



NRL/MR/6790--12-9374

On the Construction of Optimal Paths to Extinction

ANDREW KESSLER

*Northwestern University
Evanston, Illinois
Office of Naval Research Summer Fellow*

IRA B. SCHWARTZ

*Beam Physics Branch
Plasma Physics Division*

LEAH B. SHAW

*College of William and Mary
Williamsburg, Virginia*

January 3, 2012

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 03 JAN 2012		2. REPORT TYPE		3. DATES COVERED	
4. TITLE AND SUBTITLE On the Construction of Optimal Paths to Extinction			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Andrew Kessler; Leah Shaw; Ira Schwartz			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory, 4555 Overlook Avenue SW, Washington, DC, 20375-5320			8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/6790--12-9374		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT One of the major problems in dealing with interacting finite populations of agents, such as molecules in chemistry or people in populations, is that there always exists a probability of one species or state going extinct. Predicting the probability of extinction requires a knowledge of how the dynamics progresses towards extinction. The path that optimizes the probability to extinction is deemed to be the optimal path. Here we present an algorithm for constructing, or growing, the optimal path to extinction in systems of arbitrary dimensions. The algorithm relies on the calculation of finite-time Lyapunov exponents (FTLE), which provide a quantitative measure of how sensitively a system's behavior depends on initial conditions [1]. We also present an efficient method for approximating the FTLE field of a dynamical system.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 18	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

On the Construction of Optimal Paths to Extinction *

Andrew Kessler^{1,*}, Leah B. Shaw² and Ira B. Schwartz¹

¹ U.S. Naval Research Lab, Code 6792, Nonlinear Systems Dynamics Section,
Washington, DC 20375

²Department of Applied Sciences, College of William and Mary,
P.O. Box 8795, Williamsburg, VA 23187-8795

Abstract

One of the major problems in dealing with interacting finite populations of agents, such as molecules in chemistry or people in populations, is that there always exists a probability of one species or state going extinct. Predicting the probability of extinction requires a knowledge of how the dynamics progresses towards extinction. The path that optimizes the probability to extinction is defined to be the optimal path. Here we present an algorithm for constructing, or growing, the optimal path to extinction in systems of arbitrary dimensions. The algorithm relies on the calculation of finite-time Lyapunov exponents (FTLE), which provide a quantitative measure of how sensitively a system's behavior depends on initial conditions [1]. We also present an efficient method for approximating the FTLE field of a dynamical system.

Manuscript approved October 31, 2011

*AK currently is a student in the Mechanical Engineering Department at Northwestern University. This work was done while he was on an ONR summer fellowship at NRL. The research was supported by the Office of Naval Research and the National Institutes of Health. The project described was supported by Grant Number R01GM090204 from the National Institute Of General Medical Sciences. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute Of General Medical Sciences or the National Institutes of Health

1 Introduction

Undoubtedly, the key to understanding effective strategies for expediting the extinction of an epidemic lies in developing an accurate model of disease evolution within a population. Traditionally, the spread of an epidemic within a population has been modeled as a stochastic system, where noise arises from discrete interactions between individuals.

We will instead employ an eikonal approximation to recast the problem in terms of an effective classical Hamiltonian system [1]. It turns out that there exists a specific sequence of noise that transitions the system from an endemic to extinct state, representing the most probable, rare sequence of such events. This heteroclinic trajectory in the phase space of the Hamiltonian flow is known as the optimal path to extinction (OPE).

Analysis of optimal paths to extinction has the potential to reveal new strategies for controlling the spread of disease and monitoring the progress of existing protocols [1].

This report presents an algorithm for growing the optimal path to extinction in systems of arbitrary dimensions. The algorithm relies on the calculation of finite-time Lyapunov exponents (FTLE), which provide a quantitative measure of how sensitively a system's behavior depends on initial conditions [1]. We will also present an efficient method for approximating the FTLE field of a dynamical system.

The algorithm presented has successfully grown optimal paths to extinction for a number of two dimensional systems, and shows promise for working with four dimensional systems as well. However, additional experiments are required before accuracy in higher dimensional systems can be confirmed.

2 Finite-Time Lyapunov Exponent Approximation

Continuous dynamical systems have quantities, known as Lyapunov exponents, which provide a quantitative measure of how sensitively a system's behavior depends on small perturbations to initial conditions. Traditionally, Lyapunov exponents measure sensitivity in the infinite time limit; finite time Lyapunov exponents are simply Lyapunov exponents evaluated on a finite time interval [4].

A detailed exploration of FTLE calculations can be found in [1] and [4]. Below, we present a method for approximating FTLE quickly and efficiently, referred to as the "Poor Man's FTLE".

2.1 Poor Man's FTLE Algorithm

Assume x and p are state variables, representing a point in a Hamiltonian system's phase space.

1. Evaluate equations of motion starting at (x, p) for time T with integration time step Δt , and record the final position as (x', p') .

2. Pick a random point at distance ϵ away from (x, p) , called (x_ϵ, p_ϵ) , where ϵ is very small relative to the dynamics of the system.
3. Evaluate equations of motion starting at (x_ϵ, p_ϵ) for time T with integration time step Δt , and record the final position as $(x'_\epsilon, p'_\epsilon)$.
4. Let δ represent the Euclidean distance between the two end points of the trajectories, i.e. $\delta = \text{dist}[(x', p'), (x'_\epsilon, p'_\epsilon)]$.
5. Calculate the FTLE at (x, p) as

$$\sigma = \frac{1}{|T|} \ln \frac{\delta}{\epsilon}.$$

A graphical representation of this algorithm is shown in Figure 1.

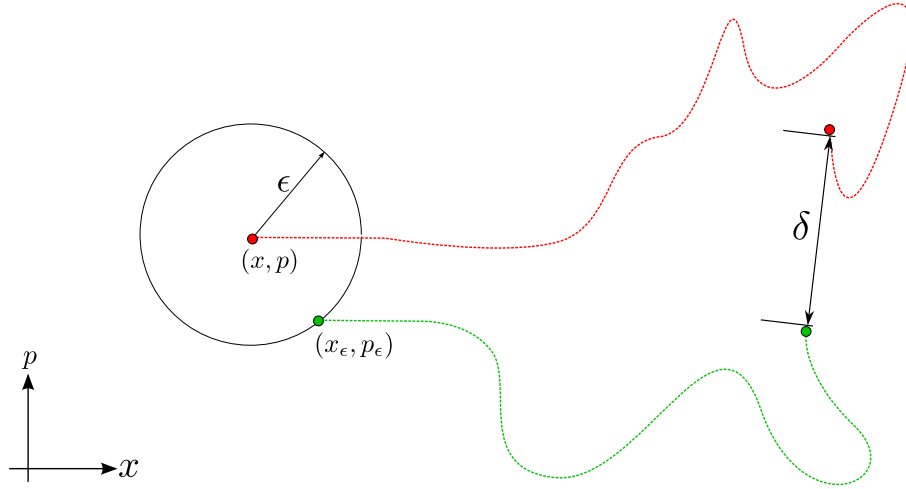


Figure 1: Graphical representation of Poor Man's FTLE algorithm. The red and green dotted lines represent the trajectories of points (x, p) and (x_ϵ, p_ϵ) through phase space over time period T .

3 Growing Optimal Path to Extinction - The Algorithm

Schwartz, et. al. have shown that the local maximums of FTLE fields correspond to the optimal path to extinction, i.e. the optimal path to extinction is the most sensitive to initial data. In fact, finding ridges of local maxima through phase space will lead us towards the optimal path to extinction, if one exists [4].

Previous work has focused on evaluating fields of FTLE values, and then picking out the optimal paths to extinction based on local maxima [1]. However, this process is a time consuming and inefficient method to detect an optimal path to extinction, because it requires evaluating FTLE's over regions of phase space that are unrelated to the OPE. Additionally, these FTLE fields are difficult to visualize in higher dimensions [5].

Instead, we present an algorithm to grow the optimal path to extinction in arbitrary n dimensions, knowing only the endemic and extinct states.

3.1 Review of Terminology and Notation

For the purposes of this explanation, we will use $\mathbf{x}$ to represent an n -dimensional vector, which defines a point in $n = d + 1$ dimensional state space. Equivalently, we can represent a point in $d + 1$ dimensional state space using hyperspherical coordinates, where $\boldsymbol{\alpha}$ is a vector of d angles, along with a radius r . We will define functions $\Phi_C(\boldsymbol{\alpha})$ and $\Phi_P(\mathbf{x})$ to represent conversions from polar to cartesian and cartesian to polar, respectively¹. For a review of hyperspherical coordinates, see Appendix A.

Let $\mathbf{P}$ be a list of equally sized surface partitions of a d-sphere, defined by intervals of hyperspherical angles². They can be thought of as *bins* for points on the surface of a d-sphere. Each $p \in \mathbf{P}$ defines an equally sized d-volume. Let ν be the length of $\mathbf{P}$ (i.e. the number of partitions).

3.2 Growing Algorithm

The OPE growing algorithm combines the fact that the path must be curve along with the knowledge that the local maximum of the FTLE falls along the path to grow it from the endemic state. The algorithm itself is presented below.

Initialize variables ...

1. Set the iterator $\kappa = 0$.
2. Choose a starting point for the algorithm, $\mathbf{x}_0$ (the endemic state).
3. Choose an ending point for the algorithm, $\mathbf{x}_f$ (the extinct state).
4. Choose search shell radial limits, $\mathbf{R}$. $R_{\max} = \frac{|\mathbf{x}_f - \mathbf{x}_0|}{100}$ is usually sufficient.

Begin procedure ...

5. Calculate the direction of the vector field at point $\mathbf{x}_\kappa$, $\hat{\mathbf{u}}$.

¹For simplicity, these functions are assumed to operate only on points where $|\mathbf{x}| = 1$.

²Recall that a d-sphere exists in d+1 dimensional space.

6. Generate S uniformly distributed points on a d -hemisphere, where the inward normal to the hemisphere is $\hat{\mathbf{u}}$.
7. Randomly choose S radii for the test points, normally distributed between $R_{\min}$ and $R_{\max}$.
8. Remove any test points that fall within a d-sphere of radius $R_{\max}$ from $\mathbf{x}_{\kappa-1}$.
9. For each³ remaining test point, $\mathbf{b}$, calculate and store the FTLE at $\mathbf{b}$ as $\sigma|_{\mathbf{b}}$ using the Poor Man's FTLE approximation.
10. Sort the test points by their FTLE values, and extract the z test points with the largest FTLE.
11. Sort the top z points into ν bins based on spherical angles from $\mathbf{x}_{\kappa}$.
12. Find the average angles $\boldsymbol{\alpha}_{\max}$ associated with the bin with most members.
13. Calculate $\mathbf{x}_{\kappa+1} = R_{\max} \cdot \Phi_C(\boldsymbol{\alpha}_{\max}) + \mathbf{x}_{\kappa}$
14. If $|\mathbf{x}_{\kappa+1} - \mathbf{x}_f| > R_{\max}$:
 - set $\kappa = \kappa + 1$
 - return to step 6

A graphical depiction of this process is presented in Fig 2, for the case $n = 2$.

³This loop can be run in parallel.

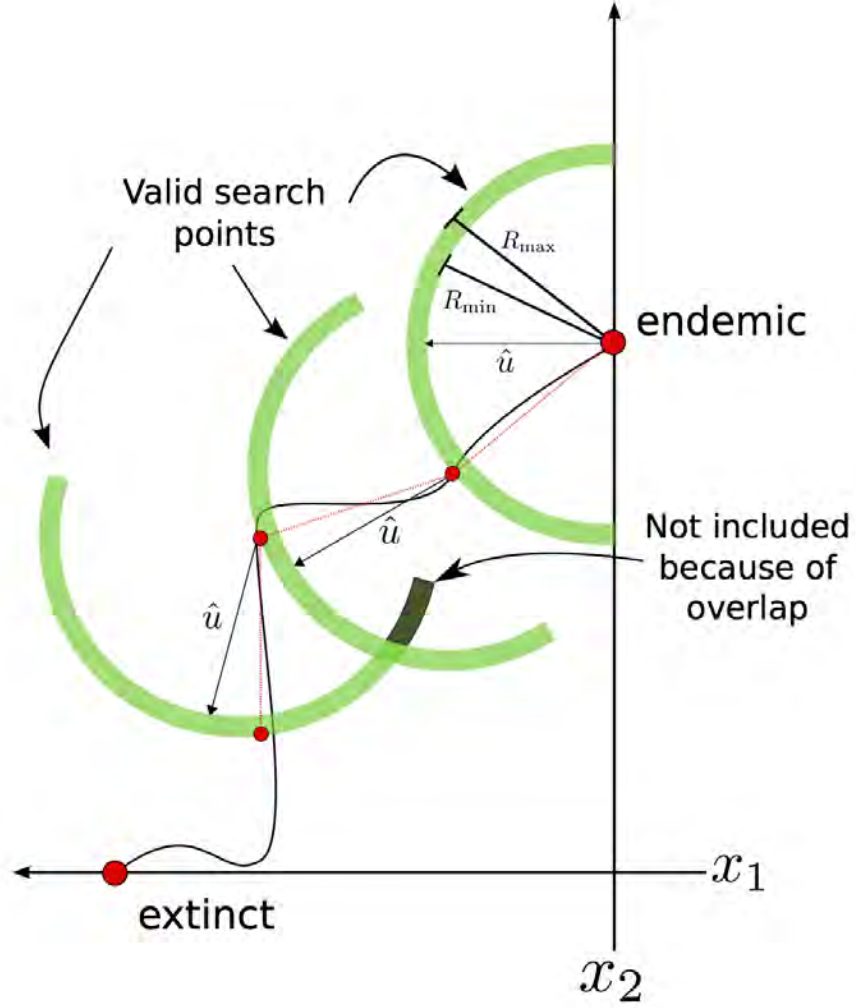


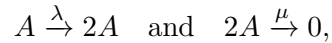
Figure 2: Sample progress of the optimal path growing algorithm in two dimensional state space. Regions of valid test points are shaded in light green for each step of the algorithm. At each step, the hemisphere of valid points has an inward normal aligned with the vector field at that point. The dotted line represents the grown path along the curve, while the solid black line represents the analytical solution.

4 Results

We now present the results of attempting to grow an optimal path to extinction using the algorithm described above on several examples, along with analytical solutions, if applicable.

4.1 Example 1 - Extinction in a Branching-Annihilation Process

A simple system with intrinsic noise fluctuations is extinction in the stochastic branching-annihilation process



where λ and $\mu > 0$ are constant reaction rates [1]. Hamilton's equations are given as

$$\begin{aligned} \dot{q} &= q [\lambda(1 + 2p) - \mu(1 + p)q] \\ \dot{p} &= p [\mu(2 + p)q - \lambda(1 + p)] \end{aligned} \tag{1}$$

where q is transformation on the number of entities, and p is the effective force of the noise.

It is simple to find the fixed points of the system analytically as $(\lambda/\mu, 0)$, $(0, 0)$ and $(0, -1)$. We can find an analytical solution for the optimal path to extinction, given as

$$q = \frac{2\lambda(1 + p)}{\mu(2 + p)}.$$

As a check of our methods, the analytical solution for the optimal path to extinction is plotted against the field of approximate FTLE values in Figure 3. We can see that the analytical solution corresponds nicely to the ridge of local maximum FTLE.

We can grow this optimal path without knowing the analytical solution. Starting at the endemic state $\mathbf{x}_0 = (\lambda/\mu, 0)$, we reach the extinct state $\mathbf{x}_f = (0, -1)$ by following the algorithm above. The successfully grown curve is shown in Figure 3.

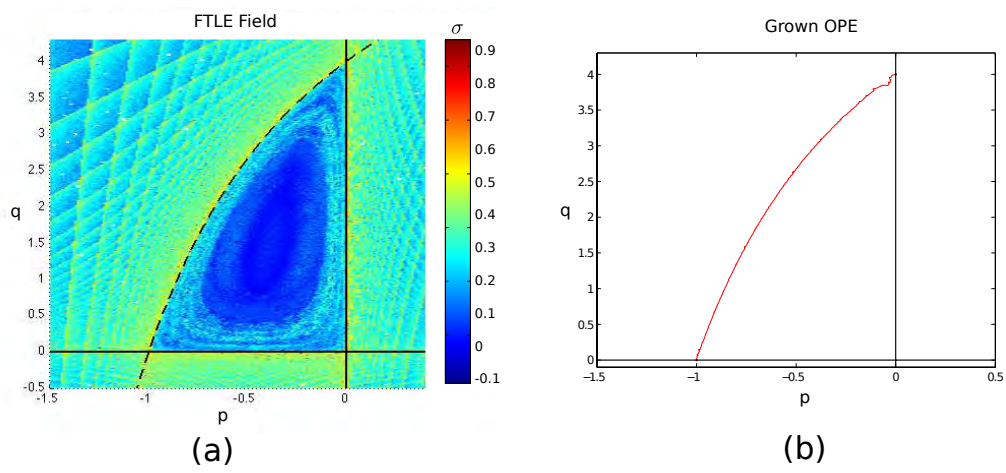


Figure 3: Graphical representation of running forward and backwards (averaged) “Poor Man’s FTLE” algorithm on (1). Analytical solution is plotted in dotted black. Axis are thickened for convenience.

4.2 Example 2 - SIS Epidemic Model with External Fluctuations

A classical Susceptible-Infectious-Susceptible (SIS) epidemiological model is represented by the following system of equations:

$$\begin{aligned}\dot{S} &= \mu - \mu S + \gamma I - \beta IS \\ \dot{I} &= -(\mu + \gamma)I + \beta IS\end{aligned}$$

where μ represents a constant birth/death rate, β represents the contact rate, and γ represents the rate of recovery. If we normalize so that $S + I = 1$, we can rewrite this as a one-dimensional problem:

$$\dot{I} = -(\mu + \gamma)I + \beta I(1 - I).$$

We can then include external fluctuations due to random migrations to/from other populations:

$$\begin{aligned}\dot{I} &= F(I) + \eta(t) \\ F(I) &= -(\mu + \gamma)I + \beta I(1 - I)\end{aligned}$$

where $\eta(t)$ is uncorrelated Gaussian noise with zero mean. Evaluating the Euler-Lagrange equation produces the desired system

$$\begin{aligned}\dot{I} &= p \\ \dot{p} &= \left(\beta I(1 - I) - \kappa I\right)\left(\beta(1 - 2I) - \kappa\right).\end{aligned}$$

where $\kappa = \mu + \gamma$ [1].

The FTLE field, along with the successfully grown curve, is shown in Figure 4.

4.3 Example 3 - SIS Epidemic Model with Internal Fluctuations

We can also look at the version of the SIS epidemic model where the form of the noise is not known beforehand. In this case, Hamilton's equations become

$$\begin{aligned}\dot{I} &= -(\mu + \gamma)I e^{-p} + \beta I(1 - I)e^p \\ \dot{p} &= -(\mu + \gamma)(e^{-p} - 1) + \beta(e^p - 1)(2I - 1).\end{aligned}$$

No analytical solution for the optimal path to extinction exists for this system, but we can use our FTLE growing method. Results obtained from running the "Poor Man's FTLE" on this system, along with the grown path, are presented in Fig. 5.

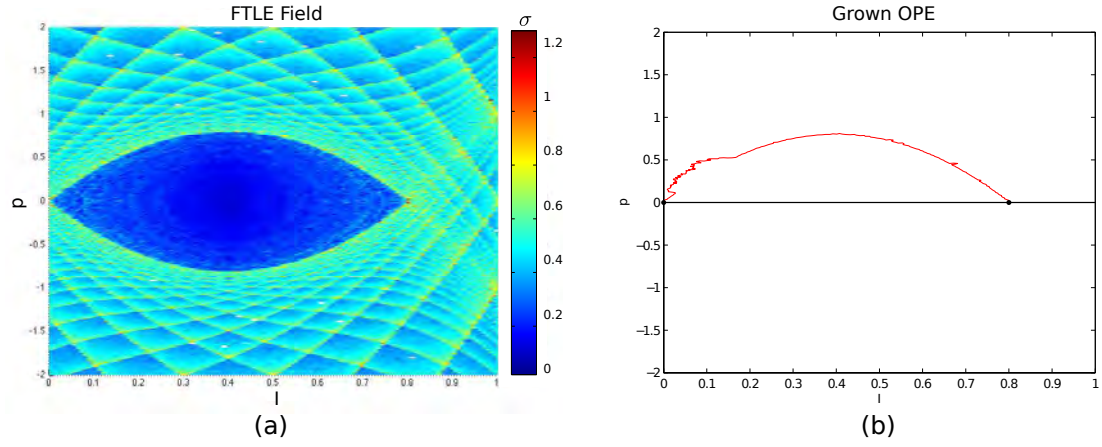


Figure 4: Graphical representation of running forward and backwards (averaged) “Poor Man’s FTLE” algorithm on SIS model with external fluctuations. The optimal path to extinction is easily distinguishable.

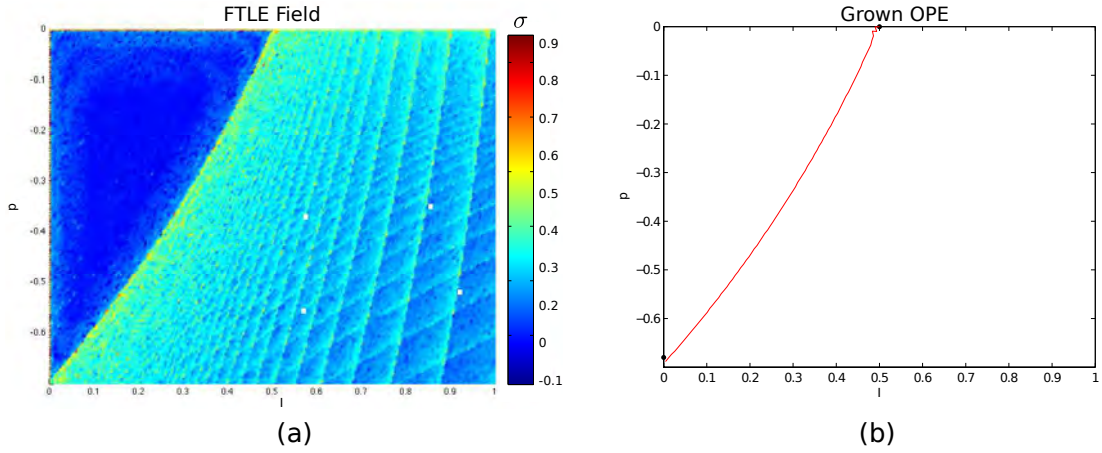


Figure 5: Graphical representation of running forward and backwards (averaged) “Poor Man’s FTLE” algorithm on SIS model with internal fluctuations. The optimal path to extinction is easily distinguishable.

4.4 Example 4 - SIS Epidemic Model without an Adiabatic Assumption

The preceding examples illustrate the effectiveness of this method in growing an optimal path to extinction for systems of two dimensions. We would now like to use this algorithm to find an OPE for a four dimensional case. We will use the SIS model with internal noise, but we will not make the adiabatic assumption that $S + I = 1$ in order to remove a variable.

We begin with the Hamiltonian

$$H(\mathbf{x}, \mathbf{p}) = \mu(e^{p_1} - 1) + \mu x_1(e^{-p_1} - 1) + \mu x_2(e^{-p_2} - 1) + \kappa x_2(e^{p_1 - p_2} - 1) + \beta x_1 x_2(e^{p_2 - p_1} - 1).$$

Hamilton's equations become

$$\dot{x}_1 = \frac{\partial H}{\partial p_1} = \mu e^{p_1} - \mu x_1 e^{-p_1} + \kappa x_2 e^{p_1 - p_2} - \beta x_1 x_2 e^{p_2 - p_1}$$

$$\dot{p}_1 = -\frac{\partial H}{\partial x_1} = -\mu(e^{-p_1} - 1) - \beta x_2(e^{p_2 - p_1} - 1)$$

$$\dot{x}_2 = \frac{\partial H}{\partial p_2} = -\mu x_2 e^{-p_2} - \kappa x_2 e^{p_1 - p_2} + \beta x_1 x_2 e^{p_1 - p_2}$$

$$\dot{p}_2 = -\frac{\partial H}{\partial x_2} = -\mu(e^{-p_2} - 1) - \kappa(e^{p_1 - p_2} - 1) - \beta x_1(e^{p_2 - p_1} - 1)$$

and the critical points of the system are

$$\mathbf{z} = \begin{pmatrix} x_1 \\ p_1 \\ x_2 \\ p_2 \end{pmatrix} \quad \mathbf{z}_{\text{endemic}} = \begin{pmatrix} 1/R_0 \\ 0 \\ (1 - R_0)/R_0 \\ 0 \end{pmatrix} \quad \mathbf{z}_{\text{extinct}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

where $R_0 = \beta/(\mu + \kappa)$.

A simpler form of these equations can be derived if we use a Taylor series expansion approximation ($e^x = x - 1$) to remove the exponentials from the Hamiltonian. We arrive at:

$$\begin{aligned} H(\mathbf{x}, \mathbf{p}) = & \mu p_1 \left(1 + \frac{p_1}{2}\right) - \mu x_1 p_1 \left(1 - \frac{p_1}{2}\right) - \mu x_2 p_2 \left(1 - \frac{p_2}{2}\right) \\ & + \kappa x_2 (p_1 - p_2) \left(1 + \frac{p_1 - p_2}{2}\right) + \beta x_1 x_2 (p_2 - p_1) \left(1 + \frac{p_2 - p_1}{2}\right) \end{aligned}$$

with differentials of

$$\dot{x}_1 = \mu(1 + p_1) - \mu x_1(1 - p_1) + \kappa x_2(p_1 - p_2 + 1) - \beta x_1 x_2(p_2 - p_1 + 1)$$

$$\dot{p}_1 = \mu p_1 \left(1 - \frac{p_1}{2}\right) - \beta x_2(p_2 - p_1) \left(1 + \frac{p_2 - p_1}{2}\right)$$

$$\dot{x}_2 = \mu x_2(p_2 - 1) - \kappa x_2(p_1 - p_2 + 1) + \beta x_1 x_2(p_2 - p_1 + 1)$$

$$\dot{p}_2 = \mu p_2 \left(1 - \frac{p_2}{2}\right) - \kappa(p_1 - p_2) \left(1 + \frac{p_1 - p_2}{2}\right) - \beta x_1(p_2 - p_1) \left(1 + \frac{p_2 - p_1}{2}\right)$$

and a new endemic critical point

$$\mathbf{z}'_{\text{extinct}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -2\frac{R_0-1}{R_0+1} \end{pmatrix}.$$

The extinct state is the same as in the full system.

Attempts at running the optimal path growing algorithm on the above systems did not terminate, although the Euclidean distance to the end point decreased steadily over time. A bug in the algorithm code may exist, preventing it from working in a higher dimensional case. Additional review of the algorithm, as well as more control cases are recommended to determine the validity of this algorithm in higher dimensions.

References

- [1] Eric Forgoston, Simone Bianco, Leah B Shaw, and Ira B Schwartz. Maximal Sensitive Dependence and the Optimal Path to Epidemic Extinction. *Bulletin of Mathematical Biology*, 73(3):495–514, 2010.
- [2] Paul Leopardi. A partition of the unit sphere into regions of equal area and small diameter. *Electronic Transactions on Numerical Analysis*, 25:309–327, 2006.
- [3] G Marsaglia. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 1972.
- [4] I B Schwartz, E Forgoston, S Bianco, and L B Shaw. Converging towards the optimal path to extinction. *Journal of The Royal Society Interface*, 2011.
- [5] Ira B Schwartz, Lora Billings, Mark Dykman, and Alexandra Landsman. Predicting extinction rates in stochastic epidemic models. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(01):P01005, 2009.

A Hyperspherical Coordinates

A point on a d -sphere, which is embedded in $(d+1)$ -space, can be defined either by $d+1$ Cartesian coordinates $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_{d+1}]$ or a distance from the origin, R , along with d polar coordinates: $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_d]$, where the longitude, $\alpha_1 \in [0, 2\pi)$, and all other colatitudes, $\alpha_i|_{i>1} \in [0, \pi)$.

We will define functions $\Phi_C(\boldsymbol{\alpha})$ and $\Phi_P(\mathbf{x})$ to represent conversions from polar to cartesian and cartesian to polar, respectively.

The polar coordinates $\boldsymbol{\alpha}$ are defined such that

$$\begin{aligned} x_1 &= R \cos \alpha_1 \prod_{j=2}^d \sin \alpha_j \\ x_2 &= R \prod_{j=1}^d \sin \alpha_j \\ x_k &= R \cos \alpha_{k-1} \prod_{j=k}^d \sin \alpha_j, \quad k \in \{3, \dots, d+1\}. \end{aligned}$$

To go the other direction (cartesian to polar), use the equations:

$$\begin{aligned} \alpha_1 &= \tan^{-1} \left(\frac{x_2}{x_1} \right) \\ \alpha_k &= \cos^{-1} \left(\frac{x_{k+1}}{\prod_{j=k+1}^d \sin \alpha_j} \right), \quad k \in \{2, \dots, d\}. \end{aligned}$$

Note that since the equation for α_k is dependent on α_{k+1} , it is most convenient to solve for $\boldsymbol{\alpha}$ backwards; i.e. $\alpha_d, \alpha_{d-1}, \dots, \alpha_1$. Also, since α_1 ranges from 0 to 2π , a correction must be made to account for the quadrant it is in. It can be corrected by

$$\alpha_1 = \tan^{-1} \left(\frac{a}{b} \right), \quad \alpha'_1 = \begin{cases} (b > 0 \ \& \ a > 0) & \alpha_1 \\ (b > 0 \ \& \ a < 0) & \alpha_1 + 2\pi \\ (b < 0) & \alpha_1 + \pi \end{cases}$$

where α' is the corrected angle. The radius, R , is analogous to the three dimensional instance

$$R = \sqrt{(\mathbf{x} \cdot \mathbf{x})} = \sqrt{x_1^2 + x_2^2 + \dots + x_d^2 + x_{d+1}^2}.$$

B Generating uniformly distributed hyperhemispheres

To find a random point $\mathbf{x}_r$ in the desired region, create a vector $\mathbf{x}$ of length n , where each entry is a random value between zero and one. Normalize $\mathbf{x}$, and multiply by the desired radius [3]. Repeat as necessary to generate more points. This can be implemented in MATLAB easily:

```
S = 1000; %the number of points on the sphere surface we want
d = 2; % we want points on a d-sphere (in d+1 dimensions)
C = randn(d+1,S); %get random Cartesian points

for ii = 1:S

    tempR = sqrt(dot(C(:,ii),C(:,ii))); %normalize each point
    C(:,ii) = C(:,ii)./tempR;

end
```

To restrict this to a hyper-hemisphere, we need only limit $0 \leq x_1 \leq 1$. If we wish to rotate this hyper-hemisphere about a particular direction, we can just use the function `rotSphere(C,a)` defined below. The argument `a` represents the polar angles of the hemisphere normal vector, $\hat{u}$.

```
function [Cp R] = rotSphere(C,a)
%ROTSPHERE Rotate hypersphere with points "C" about angles defined in "a"

Cp = zeros(size(C));
d = length(a);
R = eye(d+1);

for jj = 1:d %create the composite rotation matrix
    R = R*makeRotMat(a(jj),jj,d);
end

for ii = 1:size(C,2) %for each point, rotate
    Cp(:,ii) = R*C(:,ii);
end

function R = makeRotMat(ang,this_dim,total_d)
%make a single rotation matrix
R = eye(total_d+1);

Rsub = [cos(ang) -sin(ang);
        sin(ang) cos(ang)];

R(this_dim:this_dim+1,this_dim:this_dim+1) = Rsub;
```

C Hypersphere Partitioning and Binning

The optimal path growing algorithm assumes a method exists for partitioning the surface of a d -sphere and sorted points on the surface into such partitions [2]. In the case $d = 1$ this is trivial; one can simply divide the polar coordinate θ of a circle into linearly spaced partitions. For instance, doing `theta = linspace(0, 2*pi, nBins)` in MATLAB would result in a list of boundaries of a circumference partitioned into `nBins` sections.

Unfortunately, this is not so simple in cases where $d > 1$. An attempt to use this method in the case $d = 2$ is shown in Fig. 6. The two polar coordinates θ and ϕ are linearly spaced, however the surface elements resulting from these spacing are clearly of unequal size.

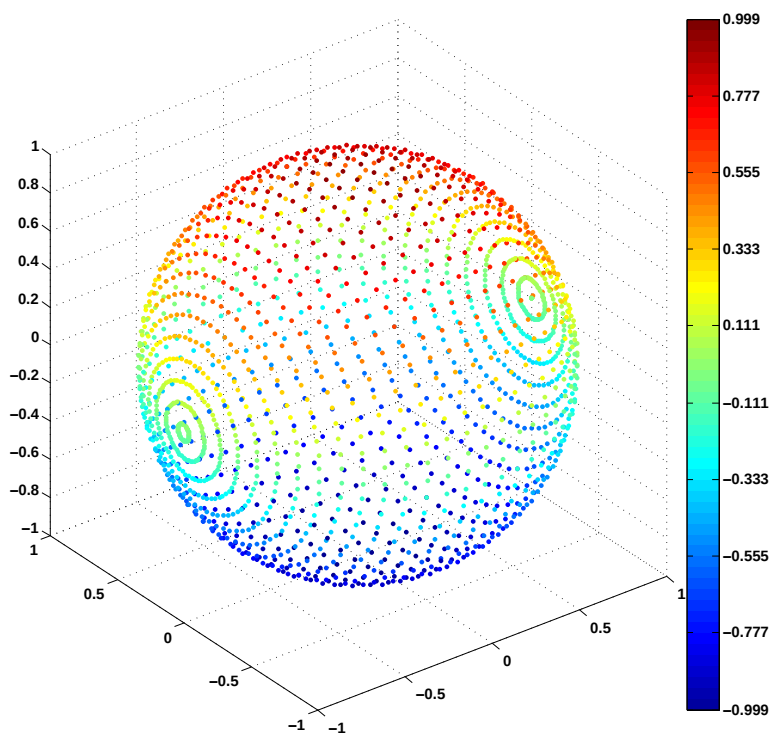


Figure 6: An example 2-sphere (in 3-dimensions) with linearly spaced polar angles. Linearly spaced polar angles lead to inconsistent surface area sizes when $d > 1$.

Instead, we need a method for partitioning the surface of a d -sphere into equally sized d -volume elements. Fortunately, a MATLAB package has been developed for this very purpose. Once the package is installed, simply use the command

```
regions = eq_regions_polar(d,nBins); %polar coordinate intervals
r_centers = eq_region_centers(d,nBins); %polar coordinate centers
```

to produce a matrix which represents `nBins` equally sized partitions of a d -sphere. A 2-sphere generated with this package is presented in Fig. 7.

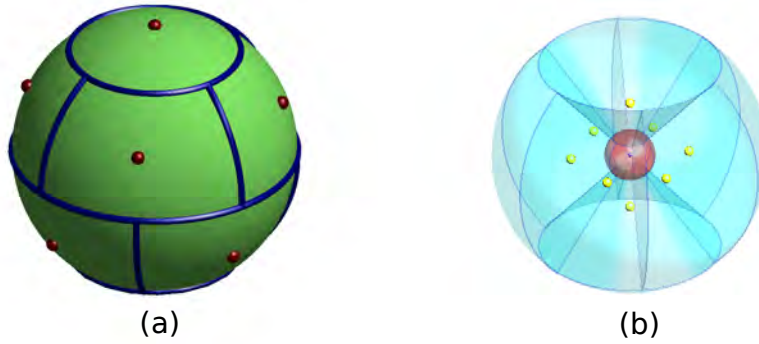


Figure 7: (a) An example 2-sphere with 10 equally sized surface partitions. (b) A stereographic projection of a 3-sphere divided into 10 equally sized surface partitions. Generated by the Recursive Zonal Equal Area Sphere Partitioning Toolbox with the command `project_s3_partition(10)` [2].

Once the partition polar coordinate intervals are known, sorted points on the surface of a d -sphere into bins is a trivial task. For example, to bin points on the surface of a 3-sphere, simply iterate through each partition until a points polar angles are within the of a partition's three intervals.