for visibility were $r_{\max} = 1.0$ with no lower limit on visibility and no limit on visibility angle.

CHAPTER 6

SPHERE SURVEILLANCE

In this chapter, we explore the special case of sphere surveillance. Using the notation and terminology developed in Chapters 4 and 5, sphere surveillance is defined by the following:

Definition 6.0.1. *The Sphere Surveillance Problem:* Take the environment to be the exterior of the unit ball, $X = \{x \in \mathbb{R}^3 : \|x\| \geq 1\}$, and the surveillance surface to be the entire boundary, $S = \partial X$. There is no limit on visibility distance or angle. Find the shortest surveillance path.

Sphere surveillance is illustrative because the goal is conceptually simple, the question has been discussed in the literature for decades, and Algorithms 5.2 and 5.3 can be modified to use geometric concepts for better theoretical applicability. The modifications will make approximation error easier to understand and showcase the ability of the SSMILE algorithm to apply to classical geometry problems more broadly.

6.1 Background

We start by defining the concept of *inradius* and the associated inradius problem which we will refer to frequently [46].

Definition 6.1.1 (Inradius). The *inradius* of a curve in \mathbb{R}^n is the radius of the largest n -sphere contained in that curve's convex hull. The *inradius problem* is to find the shortest path of inradius 1.

6.1.1 History and Results in \mathbb{R}^2

In 1956, the lost in a forest problem was proposed: If a hiker is lost in a forest and discovers a sign that informs them that they are exactly 1 mile away from a straight road, what is the

shorest path the hiker can travel that guarantees they will find the road [48]? A solution with a proof outline was proposed the following year [49] and a rigorous proof was published in 1980 [50], which was later translated to English in 2019 [57].

Since the road is straight and exactly 1 mile away, it is a tangent line of the radius 1 circle centered at the hiker's starting point. The hiker's path is guaranteed intersect with this road if and only if their path intersects every tangent of the circle. Equivalently, the circle must be contained in the convex hull of the hiker's path. An easy first attempt is to pick any direction, walk a distance 1, and then follow the circle all the way around giving a distance of $1 + 2\pi$. Figure 6.1 depicts a construction that improves on this and yields the optimal solution [50].

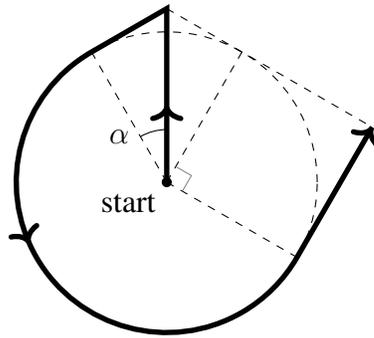


Figure 6.1: This construction yields a family of paths satisfying the conditions of the lost in a forest problem. The shortest path is obtained when $\alpha = \pi/6$ [50].

Removing the condition that the path must start at the center of the circle yields the following problem: What is the shortest curve in the plane that contains a unit circle in its convex hull? In other words, the \mathbb{R}^2 inradius problem as seen in Definition 6.1.1.

The \mathbb{R}^2 inradius problem was solved along with the lost in a forest problem [50] and the solution of length $\pi + 2$ can be seen in Figure 6.2.

Notice that the path that solves the \mathbb{R}^2 inradius problem in Figure 6.2 is also a full surveillance path for the unit disk. The path stays in the environment $X = \{x \in \mathbb{R}^2 : \|x\| \geq 1\}$ and every point of its surface is visible from somewhere along that path. Half of the surface is surveyed from points on the path that actually touch the disk. This is still

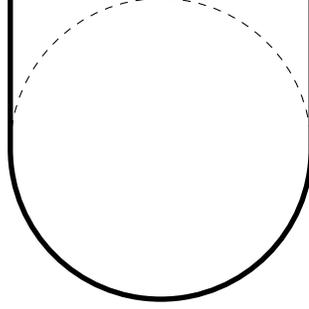


Figure 6.2: The shortest curve in the plane that contains a radius 1 circle in its convex hull.

physically reasonable from a surface surveillance perspective: If the unit disk is the view of a circular building from the top and a security guard wants to see all the way around as quickly as possible, this is the ideal path, though the physical size of the guard will force them slightly away from the ideal path.

The connection between the inradius problem and sphere surveillance is captured in the following Lemma:

Lemma 6.1.2. *In \mathbb{R}^n with $n \geq 2$, we have:*

- (1) *A minimal length inradius 1 path that does not intersect the interior of the contained sphere is a minimal length sphere surveillance path.*
- (2) *A minimal length sphere surveillance path has inradius 1.*

Proof. Take a point $x \in \mathbb{R}^n$ outside the unit sphere, $\|x\| \geq 1$, and a point $s \in \mathbb{R}^n$ on the unit sphere, $\|s\| = 1$. By Definition 4.3.7, since s is also the normal vector to the unit sphere at s , we have that s is visible from s exactly when $(x - s) \cdot s \geq 0$ (i.e., x is in the outer half-space defined by the tangent plane at s) and when the line segment \overline{xs} doesn't intersect the sphere. The first requirement implies the latter because the unit ball is convex.

From this, we see that a sphere surveillance path is a path that passes through every tangent plane of the sphere and doesn't intersect the sphere. The former requirement is precisely the same as having an inradius of at least 1. This establishes (1).

Suppose Γ is a sphere surveillance path with inradius $R > 1$. (2) will be proven by showing that Γ is not a minimal length surveillance path. The path $\Gamma/R = \{x/R : x \in \Gamma\}$

is shorter and has inradius 1, but it might intersect the sphere, so more care is needed. Instead, scale by

$$f(\|x\|) = \begin{cases} 1/R & \text{if } R \leq \|x\| \\ 1/\|x\| & \text{if } 1 \leq \|x\| < R \\ 1 & \text{otherwise} \end{cases} \quad (6.1)$$

to create a path $\Gamma_2 = \{xf(\|x\|) : x \in \Gamma\}$. Since f is continuous, Γ_2 is a path. Since Γ has inradius $R > 1$, there must be a point on the path x with $\|x\| \geq R$, so Γ_2 is strictly shorter. By construction of f , Γ_2 will not intersect the sphere and the inradius of Γ_2 is at least 1. This demonstrates that Γ was not a minimal length sphere surveillance path. Thus, a minimal length sphere surveillance path must have inradius exactly 1.

□

Note that *a priori* the solution to the \mathbb{R}^2 inradius problem could have intersected the unit disk. Since *a posteriori* we see that it does not, Lemma 6.1.2 proves that the path in Figure 6.2 solves the \mathbb{R}^2 sphere surveillance problem.

6.1.2 The Inradius Problem in \mathbb{R}^3

In \mathbb{R}^3 , we consider the inradius problem for both open paths and closed paths (paths where the end points meet). In \mathbb{R}^2 , closed paths are not interesting because the solution is simply the unit circle itself. The \mathbb{R}^3 inradius problem is also known as the asteroid surveying problem [44].

The \mathbb{R}^3 inradius problem has not been solved for open paths. The shortest known conjecture is given by Zalgaller [45] and shown in the left graphic of Figure 6.3. We describe Zalgaller's construction of the path using the right graphic of Figure 6.3.

Place the center of the unit sphere C at the origin and choose an angle $0 < \alpha < \pi/4$. Consider the cone with vertex $P_5 = (1, 0, -\cot \alpha)$, opening angle α , and center passing through C . The sphere sits inside of this cone with the two intersecting on a great circle

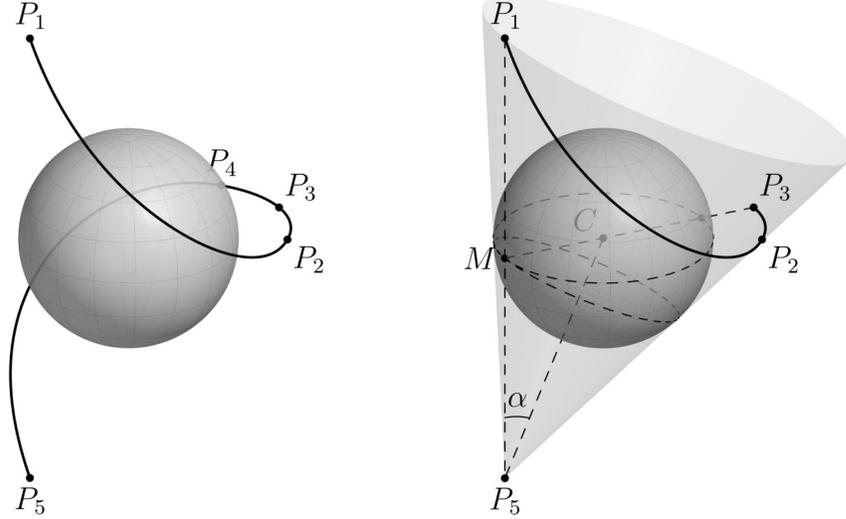


Figure 6.3: Zalgaller's conjectured optimal inradius 1 path in \mathbb{R}^3 [45].

plotted as a dashed line. We define the point $P_1 = (1, 0, \cot \alpha)$ so that $M = (1, 0, 0)$ is the midpoint of $\overline{P_1 P_5}$ which lies entirely on the cone. The equator of the sphere lying in the xy -plane has also been plotted as a dashed line. The point P_3 is defined to lay on the cone, above the equator, π radians around the sphere from P_1 , which implies P_3 lies on the x -axis.

With this in place, we can describe the path. The curve from P_1 to P_3 is a geodesic on the intersection of the cone and the upper half-space $z \geq 0$. The rest of the curve, from P_3 to P_5 , is the the arc from P_1 to P_3 rotated π radians around the x -axis. Since P_3 lies on the x -axis, this creates a single continuous path from P_1 to P_5 .

The point P_2 and its rotated copy P_4 also appear in Figure 6.3. P_2 is the first point along the path starting from P_1 that touches the xy -plane, and the part of the path from P_2 to P_4 lies entirely in the xy -plane. Without requiring the geodesic to be in the upper half-space, it would drop below the equator and Zalgaller shows that this would leave portions of the upper hemisphere unsurveyed [45]. The resulting path length is minimized when $\alpha \approx \pi/2 - 1.1421$, attaining a value of $L \approx 9.576778$.

The solution to the \mathbb{R}^3 inradius problem for closed paths is shown in the right graphic of Figure 6.4. This path was originally conjectured by Zalgaller in 1997 [58]. Progress was

made in 2018 [46] and in 2021 this path was shown to be the unique (up to rigid motion) optimal solution [47].

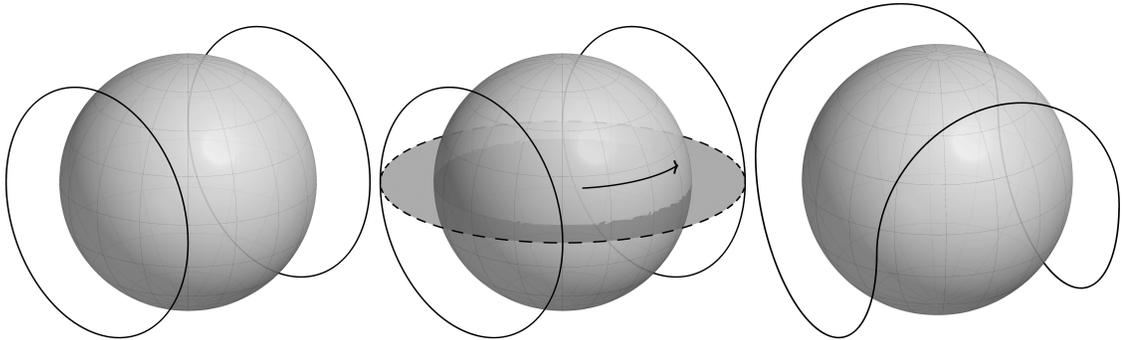


Figure 6.4: The construction of the shortest closed path of inradius 1 in \mathbb{R}^3 .

To construct this path, start with two disjoint unit circles with centers laying on antipodal points of the unit sphere. The top hemisphere is surveyed by the top semicircles and the bottom hemisphere is surveyed by the bottom semicircles. Rotate the top half by $\pi/2$ radians to create a single closed path of length 4π as seen in Figure 6.4. By construction, this clearly surveys the sphere. The proof that this path is optimal can be seen in [47].

6.2 SSMILE Implementation for Inradius

The SSMILE algorithm introduced in Chapter 5 can be directly applied to the inradius problem by triangulating the sphere. However, here we take the opportunity to showcase the wide applicability of SSMILE by using the definition of inradius directly. By directly calculating the inradius of paths, the sphere doesn't need to be modelled at all.

6.2.1 Computing the Inradius

We start with a method to calculate the inradius of a path. Recall Definition 6.1.1: The inradius of a path Γ is the radius of the largest sphere that sits inside the convex hull $\text{conv}(\Gamma)$. Our strategy will be to construct the convex hull of the path, and then to use linear programming to find the radius of the largest sphere that sits inside that convex hull.

Recall that in SSMILE, paths are discretized by a list of junction points $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^{3N}$. The path $\Gamma[\mathbf{x}]$ is constructed by connecting x_i to x_{i+1} via the shortest path, which is always a line segment since we are considering the inradius of curves in space and thus have no obstacles. Since the connecting line segments are themselves the convex hull of the connecting points, we have established the following:

Lemma 6.2.1. *The convex hull of the discretized path $\Gamma[x_1, \dots, x_N]$ is the convex hull of the junction points $x_1, \dots, x_N \in \mathbb{R}^3$.*

We can apply any technique to build the convex hull of the point cloud $x_1, \dots, x_N \in \mathbb{R}^3$. We use an incremental method that is a simplified version of the Quickhull algorithm [59].

The convex hull of a finite set of points is always a polytope which can be represented as an intersection of half-spaces. A half-space is most easily described by a normal vector $n \in \mathbb{R}^3$ and an offset $b \geq 0$ via $\{x \in \mathbb{R}^3 : n \cdot x \leq b\}$. Then the polytope \mathcal{P} defined by the intersection of m half-spaces will have the form

$$\mathcal{P} = \{x \in \mathbb{R}^3 : n_i \cdot x \leq b_i, i = 1, \dots, m\}. \quad (6.2)$$

Given the representation in Equation (6.2) of the convex hull \mathcal{P} of a path $\Gamma[\mathbf{x}]$ with $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^{3N}$, we find the radius of the largest sphere that sits inside \mathcal{P} using linear programming. This linear program is often used to solve for the Chebyshev center of \mathcal{P} [60] which is the center of the largest sphere, but the radius is also found. The linear program for the Chebyshev center and inradius is:

$$\begin{aligned} & \text{maximize} && R \\ & \text{subject to} && a_i \cdot c + R \leq b_i, i = 1, \dots, m. \end{aligned} \quad (6.3)$$

where $c \in \mathbb{R}^3$ is the center of the largest sphere inside \mathcal{P} and $R \geq 0$ is the radius. Thus, for the junction points $\mathbf{x} \in \mathbb{R}^{3N}$, we can compute the inradius $R(\Gamma[\mathbf{x}])$ of the path.

6.2.2 Modified Local Optimization Strategy

Now that we can compute the inradius, we could apply SSMILE now by replacing the projection to full surveillance step with gradient ascent of $R(\Gamma[\mathbf{x}])$ with respect to \mathbf{x} . This does work, but we have developed a better strategy.

We start with the observation that we can rescale paths to inradius 1 easily:

Lemma 6.2.2 (Inradius Rescaling). *For any path Γ with positive inradius, the rescaled path $(1/R(\Gamma)) \cdot \Gamma$ has inradius 1 and its length is given by*

$$L\left(\frac{1}{R(\Gamma)} \cdot \Gamma\right) = \frac{L(\Gamma)}{R(\Gamma)}. \quad (6.4)$$

Proof. Let Γ be a path with positive inradius $R(\Gamma)$ (which implies the length $L(\Gamma)$ is also positive). Let $\alpha > 0$. Both L and R are homogeneous functions, i.e., $L(\alpha\Gamma) = \alpha L(\Gamma)$ and $R(\alpha\Gamma) = \alpha R(\Gamma)$. This is well known for L . For R , note that scaling the path scales the largest sphere in its hull with it, and so it scales the radius of that sphere.

This proves Equation (6.4), and a simple calculation shows that the rescaled path has inradius 1: $R(\Gamma/R(\Gamma)) = R(\Gamma)/R(\Gamma) = 1$.

□

Lemma 6.2.2 gives a global method for projecting a path Γ to inradius 1: Simply rescale it by $1/R(\Gamma)$. This was attempted as a replacement in SSMILE, and it was found that a local length reduction followed by a global inradius rescaling increases length. This process does not converge and leads to an unbounded length. Figure 6.5 shows why this happens.

Since local decrease in length followed by global projection to inradius 1 does not work, we instead consider the quantity $L(\Gamma)/R(\Gamma)$. The following theorem shows that this ratio is an ideal object for optimization.

Theorem 6.2.3. *If a path Γ^* minimizes $L(\Gamma)/R(\Gamma)$ subject to $R(\Gamma) > 0$, then $(1/R(\Gamma^*))\Gamma^*$ minimizes $L(\Gamma)$ subject to $R(\Gamma) = 1$.*

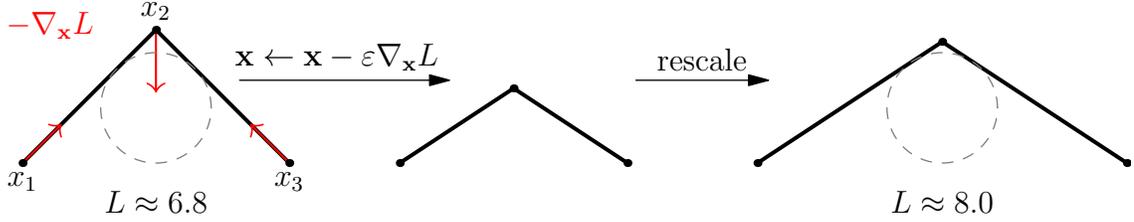


Figure 6.5: This depicts locally reducing the length by following the gradient and then globally rescaling to inradius 1. This results in increased length. Here $\varepsilon = 1/2$ so that the effect can be easily visualized, but this occurs for any $\varepsilon > 0$. The dashed circle is the radius 1 circle that sits inside the convex hull.

Proof. Note that L/R is scale invariant, i.e., for a path Γ with positive inradius and $\alpha > 0$,

$$\frac{L(\alpha\Gamma)}{R(\alpha\Gamma)} = \frac{\alpha L(\Gamma)}{\alpha R(\Gamma)} = \frac{L(\Gamma)}{R(\Gamma)}.$$

This combined with Lemma 6.2.2 establishes the theorem. □

Because of Theorem 6.2.3, we can minimize L/R instead of L . Almost all possible modifications to Γ will maintain a positive inradius, so maintaining a feasible path becomes trivial and the projection step of SSMILE is no longer necessary. Further, this allows us to use backtracking line search [61] for local minimization to reduce the number of gradient calculations.

Minimizing L/R means that we are focused on the shape of paths without regard for scale. Intuitively, this means the gradient direction should improve the shape of the curve faster, and so it should take fewer steps than if we only considered the gradient of L .

6.2.3 Modified SSMILE Algorithm

We are now able to present Algorithm 6.1 for the inradius problem.

Algorithm 6.1 Modified SSMILE for the inradius problem

Require: An initial path given by junction points $\mathbf{x} \in \mathbb{R}^{3N}$

- 1: **repeat**
 - 2: Perturb \mathbf{x} with Algorithm 5.4 replacing L with L/R and skipping step 7
 - 3: **repeat**
 - 4: Calculate $g = -\nabla_{\mathbf{x}}(L(\Gamma[\mathbf{x}])/R(\Gamma[\mathbf{x}]))$ using a finite difference method
 - 5: Use backtracking line search on \mathbf{x} in the direction g to reduce $L(\Gamma[\mathbf{x}])/R(\Gamma[\mathbf{x}])$
 - 6: **until** convergence
 - 7: Save the rescaled local optima $(1/R(\Gamma[\mathbf{x}])) \cdot \mathbf{x}$
 - 8: **until** the desired number of optima have been found
-

6.3 Numerical Results, Validation, and Prediction

We applied Algorithm 6.1 to both open and closed paths across a wide range of number of junction points. For closed paths, it is known that 4π is the optimal length and so we can validate our method with this knowledge and visually ensure that the paths we compute look like the known unique (up to rigid motion) optimal path [47]. For open paths, we can check if our results match the current best conjecture [45] and thereby offer a prediction about whether the conjecture is optimal.

First, we check the relative error on a log-log plot for different numbers of junction points, shown in Figure 6.6. If the relative error is converging to 0 in the number of junction points, we expect the points to fall on a line. This is what we see. The slope of the best fit line (using least squares) for each is approximately -2 , meaning that we have order 2 convergence in the number of junction points.

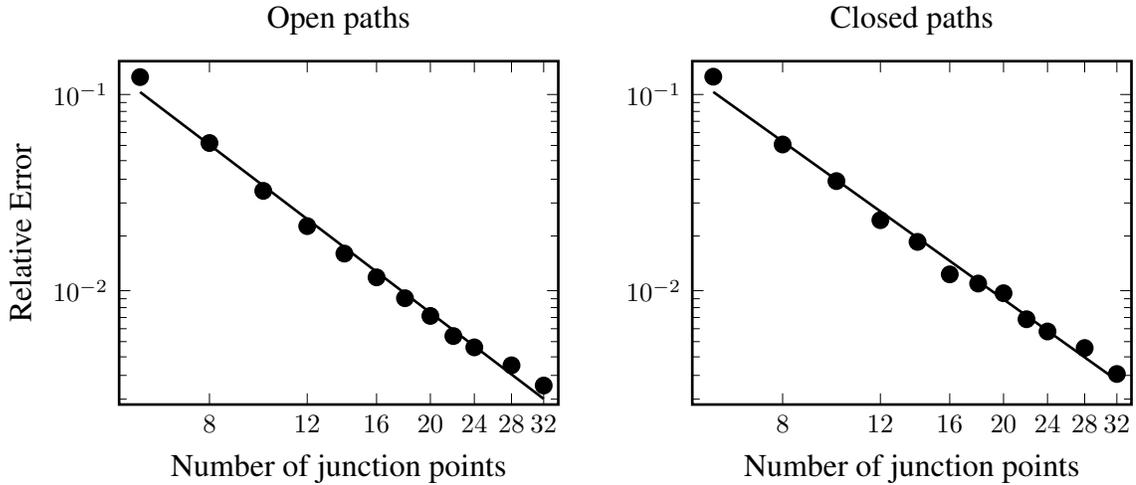


Figure 6.6: For open paths, the error is relative to the conjectured optimum. For closed paths, the error is relative to the known optimum. The slopes of the plotted best fit lines are -2.15 (open) and -2.02 (closed) implying approximately order 2 convergence in the number of junction points.

Figure 6.6 provides good evidence that Zalgaller’s conjecture for open paths is the correct length. It is possible that the path he provides is not the only solution attaining this length, therefore we still need to inspect the paths. Note that since the sphere has rotational and reflectional symmetry, uniqueness of solutions can only be found up to rotation and reflection. We standardize all paths to account for this.

In Figure 6.7 we see the shortest inradius 1 open paths found via Algorithm 6.1 for a variety of numbers of junction points N . There are no examples in the data of any other configuration attaining optimal length for any N . Therefore, our results provide evidence that the Zalgaller conjecture is not merely optimal, but is also the *unique* optimal inradius 1 path in \mathbb{R}^3 .

Figure 6.8 shows a collection of closed paths resulting from Algorithm 6.1. Again, after rotating (in this case, reflection is actually unnecessary) we see that all paths have the same configuration as the known unique solution. Along with the relative error graph in Figure 6.6, this helps validate our method.

Now that we’ve verified the algorithm produces the expected result for closed paths

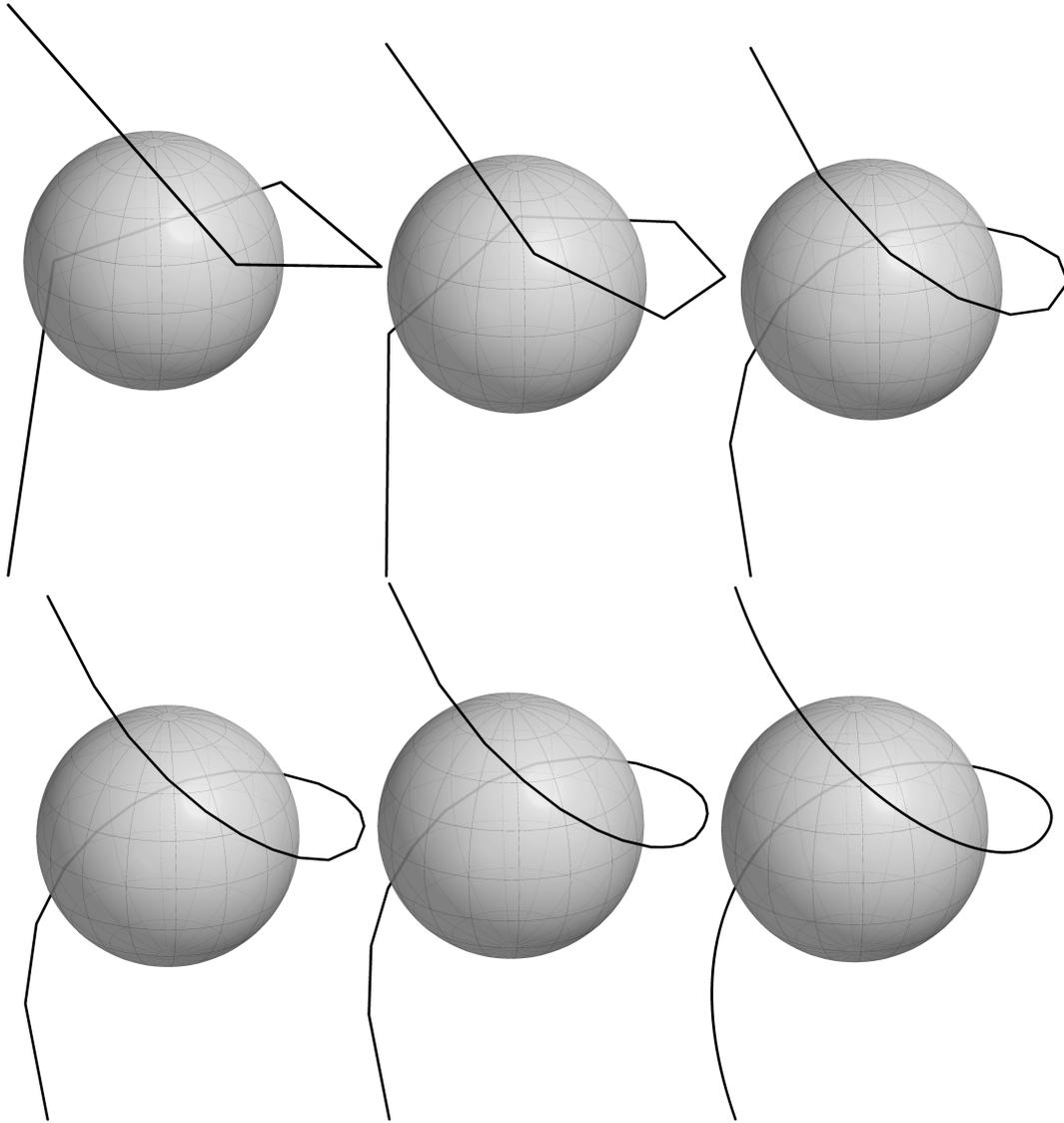


Figure 6.7: The result of Algorithm 6.1 for open paths with junction points $N = 6, 8, 16, 24, 32$, and finally Zalgaller's conjectured optimal path.

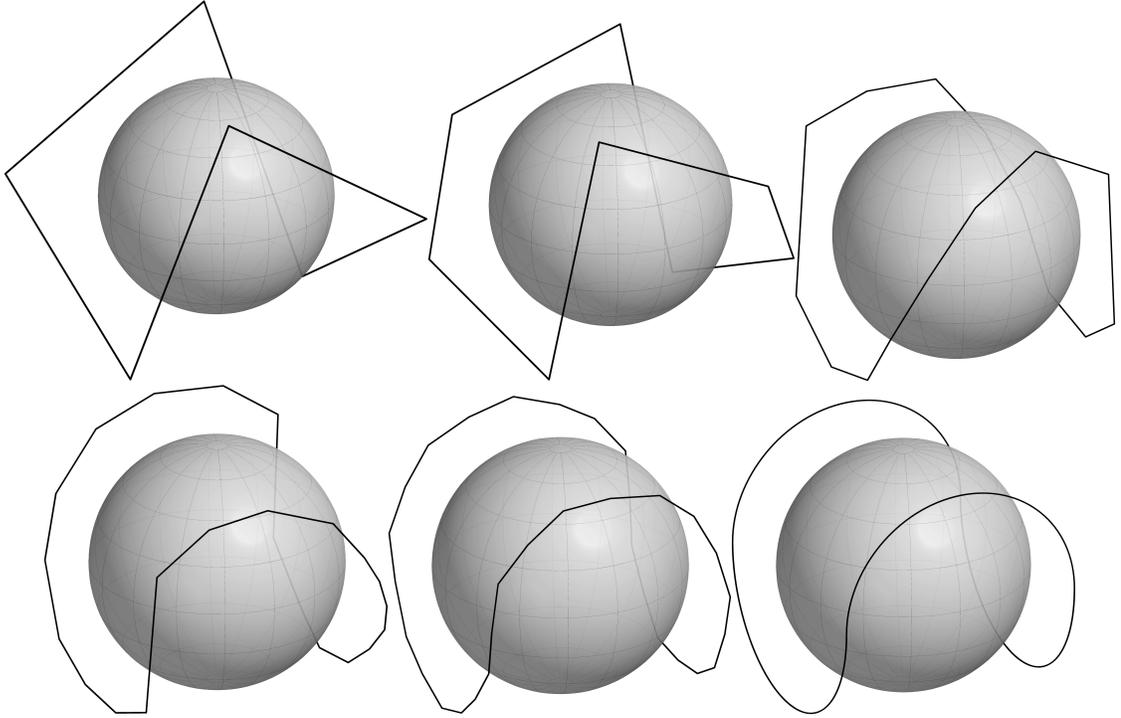


Figure 6.8: The result of Algorithm 6.1 for closed paths with junction points $N = 6, 8, 16, 24, 32$, and finally the unique (up to rotation) optimal path.

and produces the conjectured optimum for open paths, there are a few observations we can make. We were searching for inradius 1 paths and imposed no requirement that these paths not intersect the sphere. Since we see that they have not intersected the sphere, by Lemma 6.1.2, we know we have also found optimal sphere surveillance paths.

Also, we have experimentally found convergence of roughly order 2 in the number of junction points. Since we used an exact calculation for inradius, this has allowed us to learn about convergence characteristics with respect to path discretization. In Chapter 5, we couldn't do this because surfaces were also discretized.

Order 2 convergence is precisely what we would expect. For example, consider approximating the unit circle with a linear spline interpolation with N knots. The circle has perimeter 2π . The spline forms a regular polygon with N vertices which has perimeter $2N \sin(\pi/N)$. Using the Maclaurin series for \sin , we have:

$$\begin{aligned}
\text{approximation error} &= 2\pi - 2N \sin \frac{\pi}{N} \\
&= 2\pi - 2N \left(\frac{\pi}{N} - \frac{\pi^3}{N^3 3!} + \frac{\pi^5}{N^5 5!} - \dots \right) \\
&= 2\pi - 2\pi + \frac{2\pi^3}{N^2 3!} - \frac{2\pi^5}{N^4 5!} + \dots \\
&= \mathcal{O}(1/N^2).
\end{aligned} \tag{6.5}$$

Focusing on the paths themselves, there are a couple details worth noting. First, as would be expected, there are fewer junction points by length describing straighter portions of optimized paths as compared to higher curvature portions of the same paths. Second, for closed paths, particularly noticeable with $N = 8, 16, 24$ in Figure 6.8, the optimized paths have less symmetry than expected, though they are still converging to the expected result. In particular, these paths do not have the reflective symmetry found in the optimal path.

Our final observation is that the success seen in this chapter suggests a wide applicability of these ideas to other classical geometry problems. For example: The *width* of a curve in \mathbb{R}^n is the infimum of the distance between parallel hyperplanes bounding the path [46]. This could be studied in a similar way. Further, “paths” could be replaced with polytopes, e.g., opening a broad range of problems to study. This algorithm is only useful in giving evidence for pure math conjectures and can not prove anything on its own, but we believe this is a useful and underexplored area, evidenced by the results herein which strongly imply that Zalgaller has given the correct conjecture [45] and that it is better to spend time searching for a proof rather than searching for a shorter path.

CHAPTER 7

THEORETICAL EXPLORATIONS

7.1 Introduction

In this chapter we present some theoretical explorations related to surface surveillance via paths with limited visibility distance. There has been some theoretical work for polygon surveillance with unlimited visibility distance which we discussed in Chapter 4. With limited visibility distance, research has been limited to numerical approximation. Theoretical results, in addition to being intrinsically interesting, can help validate numerical methods. Thus, while we do not present major results in this chapter, it is valuable to present some possible starting points for future work.

We focus on environments in \mathbb{R}^2 , however the generalized setup presented in Section 7.2 works in \mathbb{R}^3 as well. After giving a generalized setup, we discuss surveillance within simple polygons in Section 7.3.

7.2 Generalized Setup and Observations

This setup is meant to be general enough to encompass many surveillance style problems. This could be surface surveillance, full environment surveillance, surveillance of a finite collection of points, and so on. The definition of visibility is abstract notation that can represent any possible visibility rules. Finally, we do not limit our surveillance to happen from paths, so that the problems with stationary sensors or security guards, like the art gallery problem [32], can be encoded in this framework.

We will prove two lemmas using this general setup. In the following section, we will limit the problem space to surface surveillance from paths with limited visibility distance.

Consider an environment $X \subseteq \mathbb{R}^n$ ($n = 2, 3$), a target surveillance set $S \subset X$, and a

visibility criterion $\mathcal{V}(x, y)$ that determines if $y \in X$ is visible from a point $x \in X$. S can be any subset of X . The visibility criterion is a function that encodes the visibility rules and has the form $X \times X \rightarrow \{0, 1\}$ where $\mathcal{V}(x, y) = 1$ if y is visible from x and $\mathcal{V}(x, y) = 0$ otherwise. For convenience, we also use the notation $\mathcal{V}(\Gamma, y)$ which is 1 when $\mathcal{V}(x, y) = 1$ for any $x \in \Gamma$.

A set $\Gamma \subseteq X$ surveys S if for all $y \in S$ there is an $x \in \Gamma$ such that $\mathcal{V}(x, y) = 1$. We call this a *surveillance set*. Because we have generalized to allow Γ to be a set, this framework encompasses problems where there are stationary guards or cameras as well.

When a surveillance set Γ is a path, we call it a *surveillance path*. Γ is called an *optimal path* if there are no shorter surveillance paths. I.e., if length is given by $L(\Gamma)$, the surveillance path Γ is called optimal if $L(\Gamma) \leq L(\Gamma')$ for all other surveillance path Γ' .

When Γ is a finite set of points, as in the art gallery problem [32], we define $L(\Gamma)$ to be the number of elements in Γ . We call Γ an *optimal set* if it is a finite cardinality surveillance set and there does not exist any finite surveillance set with fewer elements.

Lemma 7.2.1. *Let $T \subset S \subseteq X$. If Γ is: 1) An optimal path (or set) for T , and 2) a surveillance path (or set) for S , then Γ is also optimal for S .*

Proof. Suppose that Γ' is a surveillance path (or set) for S such that $L(\Gamma') < L(\Gamma)$. Let $y \in T$ be chosen arbitrarily. Since $T \subset S$, $y \in S$, and since Γ' surveys S , $\mathcal{V}(\Gamma', y) = 1$. Since y was an arbitrary element of T , we see that Γ' surveys T . This is a contradiction since Γ was assumed to be optimal for T , completing the proof. \square

The proof is simple, but the resulting Lemma 7.2.1 is powerful. To demonstrate the concept: Suppose S is some large and complicated set we would like to survey. If we can construct a very simple subset $T \subset S$ that has easy to find optimal paths (or sets) that happen to also survey S , then we've solved the problem for S . This will be used later.

Another way to make use of Lemma 7.2.1 is to construct a class of subsets $\mathcal{T} = \{T : T \subset S\}$ where each member of \mathcal{T} has easy to find optimal paths (or sets). Then the

optimization problem can be recast as a search over \mathcal{T} for sets T with optimal paths (or sets) that also survey S .

Next, we make an observations about self intersecting paths in \mathbb{R}^2 . Recall from Definition 4.3.4 that a path in an environment $X \subset \mathbb{R}^2$ is the image of a continuous map from the closed unit interval $[0, 1]$ to X . We will argue that self intersections can be interpreted as non-crossing using the idea of disentangling a path [35]. A visual explanation of the disentangling procedure is given in Figure 7.1.

Lemma 7.2.2 (No self-crossing). *A path Γ in an environment $X \subset \mathbb{R}^2$ with a finite number of self-intersections is the image of a continuous map $\eta : [0, 1] \rightarrow X$ with no self-crossing.*

Proof. Let $\gamma : [0, 1] \rightarrow X$ be any continuous map defining Γ . Suppose that for $t_1 < t_2$, we have the path crossing itself at $x_{\text{intersect}} := \gamma(t_1) = \gamma(t_2)$. Construct a new continuous map, $\eta : [0, 1] \rightarrow X$ that disentangles γ at the point $x_{\text{intersect}}$:

$$\eta(t) = \begin{cases} \gamma(t) & t \leq t_1 \\ \gamma(t_2 + t_1 - t) & t_1 < t < t_2 \\ \gamma(t) & t_2 \leq t. \end{cases} \quad (7.1)$$

See Figure 7.1 for a visual example of Equation 7.1. Note that the image of η is identical to the image of γ , so they are defining the same path Γ .

Repeat this process for every other point where the path crosses itself. □

7.3 Surface Surveillance in Simple Polygonal Environments

A polygon is *simple* if its boundary is formed by a single, non-self-intersecting path. We call an environment X a *simple polygonal environment* when X is a simple polygon. In particular, this means that X has no holes.

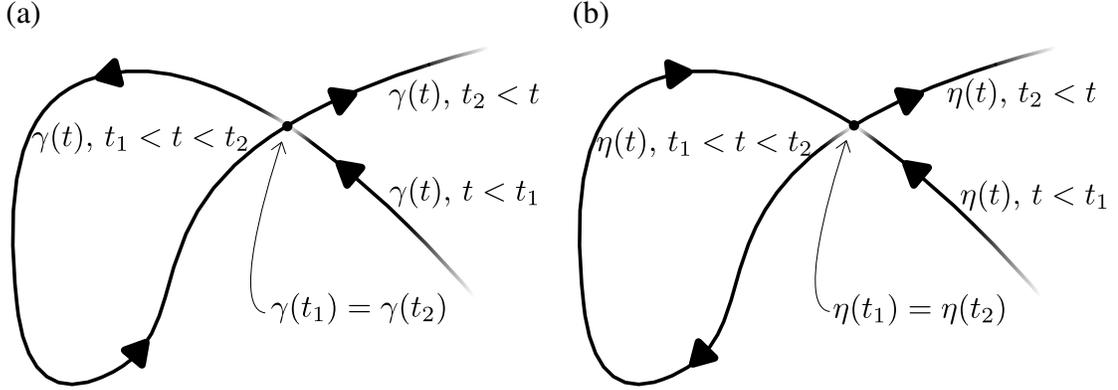


Figure 7.1: Example of disentangling a path. (a) Before and (b) after.

In this section we will explore surface surveillance within simple polygonal environments. Throughout this section, the surveillance set $S = \partial X$ is the boundary of the simply polygon X . For concreteness, the vertices of X are labeled v_1, \dots, v_n in clockwise order.

Our visibility criterion is given by a finite visibility distance. We will not consider a limited viewing angle as we did in Chapter 5. Precisely, we have $\mathcal{V}(x, y) = 1$ (y is visible from x) when $\overline{xy} \subset X$ (i.e., there is a line of sight) and $\|x - y\| \leq r$ for a finite visibility distance $0 < r < \infty$.

We will frequently refer to the region in X from which a point $y \in X$ is visible. For convenience, we name this region $\mathcal{R}(y) := \{x \in X : \mathcal{V}(x, y) = 1\}$.

7.3.1 Closed Paths

We start with closed paths. Our general strategy is to pick a finite subset T of S such that there exists an optimal path for surveillance of T also surveys S , and then apply Lemma 7.2.1. We will choose $T = \{v_1, \dots, v_n\}$, the vertices of the polygon.

We will show that there is an optimal closed path with the form $\Gamma[x_1, \dots, x_n, x_1]$ where $x_i \in \mathcal{R}(v_i)$. Recall from Definition 4.3.5 that this is the shortest path connecting the list of junction points in order. In other words, we will prove that there is an optimal path for surveying S that is formed by taking the shortest route to survey each vertex in order.

Before we prove this, first we look at an example of a polygon and a path of this form

in Figure 7.2. There is an important idea exemplified in this image. The junction points x_3 and x_4 are superfluous for defining this path, and the junction points x_5 , x_6 , and x_7 are all identical. Thus, while we will prove that a path of this form is optimal, its representation as a list of junction points is not unique, nor is the representation necessarily the shortest possible list of junction points.

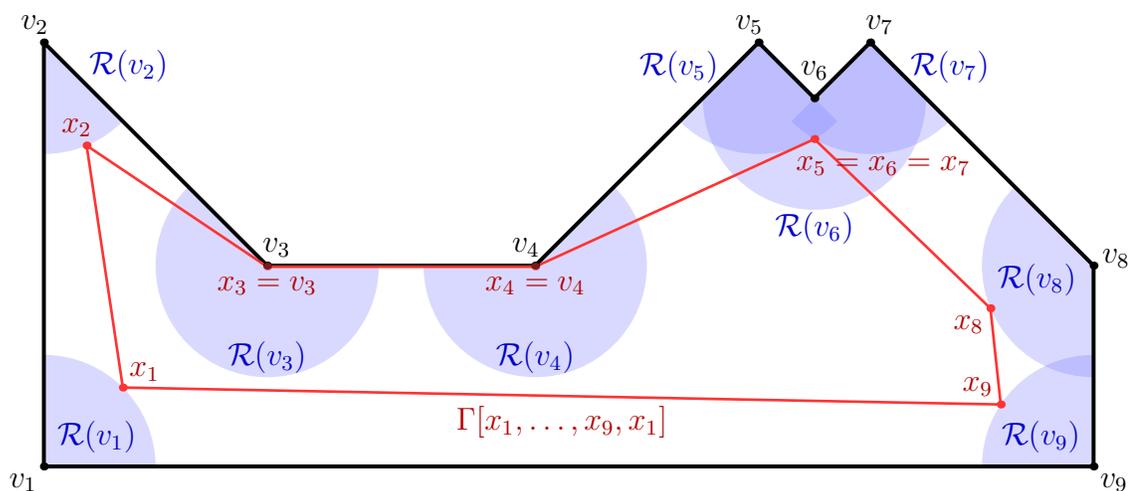


Figure 7.2: An example of a path of the form given in Theorem 7.3.1.

Theorem 7.3.1 (Closed paths in simple polygons). *Given a simple polygonal environment X with vertices given in clockwise order v_1, \dots, v_n , if an optimal path for $\{v_1, \dots, v_n\}$ exists, then there exists an optimal closed path surveying $S = \partial X$ of the form $\Gamma[x_1, \dots, x_n, x_1]$ where $x_i \in \mathcal{R}(v_i)$.*

Proof. Define the subset $T = \{v_1, \dots, v_n\} \subset S$ to be the vertices of the polygon. Our strategy is to first construct an optimal path for surveying T , then show that it also surveys S , and finally apply Lemma 7.2.1 to show that this path is optimal for surveillance of S .

Any path that surveys T must by definition pass through the regions $\mathcal{R}(v_i)$ for all $i = 1, \dots, n$. Suppose there is an optimal surveillance path Γ for T , pick an $i \in \{1, \dots, n\}$ and point x_i on the path in the region $\mathcal{R}(v_i)$. Follow the path and find the first region $\mathcal{R}(v_j)$ that does not include x_i is entered, and label the first point in this region x_j . Since Γ is supposed to be optimal, and no other new point of T could have been surveyed until reaching x_j , the

path from x_i to x_j must be the shortest possible.

Thus we can see that $\Gamma = \Gamma[x_{I_1}, \dots, x_{I_m}]$ for $m \leq n$ and $I \subseteq \{1, \dots, n\}$. If $m < n$, let $k \in \{1, \dots, n\} \setminus I$. The path must enter $\mathcal{R}(v_k)$ since $v_k \in T$, so at a point x_k where this occurs, include the (superfluous) junction point x_k in the representation of Γ . Continue this process until there is a junction point associated with each vertex.

At this point, there is some permutation p of $\{1, \dots, n\}$ such that $\Gamma = \Gamma[x_{p(1)}, \dots, x_{p(n)}, x_{p(1)}]$ where $x_i \in \mathcal{R}(v_i)$ for all $i = 1, \dots, n$.

Since we are working with closed paths and it doesn't matter where along the path we choose to start, without loss of generality we can take $p(1) = 1$. If there is an index $i > 1$ such that $p(i) \neq i$, then this will generate a crossing in the path since the vertices are given in order. By Lemma 7.2.2, we can add two junction points each at the intersection point and reorder the junction points according to Equation (7.1). By the triangle inequality, these two new junction points can be removed to shorten the path. Thus $p(i) = i$ for all $1, \dots, n$.

So far we have proven that an optimal path for surveying T has the form $\Gamma[x_1, \dots, x_n, x_1]$ with $x_i \in \mathcal{R}(v_i)$. Note that this path in fact surveys all of S because the section of the path given by $\Gamma[x_i, x_{i+1}]$ must survey the side $\overline{v_i v_{i+1}}$. Thus, by Lemma 7.2.1, this path is optimal for surveillance of S as well. \square

The disentangling argument used in the proof of Theorem 7.3.1 gives the following for open paths, which could be useful in future work:

Corollary 7.3.2. *Given any environment $X \subseteq \mathbb{R}^2$ and a finite surveillance set $T = \{v_1, \dots, v_n\}$, an optimal path to survey T will have the form $\Gamma[x_{p(1)}, \dots, x_{p(n)}]$ where p is a permutation of $1, \dots, n$ and $x_i \in \mathcal{R}(v_i)$ for $i = 1, \dots, n$, and the optimal path will not be self-crossing.*

7.3.2 Open paths

With open paths in most simple polygonal environments, the form given in Theorem 7.3.1 will not work. Consider the unit square, $X = [0, 1] \times [0, 1]$ with a maximum visibility

distance of $r = 0.2$. Figure 7.3 shows the shortest path of this form. Clearly, the majority of the side $\overline{v_1v_4}$ is not surveyed.

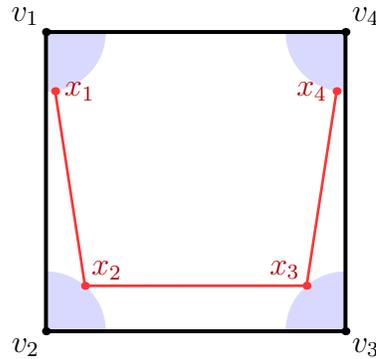


Figure 7.3: Open path in the square of the form $\Gamma[x_1, \dots, x_4]$ with $x_i \in \mathcal{R}(v_i)$ for $i = 1, \dots, 4$. This path does not survey all of $S = \partial X$.

One possible way to create a path that does survey the entire boundary of the square is to pick a point $v_5 \in \overline{v_1v_4}$, and then add two additional junction points on either end of the path that are both required to survey v_5 . This gives $\Gamma[x_0, \dots, x_5]$ with $x_i \in \mathcal{R}(v_i)$ for $i = 1, \dots, 5$ and $x_0 \in \mathcal{R}(v_5)$.

This does create a path that surveys the entire boundary. What is not clear is which point v_5 along $\overline{v_1v_4}$ we should pick, and whether or not this surveillance path is optimal. Figure 7.4 shows the resulting path when v_5 is picked to be the midpoint of $\overline{v_1v_4}$. The region $\mathcal{R}(v_5)$ is shown in green and must be entered by both ends of the path.

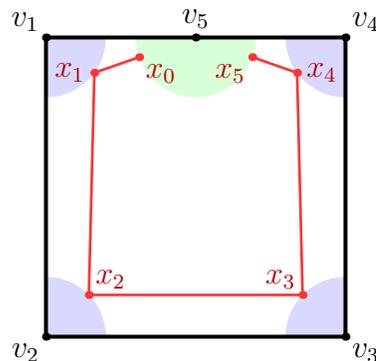


Figure 7.4: This construction gives a path that surveys the entire boundary.

We conjecture that this construction for some point $v_5 \in \overline{v_1v_4}$ does give an optimal path

for surveying the boundary of the square. Further, it is likely that including a point that must be covered by both ends of the path is a strategy that will work for many environments.

The strategy will not work for all environments. We call the following example the hallway. Take $X = [0, 1] \times [0, 5]$ with a maximum visibility distance of $r = 0.7$. It's possible to survey both sides of the hallway from near the center. In this case, the central channel from which both sides can be surveyed is 0.4 wide. Figure 7.5 shows this central channel in green, along with a path that surveys the entire boundary without needing to consider an extra point v_5 .

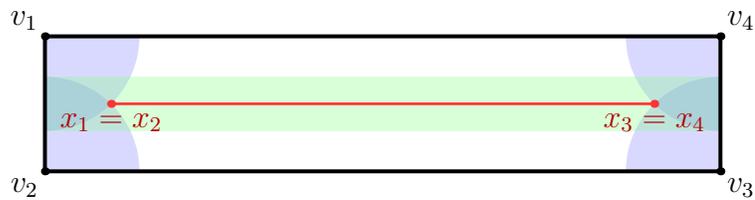


Figure 7.5: The hallway example. In green is the central channel from which some portion of either side of the hallway can be surveyed.

The full surveillance path given in Figure 7.5 is less than 4 units long. Consider what happens if we try to force some point to be surveyed from both ends of the path. Since $r = 0.7$, the endpoints would need to be within 1.4 units of each other, which would force a much longer path.

So, with examples as simple as the square and the hallway, we can see that a single general theorem similar to Theorem 7.3.1 for closed paths will not be possible.

CHAPTER 8

CONCLUSIONS

8.1 Model of the Germinal Center Reaction

The germinal center reaction is a vital process of the adaptive immune system. As such, it has received much attention in the literature. ODE models of the reaction have been important historically [9, 10, 12, 14]. More recently, the literature has been focused on computationally expensive agent-based modeling [15, 16]. This has allowed many more complex interactions to be included in models.

Our contribution is a delay differential equation model that captures the complex behavior of the agent-based models and offers new predictions. During the GC reaction, plasma B-cells are produced that manufacture antibodies. Some of these antibodies re-enter the GC and interact directly with the GC reaction that produced them. As discussed in the literature, this causes negative feedback where BCRs are competing with free antibodies to bind to epitopes on immune complex. We propose there is also positive feedback because the availability of stable immune complexes for BCRs to capture should increase with higher affinity antibodies available. We find that including this mechanism in the model produces stronger and more realistic immune responses.

Our model also resolves a puzzle related to the B-cells that seed a secondary immune response. It has been found in the laboratory that many naive B-cells are seeding the GC reactions [11]. Being a secondary response, higher affinity B-cells are available, which has been puzzling. We find that in fact, having lower affinity seeders improves the outcome of the reaction. This improvement is measured as the proportion of antigen epitopes outside of the GC that are masked by antibodies.

Having a differential equations model, as opposed to an agent-based model, is impor-

tant because it is computationally much cheaper. Modifications to the model can be iterated quickly to develop new ideas because there is no perceivable wait when running an experiment. Further, popular tools for differential equations solving can be used (e.g., [27]).

Having a computationally cheap model that represents our current understanding of a single germinal center reaction points to the possibility of coupling many reactions together in a network to model an entire lymph node or even the interactions between many lymph nodes. This is something we intend to explore in future work. This would allow us to study location dependency with vaccinations and infections, e.g., intranasal vaccines and injections are not identical in the real world, which we might be able to capture in a future networked model.

One of the novel features of our model is the mutation and division model. Now that we have seen that a normal distribution is correct for modeling affinity change in our affinity space (which is log-normal in real affinity), this allows us to consider using a heat equation for mutation. Based on this observation, in future work we intend to create a reaction-diffusion PDE model of the GC reaction. Our hope is that this will lead to a model that has more to discover through analytical study.

8.2 Surface Surveillance from Minimal Length Paths

In Chapters 4 and 5, we introduced the novel SSMILE Algorithm 5.6 to find the shortest paths from which surfaces can be surveyed in arbitrary environments. This included the option of a limited visibility distance constraint and a limited visibility angle constraint.

In Chapter 6, the SSMILE algorithm was modified to approximate solutions to the inradius problem in \mathbb{R}^3 . This was an ideal application because the theoretical solution is known for closed paths [47], but not for open paths. The algorithm converged on the true solution for closed paths in all tests which offers some validation for our method.

For open paths, the solution to the inradius problem is not known. We compared our numerical results to the best known conjecture which was offered by Zalgaller [45] and

found that we do converge to this conjecture in all tests. We therefore predict that the conjecture is correct.

In addition to this validation and prediction, the modifications made to study inradius allowed us to isolate the path discretization because no other quantity or structure required approximation. We numerically found that path length has about order 2 convergence in the number of junction points defining paths. This is the expected convergence for a polygonal path approximating a curved path, thus offering another validation of the algorithm.

The success of our application of the SSMILE algorithm to the inradius problem suggests that, in future work, we should apply this method to other classical geometry problems. It allows a systematic method for determining if a conjecture needs improvement, or of time should be spent on trying to prove a conjecture.

Finally, in Chapter 7, we discuss some potential starting points for a theoretical study of surveillance problems. Numerically, SSMILE allows us to find approximations of optimal surveillance paths, but more theoretically known true solutions would allow further validation and is interesting in its own right.

Future research directions could include: 1) Minimization of some quantity other than path length, 2) other classical geometry problems, 3) replacing path discretization with polytopes (i.e., convex surface discretization) for application to other types of problems, and 4) further theoretical work as in Chapter 7.

REFERENCES

- [1] L. M. Sompayrac, *How the Immune System Works*. Wiley-Blackwell, Apr. 2019, vol. 1, ISBN: 978-11-1954-212-4.
- [2] G. D. Victora and M. C. Nussenzweig, “Germinal centers,” *Annual Review of Immunology*, vol. 30, no. 1, pp. 429–457, 2012.
- [3] G. Nossal, “The molecular and cellular basis of affinity maturation in the antibody response,” *Cell*, vol. 68, no. 1, pp. 1–2, 1992.
- [4] M. J. Shlomchik, P. Watts, M. G. Weigert, and S. Litwin, “Clone: A monte-carlo computer simulation of b cell clonal expansion, somatic mutation, and antigen-driven selection,” in ser. *Current Topics in Microbiology and Immunology*, R. W. Compans *et al.*, Eds., Springer Berlin Heidelberg, 1998, pp. 173–197.
- [5] Y. Zhang *et al.*, “Germinal center b cells govern their own fate via antibody feedback,” *The Journal of Experimental Medicine*, vol. 210, no. 3, pp. 457–464, 2013.
- [6] S. Heinzl *et al.*, “A myc-dependent division timer complements a cell-death timer to regulate t cell and b cell responses,” *Nature Immunology*, vol. 18, no. 1, pp. 96–103, 2016.
- [7] A. D. Gitlin, Z. Shulman, and M. C. Nussenzweig, “Clonal selection in the germinal centre by regulated proliferation and hypermutation,” *Nature*, vol. 509, no. 7502, pp. 637–640, 2014.
- [8] M. Meyer-Hermann, E. Mohr, N. Pelletier, Y. Zhang, G. D. Victora, and K.-M. Toellner, “A theory of germinal center b cell selection, division, and exit,” *Cell Reports*, vol. 2, no. 1, pp. 162–174, 2012.
- [9] T. B. Kepler and A. S. Perelson, “Cyclic re-entry of germinal center b cells and the efficiency of affinity maturation,” *Immunology Today*, vol. 14, no. 8, pp. 412–415, 1993.
- [10] M. Meyer-Hermann, A. Deutsch, and M. Or-Guil, “Recycling probability and dynamical properties of germinal center reactions,” *Journal of Theoretical Biology*, vol. 210, no. 3, pp. 265–285, 2001.
- [11] J. M. J. Tas *et al.*, “Visualizing antibody affinity maturation in germinal centers,” *Science*, vol. 351, no. 6277, pp. 1048–1054, 2016.
- [12] C. Keşmir and R. J. De Boer, “A mathematical model on germinal center kinetics and termination,” *The Journal of Immunology*, vol. 163, no. 5, pp. 2463–2469, 1999.

- [13] A. S. Perelson and G. F. Oster, “Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-non-self discrimination,” *Journal of Theoretical Biology*, vol. 81, no. 4, pp. 645–670, 1979.
- [14] D. Iber and P. K. Maini, “A mathematical model for germinal centre kinetics and affinity maturation,” *Journal of Theoretical Biology*, vol. 219, no. 2, pp. 153–175, 2002.
- [15] M. MEYER-HERMANN, “Agent-based models or differential equations: Two ways to learn about selection mechanisms in germinal centres,” *Biophysical Reviews and Letters*, vol. 3, no. 01n02, pp. 313–324, 2008.
- [16] M. T. Figge, A. Garin, M. Gunzer, M. Kosco-Vilbois, K.-M. Toellner, and M. Meyer-Hermann, “Deriving a germinal center lymphocyte migration model from two-photon data,” *The Journal of Experimental Medicine*, vol. 205, no. 13, pp. 3019–3029, 2008.
- [17] M. Meyer-Hermann, “Overcoming the dichotomy of quantity and quality in antibody responses,” *The Journal of Immunology*, vol. 193, no. 11, pp. 5414–5419, 2014.
- [18] P. A. Robert, A. Rastogi, S. C. Binder, and M. Meyer-Hermann, “How to simulate a germinal center,” in ser. *Methods in Molecular Biology*, D. P. Calado, Ed., Springer New York, 2017, ch. 22, pp. 303–334, ISBN: 9781493970957.
- [19] G. D. Victora *et al.*, “Germinal center dynamics revealed by multiphoton microscopy with a photoactivatable fluorescent reporter,” *Cell*, vol. 143, no. 4, pp. 592–605, 2010.
- [20] Y. Zhang, L. Garcia-Ibanez, and K.-M. Toellner, “Regulation of germinal center B-cell differentiation,” *Immunological Reviews*, vol. 270, no. 1, pp. 8–19, 2016.
- [21] Y.-J. Liu, J. Zhang, P. J. Lane, E. Y.-T. Chan, and I. MacLennan, “Sites of specific b cell activation in primary and secondary responses to t cell-dependent and t cell-independent antigens,” *European journal of immunology*, vol. 21, no. 12, pp. 2951–2962, 1991.
- [22] I. MacLennan, Y. Liu, S. Oldfield, J. Zhang, and P. Lane, “The evolution of b-cell clones,” in *Immunological Memory*, Springer, 1990, pp. 37–63.
- [23] M. Kuraoka *et al.*, “Complex antigens drive permissive clonal selection in germinal centers,” *Immunity*, vol. 44, no. 3, pp. 542–552, 2016.
- [24] C. Berek and C. Milstein, “The dynamic nature of the antibody repertoire,” *Immunological Reviews*, vol. 105, no. 1, pp. 5–26, 1988.

- [25] M. Shannon and R. Mehr, “Reconciling repertoire shift with affinity maturation: The role of deleterious mutations,” *The Journal of Immunology*, vol. 162, no. 7, pp. 3950–3956, 1999. eprint: <http://www.jimmunol.org/content/162/7/3950.full.pdf>.
- [26] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A fresh approach to numerical computing,” *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017.
- [27] C. Rackauckas and Q. Nie, “Differentials.jl—a performant and feature-rich ecosystem for solving differential equations in julia,” *Journal of Open Research Software*, vol. 5, no. 1, 2017.
- [28] C. Tsitouras, “Runge–kutta pairs of order 5 (4) satisfying only the first column simplifying assumption,” *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 770–775, 2011.
- [29] I. Pohl, “A method for finding Hamilton paths and Knight’s tours,” *Communications of the ACM*, vol. 10, no. 7, pp. 446–449, Jul. 1967.
- [30] G. Gutin and A. P. Punnen, Eds., *The Traveling Salesman Problem and Its Variations* (Combinatorial Optimization 12). New York: Springer, 2007, ISBN: 978-0-387-44459-8.
- [31] D. S. Johnson, “The NP-completeness column: An ongoing guide,” *Journal of Algorithms*, vol. 3, no. 2, pp. 182–195, Jun. 1982.
- [32] J. O’Rourke, *Art Gallery Theorems and Algorithms* (International Series of Monographs on Computer Science, No 3). Oxford University Press, 1987, ISBN: 9780195039658.
- [33] W.-p. Chin and S. Ntafos, “Optimum watchman routes,” *Information Processing Letters*, vol. 28, no. 1, pp. 39–44, 1988.
- [34] S. Ntafos, “Watchman routes under limited visibility,” *Computational Geometry*, vol. 1, no. 3, pp. 149–170, 1992.
- [35] S. H. Kang, S. J. Kim, and H. Zhou, “Path optimization with limited sensing ability,” *Journal of Computational Physics*, vol. 299, pp. 887–901, 2015.
- [36] S. Carlsson, H. Jonsson, and B. J. Nilsson, “Finding the Shortest Watchman Route in a Simple Polygon,” *Discrete & Computational Geometry*, vol. 22, no. 3, pp. 377–402, Oct. 1999.
- [37] X. Tan, “Fast computation of shortest watchman routes in simple polygons,” *Information Processing Letters*, vol. 77, no. 1, pp. 27–33, Jan. 2001.

- [38] J. S. B. Mitchell, “Approximating Watchman Routes,” in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Jan. 2013, pp. 844–855, ISBN: 978-1-61197-251-1.
- [39] X. Tan and T. Hirata, “Shortest safari routes in simple polygons,” in *Algorithms and Computation*, G. Goos, J. Hartmanis, D.-Z. Du, and X.-S. Zhang, Eds., vol. 834, Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 523–530, ISBN: 978-3-540-58325-7.
- [40] C. Wei-Pang and S. Ntafos, “The zookeeper route problem,” *Information Sciences*, vol. 63, no. 3, pp. 245–259, Sep. 1992.
- [41] S.-N. Chow, J. Lu, and H.-M. Zhou, “Finding the shortest path by evolving junctions on obstacle boundaries (e-job): An initial value ode’s approach,” *Applied and Computational Harmonic Analysis*, vol. 35, no. 1, pp. 165–176, 2013.
- [42] Y.-H. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro, “Visibility and its dynamics in a pde based implicit framework,” *Journal of Computational Physics*, vol. 199, no. 1, pp. 260–290, 2004.
- [43] S.-N. Chow, T.-S. Yang, and H. Zhou, “Global optimizations by intermittent diffusion,” *National Science Council Tunghai University Endowment Fund for Academic Advancement Mathematics Research Promotion Center*, vol. 121, 2009.
- [44] T. M. Chan, A. Golynski, A. Lopez-Ortiz, and C.-G. Quimper, “The asteroid surveying problem and other puzzles,” in *Proceedings of the nineteenth annual symposium on Computational geometry*, ACM, 2003, pp. 372–373.
- [45] V. A. Zalgaller, “Shortest inspection curves for the sphere,” *Journal of Mathematical Sciences*, vol. 131, no. 1, pp. 5307–5320, 2005.
- [46] M. Ghomi, “The length, width, and inradius of space curves,” *Geometriae Dedicata*, vol. 196, no. 1, pp. 123–143, Oct. 2018.
- [47] M. Ghomi and J. Wenk, “Shortest closed curve to inspect a sphere,” *Journal für die reine und angewandte Mathematik*, Sep. 2021.
- [48] R. Bellman, “Minimisation problem,” *Bull. Amer. Math. Soc.*, p. 270, 1956.
- [49] J. R. Isbell, “An optimal search pattern,” *Naval Research Logistics Quarterly*, vol. 4, no. 4, pp. 357–359, 1957.
- [50] H. Joris, “Le chasseur perdu dans la forêt,” *Elemente der Mathematik= Revue de mathématiques élémentaires= Rivista di matematica elementare*, vol. 35, no. ARTICLE, pp. 1–14, 1980.

- [51] V. A. Zalgaller, “A question of bellman,” *Journal of Mathematical Sciences*, vol. 131, no. 1, pp. 5286–5306, 2005.
- [52] O. Hjelle and M. Daehlen, *Triangulations and Applications* (Mathematics and Visualization). Springer, 2006.
- [53] M. de Berg, Ed., *Computational Geometry: Algorithms and Applications*, 3rd ed. Berlin: Springer, 2008, ISBN: 978-3-540-77973-5.
- [54] Blender Online Community, *Blender - a 3d modelling and rendering package*, Blender Foundation, Blender Institute, Amsterdam, 2021.
- [55] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*. 1992.
- [56] M. Benyamin, J. Calder, G. Sundaramoorthi, and A. Yezzi, “Accelerated PDE’s for efficient solution of regularized inversion problems,” *arXiv:1810.00410 [cs, math]*, Sep. 2018.
- [57] S. Finch, “A translation of Henri Joris’ ”Le chasseur perdu dans la for\`et” (1980),” *arXiv:1910.00615 [math]*, Oct. 2019. arXiv: 1910.00615.
- [58] V. A. Zalgaller, “Extremal problems on the convex hull of a space curve,” *Algebra i Analiz*, vol. 8, no. 3, pp. 1–13, 1996.
- [59] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, Dec. 1996.
- [60] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004, ISBN: 978-0-521-83378-3.
- [61] D. P. Bertsekas, *Nonlinear Programming*, 3rd ed. Belmont, Mass: Athena scientific, 2016, ISBN: 978-1-886529-05-2.