

UNIVERSITY OF CALIFORNIA, MERCED

**A computational geometric approach for an ensemble-based  
topological entropy calculation in two and three dimensions**

A dissertation submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy  
in  
Applied Mathematics

by

Eric Roberts  
November 2019

Portions of Chapter 1:

© 2019 American Physical Society (APS)

Portions of Chapter 2:

© 2019 American Physical Society (APS)

Portions of Chapter 3:

© 2019 Nature Publishing Group

All other chapters: © Eric Roberts 2020

All Rights Reserved

The dissertation of Eric Roberts, titled A computational geometric approach for an ensemble-based topological entropy calculation in two and three dimensions, is approved, and it is acceptable in quality and form for publication.

---

(Suzanne Sindi) Principal Co-Adviser

---

Date

---

(Kevin Mitchell) Principal Co-Adviser

---

Date

---

(Francois Blanchette) Committee Chair

---

Date

---

(David Strubbe) Committee Member

---

Date

University of California, Merced

2020

Dedicated to Hope, Isabella, and William.



# Acknowledgments

To Kevin and Suzanne...

To David and Francois...

To my collaborators Spencer, Linda, and Amanda...

To my Merced family of friends...

To my mother, father, and sister...

To Hope...

To Isabella and William...

To my grandmother. Rest well.

# Curriculum Vitae

Eric Roberts

## Contact Information

LinkedIn: <https://www.linkedin.com/in/eric-roberts714/>

Website: <https://sites.google.com/view/ericrobertsmath/home>

Email: [ericroberts714@gmail.com](mailto:ericroberts714@gmail.com)

Phone: +1 714 - 325 - 5446

## Summary

Proven track record in developing and implementing algorithms for extracting meaningful information from highly dynamic and nonlinear systems from limited amounts of data. My work is generally applicable to real-world phenomena and can be integrated with other fields, as evidenced by my collaborations with biologists, theorists, and experimentalists in other domains.

## Research Interests

Nonlinear dynamical systems, topological fluid dynamics, and mixing in viscous fluids. More specifically, my interests lie in the investigation of topological approaches to chaotic mixing and numerical methods for quantifying complexity in fluid flows and low dimensional systems.

## Education

**University of California, Merced** **2014 – present**

*PhD, Applied Mathematics*

Co-Advisor: Prof. Suzanne Sindi

Co-Advisor: Prof. Kevin Mitchell

Dissertation Title: A computational geometric approach for an ensemble-based topological entropy computation in two and three dimensions.

**University of California, Los Angeles** **2011 – 2013**

*BS, Applied Mathematics*

## Publications

“*Ensemble-based Topological Entropy Calculation (E-tec) in three dimensions.*” **Eric Robert**, Suzanne Sindi, Spencer Smith, and Kevin A Mitchell. *In preparation*.

“*Investigating self-assembly of microtubule spools and the collective behavior of a biological active nematic.*” Linda Hirst, **Eric Roberts**, Amanda Tan, and Kevin Mitchell. *Nature Physics* (2019).

“*Ensemble-based Topological Entropy Calculation (E-tec).*” **Eric Robert**, Suzanne Sindi, Spencer Smith, and Kevin A Mitchell. *Chaos: An Interdisciplinary Journal of Nonlinear Science* (2019).

“*Topological chaos in a three-dimensional spherical vortex.*” Spencer A. Smith, Joshua Arenson, **Eric Roberts**, Suzanne Sindi, and Kevin A. Mitchell. *Europhysics Letters* (2017).

## Professional Experience

**Institute for Mathematics and its Applications** **2018**

Highly competitive program introduced mathematical and statistical techniques and software employed by industry researchers. Daily technical skill building modules on machine learning techniques, statistical modeling, and optimization methods for data-driven decision-making processes.

## Teaching Experience

**University of California, Merced** **2014 – 2018**

*Teaching Assistant*

Math 23 - Calculus III: Fall 2014 and Spring 2015.

Math 32 - Probability and Statistics: Summer 2015.

Math 122 - Complex Variables: Fall 2017 (lead instructor for three weeks).

Math 125 - Intermediate Differential Equations: Fall 2015.

Math 126 - Partial Differential Equations: Spring 2016, Spring 2017, Spring 2018.

Math 130 - Numerical Analysis: Fall 2018.

**University of California, Merced** **2017 – 2018**

*DESCARTES Mentor*

Funding through NSF-sponsored Data-Enabled Science and Computational Analysis Research, Training and Education for Students (DESCARTES) Program. Responsibilities include the design and implementation of a three-week 2017 Summer course in numerical analysis for second-year undergraduate students and a three-week 2018 Summer course serving as an introduction to data science and machine learning for third-year undergraduate students.

**University of California, Merced** **2014**

*Teaching Associate*

Invited to co-lecture for a single summer appointment.

## Research Funding

**Graduate Student Researcher Appointment** **2019 – Present**

Funding through NSF grant number #1808926 titled “Self-mixing Active Fluids.”

**Graduate Student Researcher Appointment** **2016**

Funding through Army Research Office’s MURI (Multidisciplinary University Research Initiative) grant titled “New Theoretical and Experimental Methods for Predicting Fundamental Mechanisms of Complex Chemical Processes.”

**Applied Mathematics Summer Research Fellowship** **2015**

Awarding institute: The University of California, Merced

## Selected Talks

“*A Framework for an Ensemble-based Topological Entropy Calculation in Three Dimensions.*” 2019 SIAM Conference on Applications of Dynamical System. Snowbird, UT. May, 2019.

“*Maximal Topological Entropy of Braids on a Torus .*” 2019 APS Annual March Meeting. Boston MA. March, 2019.

“*Ensemble-based Topological Entropy Calculation in Three Dimensions.*” 71st Annual Meeting of the APS Division of Fluid Dynamics. Atlanta, GA. November, 2018.

“*Chaotic Advection in Soft Matter Physics.*” Carolina Dynamics Symposium at Agnes Scott College. Decatur, GA. April, 2018.

“*Introducing E-tec: Ensemble-based Topological Entropy Calculation.*” 70th Annual Meeting of the APS Division of Fluid Dynamics. Denver, CO. November, 2017.

“*Introducing E-tec: Ensemble-based Topological Entropy Calculation.*” 2017 Annual Meeting of the Far West Section, Merced, CA. November, 2017.

“*What is Chaos – A Brief History.*” University of California, Merced SIAM Student Chapter SAMPLe Seminar Series. October, 2015.

## Poster Presentations

“*Chaotic Advection in Active Nematics.*” Dynamics Days 2018. Denver, CO. January, 2018.

“*Ensemble-based Topological Entropy Calculation.*” 2017 SIAM Conference on Applications of Dynamical Systems. Snowbird, UT. May, 2017.

“*Computational Methods for Entropy.*” 11th AIMS Conference on Dynamical Systems, Differential Equations and Applications. Orlando, FL. July, 2016.

“*Identifying Chaotic Structure Through Expansion Entropy.*” California Central Valley Regional SIAM Student Chapter Conference. Merced, CA. April, 2016

## Selected Honors and Awards

Summer 2019: University of California, Merced Applied Mathematics Research Travel Fellowship for oral presentation at *SIAM DS 19*, awarded via competitive application.

March 2019: American Physical Society-funded student travel grant awarded for oral presentation at *2019 APS March Meeting*.

January 2018: NSF-funded competitive travel grant awarded for oral presentation and poster presentation at *Dynamics Days 2018* annual meeting.

November 2017: American Physical Society-funded student travel grant awarded for oral presentation at *APS DFD 2017* annual meeting.

Summer 2017: Society of Industrial and Applied Mathematics-funded student travel grant for poster presentation at *SIAM DS 17*.

Summer 2016: University of California, Merced Applied Mathematics Research Travel Fellowship for poster presentation at *11th AIMS Conference on Dynamical Systems*, awarded via competitive application.

Summer 2016: Sindi Lab Travel/Research Award.

UCLA Achievement Scholarship recipient for academic excellence, 2011 and 2012.

UCLA Chancellor’s Blue and Gold Scholarship, awarded to top 300 incoming students, 2011.

Fullerton Community College Person of the Year Finalist, 2011.

## **Leadership and Service**

### **UC Merced Women in STEM**

**2017 – 2019**

2018 University California Student Chapter representative at SIAM Annual Meeting.

Planned and coordinated the second annual California Central Valley SIAM Student Conference.

### **UC Merced SIAM Student Chapter**

**2016 – 2018**

Planned and coordinated the inaugural California Central Valley SIAM Student Conference.

Running SIAM-sponsored 'Math Mingle', a casual weekly meeting designed for providing mentorship and guidance to undergraduate students interested in mathematics.

## **Skills**

Languages (Proficient Knowledge): Matlab/Octave, Python **R**, C++, AMPL.

Environments and Applications: Matlab, L<sup>A</sup>T<sub>E</sub>X, Visual Studio, Bash Scripting

Operating Systems: Unix/Linux, Windows.

# Abstract

A Computational Geometric Approach for an Ensemble-based Topological Entropy Calculation in Two and Three Dimensions

by

Eric John Roberts

Doctor of Philosophy in Applied Mathematics

University of California, Merced

Professor Suzanne Sindi, Chair

From the stirring of dye in viscous fluids to the availability of essential nutrients spreading over the surface of a pond, nature is rife with examples of mixing in two-dimensional fluids. The long-time exponential growth rate of a thin filament of dye stretched by the fluid is a well-known proxy for the quality of mixing in two dimensions. This growth rate in turn gives a lower bound on the flow's topological entropy, a measure quantifying the complexity of chaotic dynamics. In the real-world study of mixing, topological entropy may be hard to compute; the velocity field may not be known or may be expensive to recover or approximate, thus limiting our knowledge of the governing system and underlying mechanics driving the mixing. Central to this study are two questions: *How can stretching rates in two-dimensional planar flows best be computed using only trajectory data?*, and *Can a method for computing stretching rates in higher dimensions from only trajectory data be developed?*

In this spirit, we introduce the Ensemble-based Topological Entropy Calculation (E-tec), a method to derive a lower-bound on topological entropy that requires only finite number of system trajectories, like those obtained from ocean drifters, and no detailed knowledge of the velocity field. E-tec is demonstrated to be computationally

more efficient than other competing methods in two dimensions that accommodate trajectory data. This is accomplished by considering the evolution of a “rubber band” wrapped around the data points and evolving with their trajectories. E-tec records the growth of this band as the collective motion of trajectories strike, deform, and stretch it. This exponential growth rate acts as a lower bound on the topological entropy. In this manuscript, I demonstrate convergence of E-tec’s approximation with respect to both the number of trajectories (ensemble size) and the duration of trajectories in time.

Driving the efficiency of E-tec in two dimensions is the use of computational geometry tools. Not only this, by computing stretching rates in this new computational geometry framework, I extend E-tec to three dimensions using two methods. First, I consider a two-dimensional rubber sheet stretched around a collection of points in a three-dimensional flow. Similar to the band-stretching component of two-dimensional E-tec, a three-dimensional triangulation is used to record the growth of the sheet as it is stretched and deformed by points evolving in time. Second, I calculate the growth rates of one-dimensional rubber strings as they are stretched by the edges of this dynamic, moving triangulation.



# Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Curriculum Vitae</b>	<b>vi</b>
<b>Abstract</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Background . . . . .	1
1.2 Chaotic Advection . . . . .	2
1.2.1 Numerical Considerations . . . . .	3
1.3 Topological Entropy . . . . .	4
1.3.1 Topological Entropy from Braiding . . . . .	6
1.4 Manuscript Overview . . . . .	7
<b>2 2D Ensemble-based Topological Entropy Calculation</b>	<b>10</b>
2.1 Introduction . . . . .	10
2.2 Topological Entropy . . . . .	11
2.3 Overview of E-tec . . . . .	13
2.4 E-tec Algorithm Details . . . . .	17
2.5 E-tec Algorithm Verification . . . . .	21
2.5.1 Chaotic Lid-Driven Cavity Flow . . . . .	21
2.5.2 Period-Three Orbit and Convergence in Ensemble Size . . . . .	25
2.5.3 Topological Entropy for Range of Period Driving Parameter $\tau_f$ . . . . .	27
2.6 E-tec Robustness . . . . .	31

2.6.1	Robustness to Choice of Initial Band . . . . .	31
2.6.2	Algorithm Scaling and FTBE Comparison . . . . .	32
2.6.3	Robustness to Step Size $\Delta t$ . . . . .	34
2.7	E-tec runtime scaling with number of points . . . . .	35
2.8	Conclusion . . . . .	38
<b>3</b>	<b>2D E-tec Applications</b>	<b>40</b>
3.1	Active Matter Mixing . . . . .	40
3.1.1	Introduction . . . . .	40
3.1.2	Active Nematic Fluid . . . . .	41
3.1.3	Experimental Setup . . . . .	41
3.1.4	Extracting Particle Trajectories . . . . .	43
3.1.5	Topological Entropy Calculation for 50 $\mu\text{M}$ ATP Concentration	45
3.1.6	Topological Entropy Calculation from Beads as a Function of ATP Concentration . . . . .	45
3.1.7	Topological Entropy Calculation from Defect Tracking as a Function of ATP Concentration . . . . .	47
3.1.8	Acknowledgments . . . . .	49
3.2	E-tec and Coherent Sets . . . . .	50
3.3	E-tec, Escape Rates, and Unbounded Flows . . . . .	51
3.3.1	Unbounded E-tec Entropy, Bounded E-tec Entropy . . . . .	53
3.3.2	Escape Rates . . . . .	54
3.3.3	E-tec and Trajectory Replacement . . . . .	55
3.4	Conclusion . . . . .	56
<b>4</b>	<b>3D E-tec</b>	<b>58</b>
4.1	Introduction . . . . .	58
4.1.1	Troubles in 3D . . . . .	59
4.1.2	A New E-tec Scheme . . . . .	61
4.1.3	Chapter Summary . . . . .	63
4.2	Dual E-tec . . . . .	63
4.2.1	2D Collapse Event Weight Update Rule . . . . .	65

4.2.2	3D Collapse Event Weight Update Rule . . . . .	70
4.3	Kinetic Delaunay Triangulation Maintenance . . . . .	78
4.3.1	Flips . . . . .	81
4.3.2	Stars, Links, and Ears . . . . .	82
4.3.3	Real and Imaginary Events . . . . .	84
4.3.4	3D Delaunay Maintenance Algorithm Details . . . . .	86
4.4	Topological Entropy Calculation . . . . .	89
4.4.1	2D Dual E-tec Results Using Delaunay Triangulation Maintenance . . . . .	91
4.4.2	3D Dual E-tec Results Using Delaunay Triangulation Maintenance . . . . .	93
4.4.3	3D Dual E-tec Results Using Stirring Rods of the Chaotic Lid-driven Cavity Flow . . . . .	98
4.4.4	3D Dual E-tec's Loss of Information . . . . .	99
4.5	Stretching Strings in 3D Using E-tec . . . . .	102
4.5.1	Encoding the Strings . . . . .	102
4.5.2	Flip23 Event Update . . . . .	106
4.5.3	Flip32 Event Update . . . . .	108
4.5.4	String Stretching Results . . . . .	114
4.6	Future Considerations . . . . .	115
4.6.1	Coupling 2D Sheet Stretching and Cylindrical Surface Tracking . . . . .	115
4.6.2	Parallelizing the Re-triangulation Process . . . . .	116
4.7	Conclusion . . . . .	118
4.8	Acknowledgments . . . . .	118
<b>5</b>	<b>Conclusions and Future Direction</b>	<b>119</b>
5.1	Conclusion . . . . .	119
5.2	Future Work and Considerations . . . . .	120
5.2.1	Three Dimensional Considerations . . . . .	120
5.2.2	E-tec in Higher Dimensions . . . . .	121

<b>A</b>	<b>Expansion Entropy</b>	<b>122</b>
A.0.1	Introduction . . . . .	122
A.0.2	Expansion Entropy Theory . . . . .	123
A.0.3	Numerical Computation . . . . .	124
A.0.4	Expansion Entropy of the Chaotic Lorenz System . . . . .	125
A.0.5	Published Result: Expansion Entropy as an Upper Bound for Topological Entropy . . . . .	128
<b>B</b>	<b>1D Interval Stretching</b>	<b>130</b>
B.1	Chaotic Logistic Map . . . . .	132
B.2	Chaotic Gauss Map . . . . .	134
	<b>Bibliography</b>	<b>137</b>

# List of Figures

1.1	Comparison of two mixing processes, one with zero topological entropy (above) and one with high topological entropy (below). Dye-stirring images adapted from ref. [1], Cambridge Univ. Press. . . . .	5
1.2	A loop entangled between points. The braid operator $\sigma_{2,3}^{-1}$ acts on the loop, deforming it accordingly. The operator subscript indicates the second and third points swapped orderings on the projection axis in black. The superscript indicates the two points swapped in a counter-clockwise fashion (the right-most point passes above). . . . .	7
2.1	<b>Band Deformation.</b> a) The white point strikes and deforms the band (red). b) The white point detaches from the band. Notice the band edge is taut after detachment. c) An initial rubber band stretched between two points on the left with edge weights displayed. A more complicated band on the right. The edge weights correspond to the number of times the band crosses an edge. . . . .	12
2.2	<b>Edge Weights.</b> E-tec counts the number of edges of a rubber band as it is stretched by moving points. As the two bottom points rotate, the red band, initially wrapped around two points, is stretched and folded (left to right). E-tec tracks the growth of this band by assigning a weight to each edge corresponding to the number of times the band passes over this edge. . . . .	14

2.3	<b>Events of E-tec Algorithm.</b>	a) As point 5 moves right, triangle (2, 4, 5) collapses and inverts orientation. Two core triangles are retriangulated, with the new edge shown as dashed. The initial edge weight of 2 for segment (2, 4) is shifted to segments (2, 5) and (4, 5). The blue-highlighted triangle (2, 4, 5) is the new outer triangle of point 5. It records which triangle collapse would be needed for the band to “snap back” taut, thereby undoing the collision. b) As point 7 moves to the right, outer triangle (2,5,7) collapses and the band edges (2, 7) and (5, 7) straighten into (2, 5). The three core triangles within pentagon (1,2,7,5,6) are reconfigured into three new core triangles (1,2,6), (2,5,6), and (2,5,7). Point 7 is still a candidate for future detachment, with new outer triangle (3,4,7), which also happens to be a core triangle. c) In blue is a combined core and outer triangle (2,4,5). As point 5 moves to the right and this triangle collapses, the band returns taut around segment (2,4). Three core triangles (1,2,4), (2,3,5), and (3,4,5) are reconfigured, with the new edge shown as dashed. Collapsed triangle (2,4,5) (previously shaded) remains as a core triangle. . . . .	16
2.4	<b>E-tec Algorithm Flowchart.</b>	As described in Sec. 2.3, E-tec employs computational geometry techniques for tracking the evolution of a piecewise-linear band. Full details are given in Sect. 2.4. . . . .	18
2.5	<b>Numerical Example of an E-tec Implementation.</b>	a) Initial data points with the band wrapped around two points (in red). The core Delaunay triangulation (in blue, dotted) is constrained to include the red band edge. b) Final data point positions at $T = 20$ , the triangulation, and the stretched band evolved under the motion of the trajectories. Dynamics is given by model in Sect. 2.5 with $\tau_f = 0.96$ . c) E-tec output: the number of band edges as a function of time (blue). The slope of the best-fit line (red, dashed) is the topological entropy estimate. . . . .	22

- 2.6 **Dynamics of Chaotic Lid-Driven Cavity Flow.** We depict streamlines of the flow, Eq. (2.3). a) Motion under the first half-period,  $n\tau_f \leq t < (n + 1/2)\tau_f$ . b) Motion under the second half-period,  $(n + 1/2)\tau_f \leq t < (n + 1)\tau_f$ . c) Illustration of a period-three orbit  $r_i$ . Each color (blue, green orange) represents the trajectory evolving forward one period. . . . . 24
- 2.7 **E-tec Analysis of the Chaotic Lid-Driven Cavity Flow.** We show E-tec results on trajectories governed by Eq. (2.3) with  $\tau_f = 0.96$ , guaranteeing the existence of a period-three orbit, seen in Fig 2.6c. a) We show E-tec results when considering only 3 points close to the period-three orbit and contained in period-three islands. We consider an initial band around the two right points (top). This band evolves (bottom) into a highly stretched band (edge weights in red) around all three points by  $T = 15$ . b) We consider the same 3 initial points, but add 72 random trajectories (top). The dynamics are more complex (bottom, weights omitted). c) The growth rate in the number of edges, i.e. our estimate of the topological entropy, for (a, red) and (b, blue) is the same. This indicates the entropy is driven by the period-three islands as also shown by Ref. 2. d) The coherent period-three islands, noted in Refs. 2, 3, and 4, are clearly seen in the Poincaré return map of a long-lived trajectory in blue. Ten long-lived trajectories inside the islands are shown in red. . . . . 26

2.8	<b>Convergence of E-Tec in the Length and Number of Trajectories.</b>	<p>a) We demonstrate convergence of E-tec to <math>h_{po3} = 0.9624</math> with increasing sample size and trajectory duration. For E-tec, the same initial band is stretched under ensembles of increasing size. All trajectories are sampled from outside the islands in the chaotic lid-driven flow with period-driving parameter <math>\tau_f = 0.96</math>. The entropy reported at time <math>T</math> is the fitting slope and 95 percent confidence interval to the log of the total number of edge weights over time <math>t</math> for the range <math>t \in [5, T]</math>. b) We demonstrate consistency between E-tec and FTBE results (calculated using the freely available Matlab package <code>braidl</code> [5]). The same ensembles of trajectories are used for both. To stay consistent with the <code>braidl</code> calculations, E-tec reports the entropy as the fit from initial time to reported time (or rather, it is the fit from <math>t \in [0, T]</math>). c) E-tec output using the 100 point ensemble with a single trajectory added into one, two, and three of the periodic islands.</p>	28
2.9	<b>Verification of E-tec for Increasing <math>\tau_f</math>.</b>	<p>E-tec topological entropy results over a range of <math>\tau_f</math> values using increasing ensemble sizes. We compare to the estimate of topological entropy from directly stretching a material line [2] and through another topological technique, homotopic lobe dynamics [6, 7]. . . . .</p>	29
2.10	<b>Stretched Band Visualization.</b>	<p>E-tec band stretching due to flow advection for period-driving parameters a) <math>\tau_f = 0.80</math>, b) <math>\tau_f = 0.96</math>, and c) <math>\tau_f = 1.05</math>. Colorbar corresponds to edge weights. All bands are stretched by ensembles of 1000 uniformly distributed trajectories.</p>	30



2.11	<b>Initial Bands.</b> E-tec output is the logarithm of the sum of edge weights as a function of time. E-tec’s estimate of topological entropy is the best-fit linear slope through this data. Here we show 10 different outputs from E-tec for the same set of 100 trajectories. In each case, we chose a different pair of points around which to stretch our band. Despite some initial differences in the increase in edge weights due to initial adjacent points staying close to one another ( <i>left</i> inset), eventually all the bands grow at similar rates ( <i>right</i> inset). When we fit the exponential growth rate, starting at $T = 5$ , we find the values for each of the 10 bands agree within 5 decimal places and average out to 0.9617.	32
2.12	<b>E-tec Runtimes.</b> Runtime comparison of E-tec and <code>braidlab</code> , a freely available Matlab package implementing the FTBE calculation. Both used the same trajectories from the chaotic model flow for $\tau_f = 0.96$ . All computations were completed using a 2.8 GHz Intel Core i7 processor. . . . .	33
2.13	<b>Relative Error as a Function of Step Size.</b> The effect of time step $\Delta t$ on the relative error in the topological entropy calculation with respect to the reference time step $\Delta t^* = 10^{-4}$ . Graph displays calculations done on two separate ensembles of size 1000 and 100. . .	35
2.14	Schematic of the movement of a single point (circled in red) against a background of stationary points. The number of E-tec collapse events is proportional to the number of average core triangles that would have an equivalent area to that of the diagonal blue rectangle. For FTBE calculations, the number of braid generators created by this same process is equal to the number of points in the larger black rectangle.	36
2.15	The collapse event rate scaling for three numerical examples: random motion, chaotic lid-driven cavity flow (for $\tau = 0.96$ ), and rigid body rotation. . . . .	37

3.1	Fluorescence microscopy image of the active nematic fluid consisting of a densely-packed microtubule network in 2D. The two types of topological defects that emerge are pictured in the inset. <i>Image credit:</i> Amanda Tan [8]. . . . .	42
3.2	a) Bright field image of tracked bead trajectories that remain unbroken (colored in for effect). b) Lagrangian particle image tracking results. All bead particles found in the frame are highlighted in dark blue. All bead particles belonging to unbroken trajectories used for later analysis are highlighted in cyan. Pinned particles near the boundary are those that leave the field of view. c) Number of trajectories that are actively moving in the field at any given time for one data set (red) and the union of data sets (blue). . . . .	43
3.3	E-tec results for the 50 $\mu\text{M}$ ATP concentration. a) Initial bead positions with the rubber band (red) stretched between two of them. b) At the final time, the band is stretched and folded over itself in an intricate pattern. c) The number of band edges grows exponentially in time, giving the topological entropy $h_{top}$ . . . . .	46
3.4	E-tec topological entropy measurements using bead trajectories in blue. Also pictured are the Lyapunov exponents found by averaging the exponential rate of separation of 10 bead pairs for each concentration. . . . .	47
3.5	A comparison between the active nematic $+\frac{1}{2}$ defects and period-3 coherent islands from the chaotic test flow used in Ch. 2 a) Fluorescence microscopy image of the active nematic fluid with topological defects marked. White circles denote $+\frac{1}{2}$ defects. A yellow triangle denotes a $-\frac{1}{2}$ defect. b) A band stretched by the braiding of the three period-3 islands (circled). <i>Image a) credit:</i> Amanda Tan [8]. . . . .	48
3.6	E-tec topological entropy measurements using defect trajectories in yellow. Also pictured are the same Lyapunov exponent averages and bead trajectory topological entropies from Fig. 3.4. . . . .	49

3.7	Displayed are bands around three coherent sets a) Initial bands placed around a three subset of points that lie in the interior of the coherent sets. (Blowup in inset.) b) The bands evolved forward in time, with negligible stretching. c) The number of edges in the band exhibits no net growth in time, indicating that the band encloses a coherent set. .	51
3.8	(Left) Unbounded trajectories initially seeded in circle of radius 5. (Right) Same initially-seeded trajectories are frozen once leaving the boundary. . . . .	52
3.9	Blue, red, and yellow solid curves are E-tec results using bounded and unbounded trajectories as input. The diamonds represent the sum of the E-tec results and the bounding regions' escape rates. In purple is the independent expansion entropy calculation which acts as an upper bound for the topological entropy. . . . .	54
3.10	The natural log of the number of remaining trajectories in each bounding region as a function of time. Bounding boxes are in a) and bounding circles are in b). On the right are tables summarizing the escape rate data extracted from the plots. Listed data includes $N$ , the number of initially seeded trajectories, and the escape rates and corresponding escape rate errors. . . . .	55
3.11	Updated Fig. 3.9 from above, with new E-tec topological entropy estimates (in black) for the trajectory sets with replacement. . . . .	56
3.12	Final configuration of a band stretched by E-tec. Edge weights are colored by weight, with color bar shown to right (log scale coloring). Trajectory replacement is used, so many trajectories are frozen just outside of boundary, as seen in subsequent zooming . . . . .	57

4.1	<b>Collisions in 3D:</b> Depiction of the two collisions (point-face and edge-edge) that may occur among points, edges, and faces of a 3D triangulation as points move. Triangulation faces are colored to help better visualize the stretching of potential elastic sheets. a) Point 5 (hollow) moves up and collides with the triangular face (1, 2, 3). As a result, tetrahedra (1, 2, 4, 5), (1, 3, 4, 5), and (2, 3, 4, 5) are added to the triangulation to replace the newly invalid tetrahedron (1, 2, 3, 4). A potential elastic sheet wrapped around face (1, 2, 3) is now represented with weights on faces (1, 2, 5), (1, 3, 5), and (2, 3, 5). b) From the same left configuration as above, point 5 moves to the right and into the page, resulting in an intersection between edges (1, 5) and (2, 3). The resulting re-triangulation depends on the number of tetrahedra sharing edge (2, 3). Furthermore, the new weight representation of a potential red sheet is left ambiguous. . . . .	60
4.2	<b>Strings Anchored by Edges:</b> Pieces of a rubber string (in red) are wrapped around triangulation edges (a, c) and (c, d). . . . .	61
4.3	<b>Encoding Rubber Sheets in the Triangulation:</b> Two types of two-dimension rubber sheets and their Dual E-tec weight representation. a) A rubber sheet folded around the middle triangular face on the left (the folding over points is not drawn in 3D). This sheet is represented in Dual E-tec by weights along each edge it passes through. Weighted edges are in red. b) A rubber sheet in the form of a plane is on the left. Weighted edges that represent the sheet are shown on the right. . . . .	62
4.4	A Delaunay triangulation of forty points. Eight triangle circumspheres with empty interiors are shown. . . . .	63

4.5	<b>Comparison of Original and Dual E-tec:</b> The same band and point configuration are displayed in all three sub-figures. However, the band representation among the edges and weights in the original scheme in a) differ from Dual E-tec representations in b) and c) a) In the original E-tec scheme, this is the only possible triangulation among these five points in which the band may be represented. b) The same triangulation as the left, this time with the triangulation's Dual-E-tec weight representation. c) A second possible triangulation of the five points and corresponding edge weights, with the changed edge dashed for emphasis. . . . .	64
4.6	<b>E-tec and Dual E-tec Numerical Band Initialization:</b> a) The original E-tec representation of an initial band (in red). Notice the constrained Delaunay triangulation. b) The Dual E-tec representation of the band anchored by same two points in a). The triangulation edges passing through the band are shown in red. Notice this is an unconstrained Delaunay triangulation. . . . .	65
4.7	<b>2D Collapse Event:</b> Triangles, edges, and labeled variables before and after a collapse event. Edge weights (in red) and refer to the number of times a band passes through the labeled edge. Auxiliary lower case variables at points refer to the number of times a band passes through the two edges adjacent to the labeled point. . . . .	66
4.8	<b>Example of a 2D Collapse Event:</b> A specific example displaying new edge weight $C'$ . . . . .	67
4.9	<b>Dual and Original E-tec Numerical Comparison:</b> Comparison of the two methods using the same 100 point trajectory set and initial band. The confidence intervals for both slopes (starting at fit time $T = 5$ ) overlap at (0.9620, 0.9624). . . . .	70

4.10 **3D Dual E-tec Collapse Event Update:** A point-face collision and subsequent re-triangulation is depicted. As the hollow point 5 move north, edge (4, 5) is added and tetrahedra  $T$  and  $B$  are converted into the three final tetrahedra  $X$ ,  $Y$ , and  $Z$ . Sheets wrapped taut around points 1 and 2 are shown in red. The Dual E-tec representation of this sheet is given by weights along edges in which the sheet passes through. 71

4.11 **Adjacent Tetrahedra Vertex Partition Variables:** The top and bottom tetrahedra involved in the collision event. In this projection, two of the shared vertices are on the left and one is on the right. This represents one of three combinatorially possible projective side views of the pair of tetrahedra. The red lines are possible ways that a surface could pass through both tetrahedra (viewed edge-on).  $L$  and  $M$  represent two of the  $V^T$  variables (partitioning the top vertices 2-2 and 1-3 respectively).  $N$  and  $O$  are the analogous  $V^B$  variables. The variables  $a, b, c$ , and  $d$  encode how  $L$  and  $M$  split and re-distribute to form  $N$  and  $O$ . . . . . 75

4.12 **Split Crossing Variables:** The three different side views of the two tetrahedra before the collapse event. Bold and boxed letters designate a new set of defined variables. . . . . 76

4.13	<b>New Local Triangulation Updates:</b> In all sub-figures, the hollow point moves north or northeast. a) Instead of waiting for the 2D point-edge collision to occur before implementing the local re-triangulation rules, depicted on the left, I apply the same local update on the right at an earlier time, specifically when the Delaunay criterion is broken, i.e. once the blue circumcircle contains the fourth point on its boundary. b) In blue, the circumsphere for points 1, 2, 3, and 4. On the right, once point 5 breaks the Delaunay criterion by moving onto the circumsphere, the tetrahedra are re-triangulated with the addition of red edge 4-5. Note that the two tetrahedra (1, 2, 3, 4) and (1, 2, 3, 5) are replaced by three that all share the newly introduced edge. c) In blue, the circumsphere for points 1, 2, 3, and 5. As point 5 moves northeast, the circumsphere dilates and moves and the Delaunay criterion is broken as point 4 enters. The tetrahedra are re-triangulated in the same manner as b). . . . .	79
4.14	<b>Flips:</b> In a), a flip22 is used to transition among the two valid triangulation configurations of four points in 2D. In b), either a flip23 (left to right) or flip32 (right to left) is used in 3D depending if the triangulation configuration of five points contains two or three tetrahedra before the flip. . . . .	82
4.15	<b>Stars and Link:</b> The stars and links for red point $p$ . (Left) The 2D star in a) contains all lines and triangles connected to $p$ . The 3D star in b) contains all lines, triangular faces, and tetrahedra connected to $p$ . (Right) The 2D link of $p$ in a) contains all adjacent points and the edges connecting them, while the 3D link in b) contains all adjacent points and their connecting edges in addition to the outer triangular faces. . . . .	83
4.16	<b>2-Ears and 3-Ears:</b> In three dimensions, a link for some vertex viewed from outside. Three adjacent faces in the link sharing a common vertex for a 3-ear in dark gray. In light gray, two adjacent faces sharing only one common edge form a 3-ear. . . . .	84

- 4.17 **3-ear and 2-ear Convexity Tests:** Point  $p$  is shown with the two types checks. a) Faces (1, 2, 3), (1, 3, 4), and (2, 3, 4) in  $\text{link}(p)$  form a 3-ear of  $p$ . This 3-ear is valid (left) when the line connecting  $p$  and 4 passes through shaded triangle and invalid (right) if not. b) Faces (1, 2, 3) and (2, 3, 4) in  $\text{link}(p)$  form a 2-ear of  $p$ . This 2-ear is valid (top) if the line connecting 1 and 4 passes through the shaded triangle and invalid (bottom) otherwise. . . . . 85
- 4.18 **Real and Imaginary Events in 2D:** Moving point  $p$  is in red. The link of  $p$  consists of points in black and the edges connecting them. The events necessary for the maintenance of a Delaunay Triangulation are detailed. a) Each gray circle represents the circumcircle through a real triangle of  $p$ . A real triangle is comprised of an edge on the link and the single hollow point outside of the link that this edge is attached to. A real event occurs when the red point  $p$  enters a gray circumcircle. A flip22 must then occur between  $p$  and the real triangle to maintain a Delaunay triangulation. b) The blue circle represents the circumcircle through an imaginary triangle consisting of three adjacent points on the link of  $p$ . An imaginary event is triggered once  $p$  leaves a blue circumcircle, resulting in a flip22 between  $p$  and the imaginary triangle. 86
- 4.19 **Moving a Single Point:** Real and imaginary events are detected and processed as point  $p$  (in red) moves along its trajectory. Circum-circles of real and imaginary triangles are shown in gray and blue. (For simplicity, only one imaginary triangle is shown.) a) The first event detected is a real event.  $p$  is moved to the intersection (middle). The triangulation is updated with a local flip22 (right) and two new real triangles emerge. Not pictured is the modification of the imaginary triangles. b) The next event detected is an imaginary event.  $p$  is moved to the intersection (middle). The triangulation is updated with another flip22 (right). . . . . 87



4.20	<b>Real Event Convexity Test:</b> A real event occurs between $p$ and real tetrahedron (1, 2, 3, 4). a) A flip 23 is performed if the line connecting $p$ and 4 passes through the shaded triangular face of the other three points. b) If the line connecting $p$ and 4 does not pass through the shaded triangular face, a third tetrahedron ( $p$ , 2, 3, 4) exists in the triangulation of the five points. A flip 32 is then performed. . . . .	90
4.21	<b>Original Motion vs. Motion Maintaining a Delaunay Triangulation:</b> An ensemble of 12 trajectories from the chaotic lid-driven cavity flow is used. The true entropy is $h_{po3} = 0.9624$ . Red line represents the stretching of a single band under the original flow evolution. All other lines represent stretching of single bands under the evolution of the flow maintaining a Delaunay triangulation. Bold blue line represents the initial band yielding the most stretching after the time $T = 4$ . . . . .	92
4.22	<b>Entropy Convergence:</b> Topological entropy $h_{top}$ results using Delaunay triangulation-maintaining event operators as a function of ensemble size. Dashed line is the true entropy result for the original motion of this flow. . . . .	92
4.23	<b>Events per Period:</b> Average number of re-triangulation events per period required to maintain a Delaunay triangulation. The $N^{1.05}$ guideline shows the event scaling for the original flow. . . . .	93
4.24	<b>3D Chaotic Lid-driven Cavity Flow:</b> Three trajectories belonging to the same chaotic flow from previous sections are used. Here, each individual point is uniformly distributed along $0 < z < 1$ for the entirety of its trajectory. Projecting all trajectories down to the $z = 0$ plane gives the original 2D dynamics. . . . .	94
4.25	<b>Total Number of Edges in the Triangulation:</b> Total edges are plotted as a function of time. Increases or decreases to the edge totals result from the flip23 and flip 32 operations. . . . .	95

4.26	<b>Sheet Initialization:</b> An initial triangulation of points and the Dual E-tec edge weight representation of two rubber sheets along the planes $x = 2.5$ and $x = 3.5$ . Weighted edges of the triangulation are shown in red. Integer weight corresponds to the number of times an edge passes through a sheet. . . . .	96
4.27	<b>Topological Entropy Results for Three Initial Sheet Configurations:</b> Three sets of initial sheets are each tested on the 20 pt., 50 pt., and 100 pt. ensembles: a) two planes at $x = 2.5, 3.5$ , b) one plane through $y = 0$ , and c) one hundred planes evenly spaced up and down the $y$ -axis. Planes in a) are chosen such that no plane crosses through the period-2 islands (or rods, in 3D). Topological entropy is stated in inset boxes and is given by the slope of the best-fit line through the natural log of edge weight sums as a function of time. Slope and line fitting begins at $T = 5$ . . . . .	97
4.28	<b>Weighted Edge Ratios:</b> As a function of time, the ratio between the number of weighted edges to the number of total edges in the triangulation. . . . .	98
4.29	<b>Chaotic Lid-driven Cavity Flow in 3D:</b> Cartoon of the 2D period-3 island motion is displayed in the $z = 0$ plane. Raising these islands in three dimensions results in three cylinders that resemble stirring rods. Trajectories in black are sampled from the islands and raised to some uniformly distributed $z$ -coordinate. . . . .	99
4.30	<b>Stirring Rod Entropy Results:</b> Logarithmic sum of edge weights as a function of time for the three initial sheet configurations. For reference, in black is the best result obtained using random ensembles. . . . .	100

4.31 **Dual E-tec Loss of Information:** A cylindrical surface and corresponding Dual E-tec weight representations through a three-tetrahedra configuration in a) and after a flip32 event in b). Likewise, two sheets passing through the same tetrahedra configurations in c) and d). While the two-tetrahedra configurations have differing weights in a) and c) (boxed for emphasis), all edges have the same weight after the flipping events in b) and d). In e) is a cylindrical surface passing through a tetrahedron while intersecting no edges. It has zero edge weight and thus cannot be encoded into the triangulation. . . . . 101

4.32 **Encoding Strings in a Single Tetrahedron:** Pieces of a rubber string (in red) are wrapped around edges (a, d) and (c, d). The string is represented with nonzero edge weights of 2 and 1 corresponding to the number of times the string wraps around each edge. All other edges have zero weight. . . . . 103

4.33 **Topological Diagram of Neighboring Tetrahedra:** a) On the left is the two tetrahedra configuration with labeled faces and shared face  $T$ . On the right is the simplified topological diagram of this configuration. Here, open holes represent the faces and the two full points represent the two tetrahedra. b) On the left is the three tetrahedra configuration that share a single edge running vertical. Exterior faces have integer names and are the same as a). The interior faces adjacent to the shared edge are labeled  $u_1, u_2$ , and  $u_3$ . (Faces  $u_2$  and  $u_3$  are colored for clarity.) . . . . . 105

4.34 **Flip23 Update Rule Example:** On the left are three colored pieces of string entwined in the two tetrahedra configuration. A flip23 occurs and the string pieces are mapped to the new configuration. The update rules moves pieces in the following order: red piece is mapped from 1-3 to 1- $u_2$  and 3- $u_2$ , the blue piece is mapped from 1- $T$  and 1'- $T$  to 1-1', and the green piece is mapped from 3- $T$  and 2'- $T$  to 3- $u_1$  and 2' -  $u_1$ . 106

4.35 **Flip32 Update Rule Example:** a) Three colored pieces of string entwined in the three tetrahedra configuration. b) The update rule moves pieces in the following order: the red piece is mapped to  $1-u_3$  and  $2-u_3$ , the blue piece is mapped to  $1'-u_3$  and  $2-u_3$ , and the green piece is mapped to  $2-2'$ . c) The transition to the two tetrahedra configuration occurs. The red piece is mapped to  $1-2$ , the blue piece is mapped to  $1'-T$  and  $2-T$ , and finally the green piece is mapped to  $2'-T$  and  $2-T$ . . . . . 109

4.36 **Proof of Concept:** The growth of a rubber string stirred by the edges of an evolving triangulation of six trajectories specially curated from the chaotic lid-driven cavity flow coherent islands. For reference is a line with slope  $h_{po3} = 0.9624$ , the exact flow entropy ultimately produced by the braiding of the islands. . . . . 115

4.37 **String Stretching from Random Trajectories:** The dashed orange line shows string growth using 20 random trajectories in each of the three islands. the other results come from nested sets of trajectories that are uniformly and randomly distributed among the space in which the flow is contained. . . . . 116

4.38 **Parallelizing Procedure:** A Delaunay triangulation in 2D (containing four auxiliary boundary points) is split up into four quadrants. In this depiction, four sets of trajectories interior to the four shaded regions will be processed in parallel. After, to complete the time step, remaining trajectories (those initially on the shaded region boundaries or those exiting their boundary regions) will be processed. . . . . 117

A.1	Expansion entropy results for the chaotic Lorenz system. Plotted in blue is the computation of $\ln \hat{E}_t$ versus $T$ using 20 samples of $N = 100$ uniformly distributed initial conditions. The expansion entropy estimate is the slope of the best-fit red line. a) Best-fit line slope is $0.901 \pm .011$ . Here, the restraining region $S$ is large enough such that no trajectory leaves $S$ . b) Best-fit line is fitted until time $T = 30$ . The slope is $0.884 \pm .020$ . A smaller restraining region $S$ is used. Expelled trajectories are left out of the sum. . . . .	127
A.2	We choose $N = 1.5 \times 10^6$ points distributed randomly and a ball of radius 1.5 units centered at the origin as the restraining region $S$ . We plot the exponential growth of both the product of singular values greater than one and the single largest singular value a) Data is parsed into 10 different samples of 150000 points. We estimate expansion entropy to be between 1.0188 and 1.0554. b) Same data is parsed into 50 samples of 30000 points, yielding an estimate of .98241 and 1.0125. . . . .	129
B.1	Time series plot of three orbits $b, g,$ and $r$ . Please note, the superscript ordering of subintervals given up and down at a specific orbit $n$ are independent of previous orderings. . . . .	131
B.2	Interval stretching entropy estimate for the chaotic logistic map with parameter $\sigma = 4$ . Entropy is given as best-fit line through the solid data. a) $h_{stretch} \approx 0.32702$ for $N = 5$ orbits. b) $h_{stretch} \approx 0.68976$ for $N = 50$ orbits. c) $h_{stretch} \approx 0.69305$ for $N = 500$ orbits. d) Entropy results as a function of the number of orbits used. Correct entropy of $\log(2)$ is shown for reference. Colored lines represent the iterate $T$ at which the best-fit line is taken. . . . .	133
B.3	Interval stretching (or copying) entropy $h_{stretch}$ of the logistic map for a range of $\sigma$ values. 1000 orbits are used for each computation of $h_{stretch}$ , with different ending fit times $T$ shown in differing colors. The expansion entropy for the same parameter values are shown. Exact topological entropy values are shown in green markers. . . . .	134

B.4	Graph of discrete Gauss Map with fixed point line as reference. . . .	135
B.5	a) Expansion Entropy results using 20 samples of $N = 100, 1000,$ and 10,000 orbits. b) Interval stretching entropy $h_{stretch}$ as a function of sample size $N$ . Expansion Entropy results from a) are shown starred.	136

# Chapter 1

## Introduction

### 1.1 Motivation and Background

A common approach to studying physical systems is through their representation as dynamical systems. A dynamical system represents a fundamental tool for analyzing and modeling at every length scale. In simplest terms, they describe how a system evolves over time. For example, dynamical systems modeling is instrumental in studying population growth in landscapes with limited resources [9–11], analyzing long-time behavior of predator-prey interactions [12, 13], predicting stock market fluctuations [14–16], and discovering bacterial growth patterns [17].

Intertwined in the study of dynamical systems is the study of complexity. Complexity arises in models both intricate – slight variations in experimentally-estimated parameters result in large-scale changes in the dynamics of blood coagulation models comprised of many coupled ordinary differential equations [18] – and simple – the nature of the three-body problem solution evaded the greatest scientific minds for over a century. In fact, system complexity can give rise to unpredictability and render long-term forecasts generally impossible, a fact first alluded to by Henry Poincaré in his studies of the deceptively simple three-body problem. In it, Poincaré established the existence of an infinite number of aperiodic solutions to the problem [19–21]. In truth, Poincaré’s discovery set the groundwork for the theory of chaos. The field was later formalized over half a century later with the advent of the computer. In

1961, Yoshisuke Ueda [22, 23] and Edward Lorenz [22, 24, 25], using numerical calculations, independently discovered that small perturbations in initial conditions grow exponentially and result in vastly different system behavior. Systems exhibiting this sensitivity to initial conditions are deemed chaotic.

Trajectories and orbits originating from chaotic systems are oftentimes mistaken for random behavior. In truth, chaotic systems are deterministic and all future behavior is fully determined. These two seemingly contradictory statements allude to the very difficult and tedious nature of chaotic systems research. Focused efforts in better understanding the field have led to better prediction analysis for a large number of applications, the best known being weather and climate forecasting [24, 26].

## 1.2 Chaotic Advection

Though analysis of chaotic dynamical systems is difficult, the presence of chaos affords certain advantages to particular applications. Chief among these is the field of fluid dynamics. Fluid systems undergo efficient mixing and scalar transport in the presence of chaos. This is especially useful for fluids at the low Reynolds number regimes where viscosity or small length scales govern the dynamics (think peanut butter or microfluids). In this low Reynolds setting, turbulent flow, a well-known facilitator of mixing (think cream mixing in coffee with the single whisk of a spoon), is impossible to induce.

Shifting the focus to deterministic fluid dynamics, flow complexity is connected to the chaotic nature of particle trajectories – that is, chaotic flow trajectories have a sensitive dependence to initial conditions, one of the hallmarks of chaos theory. This phenomenon is called chaotic advection [27, 28]. Though flows undergoing chaotic advection are deterministic and complexity drives uncertainty, their study sheds light on the mixing and transport processes that occur in a wide range of natural phenomena. Such knowledge aids greatly in a wide variety of natural and industrial fluid systems, including the large-scale dispersion of pollutants in the Earth’s atmosphere and oceans [29]; for example, understanding how regions of fluid remain isolated from each other helps predict the fate of oil spills [30, 31]. Understanding mixing in the



rapidly developing field of microfluidics [32,33] could lead to new classes of self-mixing active solvents that further our understanding of the kinetics of mass transport and chemical reactions. Obvious industrial applications include the optimization of stirring devices in viscous fluids, such as the rod-stirring devices used to effectively knead dough, pull taffy [34,35], or manufacture glass compounds [34,36].

### 1.2.1 Numerical Considerations

A variety of techniques have been used to quantify complexity and uncertainty in chaotic dynamical systems theory. These tools include the finite-time Lyapunov exponent (FTLE) field [37, 38], which measures the exponential rate of separation between points in a small neighborhood; the finite-time entropy (FTE) field [39], a probabilistic approach to measuring local stretching and determining the uncertainty in a trajectory’s final position; and operator-theoretic methods, such as the eigenfunctions and eigenvalues of the Koopman operator [40].

However, a problem remains - a prohibitively large number of trajectories may be required if the phase space is too large or trajectories and linearizations may simply be too expensive to compute, as with some experimental measurements. The critical challenge for extracting statistically meaningful information for numerous physical problems is often the adequate sampling of system trajectories, either from experimental data or from large-scale simulations. When only particle trajectories are available, like those collected from oceanic floats or airborne tracers, data sets are oftentimes sparse. Few methods give adequate topological entropy results for sparse input. Fewer methods yet are able to detect topological structures that may drive the mixing. Fuzzy clustering detection of Lagrangian coherent structures [41] and the braiding technique for topological entropy calculations come to mind [5, 42], though these methods fail to scale linearly in the number of particle trajectories, thus rendering them inefficient when the analysis of a highly complex system requires a high point density.

To develop an algorithm that succeeds where others have failed, we aim to carefully extract the maximum information from the limited number of trajectories available.

The collective motion of all trajectories moving through phase space in tandem encodes global information that is not contained in any individual trajectory. That is, extra information is “hiding” in an ensemble of classical trajectories, which is not exploited in a trajectory-by-trajectory approach. We exploit this idea by focusing on the numerical evolution of a two-dimensional material-curve, whose growth rate is shown to be equivalent to the topological entropy [43–47], which measures the exponential proliferation of distinguishable orbits [48].

### 1.3 Topological Entropy

In deterministic dynamical system theory, the amount of complexity that is present in a  $d$ -dimensional system may be quantified by entropy. Two common entropies, metric and topological, can be formally described by considering two trajectories of length  $T$  that can only be distinguished if they are, at any point in time, further than some resolution  $\epsilon$  apart. We expect the number of these  $\epsilon$ -distinct trajectories to increase as both  $T \rightarrow \infty$  and  $\epsilon \rightarrow 0$ . If this growth is asymptotically exponential, the rate  $h$  is the entropy. We distinguish between metric and topological entropy in that topological entropy counts *all*  $\epsilon$ -distinct trajectories, while metric entropy counts those trajectories that are *typically*  $\epsilon$ -distinct, or rather, the set of orbits that have nonzero measure with respect to an invariant measure [49]. In this manner, the topological entropy is the upper bound for the metric entropy.

The formal topological entropy definition above is ill-suited in capturing the second hallmark of chaotic fluid dynamics: the exponential stretching and folding of packets of fluid into long, thin fluid filaments. In a manner more consistent with this idea, topological entropy  $h$  can be estimated in two-dimensional flows by embedding an initial material-curve, analogous to a line of dye, of length  $L_0$  in the system and estimating its growth under the evolution of the flow [46]. At long times, the length  $L(t)$  of the curve as a function of time  $t$  grows exponentially as

$$L(t) \approx L_0 e^{ht}. \tag{1.1}$$

Because positive topological entropy captures the strength of the stretching and folding processes inherent to chaotic flows, it is often used as a common proxy for the quality of mixing occurring in a flow. This is displayed in the two mixing protocols in Fig. 1.1. Here, the lines of dye are the material-curves being stretched. The top protocol results in dye that stretches and folds and grows in time, though this growth is not exponential yields a low topological entropy. However, we find the stirring protocol on the bottom results in exponentially stretched line of dye.

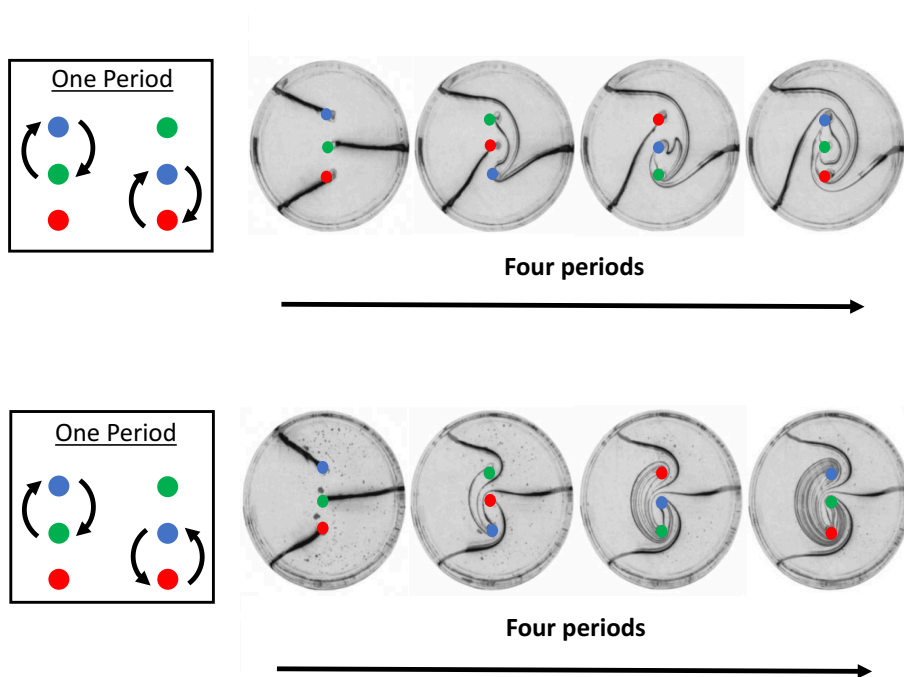


Figure 1.1: Comparison of two mixing processes, one with zero topological entropy (above) and one with high topological entropy (below). Dye-stirring images adapted from ref. [1], Cambridge Univ. Press.

Direct computation of the curve's evolution is troublesome in chaotic flows, requiring an exponentially growing number of trajectories to maintain sufficient point density of the curve. This problem is only exacerbated in three dimensions and higher. A generalization of topological entropy called expansion entropy [50] considers the ensemble-averaging of Jacobian singular values scales to higher dimensions for all flows and requires no computing or measuring of multidimensional surfaces.

However, this method requires many millions of trajectories and their subsequent linearizations. Finite-time Lyapunov exponent fields remain popular for analysis of chaotic flows and detecting special flow structures, though intimate knowledge of governing equations is required for generating many trajectory pairs. Worse yet, many of these techniques are not well-suited for the analysis of real-world experiments, as data is oftentimes some combination of noisy, sparse, and short-lived.

### 1.3.1 Topological Entropy from Braiding

A natural alleviation to these restriction is to use methods requiring only potentially sparse sets of particle trajectories as input, with no reliance on any detailed knowledge of the velocity field; namely, topological methods. In particular, braiding techniques were first applied to two-dimensional fluids to compute the topological entropy of either flows containing periodic orbits [51] or of particular multi-rod stirring protocols [1, 52] that induce topological chaos. The single Artin algebraic braid [53] representation is extracted from 2D point trajectory data. This braid uniquely encodes how periodic trajectories or rods exchange positions with respect to an arbitrary projection axis, thus capturing the dynamics of time-ordered trajectory data. Braid operators are collected by ordering points along an arbitrarily chosen projection axis. As ordered points  $(p_1, p_2, p_3, \dots)$  evolve forward, neighbors swap places along the projection axis. At each event, neighboring points  $p_i$  and  $p_j$  that swap places are recorded as an operator  $\sigma_{p_i, p_j}^{\pm 1}$ , where the superscript denotes a clockwise or counterclockwise swap. Braids are then simply a collection of time-ordered operators. The topological entropy of the braid, typically calculated using braid’s matrix representation [1, 54].

Topological entropy calculations of braids were later generalized to any arbitrary, two-dimensional, aperiodic point trajectory data in chaotic flows by Thiffeault, Finn, Allshouse, and Budišić [55–58]. They introduce Finite-Time-Braiding Entropy [5, 42], an estimation of the topological entropy, which may be calculated by replacing the initial material line  $L_0$  in Eq. 1.1 with a “loop” anchored between two points that acts as a rubber band. The Dynnikov coordinate representation [59] is used to encode entangled loops by listing the number of times the loop passes above and below each

point and the number of loop strands between points. (For an arbitrary set of points, each possible entangling of a loop has a unique Dynnikov representation.) Using the coordinates, loops are deformed under the action of braid operators. A cartoon depicting a braid operator acting on a loop is pictured in Fig. 1.2. The exponential rate of deformation of the loop acts as a generalization to the material curve stretching in Eq. 1.1 and is denoted the Finite-Time Braiding Exponent.

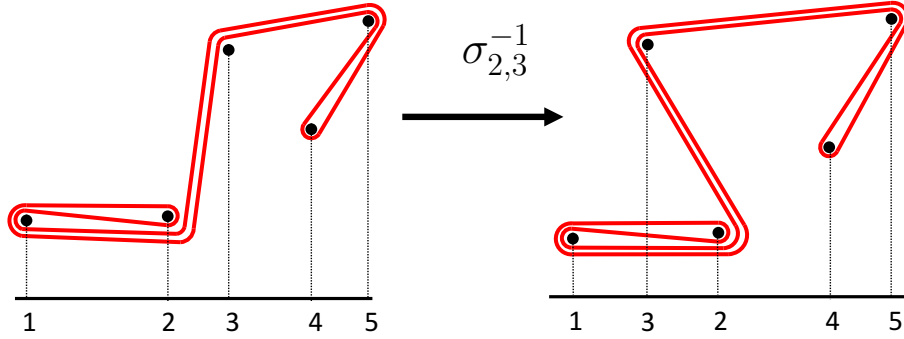


Figure 1.2: A loop entangled between points. The braid operator  $\sigma_{2,3}^{-1}$  acts on the loop, deforming it accordingly. The operator subscript indicates the second and third points swapped orderings on the projection axis in black. The superscript indicates the two points swapped in a counterclockwise fashion (the right-most point passes above).

Though a popular metric for computing topological entropy from trajectory data, Braiding Exponent calculations scale quadratically in the number of points, rendering them unwieldy for systems requiring many points. Worse yet, this method has no current extension to three dimensions or higher [41], as two dimensional lines are not obstacles for points in three dimensions.

## 1.4 Manuscript Overview

We use this as motivation for a new data-driven topological entropy calculation from system trajectories. To develop a method for computing topological entropy that will efficiently scale up to higher dimensions in both system size and modeling time, one must be clever about extracting the maximum information from the limited trajectories available. To achieve this, much like Braiding Exponent theory, the focus needs

to be on topological information, not geometric information. In essence, the relative motion of trajectories through phase space encodes global information that is not contained in any one individual trajectory. That is, extra information is “hiding” in an ensemble of classical trajectories, which is not exploited in a trajectory-by-trajectory or Monte Carlo approach.

I introduce a new algorithm based on topological analysis designed to take advantage of such additional information: the Ensemble-Based Topological Entropy Calculation, or E-tec. Much like the work of Thiffeault, Finn, Allshouse, and Budišić in two dimensions, E-tec requires only system trajectories as input and computes a lower bound on the topological entropy by deforming a loop, hereby referred to as a band. However, by forgoing the stretching of bands by braid operators and adopting a computational geometry framework, E-tec scales much more favorably in the number of trajectories than its braiding predecessor.

In Chapter 2, I outline the procedural steps of the algorithm. Using a time-periodic and chaotic lid-driven cavity flow, I demonstrate E-tec convergence to topological entropy in ensemble size  $N$  and near- $N \log N$  runtime scaling, representing a significant improvement over the quadratic runtime scaling of the finite-time braiding exponent technique that motivated E-tec research. Furthermore, I demonstrate that E-tec can be integrated with other fields by applying it to real-world phenomena. In Chapter 3, I use experimentally obtained trajectories to verify the mixing properties of a microfluidic active matter system.

Next, motivated by the study of advective transport in complex three-dimensional fluid flows and higher dimensional applications, I recognize a critical need for new topological approaches capable of analysis in higher-dimensional systems, particularly when sampling of phase space is sparse. In Chapter 4, the computational geometry foundation on which E-tec is founded is scaled up to three dimensions and represents the first three-dimensional topological entropy calculation requiring only system trajectories and no linearizations or knowledge of the governing equations. Two methods for capturing stretching are described, implemented, and discussed here. In one, a two-dimensional rubber sheet is deformed and stretched under the evolution of points. In the second, a one-dimensional rubber string is stretched and folded by the edges of

a moving, dynamic triangulation. The growth of the sheet and string both represent a lower bound on the topological entropy. Procedural details for both methods are given in the chapter, including the derivation of a second E-tec scheme for recording re-triangulation events.

Lastly, a summary of contributions and a discussion of future research directions are presented in Chapter 5.

In Appendix A, I highlight my numerical implementation of expansion entropy, the generalization of topological entropy that is capable of higher dimensional analysis. Requiring millions of trajectories, I verify the expansion entropies of a number of discrete maps and the three dimensional chaotic Lorentz system with dual attractors.

In Appendix B, I introduce a precursor to E-tec which accounts for the stretching of bands anchored between one dimensional orbits projected as a time series in two dimensions (orbit position as a function of time).

# Chapter 2

## 2D Ensemble-based Topological Entropy Calculation

### 2.1 Introduction

Our goal is to compute material-line stretching rates using only 2D particle trajectories, like those collected from oceanic floats [56,60] or fluorescent beads in microfluidic systems [61,62]. These data sets may be sparse, and hence may not fully sample all of the 2D space. We are motivated by Budišić, Allshouse, and Thiffeault [42,56,63], who use braiding theory to compute a lower bound for topological entropy of flows from such data sets. The initially embedded material-curve is thought of as an elastic line whose growth rate is computed using the collective motion of all available trajectories moving through space in concert. In essence, the relative motion of an ensemble of trajectories in space encodes global information that is not contained in any one individual trajectory. That is, extra information is “hiding” in an ensemble of trajectories, which is not exploited in a trajectory-by-trajectory approach.

In this paper, we focus on these underlying stretching and folding processes that drive mixing in two dimensional fluids. We apply computational geometry techniques to develop a 2D algorithm titled the Ensemble-based Topological Entropy Calculation (E-tec), which may be downloaded at [10.5281/zenodo.1405656](https://zenodo.org/record/1405656). E-tec achieves three main goals: a) estimation of a lower bound to the topological entropy on data sets,



b) convergence to the topological entropy as ensemble size increases, c) linear scaling in runtime with the length of trajectories and  $N^k \log N$  scaling with the number of trajectories  $N$ . (Values of  $k$  range from  $1/3 \leq k \leq 3/2$  and typically  $k \lesssim 1$ . We point the reader to Sect. 2.7 for a discussion.) E-tec does not require the flow to be area preserving or incompressible.

The remainder of this paper is broken up into six sections. We first review topological entropy (Sect. 2.2) and then summarize (Sect. 2.3) and give procedural details (Sect. 2.4) of our E-tec algorithm. E-tec performance is then evaluated on a chaotic, lid-driven cavity flow as a test case (Sect. 2.5) and we show that results are consistent with the braiding approach. Next, we demonstrate E-tec's robustness and show evidence that the E-tec runtime compares favorably to braiding algorithms (Sect. 2.6). Finally, we discuss details regarding E-tec's runtime scaling and computational bottlenecks (Sect. 2.7).

## 2.2 Topological Entropy

As discussed in Sect. 1.3, topological entropy is the growth rate of the number of distinguishable orbits [49]. In two-dimensional flows, topological entropy  $h$  can be estimated by embedding an initial material-curve, e.g. a line of dye, of length  $L_0$  in the system and estimating its growth under the evolution of the flow [46]. At long times, the length  $L(t)$  of the curve as a function of time  $t$  grows exponentially as

$$L(t) \approx L_0 e^{ht}. \quad (2.1)$$

Thus, direct computation of the curve's evolution is troublesome in chaotic flows since the length is expected to grow exponentially fast, which requires an exponentially growing number of trajectories to maintain sufficient point density of the curve. Other techniques for extracting topological entropy operate on a trajectory-by-trajectory basis, i.e. ensemble-averaging some quantity (such as the Jacobian singular values) computed one trajectory at a time. This is the approach taken in recent work on *expansion entropy* [50], a generalization of topological entropy, which unlike Eq. 2.1,

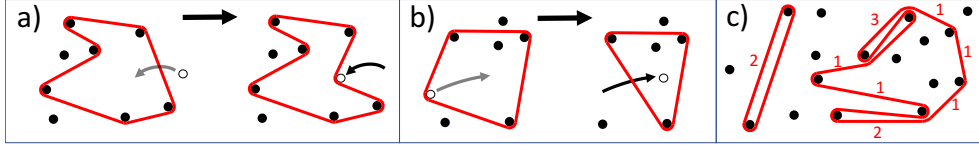


Figure 2.1: **Band Deformation.** a) The white point strikes and deforms the band (red). b) The white point detaches from the band. Notice the band edge is taut after detachment. c) An initial rubber band stretched between two points on the left with edge weights displayed. A more complicated band on the right. The edge weights correspond to the number of times the band crosses an edge.

scales to higher dimensions for all flows and requires no computing or measuring of multidimensional surfaces.

As an alternative approach for 2D systems, a lower bound to the topological entropy may be computed with a finite number of trajectories and no detailed knowledge of the velocity field. The material-curve to be advected is represented by a taut elastic loop that wraps tightly around trajectories that strike it. Since an advected material-curve may be continuously deformed into this taut loop given the same trajectory evolution, the need for maintaining material-curve point density is eliminated. The loop is stretched and folded over itself exponentially many times in a chaotic flow. Its exponential growth rate is a lower bound to the full system's topological entropy [56].

In this more topological setting, braiding theory has been used to compute this lower bound. The Finite-Time-Braiding-Exponent (FTBE) method [42] evolves the loop forward using the entanglement of a finite number of trajectories. However, this method scales quadratically in the number of points  $N$  due to the braid approach requiring  $\mathcal{O}(N^2)$  algebraic generators per unit time. This renders braiding exponent calculations unwieldy for systems requiring many trajectories.

To develop a computationally efficient method to estimate a lower bound on the topological entropy of a planar flow that scales sub-quadratically in the number of points  $N$ , we compute the stretching rate of an advected elastic curve directly. Referring now to the elastic curve or loop as a rubber band, we use the same FTBE idea of trajectories working in concert to stretch and fold the band. The E-tec algorithm achieves this using the same input: i) a set of (typically aperiodic) trajectories

$\{x_i(t), y_i(t)\}$  that are discretized over time  $t_1, t_2, \dots$  and ii) a user-specified, non-self-intersecting elastic band which wraps around a set of trajectories. The output is the number of edge segments in the band as a function of time. However, instead of using a braid representation to compute the stretching of the band, E-tec computes this stretching, and thus the topological entropy, *directly* by using a triangulation to detect all point-band collisions.

In summary, E-tec tracks the crossing of a trajectory with *only* its neighboring edges in the triangulation, unlike the braiding method which concerns itself with *each* trajectory’s relative position with every *other* trajectory along a projection axis. This idea leads to a more favorable sub-quadratic runtime scaling of  $\mathcal{O}(N^k \log N)$ , where  $1/3 \leq k \leq 3/2$ . (For a detailed discussion about the two methods’ runtime scaling in the number of points, we refer the reader to Sect. 2.7.) The idea of using an advected dynamic triangulation to compute topological entropy was first proposed by Marc Lefranc [64–66]. Lefranc’s work was restricted to the entropy generated by periodic orbits, and he did not develop a general algorithm to implement this. To our knowledge, this work is the first attempt to fully generalize Lefranc’s ideas to aperiodic orbits.

## 2.3 Overview of E-tec

We first give an overview of E-tec and forgo the details to the next Section (Sect. 2.4). E-tec computes how an initial, closed, piecewise linear, non-self-intersecting rubber band in  $\mathbb{R}^2$  evolves under an ensemble of trajectories. The vertices of the band coincide with trajectories from the ensemble. When trajectories strike the band, they do not penetrate it but stretch it like a piece of elastic (Fig. 2.1a). In this manner, the band is stretched and folded, typically producing a growing number of edges wrapping around each other. Our algorithm tracks the configuration of the band. Care must also be taken in finding when and how a trajectory detaches from an edge. This detachment results in two band edges returning taut (Fig. 2.1b), in much the same way a tight string will return taut once plucked (stretched) and released (undoing the stretching). Each band edge is assigned an integer weight  $\omega$  indicating the number of times the

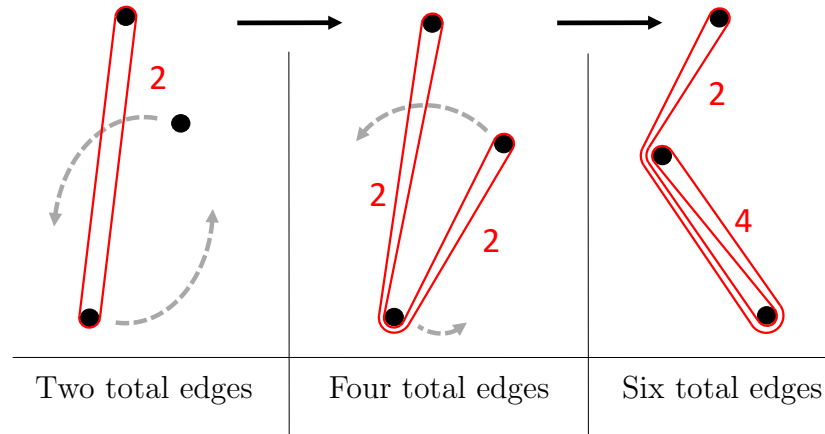


Figure 2.2: **Edge Weights.** E-tec counts the number of edges of a rubber band as it is stretched by moving points. As the two bottom points rotate, the red band, initially wrapped around two points, is stretched and folded (left to right). E-tec tracks the growth of this band by assigning a weight to each edge corresponding to the number of times the band passes over this edge.

band stretches across it (Fig. 2.1c and Fig. 2.2). For chaotic advection, the total weight of the band will grow exponentially, as shown in Sect. 2.5. This exponential growth rate is a lower bound to the true topological entropy of the dynamical system. Even though the weight of all the edges grows exponentially, the number of unique edges is bounded.

E-tec efficiently tracks band growth by simply shifting edge weights to the appropriate edges when a point collides with, or detaches from, the band. A key component of the algorithm is the detection of all relevant point-edge collisions. We achieve this by maintaining a triangulation of all trajectories for all times. First, edge weights are determined corresponding to the initial placement of the band. Next, the data points may be triangulated in any manner consistent with the initial placement of the band. We choose a Delaunay triangulation [67] constrained such that each weighted initial edge representing the initial band is included in the triangulation. For any initial band, E-tec's computation of the evolved band growth is independent of the initial triangulation. Note that the algorithm is fast enough to run and compare many different initial bands in a reasonable time.

The triangles that make up the triangulation are called *core* triangles. Each edge of the stretched band lies within the triangulation, so that each time a point strikes the band, the orientation of one of the core triangles will be inverted. We refer to this inversion as a triangle collapse. All band deformations will be detected since band edges remain in the core triangulation. The triangulation must be updated upon any triangle collapse. This update is *local* to the detection of each event, resulting in the rearrangement of edges and triangles near the collision only (illustrated in Fig. 2.3a). Similarly, the only edge weights that are shifted are those involved in the collision. The update process is independent of both the number of points  $N$  and the number of triangles.

In addition to collisions, we need to detect when a trajectory detaches itself from a band edge. E-tec records which edges of the band are candidates for detachment by storing the triangle made up of the outer-most band edges attached to each point, i.e. typically the most recent edges to have struck a point. These triangles are called *outer* triangles and are shown in blue in Fig. 2.3. Unlike the core triangles, the outer triangles do not form a triangulation of space. Rather, there is simply one outer triangle for each vertex crossed by the band. When a point detaches from the band, its corresponding outer triangle collapses and inverts its orientation. After the outer band edge peels off the point, there may remain other band edges still wrapped around the point. (Follow point 7 in Fig. 2.3b for an example.) In this case, E-tec recalculates and stores the new outer triangle. Note that the outer triangle of a given point can always be recalculated from just the weights of all edges adjacent to the point. Thus, E-tec must track when both core and outer triangles collapse.

The triangulation update process following an outer triangle inversion remains local, though the process differs from the core triangle inversion update in one fundamental aspect: the local re-triangulation is constrained to contain the band that remains taut. This creates a possibly non-unique choice in edges needed to complete the triangulation. As an example, notice that edge (1, 5) could have replaced edge (2, 6) to complete the triangulation in Fig. 2.3b. Because of this, E-tec will not generally recover the initial triangulation away from the band if trajectories are run forward and then exactly backward in time. However, the algorithm *is* time-reversible in that

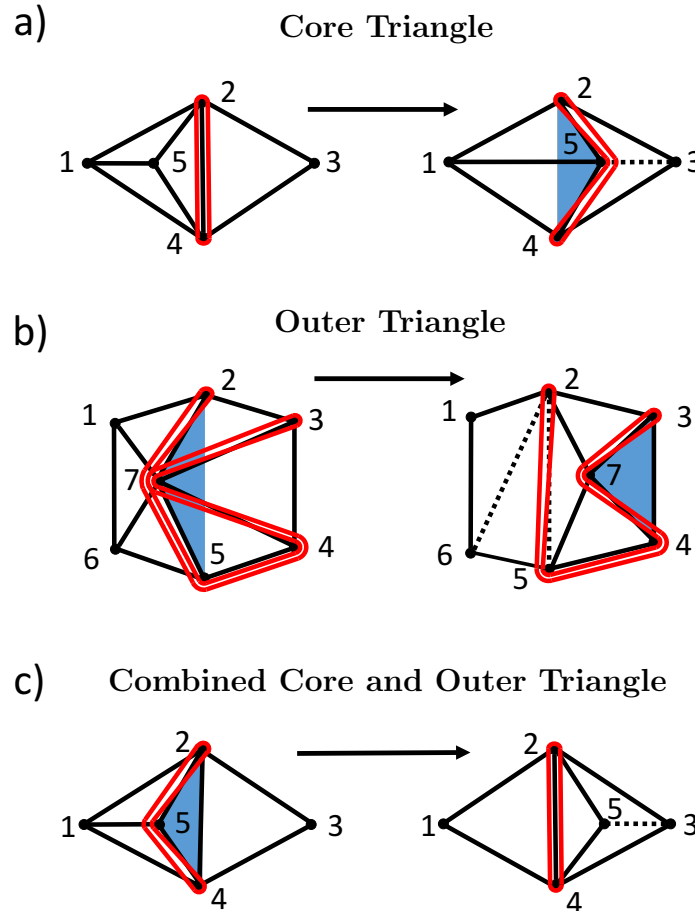


Figure 2.3: **Events of E-tec Algorithm.** a) As point 5 moves right, triangle  $(2, 4, 5)$  collapses and inverts orientation. Two core triangles are re-triangulated, with the new edge shown as dashed. The initial edge weight of 2 for segment  $(2, 4)$  is shifted to segments  $(2, 5)$  and  $(4, 5)$ . The blue-highlighted triangle  $(2, 4, 5)$  is the new outer triangle of point 5. It records which triangle collapse would be needed for the band to “snap back” taut, thereby undoing the collision. b) As point 7 moves to the right, outer triangle  $(2, 5, 7)$  collapses and the band edges  $(2, 7)$  and  $(5, 7)$  straighten into  $(2, 5)$ . The three core triangles within pentagon  $(1, 2, 7, 5, 6)$  are reconfigured into three new core triangles  $(1, 2, 6)$ ,  $(2, 5, 6)$ , and  $(2, 5, 7)$ . Point 7 is still a candidate for future detachment, with new outer triangle  $(3, 4, 7)$ , which also happens to be a core triangle. c) In blue is a combined core and outer triangle  $(2, 4, 5)$ . As point 5 moves to the right and this triangle collapses, the band returns taut around segment  $(2, 4)$ . Three core triangles  $(1, 2, 4)$ ,  $(2, 3, 5)$ , and  $(3, 4, 5)$  are reconfigured, with the new edge shown as dashed. Collapsed triangle  $(2, 4, 5)$  (previously shaded) remains as a core triangle.

the band returns to its initial configuration after running the trajectories backwards to their initial positions.

In summary, there are two kinds of events that must be detected: the collapse of either a core or outer triangle. In the given time interval, these events are detected by finding the time for which their area first goes through zero. This time of first collapse is simply the appropriate root of the area quadratic polynomial, which is formed from the linear interpolation of triangle point positions. (For any reader interested in the scaling of the number of events with the number of trajectories used, we refer them to Sect. 2.7.) Once these events are detected, they are put in a time-sorted list and processed in order. Each event is “fixed” by locally updating the core triangulation, outer triangles, and edge weights. In the course of fixing an event, we may need to add or remove events from the event list. Event lists become large for densely-packed ensembles, though E-tec parses through each event and performs each subsequent triangulation update efficiently, as verified in Sect. 2.5. A flowchart summarizing the E-tec algorithm is given in Fig. 2.4. The algorithm steps found here are detailed in the following section.

## 2.4 E-tec Algorithm Details

This section details our implementation of the E-tec algorithm.

**Input:** The following inputs are required by the algorithm:

1. The precomputed (or experimentally measured) trajectories.
2. An initial, non-self-intersecting rubber band stretched around a sequence of data points, specified by the set of edges connecting pairs of data points. This is represented as a counterclockwise ordering of this set of points. It is often convenient to choose an initial band that encloses two distant points.

**Output:** E-tec tracks the evolution of the band, as we will describe below, and outputs:

1. The state of the stretched rubber band as a function of time, recorded as a (core) triangulation of all data points and a set of edge weights of this triangulation.
2. The sum of all band edge weights  $\omega$  as a function of time.

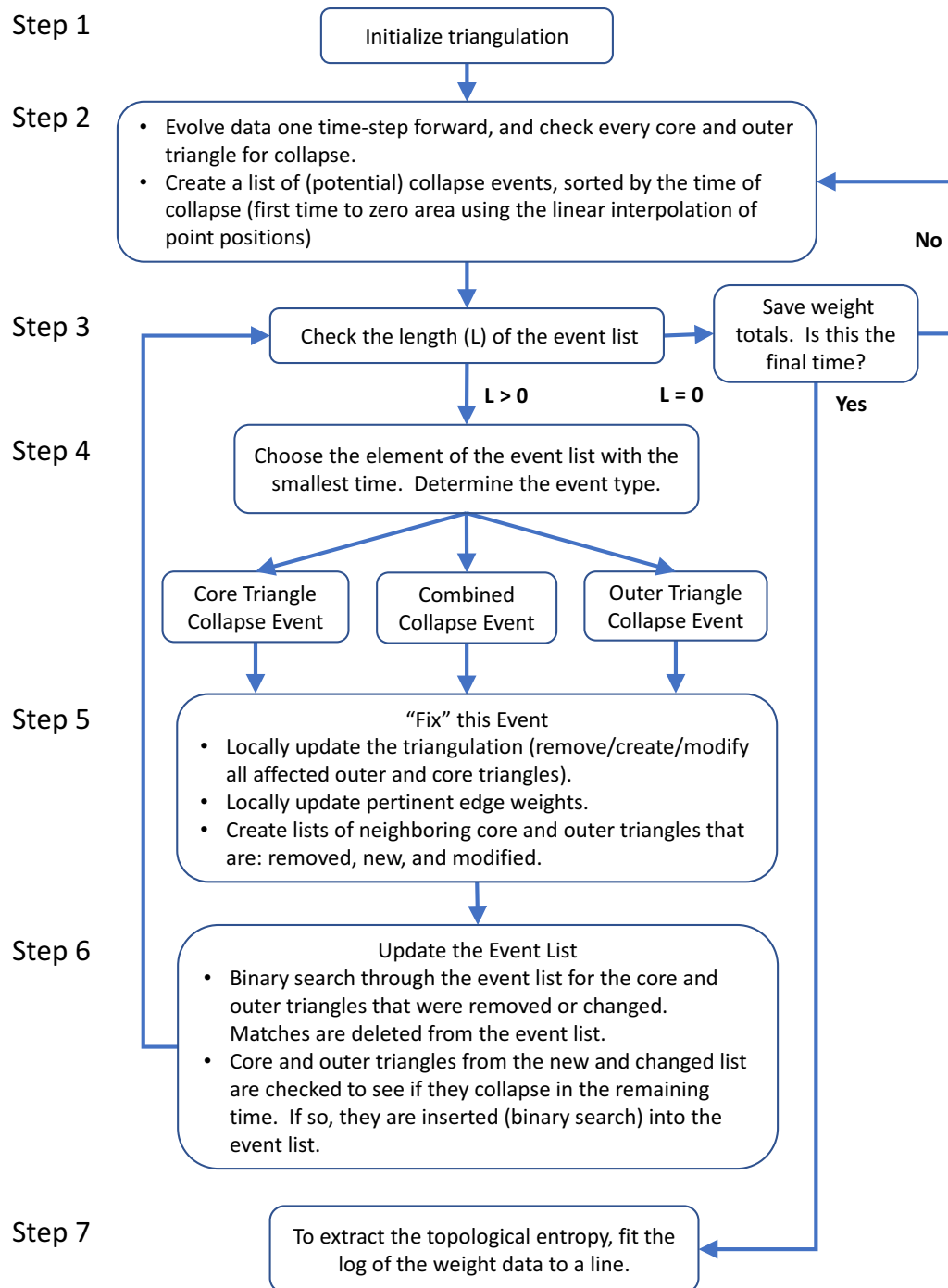


Figure 2.4: **E-tec Algorithm Flowchart.** As described in Sec. 2.3, E-tec employs computational geometry techniques for tracking the evolution of a piecewise-linear band. Full details are given in Sect. 2.4.



3. The exponential growth rate of the band (topological entropy), determined by the slope of the best fit line for the  $\ln(\omega)$  vs. time graph.

**Data structures:** E-tec maintains the following data structures as a function of time:

1. A *core* triangulation of all data points in the plane.
2. The weights on each edge in the triangulation. (Non-zero weighted edges constitute the stretched rubber band.)
3. For each relevant data point, the *outer band* triangle (abbreviated *outer* triangle) records the outermost wrapping of the rubber band around that point. (See the blue shaded triangles in Fig. 2.3.) During the algorithm's run, the outer triangle represents the piece of rubber band that has struck the point most recently and hence is a candidate for detachment at a future time. For example, upon inspection of vertex 7 in Fig. 2.3b, we may deduce that of all the red band edges attached to it, the two that created the largest angle would be the ones to snap back and revert to a single edge. Specifically, edge (2, 5) will snap back taut if triangle (2, 5, 7) changes orientation. Notice that outer triangles are not necessarily contained in the set of all core triangles.

**Steps:** We outline the key steps taken by E-tec in tracking the evolution of a rubber band. These steps are summarized in the Fig. 2.4 flowchart.

1. We first initialize the core triangulation using a constrained Delaunay triangulation [67] of the initial points (with the initial placement of the rubber band as the constraint). See Fig. 2.5a.

In steps (2-6) we evolve the state of the system (core triangulation, weights, and outer triangles) forward using the next time-slice in the trajectory data as input. Notice that E-tec does not need the whole trajectory at once in order to evolve the triangulation forward, and therefore could be used in real-time during experimental data collection.

2. For each core and outer triangle in the current state of the system, we use the linear interpolation of point positions to determine if and when a triangle will pass through zero area during this time step. These collapse events are sorted by time into an event list.

3. If the event list is non-empty, we go to step 4 and determine the event type of the next collapse event. If the event list is empty, we then add up the weights of every edge to get the current total weight  $\omega$  of the band, and store this value. This acts as a proxy for the length of the band, and grows with the same exponential rate in time. If we are at the final time of the trajectory data, we end by analyzing the accumulated weight data in step 7. Otherwise, we move on to the next trajectory time in step 2.
4. A collapse event can be one of three general types: a core triangle collapse (Fig. 2.3a), an outer triangle collapse (Fig. 2.3b), or a combined core and outer triangle collapse (see Fig. 2.3c for an illustration). While the specifics of how the three types of collapse events are handled are different, the broad strokes, as seen in step 5, are the same.
5. For each collapse event type, there is a general template for adding, removing, and/or modifying the core and outer triangles that are adjacent to the collapsing triangle. Crucially, this process is local, and the number of operations is bounded and does not grow with the number of trajectories.
6. The local deletions, creations, and modifications of core and outer triangles that result from handling a collapse event potentially affect the overall event list for this time-step. First we consider the deleted and modified core and outer triangles. If, before modification, they have a time-to-zero-area that is in the remaining fraction of the current time-step, then we search for and remove them from the event list. Next we consider the new and modified core and outer triangles. If, after modification, they will collapse in the remaining time-step, we search for the proper position to insert them into the sorted event list. Both searches are binary, and constitute one of the two aspects of the algorithm that give us  $\mathcal{O}(N^k \log N)$  computational complexity, where  $1/3 \leq k \leq 3/2$  ( $\mathcal{O}(\log N)$  for binary search and  $\mathcal{O}(N^k)$  searches per time-step). Here, the value  $k$  is determined by the scaling of the collapse event rate and depends heavily on the type of flow producing the trajectories. (Please see Sect. 2.7 for an explanation of the collapse event rate scaling.) After modifying the event list, we return to step 3.

7. Approximate the topological entropy by computing the exponential growth rate for the total weight over time.

## 2.5 E-tec Algorithm Verification

In this section, we verify the E-tec algorithm by running E-tec on numerical trajectories sampled from a chaotic lid-driven cavity flow used to study chaotic advection [3]. A numerical example of E-tec applied to real trajectory data (requiring only seconds to run) is shown converging to the theoretical topological entropy lower bound of the flow in Fig. 2.5. In later subsections, we compare our results to lower bounds on topological entropy computed from two different methods; first, by a direct application of Eq. (2.1) to a growing material-line, and second, by a technique called homotopic lobe dynamics (HLD), which extracts symbolic dynamics from finite-length pieces of stable and unstable manifolds attached to fixed points of the fluid flow [2, 6, 7].

### 2.5.1 Chaotic Lid-Driven Cavity Flow

The chaotic lid-driven cavity model [3, 4, 68, 69] is a two-dimensional area-preserving flow defined over a 2D vertical cross-section of a rectangular cavity, extending vertically from  $-b \leq y \leq b$  and horizontally from  $0 \leq x \leq a$ . The flow,

$$\mathbf{V}(x, y, t) = \left( \frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \quad (2.2)$$

is defined in terms of a stream function  $\psi(x, y)$ . The stream function is an exact solution of the biharmonic equation  $\nabla^2 \nabla^2 \psi(x, y) = 0$  defined on the rectangular domain. The stream function is time-periodic with period  $\tau_f$  and is given explicitly by

$$\psi(x, y, t) = \begin{cases} U_1 C_1 f_1(y) \sin\left(\frac{\pi x}{a}\right) + U_2 C_2 f_2(y) \sin\left(\frac{2\pi x}{a}\right), \\ \quad \text{for } n\tau_f \leq t < (n + 1/2)\tau_f, \\ -U_1 C_1 f_1(y) \sin\left(\frac{\pi x}{a}\right) + U_2 C_2 f_2(y) \sin\left(\frac{2\pi x}{a}\right), \\ \quad \text{for } (n + 1/2)\tau_f \leq t < (n + 1)\tau_f, \end{cases} \quad (2.3)$$

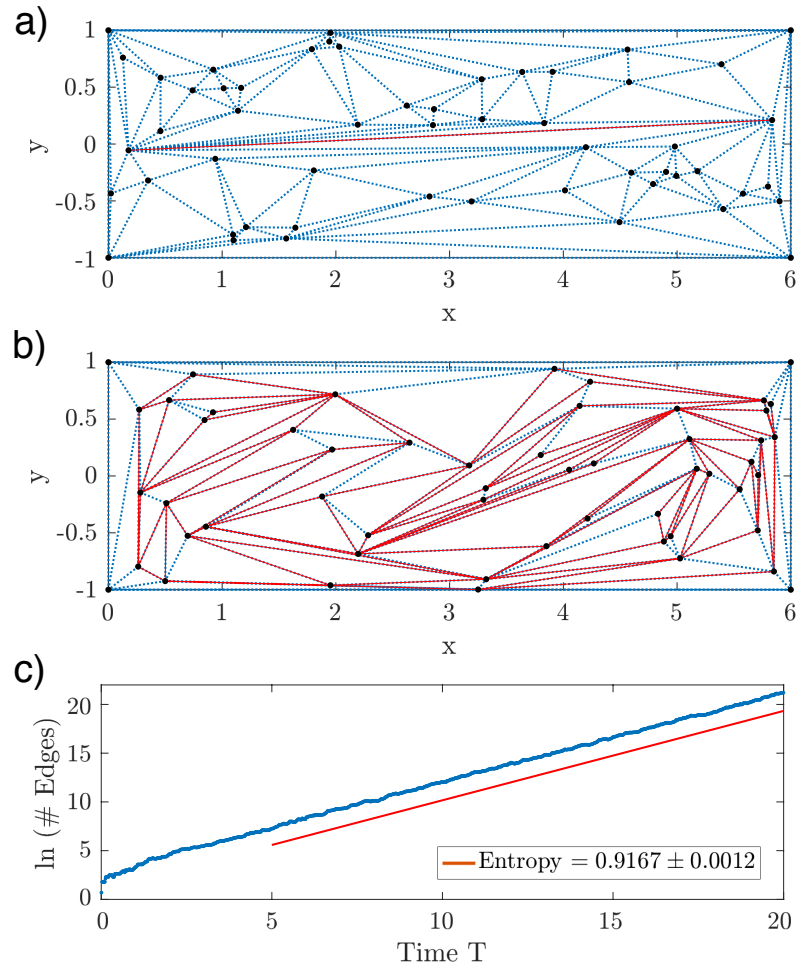


Figure 2.5: **Numerical Example of an E-tec Implementation.** a) Initial data points with the band wrapped around two points (in red). The core Delaunay triangulation (in blue, dotted) is constrained to include the red band edge. b) Final data point positions at  $T = 20$ , the triangulation, and the stretched band evolved under the motion of the trajectories. Dynamics is given by model in Sect. 2.5 with  $\tau_f = 0.96$ . c) E-tec output: the number of band edges as a function of time (blue). The slope of the best-fit line (red, dashed) is the topological entropy estimate.

where

$$f_k(y) = \frac{2\pi y}{a} \cosh\left(\frac{k\pi b}{a}\right) \sinh\left(\frac{k\pi y}{a}\right) - \frac{2k\pi b}{a} \sinh\left(\frac{k\pi b}{a}\right) \cosh\left(\frac{k\pi y}{a}\right), \quad k = 1, 2,$$

and

$$C_k = \frac{a^2}{2k\pi^2 b} \left[ \frac{a}{2k\pi b} \sinh\left(\frac{2k\pi b}{a}\right) + 1 \right]^{-1}, \quad k = 1, 2.$$

We follow Grover et al. [3] and assign  $U_1 = 9.92786$ ,  $U_2 = 8.34932$ ,  $a = 6$ , and  $b = 1$ . Fig. 2.6a and Fig. 2.6b show streamlines for the two steady flows in Eq. (2.3). Each flow is separately integrable and is asymmetric in  $x$ , with a large vortex on one side and a smaller vortex on the other. The system alternates between each flow for a half-period  $\tau_f/2$ . It is this alternating flow that introduces positive topological entropy into the system.

When  $\tau_f$  is sufficiently large,  $\tau_f \geq \tau_f^* \approx 0.9553$ , there exists a period-three orbit,  $r_i$ ,  $i = 1, 2, 3$ , such that

$$M(r_1) = r_2, \quad M(r_2) = r_3, \quad M(r_3) = r_1, \quad (2.4)$$

where  $M$  is defined to be the flow map that evolves a point  $(x, y)$  forward to the point  $(x', y') = M(x, y)$  after a single period  $\tau_f$ . Fig. 2.6c shows the points  $r_i$  and their time evolution over one period. In the first half-period,  $n\tau_f \leq t < (n + 1/2)\tau_f$ , the two trajectories on the left swap positions in a clockwise fashion, while in the second half-period,  $(n + 1/2)\tau_f \leq t < (n + 1)\tau_f$ , the two trajectories on the right swap positions in a counterclockwise fashion. Grover et al [3] characterize the  $r_i$  as a set of three strands braiding around one another in a nontrivial fashion. The presence of this braid guarantees the topological entropy is at least  $h_{po3} = 0.9624$ , the topological entropy which Boyland et al. [1, 51, 70] computed using the Bestvina-Handel train-track algorithm [71]. We note that this period-three orbit lives within a larger coherent set, a period-three island chain [4] when  $\tau_f$  is strictly greater than  $\tau_f^*$ .

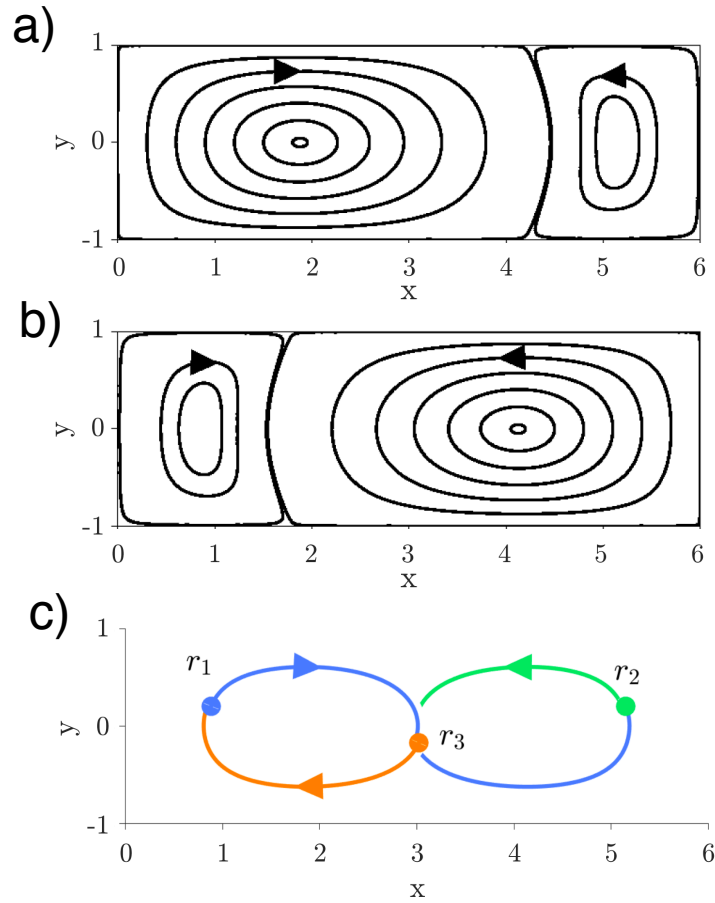


Figure 2.6: **Dynamics of Chaotic Lid-Driven Cavity Flow.** We depict streamlines of the flow, Eq. (2.3). a) Motion under the first half-period,  $n\tau_f \leq t < (n + 1/2)\tau_f$ . b) Motion under the second half-period,  $(n + 1/2)\tau_f \leq t < (n + 1)\tau_f$ . c) Illustration of a period-three orbit  $r_i$ . Each color (blue, green orange) represents the trajectory evolving forward one period.

## 2.5.2 Period-Three Orbit and Convergence in Ensemble Size

Here we investigate the convergence of the E-tec algorithm by studying trajectories from the chaotic lid-driven flow with period  $\tau_f = 0.96$ , where we are guaranteed the existence of a period-three island chain [3, 72, 73]. As illustrated in Fig. 2.7d, no trajectory starting in an island leaves the island, and no trajectories enter. These islands braid around one another as they swap places in the same fashion depicted in Fig. 2.6c. In the analysis of Sect. 2.5.1, each trajectory is sampled with time step  $\Delta t = 10^{-2}$  between points. This choice of  $\Delta t$  will be shown to be sufficient in Sect. 2.6.3.

First, we run E-tec on a set of three trajectories with the initial condition for each trajectory chosen in a different period-three island (Fig. 2.7a). We place an initial band around the right two points and observe exponentially growing band weights (Fig. 2.7b). At  $T = 15$  our estimate for the topological entropy is within 0.1% of the topological entropy guaranteed by the braid (Fig. 2.7c).

Next, we run E-tec on a set of 75 trajectories consisting of the 3 previously selected trajectories along with 72 randomly chosen ones. We calculate topological entropy by considering the time evolution of the same initial band (Fig. 2.7b). While the dynamics appear far more complicated than in Fig. 2.7a, our estimate of topological entropy is within fitting error to  $h_{po3} = 0.9624$  (Fig. 2.7c). Our results demonstrate that the periodic islands, and their braiding, are what drives most of the system entropy [74, 75]. Furthermore, this demonstrates that for certain systems, topological approaches such as E-tec (as well as braiding approaches) are capable of producing accurate estimates of topological entropy with only a small set of carefully chosen trajectories.

Although the coherent sets for our example were straightforward to locate, for other examples and practical applications, coherent sets may be harder to identify. As such, there is no guarantee trajectories from coherent sets, whose dynamics might be governing the topological entropy of the system, will be sampled appropriately. To investigate how E-tec would perform under conditions like this, we examine our ability to accurately recover the topological entropy when randomly sampling initial

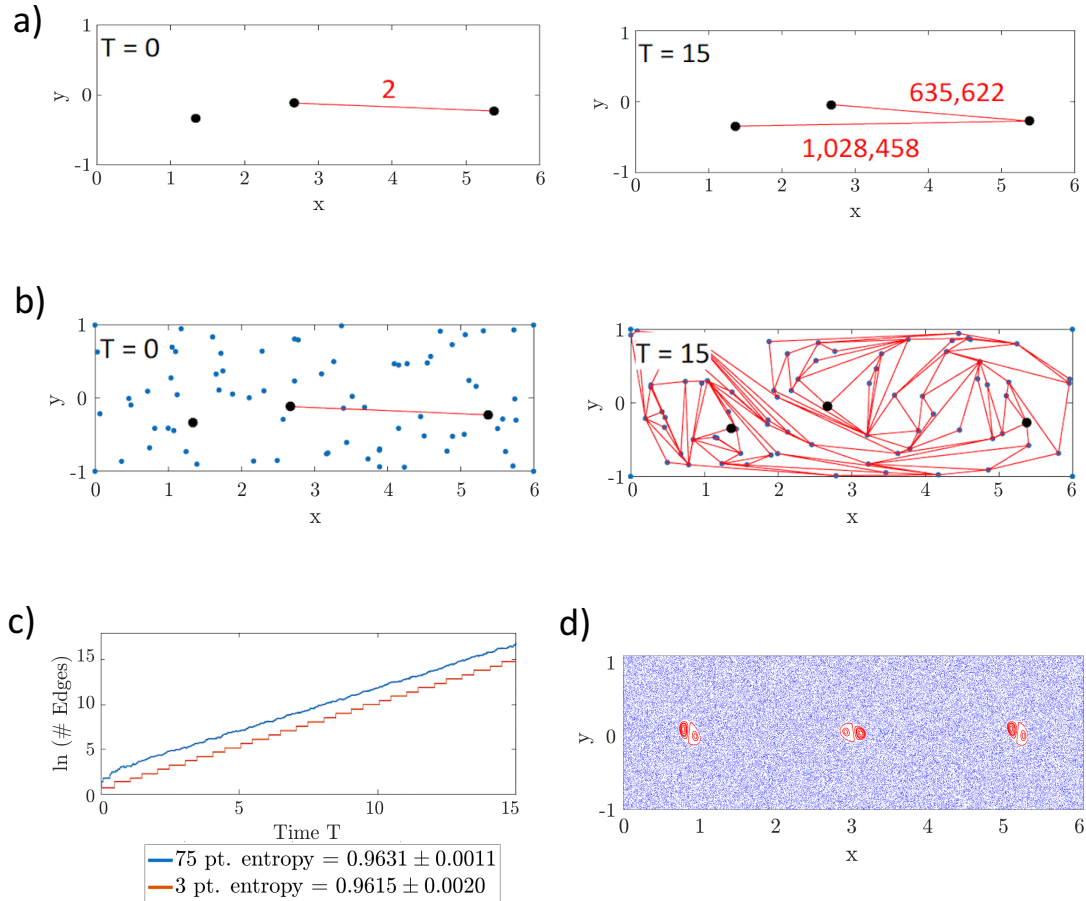


Figure 2.7: **E-tec Analysis of the Chaotic Lid-Driven Cavity Flow.** We show E-tec results on trajectories governed by Eq. (2.3) with  $\tau_f = 0.96$ , guaranteeing the existence of a period-three orbit, seen in Fig 2.6c. a) We show E-tec results when considering only 3 points close to the period-three orbit and contained in period-three islands. We consider an initial band around the two right points (top). This band evolves (bottom) into a highly stretched band (edge weights in red) around all three points by  $T = 15$ . b) We consider the same 3 initial points, but add 72 random trajectories (top). The dynamics are more complex (bottom, weights omitted). c) The growth rate in the number of edges, i.e. our estimate of the topological entropy, for (a, red) and (b, blue) is the same. This indicates the entropy is driven by the period-three islands as also shown by Ref. 2. d) The coherent period-three islands, noted in Refs. 2, 3, and 4, are clearly seen in the Poincaré return map of a long-lived trajectory in blue. Ten long-lived trajectories inside the islands are shown in red.



conditions uniformly in space, but removing any point chosen in the period-three islands. E-tec was run on increasingly larger but nested sets of such trajectories. That is, the points chosen in the 20 trajectory analysis contain all of the points in the 10 trajectory analysis, and so forth. As shown in Fig. 2.8a, E-tec converges rather quickly in the number of points to the topological entropy lower bound guaranteed by the period-three islands. Estimates may fluctuate based on the interval used to fit, especially when fewer trajectories are used. In Fig. 2.8, we see apparent oscillatory behavior, though we expect these to dampen at longer times and for results to converge if taken to infinite time. We note that in the above figure that E-tec does not require many long trajectories to compute a reasonable approximation to the topological entropy.

Finally, in Fig. 2.8b, we investigate the E-tec convergence using the 100 point ensemble in Fig. 2.8a by adding additional points in each of the three islands. E-tec performs increasingly better as the island points are added. The result with no island points, given in Fig. 2.8a, is then taken as a worst-case scenario. This assures our confidence in E-tec results as ensemble sizes are increased in Sect. 2.5.3.

### 2.5.3 Topological Entropy for Range of Period Driving Parameter $\tau_f$

With confidence in E-tec's ability to characterize topological entropy when  $\tau_f = 0.96$ , we next explore how the topological entropy changes as  $\tau_f$  varies. As mentioned previously, the period-three orbit is born at  $\tau_f^* \approx 0.9553$  and persists for larger values. Thus, entropy for values  $\tau_f < \tau_f^*$  will be bounded above by the braiding entropy of  $h_{po3} = 0.9624$ , while  $h_{po3}$  remains a lower bound for  $\tau_f > \tau_f^*$ . In all cases, the same initial band is chosen and evolved forward.

As shown in Fig. 2.9, our estimate of topological entropy using E-tec is within error of the direct calculation of material-line stretching when  $0.85 \leq \tau_f \leq 0.98$  and the number of data points is at least 1000. For  $\tau_f < \tau_f^*$ , there are no known island chains that drive the complexity. Despite this, E-tec performs well here, as shown in Fig. 2.9. For low values of  $\tau_f$ , when  $\tau_f < 0.85$ , E-tec produces an estimate slightly

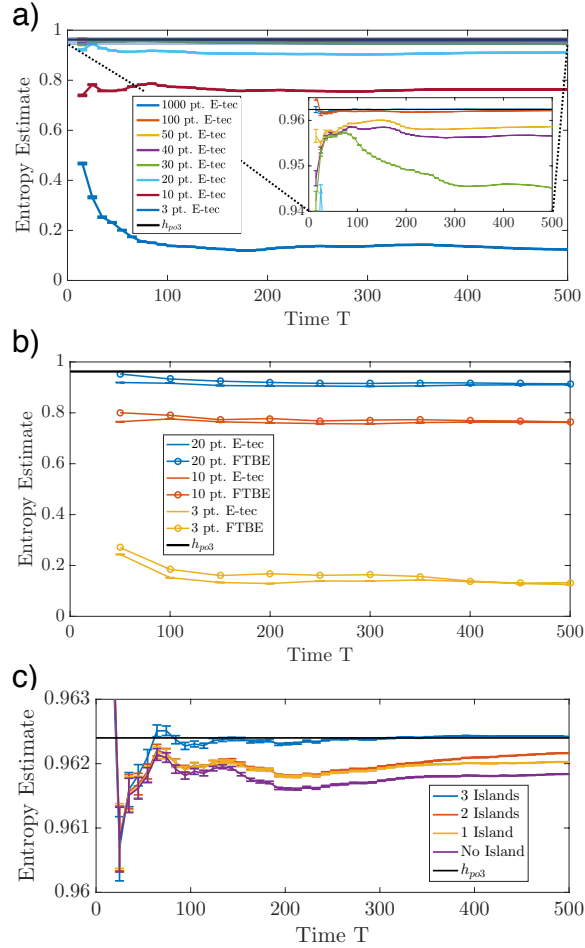


Figure 2.8: **Convergence of E-Tec in the Length and Number of Trajectories.**

a) We demonstrate convergence of E-tec to  $h_{po3} = 0.9624$  with increasing sample size and trajectory duration. For E-tec, the same initial band is stretched under ensembles of increasing size. All trajectories are sampled from outside the islands in the chaotic lid-driven flow with period-driving parameter  $\tau_f = 0.96$ . The entropy reported at time  $T$  is the fitting slope and 95 percent confidence interval to the log of the total number of edge weights over time  $t$  for the range  $t \in [5, T]$ . b) We demonstrate consistency between E-tec and FTBE results (calculated using the freely available Matlab package `braidlab` [5]). The same ensembles of trajectories are used for both. To stay consistent with the `braidlab` calculations, E-tec reports the entropy as the fit from initial time to reported time (or rather, it is the fit from  $t \in [0, T]$ ). c) E-tec output using the 100 point ensemble with a single trajectory added into one, two, and three of the periodic islands.

less than that of direct stretching but consistent with the value produced by HLD. But E-tec’s discrepancy becomes smaller with increasing numbers of samples. For high values of  $\tau_f$ , when  $\tau_f > 0.98$ , both E-tec and HLD produce lower estimates

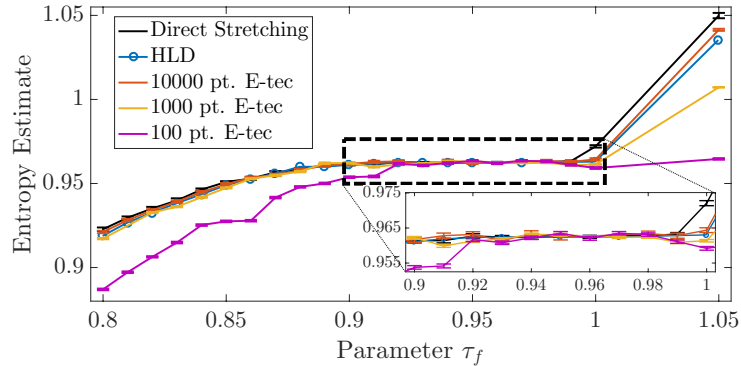


Figure 2.9: **Verification of E-tec for Increasing  $\tau_f$ .** E-tec topological entropy results over a range of  $\tau_f$  values using increasing ensemble sizes. We compare to the estimate of topological entropy from directly stretching a material line [2] and through another topological technique, homotopic lobe dynamics [6, 7].

for topological entropy than the calculated direct stretching value. We note that E-tec with 1000 trajectories still produces estimates consistent with HLD, and with 10,000 trajectories E-tec exceeds the HLD estimate but is still below the direct material-line stretching.

To more clearly see what drives the increase in entropy for high values of  $\tau_f$ , we show the band stretched by E-tec for three different values of  $\tau_f$  each computed from a set of 1000 independently chosen trajectories (see Fig. 2.10). Exponential stretching and folding is present in all tested parameter values, though Fig. 2.10 shows the band is stretched in a more complex fashion at higher  $\tau_f$  values. Here, additional island chains emerge [4] resulting in secondary folding [76] that seems less “smooth.” This secondary folding results in kinks near the islands that propagate forward, which in turn are further stretched under the dynamics. These small areas with kinks give significant contribution to the topological entropy, but because the entropy estimates (Fig. 2.9) were generated from uniformly random samples, these highly-kinked regions may remain undersampled. As such, a good portion of the stretching may remain undetected by E-tec in Fig. 2.10c.

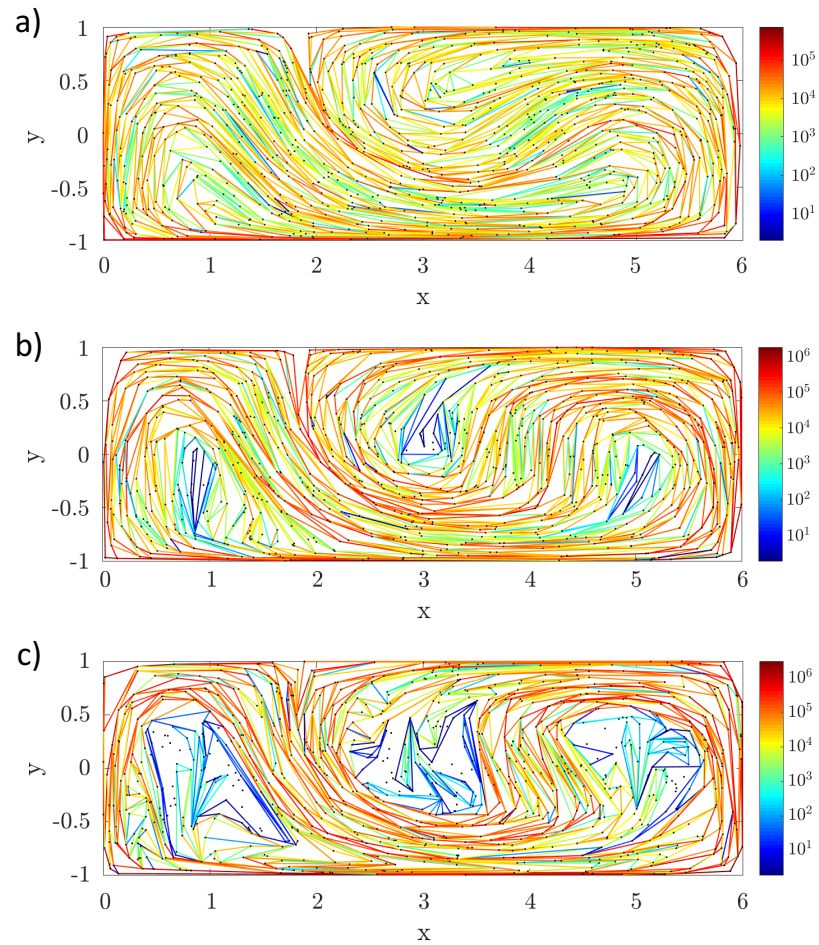


Figure 2.10: **Stretched Band Visualization.** E-tec band stretching due to flow advection for period-driving parameters a)  $\tau_f = 0.80$ , b)  $\tau_f = 0.96$ , and c)  $\tau_f = 1.05$ . Colorbar corresponds to edge weights. All bands are stretched by ensembles of 1000 uniformly distributed trajectories.

## 2.6 E-tec Robustness

In this section, we investigate the robustness of E-tec’s results. More specifically, we examine how E-tec’s ability to correctly estimate topological entropy is impacted by the choice of initial band and the time-step associated with trajectories. Finally, we discuss how the E-tec algorithm’s run-time scales with the duration and number of sampled trajectories.

### 2.6.1 Robustness to Choice of Initial Band

We make the following conjecture: if all trajectories reside in the same ergodic component then the choice of initial band does not affect the topological entropy computed by E-tec as long as the trajectories are sufficiently long. Figure 2.11 supports this conjecture. All initial bands eventually become stretched at the same rate despite some differences at early times. Adjacent points may remain close for some time, though the chaotic nature of the flow causes nearby trajectories to eventually diverge, thereby making the band’s deformation inevitable. Thus, as long as it is possible to obtain sufficiently long trajectories within a single ergodic component, E-tec’s topological entropy calculation appears to be invariant to the choice of initial band.

Some chaotic flows have more than one ergodic component, or a mixture of ergodic and non-ergodic regions. This is true of the model flow in Fig. 2.7d. In such systems, the choice of initial band will impact the topological entropy estimate. For example, a band placed entirely in one of the test flow’s period-three islands (Fig. 2.7d) will undergo no significant stretching under the flow and thus yield zero topological entropy.

In practice, to make sure all ergodic components are sampled, it is prudent to check that the final band stretches around nearly all of the data points. Alternatively, one could sample many initial bands taking the maximum growth rate of all sampled bands as the best estimate of the entropy [55]. E-tec is fast enough to run multiple bands, each with a different initial triangulation constrained to the initial band choice, in ensembles of fewer than  $10^6$  trajectories in a reasonable time. An alternative approach to choosing a single initial band is to evolve a “web” of initial bands that

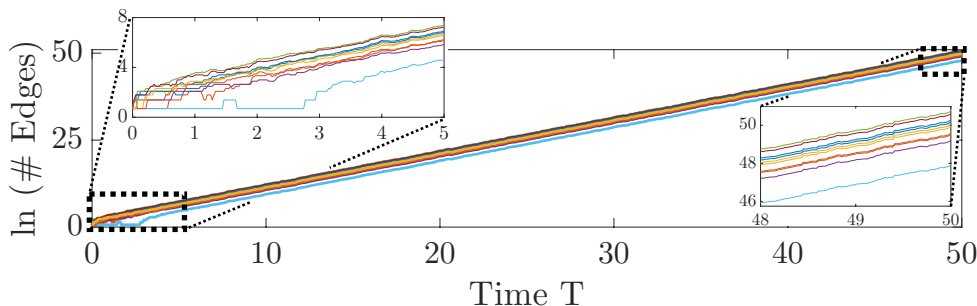


Figure 2.11: **Initial Bands.** E-tec output is the logarithm of the sum of edge weights as a function of time. E-tec’s estimate of topological entropy is the best-fit linear slope through this data. Here we show 10 different outputs from E-tec for the same set of 100 trajectories. In each case, we chose a different pair of points around which to stretch our band. Despite some initial differences in the increase in edge weights due to initial adjacent points staying close to one another (*left inset*), eventually all the bands grow at similar rates (*right inset*). When we fit the exponential growth rate, starting at  $T = 5$ , we find the values for each of the 10 bands agree within 5 decimal places and average out to 0.9617.

covers the entire initial triangulation. This guarantees that all ergodic components sampled by the data will be included. As opposed to the initial triangulation being constrained to the choice of initial band, the initial “web” is constrained to the edges of the choice of initial triangulation.

## 2.6.2 Algorithm Scaling and FTBE Comparison

The computational runtime of E-tec is linearly proportional to the *duration* of the trajectories. This is because the number of edges tracked by E-tec is constant, and it is only the values of the weights that grow exponentially in time. This scaling is the same as the FTBE calculation and stands in contrast to algorithms that precisely evolve a material-curve forward, which requires inserting exponentially more points to maintain sufficient point density [77].

One advancement we have made over the FTBE calculation is the run-time scaling with respect to the number of trajectories used (see Fig. 2.12). The FTBE calculation

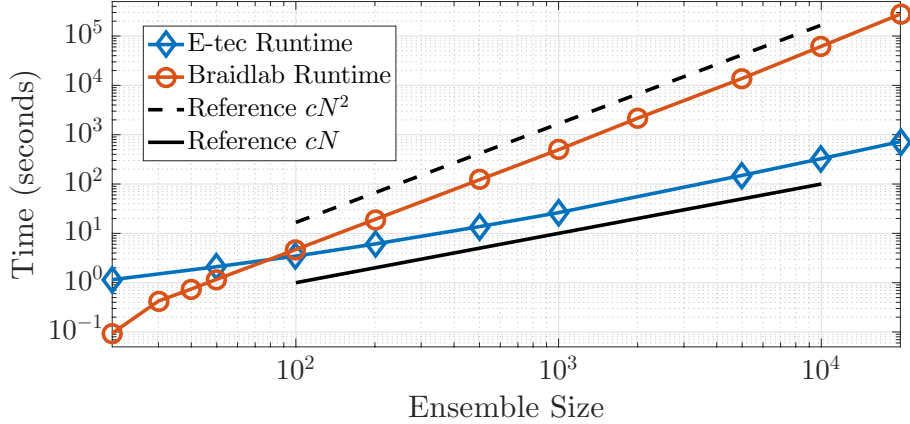


Figure 2.12: **E-tec Runtimes.** Runtime comparison of E-tec and `braidlab`, a freely available Matlab package implementing the FTBE calculation. Both used the same trajectories from the chaotic model flow for  $\tau_f = 0.96$ . All computations were completed using a 2.8 GHz Intel Core i7 processor.

scales quadratically in the number of trajectories  $N$  due to the braid approach requiring  $N^2$  algebraic generators per unit time step [42]. Overall, E-tec runtime scales as  $\mathcal{O}(N + N^k \log N)$ , where  $k$  is the collapse event rate scaling factor. In general, the value of  $k$  largely depends on the complexity of the flow being studied. For the chaotic, lid-driven cavity flow trajectories, we find  $k \approx 1.05$ , though we find values as low as  $k = 1/3$  for trajectories with highly correlated movement and as high as  $k = 3/2$  for random trajectories. We refer the reader to Sect. 2.7 for more details. As a practical matter, the E-tec runtime for small to moderate ensembles (roughly up to 5,000 trajectories) is dominated by the linear behavior in Fig. 2.12.

One illustrative example highlighting the runtime difference between the two algorithms is rigid rotational flow. While an admittedly special case, there would be no new collapse events (except for ones associated with the boundary) making E-tec very fast, whereas the number of braid generators needed would be proportional to  $N^2$ . However, one advantage the braid approach has over E-tec is that once the braid is extracted from the trajectory data, it may be applied to any initial band. E-tec only propagates a single curve forward. However, for topological entropy calculations, a single sufficiently long curve is typically sufficient (as evidenced in Fig. 2.11).

### 2.6.3 Robustness to Step Size $\Delta t$

Because E-tec is based on the computational analysis of evolving trajectories, it is necessary to consider discretized time. We next investigate how the trajectory time step  $\Delta t$  affects the entropy calculation and show that E-tec returns trustworthy results even when poorly resolved trajectories are used as input. We use two ensembles of trajectories (of sizes 100 and 1000) sampled at a fine scale using the same reference time step of  $\Delta t^* = 10^{-4}$  to generate two reference topological entropies  $h_t^*$ . We then vary the time step  $\Delta t$  (keeping the trajectories the same) and compute both ensembles' corresponding  $h_t$ . The effect of time step  $\Delta t$  is quantified by computing the relative error

$$\left| 1 - \frac{h_t}{h_t^*} \right|, \quad (2.5)$$

which is plotted in Fig. 2.13. Independent of ensemble size, the data shows that relative error grows linearly with the time step  $\Delta t$ . As the trajectory information is input into E-tec using larger step sizes, we detect more events between steps. It remains to be seen if the relationship between relative error and  $\Delta t$  scales linearly for all ensemble sizes.

E-tec detects events individually for all values of  $\Delta t$ , but the order in which these events are detected is potentially different as  $\Delta t$  increases, due to the differences in the interpolation of trajectories. In fact, undersampled trajectory data may lead to entirely different events. This explains the larger relative errors for the 1000 trajectory ensemble; at higher point densities, there are simply more events that E-tec must resolve, resulting in more erroneous and misordered event detections. Despite this, Fig. 2.13 shows that the E-tec error due to step size is still relatively small. It is comparable to (or smaller than) the error due to other sources, such as trajectory length and ensemble size (Fig. 2.8a), for  $\Delta t < 10^{-2}$ , at least for smaller ensemble sizes.



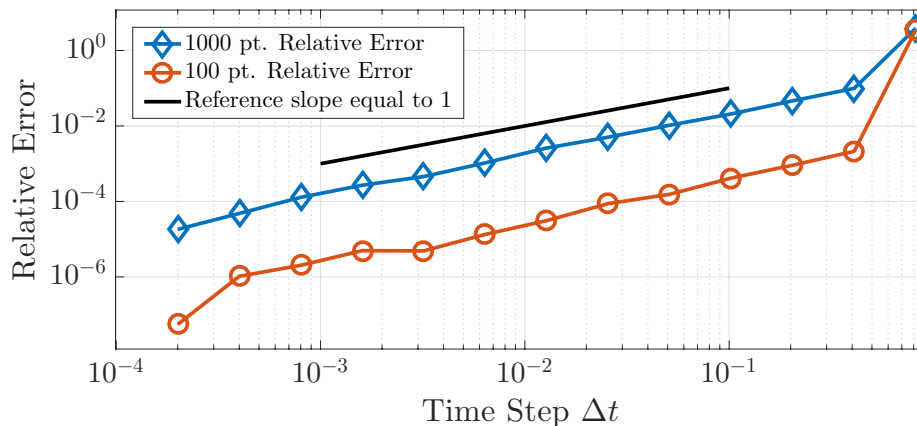


Figure 2.13: **Relative Error as a Function of Step Size.** The effect of time step  $\Delta t$  on the relative error in the topological entropy calculation with respect to the reference time step  $\Delta t^* = 10^{-4}$ . Graph displays calculations done on two separate ensembles of size 1000 and 100.

## 2.7 E-tec runtime scaling with number of points

The main bottleneck in the computational complexity of E-tec comes from the creation and maintenance of a time-sorted collapse-event list at each time-step. Since every core and outer triangle, of which there are  $\mathcal{O}(N)$ , is checked for collapse in this process, E-tec will scale no better than linear in  $N$ . Sorting is a worst-case and average-case  $\mathcal{O}(n \log n)$  process, for  $n$  items to sort. Assuming that the number of collapse events per unit time scales as  $\mathcal{O}(N^k)$  for some  $k$ , the sorting bottleneck implies an E-tec scaling of  $\mathcal{O}(N^k \log N)$ . A similar scaling comes from the maintenance of this event list. During the handling of a collapse event, core and outer triangles may be created, modified, or deleted. Importantly, this process is local, and the time for handling one event does not change with an increasing number of points. However, these amendments to the triangulation necessitate adding or removing events from the time-sorted event list. This is achieved with a binary search, which is an  $\mathcal{O}(\log n)$  routine for a list length of  $n$ . Given a list length that scales with the number of collapse events per unit time, this constitutes a second avenue for the  $\mathcal{O}(N^k \log N)$  scaling. Overall, the E-tec runtime scales as  $\mathcal{O}(N + N^k \log N)$ , where  $k$  is determined by the collapse event rate scaling.

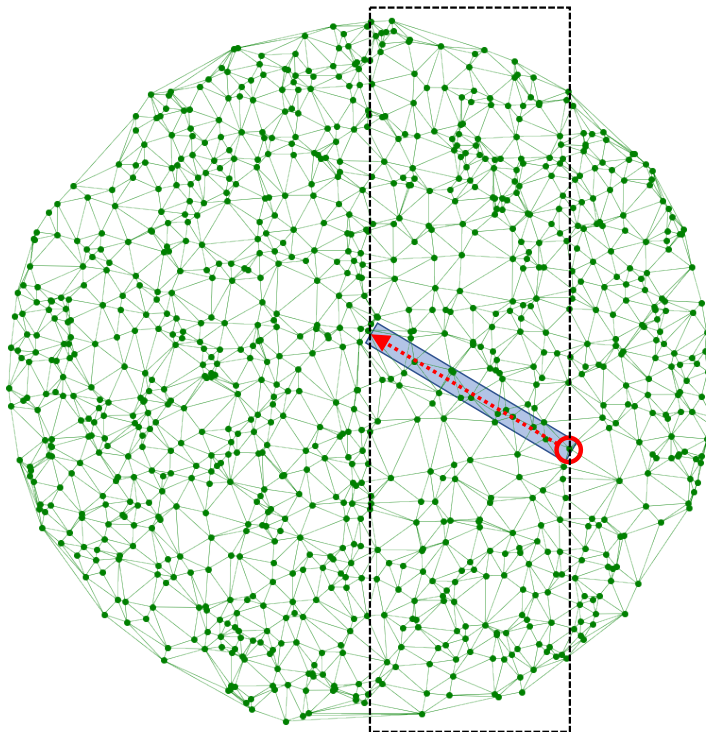


Figure 2.14: Schematic of the movement of a single point (circled in red) against a background of stationary points. The number of E-tec collapse events is proportional to the number of average core triangles that would have an equivalent area to that of the diagonal blue rectangle. For FTBE calculations, the number of braid generators created by this same process is equal to the number of points in the larger black rectangle.

The scaling of the collapse event rate depends heavily on the type of flow that produced the trajectory data. If there is no correlation between the velocities of neighboring points, then it can be as high as  $k = 3/2$ . If they are highly correlated (e.g. rigid rotation), then it can be as low as  $k = 1/3$ . For most flows,  $k \lesssim 1$ , with  $k$  generally increasing for more complex flows.

To justify the worst-case scaling of  $k = 3/2$ , consider the movement of a single point through a fixed length and against a background of stationary points, as depicted in Fig. 2.14. The number of collapse events produced by this motion will be proportional to the number of core triangles in the path of the moving point.

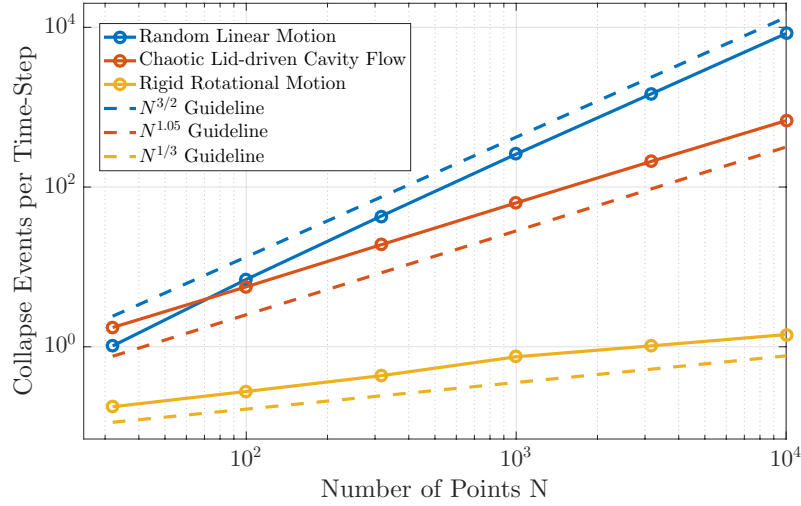


Figure 2.15: The collapse event rate scaling for three numerical examples: random motion, chaotic lid-driven cavity flow (for  $\tau = 0.96$ ), and rigid body rotation.

Given that the average area of a core triangle scales as  $\mathcal{O}(N^{-1})$ , a characteristic triangle length goes as  $\mathcal{O}(N^{-1/2})$ . Therefore, the number of characteristic lengths in the particle’s path, and from this the number of collapse events, scales as  $\mathcal{O}(N^{1/2})$ . Moving to the general case where every point is in motion, we could say that each of the  $N$  points “sees”  $\mathcal{O}(N^{1/2})$  triangles in its way, and therefore the overall scaling for collapse events would be  $\mathcal{O}(N^{3/2})$ . For comparison with FTBE calculations, the same one-point motion produces a number of braid generators equal to the number of points in the black rectangle of Fig. 2.14. Since this scales as  $\mathcal{O}(N)$ , the general case where every point is in motion produces a braid generator production rate that scales as  $\mathcal{O}(N^2)$ .

The  $k = 3/2$  scaling is also borne out in a numerical experiment, see Fig. 2.15. Here we track the collapse event rate for  $N$  trajectories, whose initial and final positions are chosen randomly within a fixed square, and whose intermediate positions are given by linear interpolation.

However, in most cases of interest the trajectory motion is generated by or sampled from an underlying flow, and there will be substantial correlations between the movement of nearby points. Points advected together can significantly suppress the

collapse event rate scaling. At the other extreme, consider the case of points undergoing rigid-body rotation. None of the triangles in the bulk will collapse, and the only contribution to the collapse event rate comes from core triangles associated with the fixed bounding auxiliary points (stationary points that are added upon initialization which help us avoid triangulation update issues at the boundary edges of the triangulation). This numerical example, see Fig. 2.15, gives a scaling value of  $k \approx 1/3$ , likely the most favorable scaling we can expect from a non-trivial flow. General flows will fall between these two extremes. Our example of a chaotic lid-driven cavity flow (see Fig. 2.15), with  $\tau = 0.96$ , gives a scaling value of  $k = 1.05$ . We have also simulated the collapse event rate scaling for linear shear flow,  $k \approx 0.66$  and an irrotational (Rankine) vortex,  $k \approx 0.77$ .

Overall, we can expect the E-tec runtime to scale as  $\mathcal{O}(N + N^k \log N)$ , with  $1/3 \leq k \leq 3/2$ , and typical flows resulting in  $k \lesssim 1$ . This favorable computational complexity, compared to  $\mathcal{O}(N^2)$  for the FTBE, comes from two sources. First, collapse events are produced locally, whereas braid generators encode more global information. Second, the correlated motion of neighboring points further reduces the scaling for trajectories derived from general flows.

## 2.8 Conclusion

We introduced the Ensemble-based Topological Entropy Calculation (E-tec), an algorithm that computes topological entropy in a planar flow from an ensemble of system trajectories. We verified E-tec's convergence to the correct topological entropy with increasing numbers of trajectories on a highly chaotic, lid-driven cavity flow. E-tec's performance was shown to be robust with respect to the choice of initial band, as well as changes in the time sampling interval ( $\Delta t$ ). Notably, we have shown that E-tec's runtime scales as  $\mathcal{O}(N^k \log N)$ , where  $1/3 \leq k \leq 3/2$  and  $N$  is the number of trajectories in the ensemble.

Our work suggests several further directions for the analysis of trajectories with E-tec, which we intend to explore in future studies. First, we shall seek to extend E-tec to three dimensions and higher. Braiding theory, the basis for FTBE calculations,

cannot be readily generalized to higher dimensions [41]. The computational geometry framework in which E-tec is based might perhaps be more naturally extended [64–66]. Instead of a rubber band in a planar flow, we would consider a two-dimensional rubber sheet stretched around a collection of points in a three-dimensional flow. A 3D triangulation may still be used to track point-face or edge-edge collisions, and the rubber sheet may be chosen as one of the faces in the initial triangulation. As the points evolve in time, they carry the sheet along with them, stretching and folding it so that its growth reflects the flow complexity. Though there clearly remain some significant challenges to executing this generalization to three dimensions, we anticipate a host of interesting theoretical opportunities that this route may provide. Finally, by tracking all the trajectories in concert, we believe E-tec’s algorithm may be naturally adapted towards identifying and tracking coherent sets and other emergent structures.

## Acknowledgments

This work was supported in part by the US DOD, ARO grant W911NF-14-1-0359 under subcontract C00045065-4. The author also received support from the National Science Foundation under grant number DMS-1331109.

# Chapter 3

## 2D E-tec Applications

### 3.1 Active Matter Mixing

#### 3.1.1 Introduction

E-tec is a Lagrangian-based method, meaning its implementation requires no knowledge of the velocity field or of the governing equations of motion. This makes E-tec an appealing algorithm to use for topological entropy calculations in real world applications of chaotic advection, mixing, and fluid transport. In many of these situations, observed data may be limited to only a small number of potentially-sparse trajectory data. E-tec succeeds in these situations where many techniques fail, as standard tools for studying such data, like finite-time Lyapunov exponent fields and velocity field estimations, oftentimes require a large number of trajectories and sophisticated trajectory-extraction techniques.

In this section, we apply E-tec to experimental data obtained from an active matter microbiological fluid in an effort to better understand the mixing efficiency of self-driven advection. Significantly, this work represents the introduction of well-known measures of chaos to the physics of biologically-active fluids. E-tec is applied to experimentally derived system trajectories and the system is verified to be chaotic. Furthermore, E-tec results are verified against Lyapunov exponent calculations on the same system trajectories. Some special challenges arise when working with these

experimentally derived trajectories, particularly when they are observed for only a portion of the time interval.

### 3.1.2 Active Nematic Fluid

Active matter is made up of many agents which each consume energy in order to collectively move and produce large-scale motion. These groups come in all shapes and sizes, ranging from the macro scale, like flocks of birds [78] and schools of fish [79], to the micro scale, like cell formation patterns [80,81] and swimming bacteria [82,83]. These self-propelling entities are able to collectively organize into patterns and exhibit chaotic flows or self-assemble themselves into higher order structures that are in or out of equilibrium [84,85]. Of interest in this study are active nematic fluids, ones composed of rod-like subunits that take on an anisotropic ordering, spontaneously aligning when packed together densely, and slide relative to each other.

The active matter fluid studied is a microtubule-kinesin-based active nematic on the length scale of 100's of microns. In biological cells, microtubules and kinesin motors play a major role in cell structure and intracellular transport. Microtubules serve as biological highways for molecular motors to transport cellular cargo within the cell's cytoskeleton. Kinesin motors step along microtubules and are essential for a variety of cell functions, including cell division and cytoplasmic streaming [86].

### 3.1.3 Experimental Setup

The experimental setup for this study is confined in two dimensions [85,87] and the resulting flow is shown in Fig. 3.1. Microtubules are densely packed together and spontaneously organize in ordered bundles that can bend, buckle, and break [88–90]. Bundles are cross-linked by clusters of kinesin motors. Powering the kinesin movement along microtubules is adenosine triphosphate (ATP) [91]. If two motors in a cluster bind to adjacent microtubules of opposite polarities, the opposing forces produce a sliding motion between these microtubules in the presence of ATP.

In the experiment, ATP can generally be thought of as an energy source, as system activity and fluid speed increases with the ATP concentration. At low concentrations,

microtubules simply glide forward in their respective opposing directions. There is no change of direction and local ordering is never broken. However, rich dynamics occur at higher concentrations of ATP. Emerging at high microtubule densities and high ATP concentrations are pairs of  $+\frac{1}{2}$  and  $-\frac{1}{2}$  topological defects that are continuously created and annihilated. These defects, pictured in the inset of Fig. 3.1, occur when local ordering breaks down. They braid around one another when alive and are responsible for driving much of the dynamics. In fact, on the macroscopic scale, the flow exhibits chaotic behavior. (This is none too surprising, as the braiding of defects is reminiscent of the braiding of period-3 islands in the chaotic lid-driven cavity flow model used to verify E-tec in Ch. 2.)

Ultimately, we wish to explore how the dynamics depend on varying ATP concentrations. Experiments are repeated at progressively higher ATP concentrations and two measures of chaos, topological entropy and the Lyapunov exponent, are estimated for each. The trajectory extraction process is explained below, followed by results for an array of ATP concentrations.

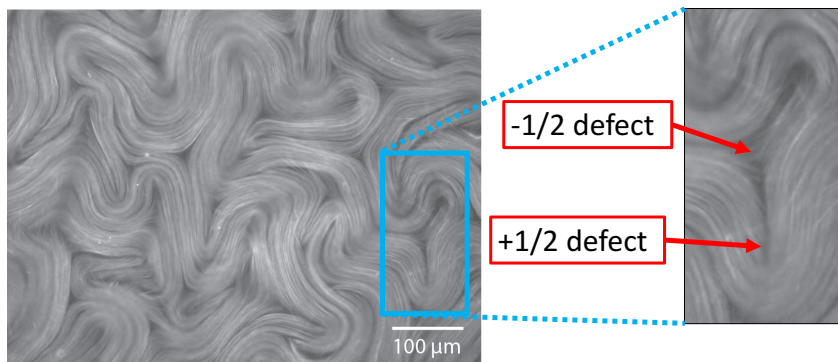


Figure 3.1: Fluorescence microscopy image of the active nematic fluid consisting of a densely-packed microtubule network in 2D. The two types of topological defects that emerge are pictured in the inset. *Image credit: Amanda Tan [8].*



### 3.1.4 Extracting Particle Trajectories

To obtain position and velocity information of the nematic flow, biotin-coated silica beads of  $2\mu\text{m}$  diameter were bound to the microtubule bundles. Bead trajectories directly following the microtubule network are seen colored-in in Fig. 3.2a. This approach differs from recent work on similar systems in which beads are not directly attached to the microtubules, but rather acted as passive tracers [85,88].

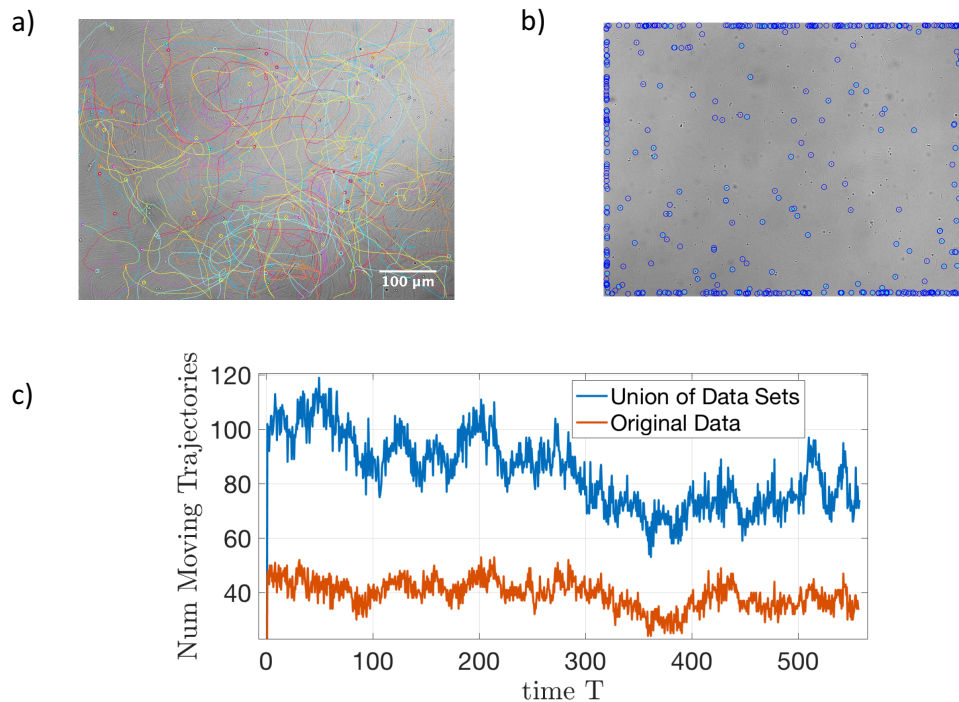


Figure 3.2: a) Bright field image of tracked bead trajectories that remain unbroken (colored in for effect). b) Lagrangian particle image tracking results. All bead particles found in the frame are highlighted in dark blue. All bead particles belonging to unbroken trajectories used for later analysis are highlighted in cyan. Pinned particles near the boundary are those that leave the field of view. c) Number of trajectories that are actively moving in the field at any given time for one data set (red) and the union of data sets (blue).

To extract the trajectories, first bright-field optical microscopy snapshots were produced to make movies. Second, bead motion was tracked using a particle-tracking algorithm developed by Crocker et al [92] and adapted by Gao et al [93]. The tracking

algorithm locates particles at each frame by identifying the center of mass based on maximum intensity for each particle. Then, trajectories are created for each particle by linking the displacement of a particle with the most probable corresponding particle in the preceding frames. A snapshot of detected bead particles in a single frame is pictured in Fig. 3.2b. In dark blue are all the bead particles the algorithm detects at the single snapshot of time. While most of these bead particles will be stitched together as bead trajectories over multiple frames, the algorithm fails to track many bead trajectories for the entirety of their run. This is mostly due to limitations in the image qualities and non-uniformity of the background darkness in time. Also marked in the Fig. 3.2b snapshot are the bead particles belonging to trajectories that the algorithm is able to track for the entirety of their run are in cyan. The algorithm does not “lose” this set of cyan trajectories. Because E-tec requires unbroken sets of trajectories, E-tec analysis using bead trajectories is limited to this smaller subset of unbroken ones.

In addition to losing track of trajectories, difficulties in the tracking algorithm arise because the camera does not capture the entire flow field. Thus, many trajectories are entering and leaving the field of view for the duration of the experiments. Any tracked trajectory that leaves the field of view is left stationary at the position it exits the boundary for all later time. Any trajectory entering is similarly placed along the boundary where it enters for all prior times. As a result, the number of trajectories that are moving in the field of view fluctuates in time.

The particle tracking algorithm is run once and Fig 3.2c shows in red the total number of unbroken trajectories moving in the field of view at any given time. Since the number of active and unbroken trajectories remains low ( $\sim 40$ ), the algorithm was run many times using different parameters. A union of trajectory sets is taken. Repeated ones are thrown out, yielding an ensemble of size 254. Unfortunately, less than half of these trajectories are moving in nearly every frame, indicated by the blue curve representing active trajectories in Figure 3.2c.

### 3.1.5 Topological Entropy Calculation for 50 $\mu\text{M}$ ATP Concentration

We use extracted bead trajectories as input to E-tec. A long initial band is placed along two trajectories and stretched by the trajectories as the triangulation is advected forward. This process is shown for a small ensemble of roughly 100 trajectories in the 50  $\mu\text{M}$  ATP solution in Fig. 3.3a and b. Figure 3.3c shows the entropy estimation for the full ensemble of 254 trajectories. The topological entropy  $h_{top}$  is estimated to be  $0.0100 \pm .0001$ . For reference, the Lyapunov exponent, a key metric of a flow's complexity measuring the exponential rate at which nearby fluid parcels separate from one another, is estimated by tracing the frame-by-frame paths of 10 pairs of nearby trajectories and calculating the rate in which each pair separates. The Lyapunov exponent  $\lambda$  of the flow is approximated to  $\lambda = 0.0120 \pm .0006$ . Both  $h_{top} > 0$  and  $\lambda > 0$  indicate the flow at the 50  $\mu\text{M}$  ATP concentration is chaotic. However, the fact that  $\lambda > h_{top}$  is problematic, as theory guarantees that the topological entropy  $h_{top}$  acts as an upper bound to the Lyapunov exponent  $\lambda$  [49].

Limitations in the number of bead trajectories that can be input into E-tec help explain the low  $h_{top}$  calculations. Further exacerbating this issue is the fact that over half of all trajectories are stationary on the boundaries at any given time (see Fig. 3.2c). Only 254 could be extracted from the experiment with 50  $\mu\text{M}$  ATP concentration. Figure 3.3d shows the topological entropy estimation as a function of ensemble size, giving us a possible gauge for the size of ensemble needed to surpass the Lyapunov exponent.

### 3.1.6 Topological Entropy Calculation from Beads as a Function of ATP Concentration

The same bead extraction technique is repeated on experiments using ATP concentrations of 75, 100, 500, and 1000  $\mu\text{M}$ s. For each, trajectories are gathered using several instances of the particle tracking software, the union of data sets is taken, and repeat trajectories are thrown out. The E-tec results for each set are recorded in blue in Fig. 3.4. Also plotted are the Lyapunov exponent averages for each concentration.

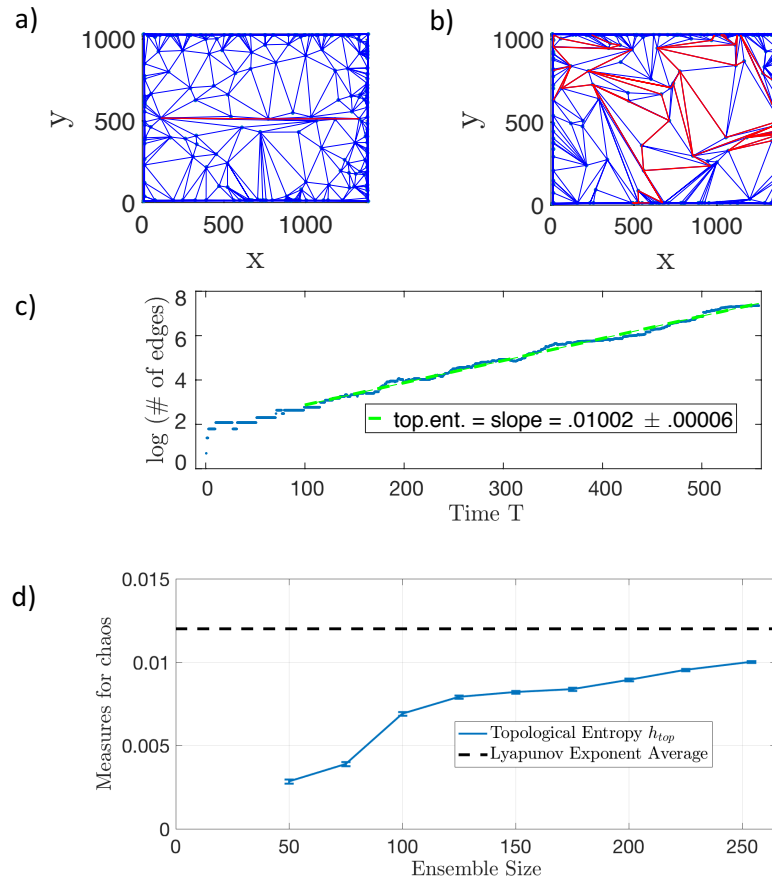


Figure 3.3: E-tec results for the 50  $\mu\text{M}$  ATP concentration. a) Initial bead positions with the rubber band (red) stretched between two of them. b) At the final time, the band is stretched and folded over itself in an intricate pattern. c) The number of band edges grows exponentially in time, giving the topological entropy  $h_{top}$ .

These measure the exponential rate of separation of nearby bead pairs and are obtained from the bright microscopy images. E-tec topological entropy stays somewhat stationary as ATP concentration increases while the Lyapunov exponents generally increase.

Because ATP concentration controls the speed at which kinesin motors step at the molecular level, we expect the topological entropy and Lyapunov exponent to both increase with increasing ATP concentration. Somewhat surprising is that this growth is not monotonic. Both measures drop in value for the 500  $\mu\text{M}$  concentration

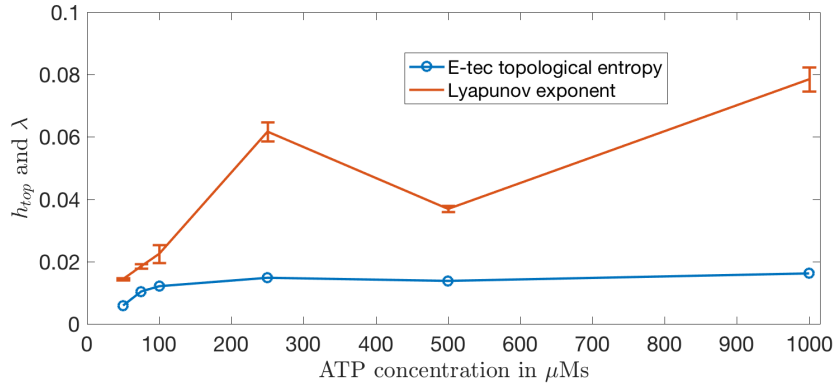


Figure 3.4: E-tec topological entropy measurements using bead trajectories in blue. Also pictured are the Lyapunov exponents found by averaging the exponential rate of separation of 10 bead pairs for each concentration.

(significantly more so for the Lyapunov exponent). We note how low the E-tec entropy results are in comparison to the Lyapunov exponents. This plot represents the best E-tec can accomplish given the system’s bead trajectory gathering restraints and the state of the bright-field optical microscopy movies. Though ensemble sizes input into E-tec are on the order of  $\mathcal{O}(10^2)$ , typically more than half of these trajectories are stationary.

### 3.1.7 Topological Entropy Calculation from Defect Tracking as a Function of ATP Concentration

Instead of tracking beads attached to microtubules that move passively in the flow, we aim to track topological defects that govern the flow dynamics. Microtubule bundles around the  $+\frac{1}{2}$  defects resemble the stretching and folding of dye by stirring rods undergoing an optimal braiding pattern [1]. Experiment defects are highlighted in Fig. 3.5a. As additional motivation for obtaining defect trajectories, the defect stretching further resembles the stretching done by the period-3 islands of the chaotic lid-driven cavity model flow [3, 4, 68, 69] used to verify E-tec in Ch. 2. The islands are highlighted in an E-tec snapshot in Fig. 3.5b. Remember that the island braiding enforced a minimum value on the topological entropy of the flow.

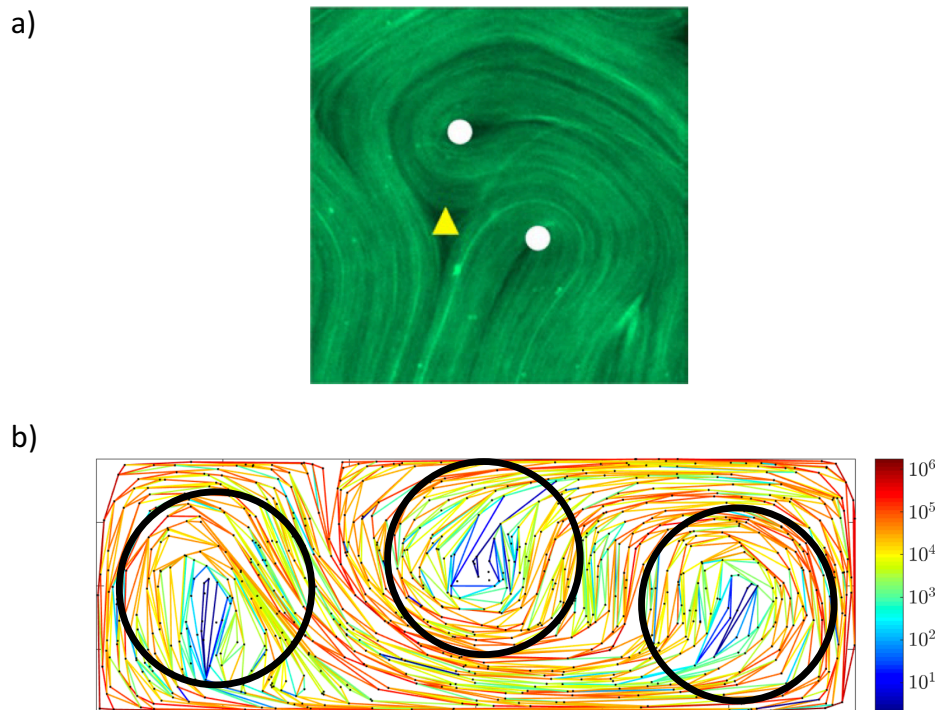


Figure 3.5: A comparison between the active nematic  $+\frac{1}{2}$  defects and period-3 coherent islands from the chaotic test flow used in Ch. 2 a) Fluorescence microscopy image of the active nematic fluid with topological defects marked. White circles denote  $+\frac{1}{2}$  defects. A yellow triangle denotes a  $-\frac{1}{2}$  defect. b) A band stretched by the braiding of the three period-3 islands (circled). *Image a) credit: Amanda Tan [8].*

As a first step to obtaining defect trajectories, the defects were tracked from particle image velocimetry (PIV) using fluorescence microscopy images of the microtubule bundles. Experimental images and tracked defects are pictured in Fig. 3.5a. Their positions are found at breakdowns in local ordering in the PIV velocity field estimations. Defect trajectories simply need to be extended for all time before and after their creation and annihilation, a task done with the PIV velocity data. Defect trajectories once again leaving or entering the field of view are left stationary on the boundary for all times they are inactive.

By applying E-tec to defect trajectories, we obtain better topological entropy results and verify that the defects act as stirring rods that drive the flow dynamics with their braiding. Figure 3.6 demonstrates remarkable consistency between the

Lyapunov exponents and the topological entropy due to the defect trajectories. Additionally, we verify once again that the active nematic flow increases in complexity with higher ATP concentrations, resulting in better global mixing.

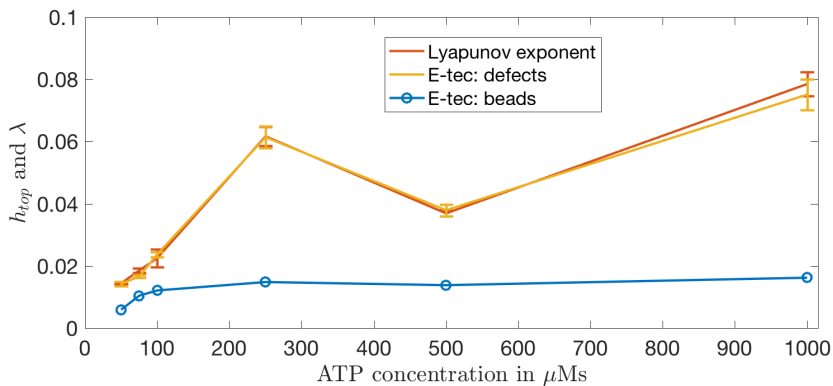


Figure 3.6: E-tec topological entropy measurements using defect trajectories in yellow. Also pictured are the same Lyapunov exponent averages and bead trajectory topological entropies from Fig. 3.4.

### 3.1.8 Acknowledgments

This project was a highly collaborative effort and the author recognizes the following for their contributions: Linda Hirst and Kevin Mitchell for designing the study; Amanda Tan for tirelessly carrying out the experiments and computing Lyapunov exponents; Spencer Smith for additional E-tec analysis; and Ulyses Olvera, Jorge Arteaga, and Sam Fortini for additional analysis. Results, authored by Linda Hirst, Kevin Mitchell, and Amanda Tan can be found in Tan et al [8]. The author wishes to acknowledge support from the National Science Foundation under grant number #1808926 titled “Self-mixing Active Fluids.”

## 3.2 E-tec and Coherent Sets

In a broad sense, coherent sets are sets of trajectories that “do the same thing.” Coherent sets exist as collections of regions in which material does not leak out. These special structures have been studied extensively in geophysical applications in recent years, as the ability to identify regions of mixing and material transport barriers has many applications: the emergence of coherent structures that help explain bacterial abundance in the Gulf of Mexico [94], the knowledge of how regions of fluid remain isolated from each other allows for invaluable prediction of the fate of oil spills [30,31], the coherent material core of Jupiter’s Great Red Spot [95], the large quantities of Indian Ocean water that is transported by the Agulhas Current up to thousands of kilometers with little to no mixing [31,96], just to name a few. Additionally, across flows in both the macro and micro scales, coherent sets can play a large role in invoking unpredictability [73], a hallmark of chaotic advection. As seen in the chaotic lid-driven cavity flow used to test and verify E-tec in 2D, coherent sets can act as stirrers, braiding material lines and topologically forcing an entropy lower bound. While the material in coherent sets might not be well mixed with the surrounding fluid, they may act as “ghost rods” that stir the fluid around them, promoting efficient mixing [73].

We verify below that E-tec will correctly handle trajectories sampled from coherent sets. As a test, three period-3 islands from the chaotic lid-driven cavity flow model defined in Ch. 2 are analyzed. Trajectories are uniformly and randomly seeded throughout the flow. The three islands are then sampled with additional random trajectories. Rubber bands placed around the convex hull of the island points are advected forward in E-tec. Though the islands all move in the “figure 8” golden braid motion, the bands do not grow in time under the flow dynamics, as shown in Fig. 3.7. The result would have been the similar for any band contained *inside* the islands rather than on the boundaries, as the points remain nearly-stationary relative to the islands’ movement.



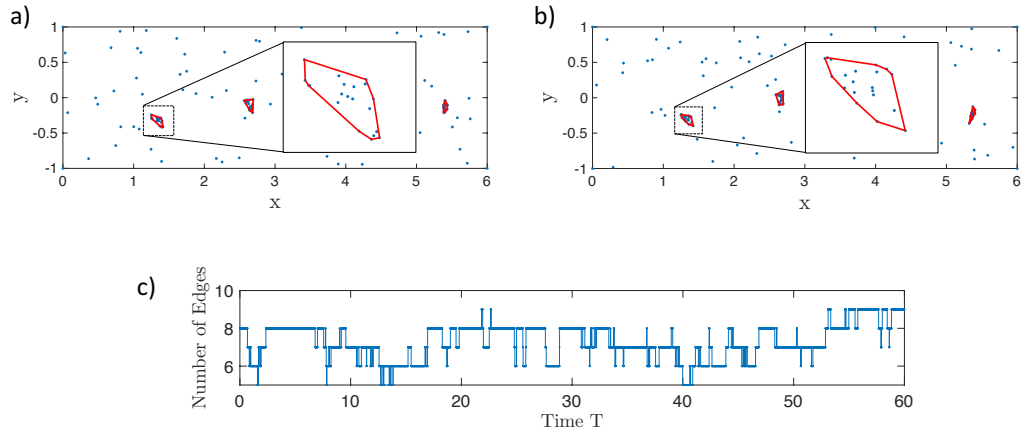


Figure 3.7: Displayed are bands around three coherent sets a) Initial bands placed around a three subset of points that lie in the interior of the coherent sets. (Blowup in inset.) b) The bands evolved forward in time, with negligible stretching. c) The number of edges in the band exhibits no net growth in time, indicating that the band encloses a coherent set.

### 3.3 E-tec, Escape Rates, and Unbounded Flows

In many real-world and experimental studies, flows are unbounded and trajectories can move apart indefinitely. This is readily seen in the active nematic study in Sect. 3.1 where trajectories continuously enter and exit the frame of view. Studies on the optimization of braiding patterns [34, 57, 58] have theoretical extensions to potentially-unbounded flows, though these works simply report on the topological entropy lower bound for planar flows.

Many questions remain regarding the nature of unbounded flows undergoing chaotic advection. The aim of this section is twofold: 1) to better understand how E-tec will perform on unbounded flows, and 2) to investigate the dependence of topological entropy on the ensemble densities and trajectory escape rates. To begin explore the relationship between E-tec applied to unbounded flows and trajectory escape rates using the following 2D time-dependent system:

$$\begin{array}{ll}
u_x = \sin(\pi x) \cos(\pi y) & u_x = -\sin(\pi(x - 1/2)) \cos(\pi(y - 1/2)) \\
u_y = -\cos(\pi x) \sin(\pi y) & u_y = \cos(\pi(x - 1/2)) \sin(\pi(y - 1/2)) \\
\text{for } n\tau_f \leq t < (n + 1/2)\tau_f & \text{for } (n + 1/2)\tau_f \leq t < (n + 1)\tau_f.
\end{array}$$

This system operates as a spatially-periodic mixer. It models a flow being mixed by rods on an infinite 2D lattice that are swapping places with their horizontal and vertical neighbors in clockwise and counterclockwise fashions, depending on the half-period. A template to help envision this motion is the braiding of the period-3 islands present in the test flow from Ch. 2, only here, rods extend infinitely in 2D and can also exchange vertically. The parameter value  $\tau_f = 2$  yields a chaotic, unbounded flow and is the parameter used for this study.

Trajectories will be initially seeded in two *bounding regions*, one consisting of a circle with radius  $d$  and another consisting of a box of side length  $d$ . For each region, the same trajectories will be advected three times, once allowing them to roam phase space unbounded (Fig. 3.8, on the left), and one for the each boundary region in which trajectories are frozen once leaving the boundary (Fig. 3.8, on the right). Trajectories exiting the regions will remain stationary for all time and will not be replaced.

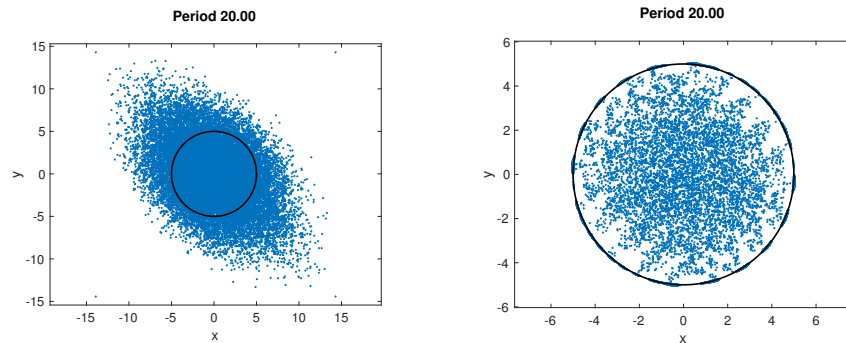


Figure 3.8: (Left) Unbounded trajectories initially seeded in circle of radius 5. (Right) Same initially-seeded trajectories are frozen once leaving the boundary.

### 3.3.1 Unbounded E-tec Entropy, Bounded E-tec Entropy

Ultimately, we aim to compare topological entropies for unbounded flows with those of the bounded flows. To accomplish this, three types of trajectory sets are used:

- i) (Unbounded) initially-seeded trajectories allowed to run unbounded,
- ii) (Bounded/Box) initially-seeded trajectories in a box that are frozen in place for all later times once leaving the box, and
- iii) (Bounded/Circle) initially-seeded trajectories in a circle that are frozen in place for all time once leaving the circle.

Bounding region of circle diameter and box length  $d = 2.5, 5, 7.5, 10, 15,$  and  $20$  are used in this study. Each bounding region is initially seeded in such a manner that point densities remain constant.

Outlined in the solid blue, red, and yellow curves in Fig. 3.9 are the topological entropy  $h_{top}$  calculations as a function of the bounding region size. The solid blue curve represents the E-tec entropy computed on trajectories from set i), those initially seeded in a region and allowed to evolve unbounded. This curve remains quite consistent throughout the size  $d$  of regions. In solid red and yellow are the E-tec entropy results from sets ii) and iii), those frozen once exiting the bounding regions. In purple is the expansion entropy calculation, defined in Appendix A, which generalizes topological and metric entropies by computing the rate at which a system expands volume elements. Most importantly, expansion entropy acts as an upper bound on  $h_{top}$  [50].

The three E-tec computations of  $h_{top}$  for the bounded and unbounded trajectory sets are consistent at larger values of  $d$ . At small  $d$ , the bounded trajectory sets result in significantly smaller values of  $h_{top}$ . This may seem surprising given that all regions start with the same ensemble densities, but an explanation to this discrepancy materializes when one considers that bounding region areas decrease at a faster rate than the regions' perimeters as  $d$  is tuned down; while boundary perimeters decrease linearly in  $d$ , the areas decrease quadratically, resulting in more opportunities for trajectories to escape.

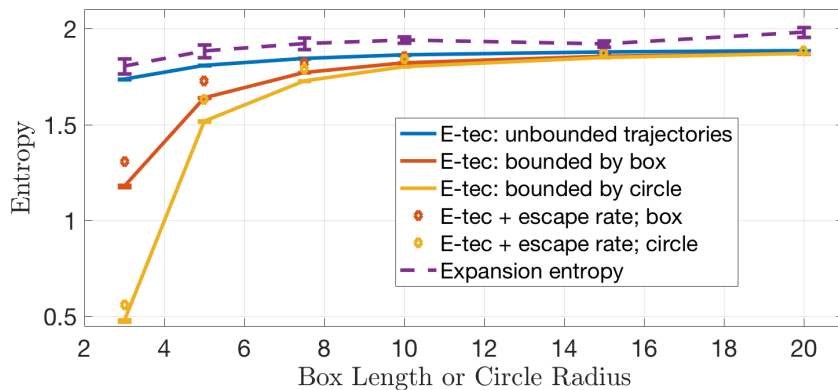


Figure 3.9: Blue, red, and yellow solid curves are E-tec results using bounded and unbounded trajectories as input. The diamonds represent the sum of the E-tec results and the bounding regions’ escape rates. In purple is the independent expansion entropy calculation which acts as an upper bound for the topological entropy.

### 3.3.2 Escape Rates

We next compute the exponential rates in which trajectories escape the initial bounding regions. The number of trajectories remaining in the bounding circles and boxes is tracked for all times and plotted on a log (base  $e$ ) scale in Fig. 3.10. At later times, the slope of the best-fit lines estimates the exponential decay rate for the number of active trajectories. In turn, the magnitude of this decay rate gives the exponential in which trajectories escape their regions, i.e. the escape rates.

Escape rate data is summarized in the Fig. 3.10 tables. As predicted, smaller values of  $d$  tend to yield larger escape rates. These are added to the bounded trajectory E-tec calculations of  $h_{top}$  in the Fig. 3.9 diamonds. For large values of  $d$ , the bounded trajectory entropy and escape rate sums are once again consistent with the E-tec computation of unbounded trajectories; at no point is the unbounded entropy in blue surpassed. However, the escape rates for smaller values of  $d$  are not large enough to compensate for the discrepancy between bounded and unbounded topological entropies. This is a surprising results, since theory suggests that the escape rates should fully compensate for the gap in entropies [97], i.e. we expect the unbounded topological to be equal to the sum of the bounded topological entropy plus the escape rates for all values of  $d$ .

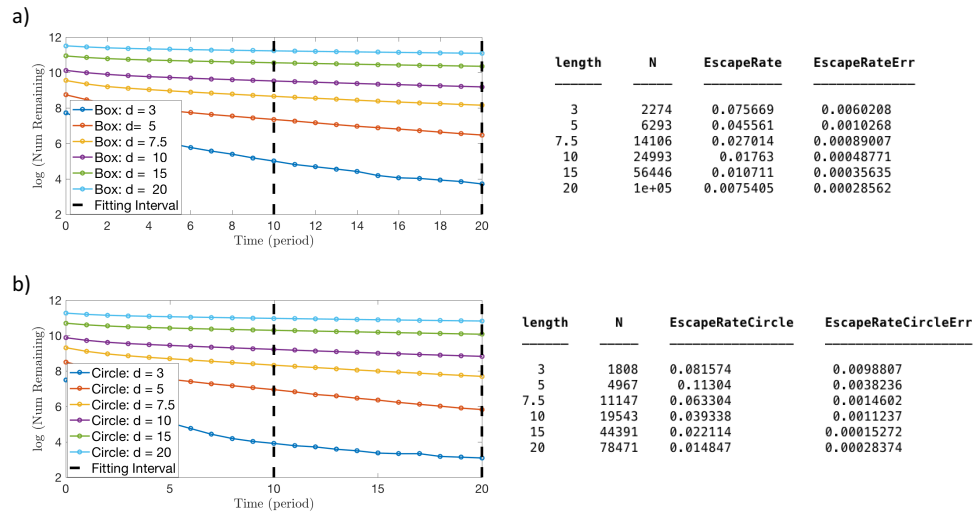


Figure 3.10: The natural log of the number of remaining trajectories in each bounding region as a function of time. Bounding boxes are in a) and bounding circles are in b). On the right are tables summarizing the escape rate data extracted from the plots. Listed data includes  $N$ , the number of initially seeded trajectories, and the escape rates and corresponding escape rate errors.

### 3.3.3 E-tec and Trajectory Replacement

To investigate the small entropy and escape rate values at small  $d$ , we maintain point densities in the bounding regions by replacing any trajectory that exits. In this setup, a new trajectory is randomly placed on the boundary and enters the bounding region at the instance another trajectory leaves. Since E-tec cannot handle broken trajectories, those entering are held stationary at the boundary for all times before flowing inward. If the E-tec calculations were to remain consistent for varying  $d$  with trajectory replacement, the escape rates would need to be further analyzed. In terms of applications to experimental data, this type of analysis would give valuable insight on E-tec's ability to calculate topological entropy when trajectory replacement is available. E-tec results for ensembles with trajectory replacement are displayed in Fig. 3.11. In this update to the entropy summary in Fig. 3.9, the new entropy result in black is noticeably larger than the expansion entropy result in dashed purple. This result is incorrect, as the new expansion entropy acts as the theoretical upper bound.

An explanation for the increase in entropy is as follows. For one, trajectory

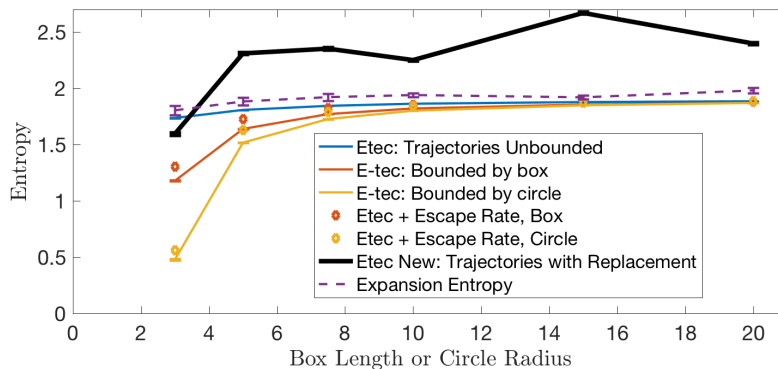


Figure 3.11: Updated Fig. 3.9 from above, with new E-tec topological entropy estimates (in black) for the trajectory sets with replacement.

replacement requires E-tec to handle ensembles containing many more trajectories. The bounding circle diameters of 3,5,7.5,10,15, and 20 require an ensemble of size 1808, 4967, 11147, 19543, 44391, and 78471 respectively to maintain, at all times, 360, 1000, 2250, 4000, 9000, and 16000 trajectories moving. In other words, for every single trajectory moving in the interior, there are roughly five frozen on the boundary at any given time.

Further investigation into the stationary boundary trajectories leads us to believe that E-tec is ill-suited to handle such large, static amounts of data on the boundary. Figure 3.12 shows a band stretched by E-tec with final band edges colored by weight. The color bar is on a logarithmic scale, so differences in color indicate exponentially more stretching. Note the abundance of black edges populating the boundary regions. This is explained by the large number of entering and exiting trajectories crossing over band edges attached to moving points. An edge anchored by two points in the boundary can be struck many times by points entering and leaving even though the anchors are stationary.

### 3.4 Conclusion

We verify that E-tec produces accurate and consistent results on experimental data. E-tec is applied to experimentally gathered trajectories of an active nematic flow on

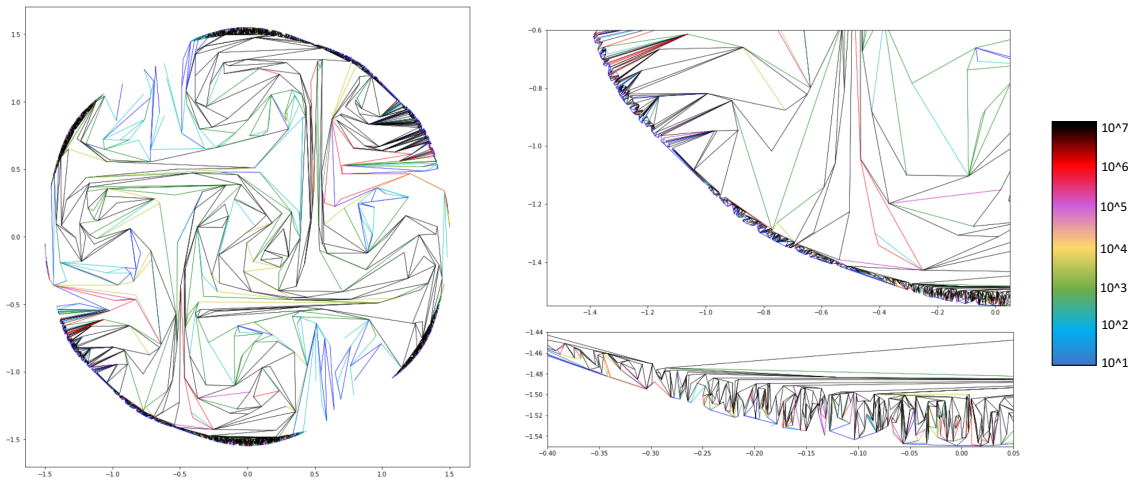


Figure 3.12: Final configuration of a band stretched by E-tec. Edge weights are colored by weight, with color bar shown to right (log scale coloring). Trajectory replacement is used, so many trajectories are frozen just outside of boundary, as seen in subsequent zooming

the length scale of 100's of microns. The resulting topological entropy calculations demonstrate that the flow exhibits chaotic advection, representing the first known study at the intersection of mixing dynamics and active matter nematic fluids. Lastly, E-tec results are shown to be consistent with the Lyapunov exponents of nearby trajectory pairs, further validating E-tec.

Next, we apply E-tec to coherent set trajectories. Bands are shown not to grow exponentially or algebraically, an expected but important result. Lastly, E-tec is used to investigate the entropy differences between bounded and unbounded trajectories. Mixed results are reported here, with the E-tec entropy difference between bounded and unbounded sets not fully recovered by the computed exponential escape rates. Further investigations reveal that implementing trajectory replacement to maintain consistent point densities among unbounded flows introduces superfluous edge weight growth near the boundaries.

# Chapter 4

## 3D E-tec

### 4.1 Introduction

In 3D, rich dynamics can occur, giving rise to more complex flow behavior and new phase space structures that may invoke new transport and mixing mechanics [98]. For some examples, the exponential rates of separation between nearby trajectories is commonly utilized in the form of finite-time exponent fields (FTLEs) [37, 38] for the discovery of 3D transport barriers and are commonly applied in 3D atmospheric and oceanic studies [99–103]. Normally hyperbolic invariant manifolds have been largely theorized to play a large role in 3D fluid transport for decades [104]. Recent studies of the development and evolution of stratospheric polar vortices that utilize FTLEs and Poincaré sections of observational data have since verified this to be the case [105, 106, 106]. Lastly, a new discovery in the field of microfluidics: new topological defects in 3D active nematic fluids comprised of microtubule bundles and molecular motors [107, 108].

However, many techniques used for studying 3D require high point densities or long-lived trajectories. The studies above utilizing FTLE fields required millions of trajectories and orders of magnitude more operations to reveal the finer-scale structure [100, 101]. Worse yet, studies involving velocity field estimations may require the linearizations of many millions of trajectories at each time step, severely hampering numerical costs. A natural remedy to these constraints may lie in the development



of a three dimensional extension to the E-tec algorithm.

We see in previous chapters that the computation geometric properties of E-tec are advantageous to a 3D extension as triangulations can be readily generalized to three dimensions and higher. The idea of using an advected dynamic triangulation in 3D to compute topological entropy was first proposed by Marc Lefranc [64–66] in 2006. Lefranc’s work was restricted to rubber band stretching by 2D periodic orbits. While no general algorithm for 3D was ever implemented, our work here represents the first implementation of his ideas.

Instead of rubber bands anchored between points in a planar flow, we consider two objects to stretch under the evolution of an ensemble of trajectories and the resulting dynamic triangulations. First are two-dimensional rubber sheets stretched around ensemble points. As points evolve in time, they carry the sheet along with them, stretching and folding it so that its growth reflects the flow complexity. This can be visualized in Fig. 4.1a. A red sheet encasing the single triangular face (1, 2, 3) of a 3D triangulation will be stretched upward as point 5 collides with it.

The second object to be stretched in three-dimensional space are one-dimensional rubber strings. Since points are not obstacle to strings in 3D, the moving edges of the triangulation act as stirrers and deform the strings. As the cartoon in Fig. 4.2 displays, the one-dimensional edges act as anchors, stretching and folding the string as the triangulation evolves in time with the ensemble. In a manner similar to the rubber band growth in 2D and the manifold stretching in 3D, the growth rate of the string reflects the flow complexity and acts as a lower bound for the topological entropy.

### 4.1.1 Troubles in 3D

In 2D, the only type of band deformations to track were point-band collisions. This was sufficient to record whether a point was striking or releasing from the band. Point-face collisions are analogous to this in 3D and remain rather intuitive. A point strikes a triangular face of the sheet and it grows. A point releases from sheet and it returns taut. The triangular edges stretch strings in a similar fashion. However, a significant

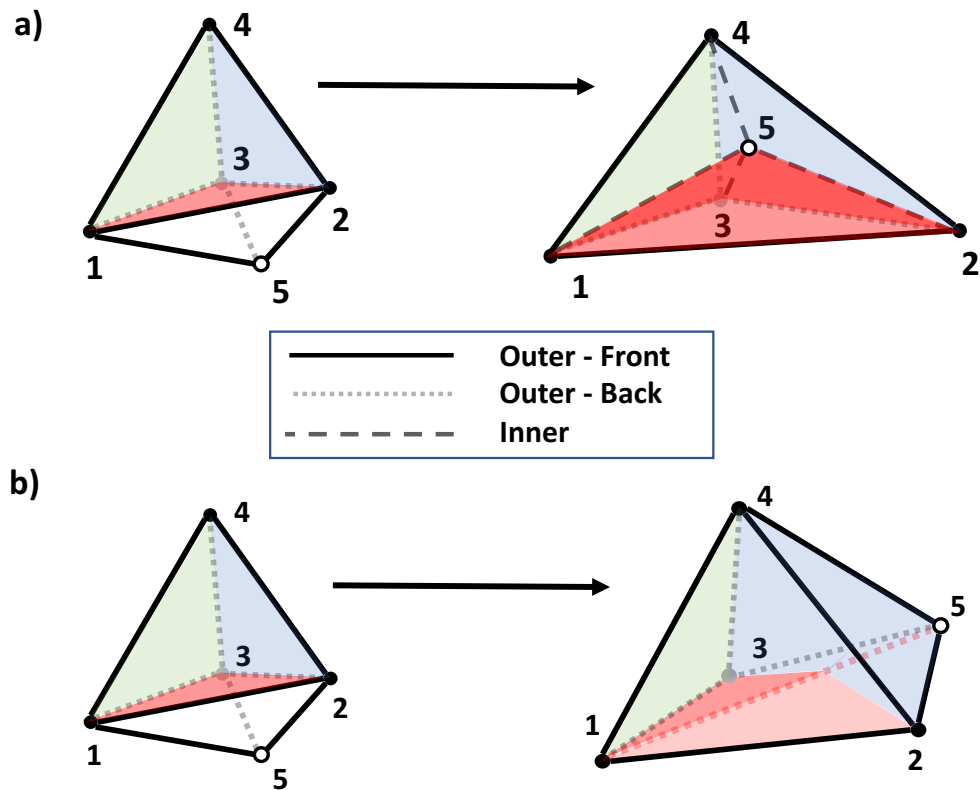


Figure 4.1: **Collisions in 3D:** Depiction of the two collisions (point-face and edge-edge) that may occur among points, edges, and faces of a 3D triangulation as points move. Triangulation faces are colored to help better visualize the stretching of potential elastic sheets. a) Point 5 (hollow) moves up and collides with the triangular face (1, 2, 3). As a result, tetrahedra (1, 2, 4, 5), (1, 3, 4, 5), and (2, 3, 4, 5) are added to the triangulation to replace the newly invalid tetrahedron (1, 2, 3, 4). A potential elastic sheet wrapped around face (1, 2, 3) is now represented with weights on faces (1, 2, 5), (1, 3, 5), and (2, 3, 5). b) From the same left configuration as above, point 5 moves to the right and into the page, resulting in an intersection between edges (1, 5) and (2, 3). The resulting re-triangulation depends on the number of tetrahedra sharing edge (2, 3). Furthermore, the new weight representation of a potential red sheet is left ambiguous.

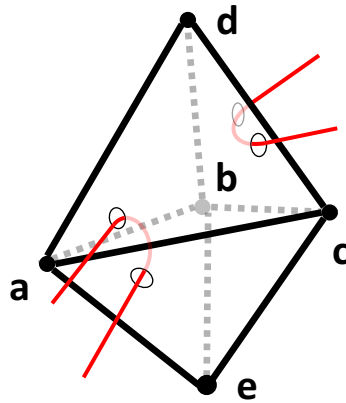


Figure 4.2: **Strings Anchored by Edges:** Pieces of a rubber string (in red) are wrapped around triangulation edges (a, c) and (c, d).

problem remains – point-face collisions are not the only type of re-triangulation event that can occur in 3D. Fig. 4.1b depicts the resulting shape of two tetrahedra that have undergone an edge-edge collision.

There remains much ambiguity on how to best re-triangulate the tetrahedra involved in edge-edge collisions. Tetrahedra not shown but adjacent to the ones drawn in Fig. 4.1b will certainly be affected, particularly all other tetrahedra containing edge 2-3. Less is known about how to best represent the sheet. The addition of an auxiliary point where the edges collide could allow for re-triangulation, but this would lead to larger ensembles and additional trajectories to compute.

### 4.1.2 A New E-tec Scheme

Motivated by the difficulty of managing edge-edge collisions and our desire for avoiding them, we introduce Dual E-tec, a modified version of E-tec that will allow us to bypass these collisions entirely. Dual E-tec works in 2D and 3D by simply encoding rubber bands and sheets into the triangulation in a new manner. Instead of representing a rubber manifold in  $d$ -dimensional space as the number of times it traverses *along* a  $(d - 1)$ -dimensional simplex (a band along an edge or a sheet along a face), Dual E-tec represents a rubber manifold by counting the number of times it crosses *through* each 1D edge of the triangulation. These two representations are dual to each

other (thus inspiring the name) in much the same way that knowledge of the Voronoi diagram between a finite set of planar points  $S$  implies knowledge of the Delaunay triangulation of  $S$ .

Two sheets are drawn on the left in Fig. 4.3 with their Dual E-tec edge weight representations on the right. This figure hints at the single most important advancement that Dual E-tec ushers in: since the rubber manifolds are no longer attached along the edges of the triangulation, the triangulation at any given time is no longer dependent on the current state of the rubber manifold. Now, since any triangulation may be used at any given time along an ensemble's evolution, we are free to seek out re-triangulation methods that eliminate edge-edge collisions entirely. Once the full evolution of the triangulation is recorded, any rubber manifold can be stretched by points and any rubber string can be stretched by the triangulation edges.

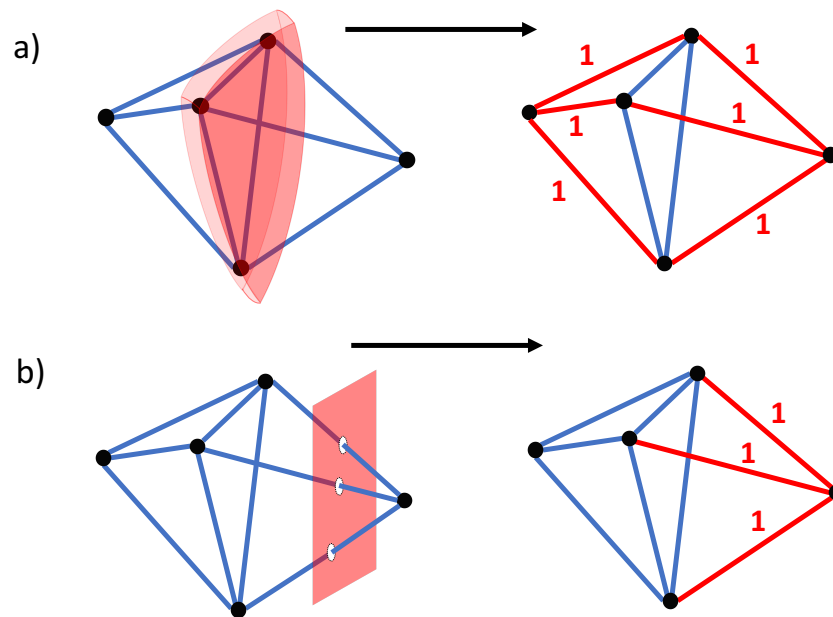


Figure 4.3: **Encoding Rubber Sheets in the Triangulation:** Two types of two-dimension rubber sheets and their Dual E-tec weight representation. a) A rubber sheet folded around the middle triangular face on the left (the folding over points is not drawn in 3D). This sheet is represented in Dual E-tec by weights along each edge it passes through. Weighted edges are in red. b) A rubber sheet in the form of a plane is on the left. Weighted edges that represent the sheet are shown on the right.

### 4.1.3 Chapter Summary

The remainder of this chapter is presented in five sections. The 2D and 3D Dual E-tec update rules for rubber manifold representation and growth are introduced and derived in Sect. 4.2. Next, our motivation and reasoning for choosing to maintain a Delaunay triangulation (pictured in Fig. 4.4) among moving, kinetic points at all times is presented in Sect. 4.3. This all culminates in Sect. 4.4 where the new Dual E-tec scheme is used in tandem with the new triangulation advection technique to build the first 3D ensemble-based topological entropy calculation. Here, 3D E-tec performance is evaluated on the familiar chaotic lid-driven cavity flow that was used in 2D, only this time with trajectories raised to a third dimension. For a second E-tec stretching scheme in 3D, one-dimensional rubber strings are anchored by triangulation edges and stretched and folded. The manner in which their growth is tracked is presented and evaluated in Sect. 4.5. Lastly, future considerations for algorithm verification are presented in Sect. 4.6, followed by a brief summary of our contributions in Sect. 4.7.

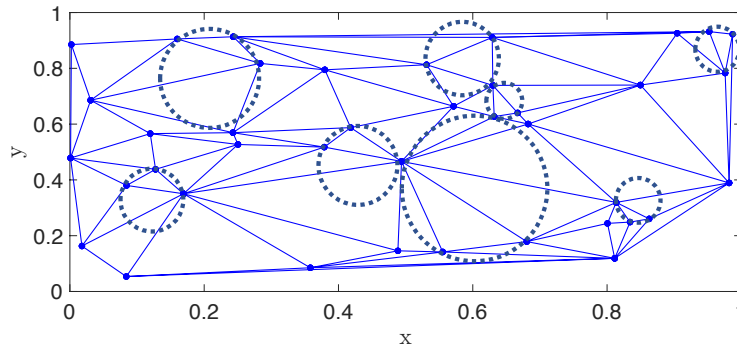


Figure 4.4: A Delaunay triangulation of forty points. Eight triangle circumferences with empty interiors are shown.

## 4.2 Dual E-tec

Figure 4.5 gives a graphical depiction of how the Dual E-tec band representation scheme represents a band in 2D by counting the number of times it crosses through each edge and how this differs from the original algorithm. Upon a collision event,

core triangulation updates are handled in the same local way among both schemes: a single interior edge is replaced with the other interior edge among different points. However, there is no need to track outer triangles using Dual E-tec; the weight update rules translating edge weight to the minimum number of taught band edges (derived below) appropriately account for the stretching, folding, and releasing of a band.

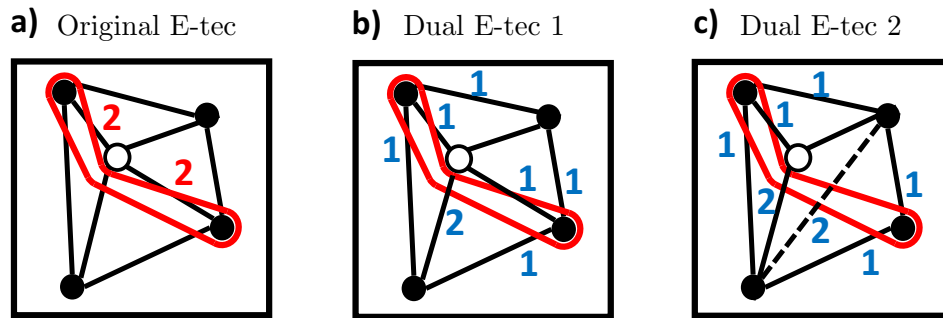


Figure 4.5: **Comparison of Original and Dual E-tec:** The same band and point configuration are displayed in all three sub-figures. However, the band representation among the edges and weights in the original scheme in a) differ from Dual E-tec representations in b) and c) a) In the original E-tec scheme, this is the only possible triangulation among these five points in which the band may be represented. b) The same triangulation as the left, this time with the triangulation's Dual-E-tec weight representation. c) A second possible triangulation of the five points and corresponding edge weights, with the changed edge dashed for emphasis.

In addition to eliminating outer triangle event tracking, the biggest advantage of Dual E-tec lies in the fact that, for a given band stretched around a configuration of points, *any* triangulation may be used to represent the band. Figures 4.5b and c depict this, showing two valid triangulations and the differing dashed edge on the right housing a different weight than the edge it replaced. In essence, the triangulation is no longer married to the current state of the band. As seen in Fig. 4.6, this allows for simpler initial band configurations that no longer rely on constrained triangulations.

Since the triangulation and initial configurations are no longer dependent on each other, the triangulation updates involving band weight are no longer dependent on the band evolution as points move around. Thus, Dual E-tec allows users to fully separate the triangulation evolution from the band stretching. By representing each collapse

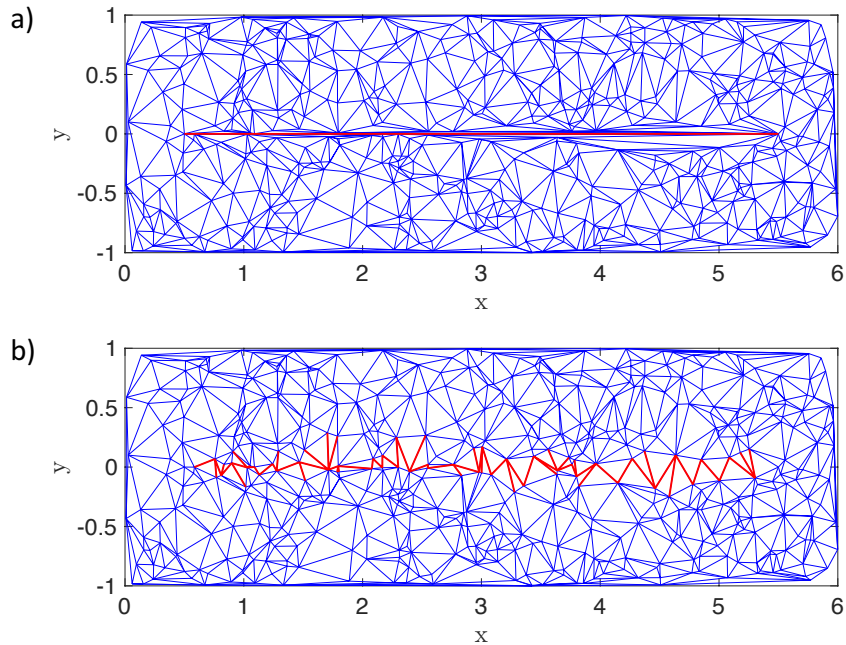


Figure 4.6: **E-tec and Dual E-tec Numerical Band Initialization:** a) The original E-tec representation of an initial band (in red). Notice the constrained Delaunay triangulation. b) The Dual E-tec representation of the band anchored by same two points in a). The triangulation edges passing through the band are shown in red. Notice this is an unconstrained Delaunay triangulation.

event as an operator, one simply needs to record the unique sequence of operators generated by the full evolution of the triangulation and then apply this sequence to any band around points in the initial triangulation. (This process is similar to the recording of Dynnikov coordinates and the separate application of their actions onto bands [5, 42].) This is a stark juxtaposition to the original evolution that would occur in Fig. 4.6a where each initial band evolution required its own unique sequence of collision update operators.

#### 4.2.1 2D Collapse Event Weight Update Rule

Dual E-tec does not detect direct collisions between moving points and the band. Instead, it seeks to translate weight crossings among edges into the minimum number of taut band edges when a re-triangulation event occurs. To illustrate how this is

done, let us define edge weight variables for a collapse event. In Fig. 4.7, we consider the pair of triangles  $T$  and  $B$  involved in a collision. Edge weights through given edges are labeled  $(T_2, T_3)$ ,  $(B_2, B_3)$ , and  $(C = T_1 = B_1)$ , where  $C$  corresponds to the shared edge. Additional auxiliary variables (in lower case) refer to the number of band edges that wrap around a particular point in a given triangle. These auxiliary variables at points represent the portion of edge weights in the two adjacent edges that are directly connected to this triangle.

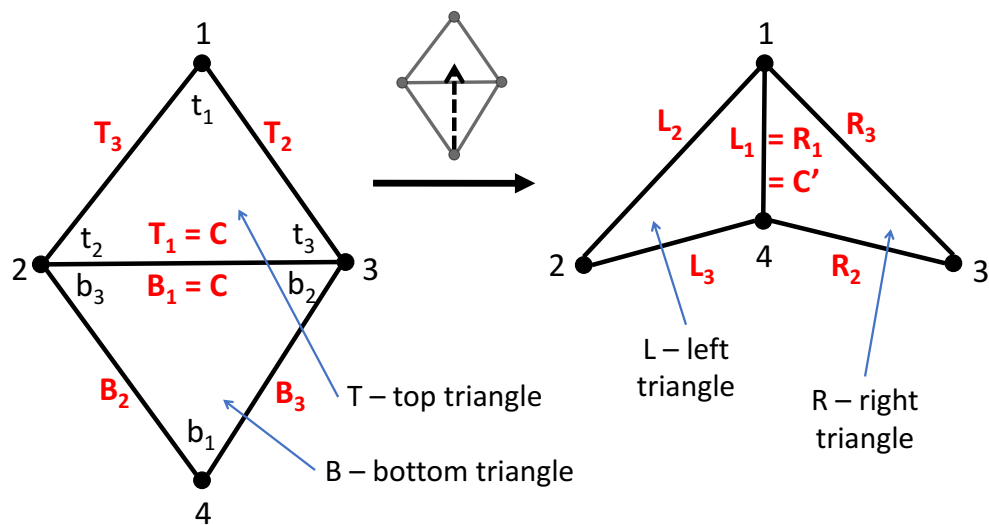


Figure 4.7: **2D Collapse Event:** Triangles, edges, and labeled variables before and after a collapse event. Edge weights (in red) and refer to the number of times a band passes through the labeled edge. Auxiliary lower case variables at points refer to the number of times a band passes through the two edges adjacent to the labeled point.

In this generalization of a collapse event in Fig. 4.7, the bottom point is moving north and collapsing  $B$ . The horizontal interior edge with weight  $C$ , shared between  $T$  and  $B$ , is struck and removed, replaced by the vertical interior edge connected by the other two points. After the re-triangulation, the weights passing through the four outer edges remain unchanged, i.e.

$$T_3 = L_2, \quad T_2 = R_3, \quad B_2 = L_3, \quad \text{and} \quad B_3 = R_2.$$

Only the new weight  $C'$  (or  $L_1$  or  $R_1$ ) lying on the new interior edge changes and is



given by:

$$C' = \frac{1}{2}(T_3 + T_2 + B_3 + B_2) - C + \frac{1}{2}|T_3 - T_2 + B_3 - B_2|. \quad (4.1)$$

To help understand how only one of the five unique edge weights changes after a collision, consider the specific weight update configuration in Fig. 4.8. Only the three bands passing through the central edge will be affected by the bottom point moving north. The single band passing through the moving point will remain un-stretched, while the bands wrapped tightly along the other three points will not interact with the moving point. This pre-collision interior band weight  $C$  is given by

$$C = \frac{1}{2}|T_3 - T_2 + B_3 - B_2|. \quad (4.2)$$

It can be thought of as the number of band edges entering and exiting the two adjacent triangles through diagonal edges that share no common points. Note this can only occur with only a single pair of opposing edges, as the band is not allowed to cross over itself.

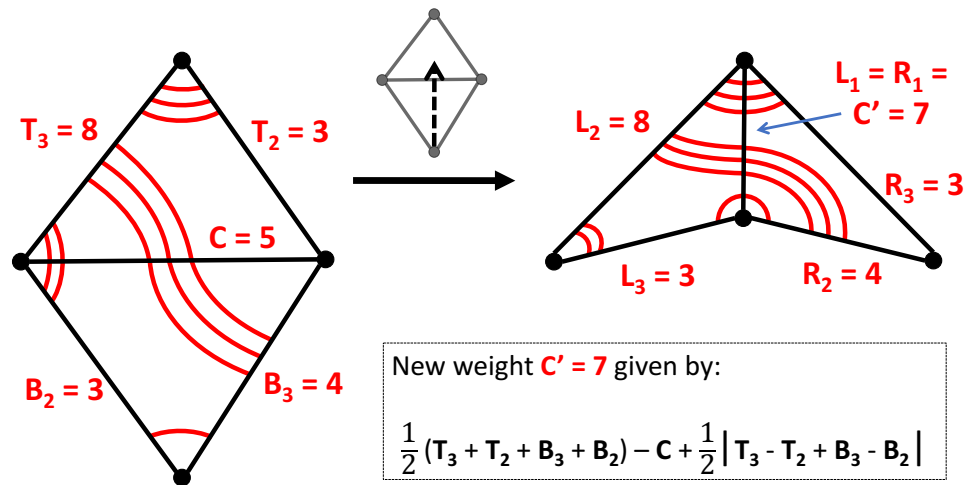


Figure 4.8: **Example of a 2D Collapse Event:** A specific example displaying new edge weight  $C'$ .

To justify the new edge weight  $C'$  in Eq. 4.1, we turn to the lower case auxiliary

variables in Fig. 4.7. For a particular triangle, the “corner weights” can be thought of as the number of band edges wrapped taut around the triangulation corners. In Fig. 4.7, the pre-collision auxiliary weights are (starting from the top and bottom points and moving clockwise) :

$$t_1 = 3, t_2 = 5, t_3 = 0,$$

and

$$b_1 = 1, b_2 = 3, b_3 = 2.$$

What follows is a linear relationship between the “corner weights” and the edge weights, given by:

$$T_1 = t_2 + t_3, T_2 = t_1 + t_3, T_3 = t_1 + t_2 \quad (4.3a)$$

for the top triangle, followed by

$$B_1 = b_2 + b_3, B_2 = b_1 + b_3, B_3 = b_1 + b_2. \quad (4.3b)$$

The new vertical edge is with weight  $C'$  will be forced to go through the top and bottom corner weights  $t_1$  and  $b_1$ , as well as the diagonal band weight given by  $|t_2 - b_3|$  (or  $|t_3 - b_2|$ ), and is expressed in terms of “corner weights” by:

$$C' = t_1 + b_1 + |t_2 - b_3|. \quad (4.4)$$

Note the absolute value is required because we do not know the direction of the diagonal band, i.e. we cannot tell if the diagonal band passes through  $t_2$  or  $b_3$ . Now, we express  $C'$  in terms of capital edge weights. Luckily, we can express  $t_1$  and  $b_1$  in

terms of these same edge weights by inverting Eqs. 4.3a and 4.3a. This yields

$$\begin{aligned} t_1 &= \frac{1}{2}(T_2 + T_3 - T_1) \\ t_2 &= \frac{1}{2}(T_1 + T_3 - T_2) \\ t_3 &= \frac{1}{2}(T_1 + T_2 - T_3) \end{aligned} \tag{4.5a}$$

and

$$\begin{aligned} b_1 &= \frac{1}{2}(B_2 + B_3 - B_1) \\ b_2 &= \frac{1}{2}(B_1 + B_3 - B_2) \\ b_3 &= \frac{1}{2}(B_1 + B_2 - B_3). \end{aligned} \tag{4.5b}$$

Subbing in  $t_1$ ,  $b_1$ ,  $t_2$ , and  $b_3$  from Eqs. 4.5a and 4.5a into the auxiliary variable form of  $C'$  in Eq. 4.4 yields:

$$\begin{aligned} C' &= t_1 + b_1 + |t_2 - b_3| \\ &= \frac{1}{2}(T_2 + T_3 - T_1 + B_2 + B_3 - B_1) + \\ &\quad \frac{1}{2}|T_1 + T_3 - T_2 - (B_1 + B_2 - B_3)|, \end{aligned}$$

which is equivalent to Eq. 4.1 once we recognize that  $T_1 = B_1 = C$ .

A numerical comparison of the two methods is found in Fig. 4.9. The weight total between the two do not match perfectly, though the ratio between the two seems to be stable. Of course, Dual E-tec scales better in runtime for this example, mainly due to stripping away the necessity of tracking outer triangles and points releasing from the band.

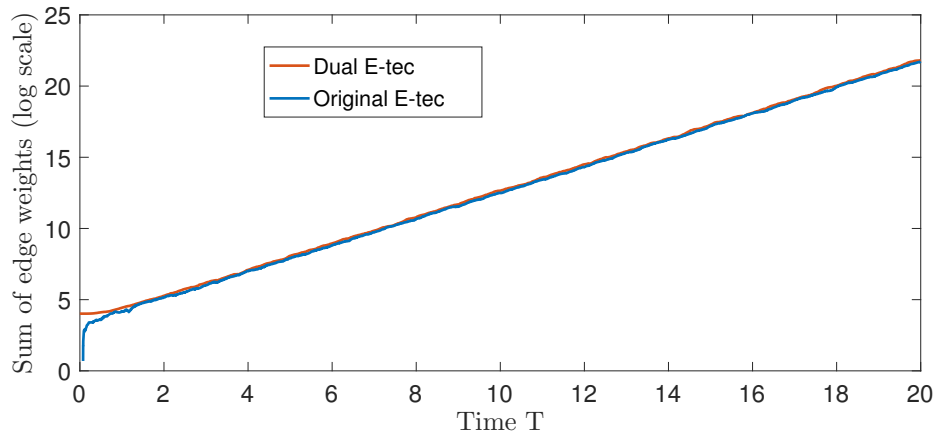


Figure 4.9: **Dual and Original E-tec Numerical Comparison:** Comparison of the two methods using the same 100 point trajectory set and initial band. The confidence intervals for both slopes (starting at fit time  $T = 5$ ) overlap at  $(0.9620, 0.9624)$ .

### 4.2.2 3D Collapse Event Weight Update Rule

Two types of collapse events occur in 3D: a point-face collision and an edge-edge collision. We discuss here only the point-face collapse events and the corresponding update rule. As depicted in Fig. 4.1a, two tetrahedra are converted into three tetrahedra when this type of collision occurs. A new edge is added to the triangulation in addition to three new faces, all of which contain the new edge. See Fig. 4.10 for the geometry and a depiction of sample rubber surface we wish to represent using edge weights.

We use four ways to encode how a rubber surface (hereby referred to as a “sheet”) intersects a tetrahedron: a) counting the number of sheet intersections with the four faces, or b) counting the number of sheet intersects with six edges, c) classifying how the vertices are partitioned from each other if the sheet were to “cut” the edges, and d) classifying how the sheet partitions the set of vertices within the faces. Using the top tetrahedron  $T$  with labeled vertices Fig. 4.10, I label and explain each variable in a bit more detail:

- a) Faces and their weights are denoted  $F_i^T$ :  $F$  for the face, subscript  $i$  for the face across from vertex  $i$ , and  $T$  for the specific tetrahedron. For the example sheet

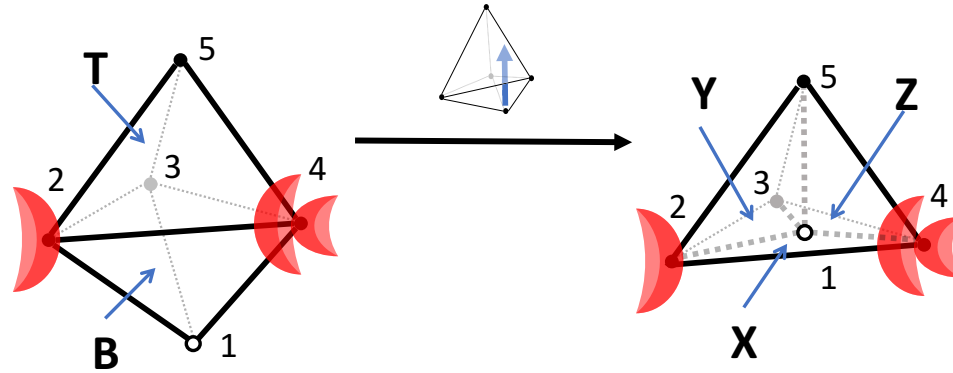


Figure 4.10: **3D Dual E-tec Collapse Event Update:** A point-face collision and subsequent re-triangulation is depicted. As the hollow point 5 move north, edge (4, 5) is added and tetrahedra  $T$  and  $B$  are converted into the three final tetrahedra  $X$ ,  $Y$ , and  $Z$ . Sheets wrapped taut around points 1 and 2 are shown in red. The Dual E-tec representation of this sheet is given by weights along edges in which the sheet passes through.

configuration in Fig. 4.10, faces have weight  $F_2^T = 2$ ,  $F_3^T = 3$ ,  $F_4^T = 1$ , and  $F_5^T = 3$ .

- b) Edges and their weights are denoted  $E_{i,j}^T$ :  $E$  for the edge, subscript  $ij$  for the two edge endpoints, and  $T$  for the specific tetrahedron. The superscript may be dropped since the vertex labeling is unique. For the example sheet configuration, the edge weights are  $E_{2,3} = 1$ ,  $E_{2,4} = 3$ ,  $E_{2,5} = 1$ , and  $E_{3,5} = 0$ .
- c) Vertex partitions are denoted  $V_{[i],[j],[k],[l]}^T$  or  $V_{[i,j],[k,l]}^T$ :  $V$  for the vertex partitioning,  $T$  for the tetrahedron, and either  $[i][j,k,l]$  or  $[i,j][k,l]$  for the partitioning. This partitioning informs us which vertices are mutually reachable by edges once the sheet has “cut” the intersecting edges. For the same example sheet,  $V_{[5],[2,3,4]}^T = 0$ ,  $V_{[3],[2,4,5]}^T = 0$ ,  $V_{[2],[3,4,5]}^T = 1$ ,  $V_{[4],[2,3,5]}^T = 1$ ,  $V_{[2,3],[4,5]}^T = 0$ ,  $V_{[2,4],[3,4]}^T = 0$ , and  $V_{[2,5],[3,4]}^T = 0$ . These variables classify how a surface intersects the tetrahedron by noting how the surface partitions the four vertices into two groups. For convenience, one half of the partitioned points will be dropped to write  $V^T = (V_2^T, V_3^T, V_4^T, V_5^T, V_{2,3}^T, V_{2,4}^T, V_{2,5}^T)$ .

d) Lastly, we focus on the sheet partitioning the set of vertices within each face. Denote each variable as  $F_i^T V_{[j],[k],[l]}$ :  $F_i^T$  for the face (as above) and  $V_{[j],[k],[l]}$  for the vertex partitioning amongst the face's vertices. (We use  $V_j$  for short.) Twelve of these variables exist for the tetrahedron  $T$ , with a few being  $F_3^T V_2 = F_3^T V_4 = 1$  and  $F_5^T V_2 = F_5^T V_1 = 1$ , indicating vertices 2 and 4 are isolated from the other vertices in both faces opposing vertex 3 and 5 (given by  $F_3^T$  and  $F_5^T$ , respectively).

The update rules are easiest to derive in terms of vertex partitions  $V^T$ , but are most simply stated and easiest to implement in terms of edge weights  $E^T$ . To connect the two representations, we examine how to go back and forth between the four variable sets. Focusing on a single tetrahedron (and this dropping the superscript  $T$ ) and its four vertices  $i, j, k$ , and  $l$ , connections between the four variable sets  $F^T, E^T, V^T$ , and  $F^T V$  are given by:

- $V^T \rightarrow E^T$

From inspection, we have

$$E_{i,j} = V_i + V_j + V_{i,j} + V_{i,m}. \quad (4.6)$$

Six of these equations exist for each tetrahedra.

- $V^T \rightarrow F^T$

There are four of the following equations:

$$F_i = V_j + V_k + V_m + V_{j,k} + V_{k,m} + V_{m,j}. \quad (4.7)$$

- $E^T \rightarrow F^T$

$$F_i = \frac{1}{2}(E_{j,k} + E_{k,m} + E_{m,j}). \quad (4.8)$$

- We aim for the connection  $E^T \rightarrow V^T$  so that we can start with  $E^T$ , translate to  $V^T$ , implement the update rules, then translate back to  $E^T$ . This is not

obvious from inspection, so we consider two intermediate steps  $E^T \rightarrow F^T V$  and  $F^T V \rightarrow V^T$ .

- $E^T \rightarrow F^T V$

This is equivalent to the 2D variable case for expressing the number of times a band wraps around a point, expressed using the triangle's edges weights:

$$F_i V_j = \frac{1}{2}(E_{j,k} + E_{m,j} - E_{m,k}). \quad (4.9)$$

- $F^T V \rightarrow V^T$

Two connections are made:

$$\begin{aligned} V_{i,j} &= (F_j V_i - F_k V_i)^+ = (F_j V_i - F_m V_i)^+ \\ &= (F_i V_h - F_k V_j)^+ = (F_i V_j - F_m V_j)^+ \\ &= (F_m V_k - F_i V_k)^+ = (F_m V_k - F_j V_k)^+ \\ &= (F_k V_m - F_i V_m)^+ = (F_k V_m - F_j V_m)^+, \end{aligned} \quad (4.10)$$

where  $S^+$  is defined as  $\max(0, S) = \frac{S}{2} + \frac{|S|}{2}$ . We also have

$$\begin{aligned} V_i &= F_j V_i - V_{i,j} \\ &= F_k V_i - V_{i,k} \\ &= F_m V_i - V_{i,m}. \end{aligned} \quad (4.11)$$

Eq. 4.10 is justified by considering the tetrahedron  $T$  in Fig. 4.10. We have the following simple relations for the face vertex partitions and the full vertex partitions about vertex 4:

$$F_1 V_4 = V_4 + V_{1,4}, \quad F_2 V_4 = V_4 + V_{2,4}, \quad \text{and} \quad F_3 V_4 = V_4 + V_{3,4}.$$

We know only one of  $V_{3,4}$ ,  $V_{2,4}$ , and  $V_{1,4}$  is nonzero. By combining in this one

example, we have:  $F_1V_4 = F_2V_4 - V_{2,4} + V_{1,4}$ . Rearranging yields

$$V_{1,4} = F_1V_4 - F_2V_4 + V_{2,4}.$$

If  $V_{1,4}$  is nonzero, then  $V_{2,4} = 0$  and  $V_{1,4} = F_1V_4 - F_2V_4$ . If  $V_{4,1} = 0$ , then either  $V_{2,4} = 0$  (resulting in  $F_1V_4 - F_2V_4 = 0$ ) or  $V_{2,4} > 0$  (resulting in  $F_1V_4 - F_2V_4 < 0$ ). All three cases are brought together in

$$V_{1,4} = (F_1V_4 - F_2V_4)^+.$$

This logic gives all parts of Eq. 4.10, while Eq 4.11 is simply a rearrangement of  $F_1V_4 = V_4 + V_{1,4}$ ,  $F_2V_4 = V_4 + V_{2,4}$ , and  $F_3V_4 = V_4 + V_{3,4}$ , defined earlier in this paragraph.

- $E^T \rightarrow V^T$

Combining Eq. 4.9 with Eqs. 4.10 and 4.11 gives two additional equations:

$$\begin{aligned} V_{i,j} &= \frac{1}{2}(E_{i,k} + E_{m,j} - E_{m,k} - E_{i,j})^+ \\ &= \frac{1}{2}(E_{i,m} + E_{k,j} - E_{m,k} - E_{i,j})^+ \end{aligned} \quad (4.12)$$

and

$$\begin{aligned} V_i &= \frac{1}{2}(E_{i,k} + E_{m,i} - E_{m,k}) - V_{i,j} \\ &= \frac{1}{2}(E_{i,j} + E_{m,i} - E_{m,j}) - V_{i,j} \\ &= \frac{1}{2}(E_{i,j} + E_{k,i} - E_{k,j}) - V_{i,j}. \end{aligned} \quad (4.13)$$

Next, we consider how the  $V^T$  and  $V^B$  partition variables for adjacent tetrahedra  $T$  and  $B$  glue together at their shared face. See Fig. 4.11 for a projection of the two tetrahedra with the shared face compressed down to a single line.



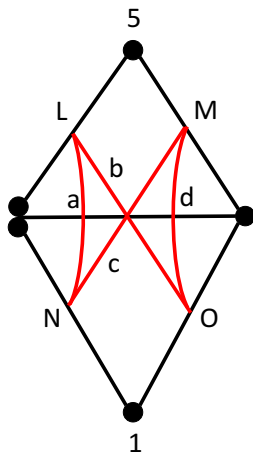


Figure 4.11: **Adjacent Tetrahedra Vertex Partition Variables:** The top and bottom tetrahedra involved in the collision event. In this projection, two of the shared vertices are on the left and one is on the right. This represents one of three combinatorially possible projective side views of the pair of tetrahedra. The red lines are possible ways that a surface could pass through both tetrahedra (viewed edge-on).  $L$  and  $M$  represent two of the  $V^T$  variables (partitioning the top vertices 2-2 and 1-3 respectively).  $N$  and  $O$  are the analogous  $V^B$  variables. The variables  $a, b, c$ , and  $d$  encode how  $L$  and  $M$  split and re-distribute to form  $N$  and  $O$ .

- The goal is to invert each of the four relationships found in Fig. 4.11:  $L = a + b$ ,  $M = c + d$ ,  $N = a + c$ , and  $O = b + d$ . Notice that  $b$  and  $c$  cannot both be zero. Consider  $(M - O + N - L)^+/2$ . Substituting this into the above relationships yields  $(c - b)^+ = c$ . Furthermore, using  $L + M = N + O$ , we arrive at  $c = (M - O)^+ = (N - L)^+$ . These inverted relationships are summarized in the following:

$$\begin{aligned}
 a &= L - b = N - c \\
 b &= (O - M)^+ = (L - N)^+ \\
 c &= (M - O)^+ = (N - L)^+ \\
 d &= M - c = O - b
 \end{aligned}
 \tag{4.14}$$

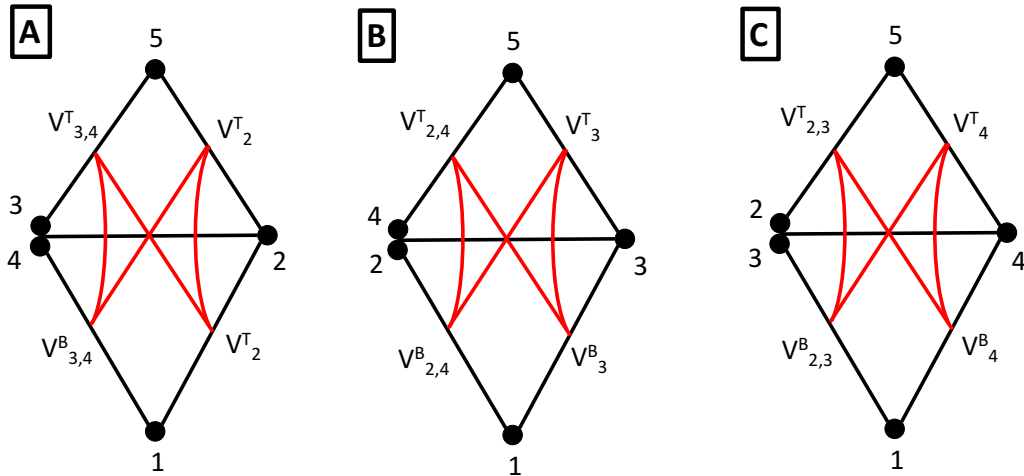


Figure 4.12: **Split Crossing Variables:** The three different side views of the two tetrahedra before the collapse event. Bold and boxed letters designate a new set of defined variables.

- The three different views Fig. 4.11 are displayed in Fig. 4.12. In it are three new sets of defined variables: **A**, **B**, and **C**. Using Eq. 4.14 above, we get the three sets of split crossing variables  $(\mathbf{A}_a, \mathbf{A}_b, \mathbf{A}_c, \mathbf{A}_d)$ ,  $(\mathbf{B}_a, \mathbf{B}_b, \mathbf{B}_c, \mathbf{B}_d)$ , and  $(\mathbf{C}_a, \mathbf{C}_b, \mathbf{C}_c, \mathbf{C}_d)$ , all in terms of the  $V^T$  and  $V^B$  variables.

Finally, we put together all of the preceding work and find the update rules for edges in terms of only the edges. First, we find  $V^X$  in terms of the crossing variables  $V^T$  and  $V^B$ . We find the weight update for the new edge  $E_{1,5}$  in the point-face collision shown in Fig. 4.10. As a reminder, the new post-update tetrahedron  $X$  is comprised of points 1, 2, 4, and 5.

- $V^T, V^B \rightarrow V^X$

The following list of variable relationships is useful for dealing with the re-triangulation of the edge-edge collapse case.

$$\begin{aligned}
V_1^X &= V_1^B + \mathbf{B}_c \\
V_2^X &= \mathbf{A}_d + \mathbf{C}_a \\
V_4^X &= \mathbf{A}_a + \mathbf{C}_d \\
V_5^X &= V_5^T + \mathbf{B}_b \\
V_{1,2}^X &= \mathbf{A}_c + \mathbf{C}_b \\
V_{1,4}^X &= \mathbf{A}_b + \mathbf{C}_c \\
V_{1,5}^X &= \mathbf{B}_a
\end{aligned} \tag{4.15}$$

We note that  $V^Y$  and  $V^Z$  are not shown, though they are related to  $V^X$  by symmetry.

- By inspection of the right-hand side of Fig. 4.10 and making use of Eq. 4.6, we arrive at

$$E_{1,5} = V_1^X + V_5^X + V_{1,2}^X + V_{1,4}^X. \tag{4.16}$$

Plugging this into the set of equalities in Eq. 4.15, we get

$$E_{1,5} = V_1^B + V_5^T + \mathbf{A}_b + \mathbf{A}_c + \mathbf{B}_b + \mathbf{B}_c + \mathbf{C}_b + \mathbf{C}_c. \tag{4.17}$$

This makes sense geometrically, as the vertical edge  $E_{1,5}$  would have to go through each of the  $b$  and  $c$  crossing variables, as well as the top partition variable  $V^T$  and the bottom partition variable  $V_1^B$ . We further utilize  $S^+ = \frac{A}{2} + \frac{|A|}{2}$  and notice that, for  $\mathbf{D}$  equal to any of  $\mathbf{A}$ ,  $\mathbf{B}$ , or  $\mathbf{C}$ ,

$$\mathbf{D}_b + \mathbf{D}_c = (L - N)^+ + (N - L)^+ = |L - N|.$$

Combining this with the proper values for  $L$  and  $N$  from the three parts of Fig. 4.12, we have

$$E_{1,5} = V_1^B + V_5^T + |V_{2,3}^T - V_{2,3}^B| + |V_{2,4}^T - V_{2,4}^B| + |V_{3,4}^T - V_{3,4}^B| \quad (4.18)$$

Finally, we use the  $E^T \rightarrow V^T$  relationships from Eqs. 4.12 and 4.13 to get the above Eq.4.18 in terms of edges only:

$$\begin{aligned} E_{1,5} = & \frac{1}{2} \min[(E_{1,2} - E_{2,3} + E_{3,1}), (E_{1,3} - E_{3,4} + E_{4,1}), (E_{1,2} - E_{2,4} + E_{4,1})] \\ & + \frac{1}{2} \min[(E_{5,2} - E_{2,3} + E_{3,5}), (E_{5,3} - E_{3,4} + E_{4,5}), (E_{5,2} - E_{2,4} + E_{4,5})] \\ & + \frac{1}{2} \text{abs}[(E_{4,2} - E_{2,3} + E_{3,5} - E_{5,4})^+ - (E_{4,2} - E_{2,3} + E_{3,1} - E_{1,4})^+] \quad (4.19) \\ & + \frac{1}{2} \text{abs}[(E_{3,4} - E_{4,2} + E_{2,5} - E_{5,3})^+ - (E_{3,4} - E_{4,2} + E_{2,1} - E_{1,3})^+] \\ & + \frac{1}{2} \text{abs}[(E_{2,3} - E_{3,4} + E_{4,5} - E_{5,2})^+ - (E_{2,3} - E_{3,4} + E_{4,1} - E_{1,2})^+]. \end{aligned}$$

Here, the min function is an expanded version of  $V_i = \min(F_j V_i, F_k V_i, F_m V_i)$  using Eq. 4.9.

### 4.3 Kinetic Delaunay Triangulation Maintenance

Let  $S$  be a set of points in  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . The Delaunay triangulation (DT) of  $S$  is the partitioning of  $S$  into  $n$ -dimensional triangles such that the  $n$ -dimensional ball circumscribed around each triangles'  $(n + 1)$  points contains no other points in the interior. Maintaining a DT of  $S$  among moving points is ideal for implementing our ensemble-based topological entropy calculation in three dimensions. This is allowed since the Dual E-tec re-triangulation is no longer forced to evolve according to point-edge or point-face. Figure 4.13a (right) and b depict re-triangulation updates in 2D and 3D that are applied at the moment the Delaunay criterion is broken. Topologically, these individual update events and subsequent local triangulations are the same as updating at collision times.

Though maintaining a global DT of moving points in  $S$  will generally require more updating of the triangulation structure and computational time, there is one

major advantage in doing so: edge-edge collisions in 3D are not possible. Two edges crossing each other in 3D require the four edge endpoints to be coplanar, resulting in a circumsphere with infinite diameter. This is an obvious and direct violation of the Delaunay criterion, one which will not occur if the proper DT maintenance updates are recorded. Figure 4.13c provides further proof of concept: as the hollow point 5 moves up and to the right, the circumsphere around points 1, 2, 3, and 5 translates and changes diameter. Before edge (1, 5) collides with edge (2, 3), the circumsphere consumes point 4 and a Dual E-tec update rule is applied.

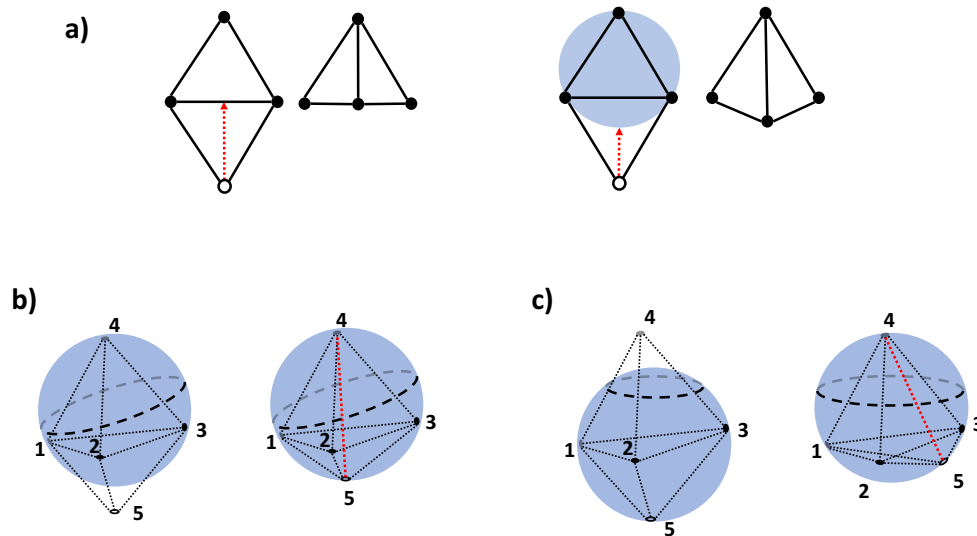


Figure 4.13: **New Local Triangulation Updates:** In all sub-figures, the hollow point moves north or northeast. a) Instead of waiting for the 2D point-edge collision to occur before implementing the local re-triangulation rules, depicted on the left, I apply the same local update on the right at an earlier time, specifically when the Delaunay criterion is broken, i.e. once the blue circumcircle contains the fourth point on its boundary. b) In blue, the circumsphere for points 1, 2, 3, and 4. On the right, once point 5 breaks the Delaunay criterion by moving onto the circumsphere, the tetrahedra are re-triangulated with the addition of red edge 4-5. Note that the two tetrahedra (1, 2, 3, 4) and (1, 2, 3, 5) are replaced by three that all share the newly introduced edge. c) In blue, the circumsphere for points 1, 2, 3, and 5. As point 5 moves northeast, the circumsphere dilates and moves and the Delaunay criterion is broken as point 4 enters. The tetrahedra are re-triangulated in the same manner as b).

Delaunay triangulations have been well-studied and their usage in both surface modeling [109–111] and simulating real-world processes [110,112–114] is robust. There exist many varied algorithms for adding/deleting vertices to/from an existing triangulation [115–118] and their higher-dimensional computations [119–126]. These algorithms involve methods based on convex hulls [117,127], dividing and conquering [128], and sweep lines [129].

The focus in this chapter is on the more difficult task of maintaining a Delaunay triangulation as points move along trajectories in two and three dimensions. Primitive algorithms accomplish this in 2D [130,131] by moving all points and finding all potential times the Delaunay criterion breaks, sorting these events in a time-ordered priority queue [132,133], updating  $DT(S)$  at each event, deleting or modifying later events, and continuing until no events exist in the queue. The event times are found by looking at every pair of adjacent triangles, translating the circumcircle equation into a polynomial, and solving for the zeros of this polynomial.

The same algorithm has theoretically been extended to 3D [134–136], however, it is much less efficient in dimensions higher than three. The polynomial form of the 2D circumcircle equation is of degree three and can be solved quickly and analytically to easily retrieve the event times. However, this polynomial in 3D is of degree eight and iterative methods must be used to approximate the roots [137,138]. Implementation [139,140] of this algorithm exists in the Computation Geometry Algorithms Library, or *CGAL*, a standard opensource C++ library.

While efforts have been made in finding the fastest times for numerically solving the eighth-order polynomial, this computational bottleneck is bypassed in a newer algorithm that seeks to maintain a DT while moving only individual points. First proposed for two dimensions by Gold [141,142] in 1990, further developed by Mostafavi [112,113] a decade later, then extended to three dimensions by Ledoux [109] in 2008, the algorithm describes a method for maintaining a DT by moving individual points forward in time while recording the times at which the Delaunay criterion is violated. Two major advantages emerge:

- By parameterizing the single trajectory of  $p$ , denoted  $\vec{p}$ , and substituting into the circumcircle and circumsphere equations, violation times are found by solving

a much simpler degree 2 polynomial.

- Delaunay criterion only needs to be checked for local triangles and tetrahedra adjacent to  $p$ , a stark contrast to the global check among all pairs of adjacent simplices in earlier algorithms.

While the method from Ledoux records Delaunay violation times among all moving  $p_i$  while holding all other  $p_{k \neq i}$  static. With all  $p_i$  moved independently, the violation times are queued in a time-ordered list of Delaunay criterion violations. Each violation time is addressed in chronological order and at each one the points involved in the violation are evolved forward, neighboring appropriate updates are made to the triangulation structure and

I implement a simpler algorithm which does not evolve many points together, instead moving only a single point  $p$  at a time through the entirety of its time step. However, my implementation remains efficient; after each new update, the number of triangles and tetrahedra requiring Delaunay criterion checks scales as  $\mathcal{O}(\log n)$ , the same as Ledoux’s method with multiple points moving at the same time [109].

### 4.3.1 Flips

First among definitions and operations needed to maintain a kinetic DT is a flip [118, 143]. This topological operation modifies the configuration of local triangles and tetrahedra. Among four points in 2D, there are only two possible triangulations. Figure 4.14a gives an example of a *flip22*, used by many of the algorithms listed above to transition a triangulation to the only other configuration possible. The ’22’ in the name comes from the fact both before and after the flip, there are two triangles.

In 3D, there are exactly two ways to triangulate five 3-dimensional points: either with two tetrahedra or with three tetrahedra [116, 118]. Illustrated in Fig. 4.14b are the two flips used to transition between the two different configurations. A *flip23* transitions the two tetrahedra sharing the triangular face (1, 2, 3) into the configuration of three tetrahedra sharing the common edge (4, 5). A *flip32* is the inverse operation, transitioning the three tetrahedra configuration into the one with two tetrahedra.

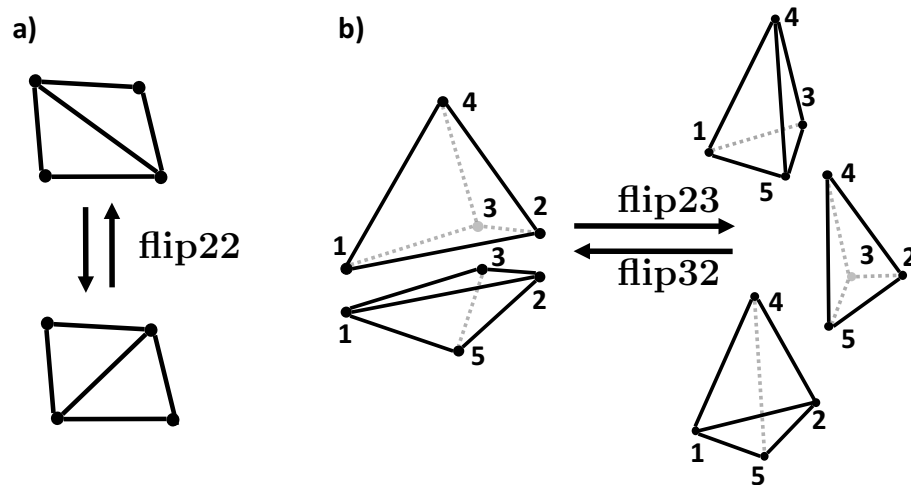


Figure 4.14: **Flips:** In a), a flip22 is used to transition among the two valid triangulation configurations of four points in 2D. In b), either a flip23 (left to right) or flip32 (right to left) is used in 3D depending if the triangulation configuration of five points contains two or three tetrahedra before the flip.

Convexity is of importance when checking if adjacent tetrahedra can be flipped. For example, in Fig. 4.14 b), a flip23 is only possible on the two adjacent tetrahedra (1, 2, 3, 4) and (1, 2, 3, 5) if and only if the union of these two tetrahedra forms a convex polyhedron. This is easily checked by verifying that edge (4, 5) passes through the shared triangular face (1, 2, 3). (See Fig. 4.17a below for an illustration of the required convexity checks.)

### 4.3.2 Stars, Links, and Ears

Next, we introduce three concepts related to triangulations, defined by Ledoux [109], all to be used for detecting when the Delaunay criterion fails in 2D and 3D. I define these concepts generally for  $d$ -dimensional space and use two and three dimensional figures to illustrate concepts. For all definitions, I denote  $p$  be a vertex in a  $d$ -dimensional triangulation.

#### Star

The star of  $p$ , denoted  $\text{star}(p)$  and shown in Fig. 4.15, consist of all the  $k$ -simplices



( $k = d, d - 1, \dots, 2, 1$ ) that contain  $p$ . No 1-simplices (vertices) other than  $p$  are in  $\text{star}(p)$ . For example, in 2D,  $\text{star}(p)$  contains all of the triangles and edges connected to  $p$ , while the  $\text{star}(p)$  contains all tetrahedra, triangles, and edges connected to  $p$ .

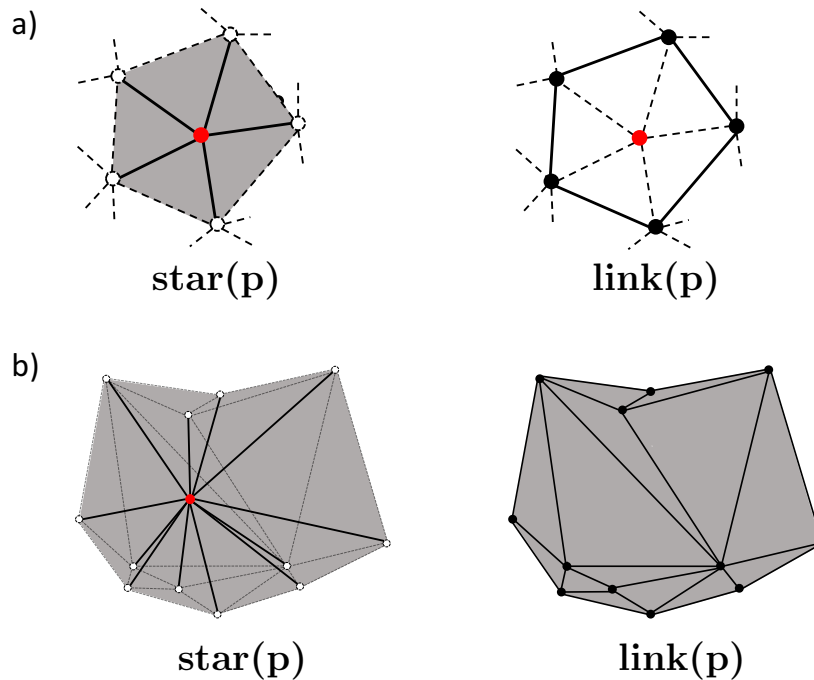


Figure 4.15: **Stars and Link:** The stars and links for red point  $p$ . (Left) The 2D star in a) contains all lines and triangles connected to  $p$ . The 3D star in b) contains all lines, triangular faces, and tetrahedra connected to  $p$ . (Right) The 2D link of  $p$  in a) contains all adjacent points and the edges connecting them, while the 3D link in b) contains all adjacent points and their connecting edges in addition to the outer triangular faces.

### Link

The link of  $p$ , denoted  $\text{link}(p)$ , consists of all the simplices incident to the simplices forming  $\text{star}(p)$ , but left out by  $\text{star}(p)$ . In other words,  $\text{link}(p)$  is the  $(d - 1)$  dimensional triangulation that forms the boundary of  $\text{star}(p)$ . Note that this boundary triangulation forms a polygon in 2D and a polyhedron in 3D, though it is not necessarily the convex hull of  $\text{star}(p)$ , as evidenced Fig. 4.15b.

## Ear

In 3D, an ear of  $\text{link}(p)$  is a potential tetrahedron that could be used in flip23 or flip32 event. These tetrahedra are referred to as imaginary because they do not exist yet in the triangulation before the flipping operation. As shown in Fig. 4.16, ears can be constructed from four points in  $\text{link}(p)$  that span either two or three triangular faces. Two faces sharing a single edge are denoted 2-ears, while three faces sharing a single, common vertex are denoted 3-ears.

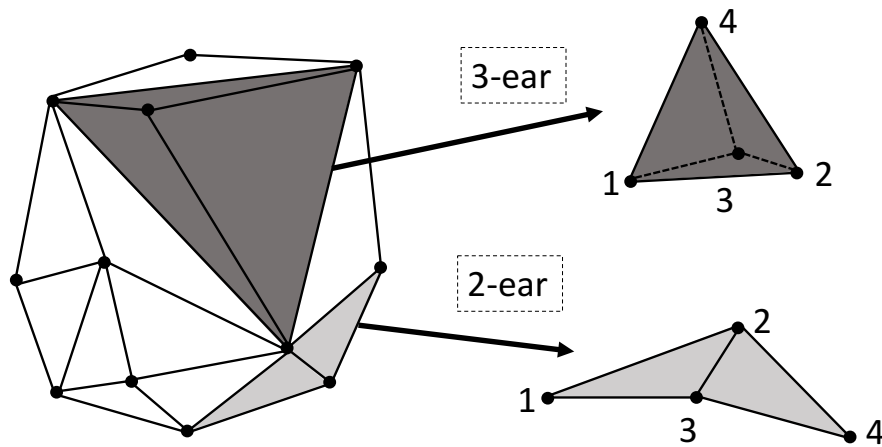


Figure 4.16: **2-Ears and 3-Ears:** In three dimensions, a link for some vertex viewed from outside. Three adjacent faces in the link sharing a common vertex for a 3-ear in dark gray. In light gray, two adjacent faces sharing only one common edge form a 2-ear.

While many ears are found in  $\text{link}(p)$ , not every ear is a potential tetrahedron for a flip23 or flip32 operation. Some adjacent faces of 2-ears and 3-ears are outside of star  $(p)$ . A simple test for convexity is done to identify invalid ears, shown in Fig. 4.17.

### 4.3.3 Real and Imaginary Events

Moving point  $p$  is evolved step-by-step to the closest time the Delaunay criterion is broken, re-triangulating each step of the way. Along the trajectory of  $p$ , a breakdown can occur among  $p$  and circumcircle/circumsphere one of two different types of triangles/tetrahedra: *real* and *imaginary*. We define these two types below, using

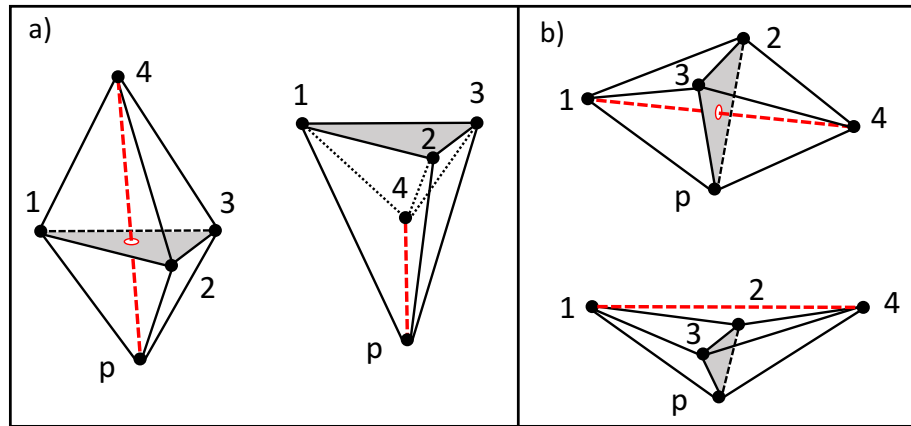


Figure 4.17: **3-ear and 2-ear Convexity Tests:** Point  $p$  is shown with the two types checks. a) Faces  $(1, 2, 3)$ ,  $(1, 3, 4)$ , and  $(2, 3, 4)$  in  $\text{link}(p)$  form a 3-ear of  $p$ . This 3-ear is valid (left) when the line connecting  $p$  and 4 passes through shaded triangle and invalid (right) if not. b) Faces  $(1, 2, 3)$  and  $(2, 3, 4)$  in  $\text{link}(p)$  form a 2-ear of  $p$ . This 2-ear is valid (top) if the line connecting 1 and 4 passes through the shaded triangle and invalid (bottom) otherwise.

the term 'event' to indicate which type of triangle or tetrahedron is involved in the breakdown.

### Real Event:

A real event involves a real triangle or tetrahedron  $\tau_i$ , those that are incident to  $\text{link}(p)$ , but outside  $\text{star}(p)$ . Figure 4.18a depicts in gray the circumcircles of  $\tau_i$  associated with red moving point  $p$ . If  $p$  moves into a gray circumcircle of a real triangle, a real event is triggered since the 2D Delaunay criterion is violated.

### Imaginary Event:

An imaginary event involves the ears  $\sigma_i$  of  $\text{star}(p)$ . These are labeled imaginary because they do not exist in triangulation before the event's subsequent re-triangulation. Re-triangulation becomes necessary once  $p$  leaves the circumcircle/circumsphere of an imaginary triangle/tetrahedron. The concept of ears is simpler in 2D. Figure 4.18b shows one of the five ears associated with  $p$  and its circumcircle in blue. Note that there is a single different ear for every point on  $\text{link}(p)$ . (For the more complicated

2-ears and 3-ears defined for 3D, see Fig 4.16.)

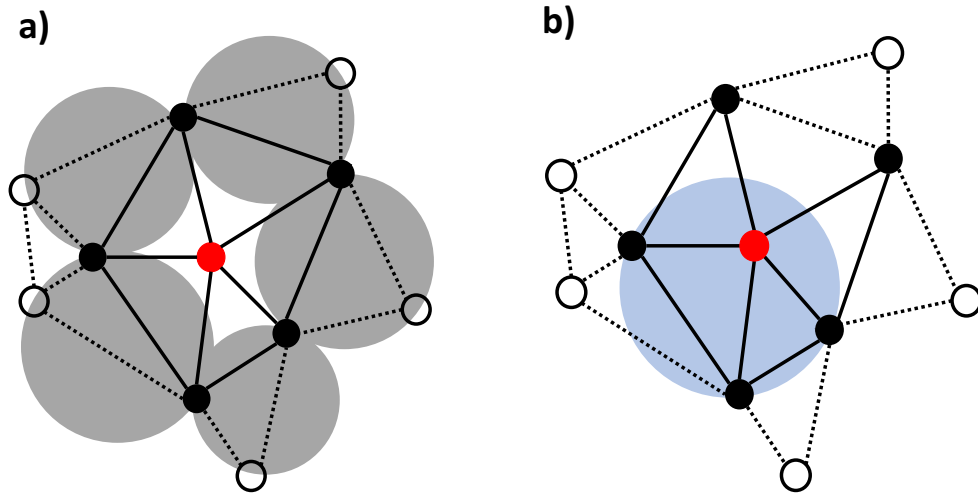


Figure 4.18: **Real and Imaginary Events in 2D:** Moving point  $p$  is in red. The link of  $p$  consists of points in black and the edges connecting them. The events necessary for the maintenance of a Delaunay Triangulation are detailed. a) Each gray circle represents the circumcircle through a real triangle of  $p$ . A real triangle is comprised of an edge on the link and the single hollow point outside of the link that this edge is attached to. A real event occurs when the red point  $p$  enters a gray circumcircle. A flip22 must then occur between  $p$  and the real triangle to maintain a Delaunay triangulation. b) The blue circle represents the circumcircle through an imaginary triangle consisting of three adjacent points on the link of  $p$ . An imaginary event is triggered once  $p$  leaves a blue circumcircle, resulting in a flip22 between  $p$  and the imaginary triangle.

#### 4.3.4 3D Delaunay Maintenance Algorithm Details

For the remainder of this section, I detail my modification to Ledoux's [109] method for Delaunay maintenance of  $S$ . For each time step in the trajectory data, each moving point  $p_i \in S$  is moved to the end of its time step while all other  $p_{k \neq i} \in S$  are held stationary. The idea is to move each  $p_i$  individually to each topological event and correct the DT with appropriate flips along the way, with the only constraint being that the DT of  $S$  must be known at the beginning of the trajectory of  $p_i$ . Once an event is detected,  $p_i$  is moved to the location of the event, the DT is corrected,

and  $\text{star}(p)$ ,  $\text{link}(p)$ , and associated ears  $\sigma_i$  are updated. A local search for the next event occurs along the remainder of the trajectory of  $p_i$ . Once no more events for  $p_i$  are found,  $p_i$  is moved to its final position in the time step, concluding a single point's movement. A cartoon depicting this process (with flip22 operators) in 2D is shown in Fig. 4.19.

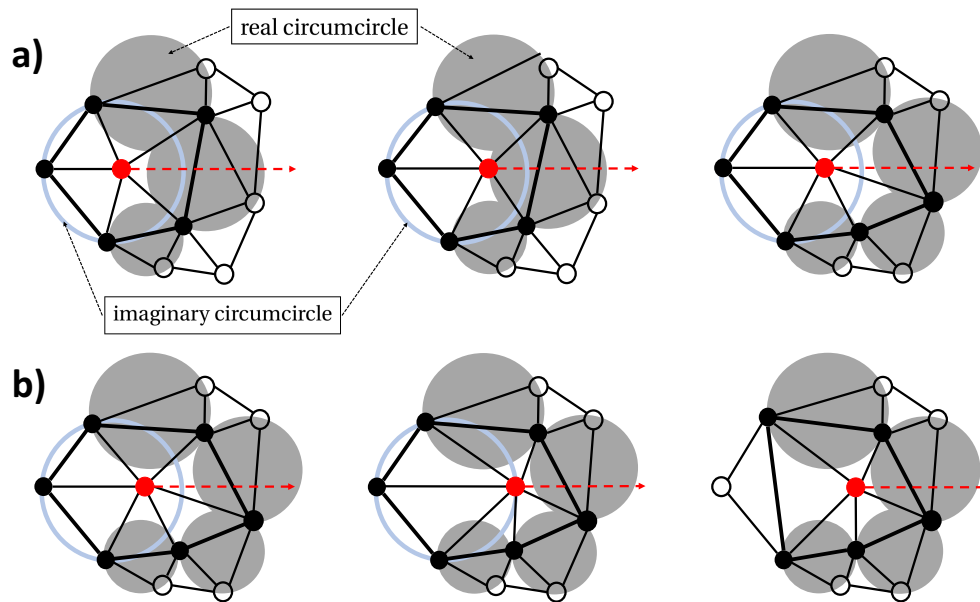


Figure 4.19: **Moving a Single Point:** Real and imaginary events are detected and processed as point  $p$  (in red) moves along its trajectory. Circumcircles of real and imaginary triangles are shown in gray and blue. (For simplicity, only one imaginary triangle is shown.) a) The first event detected is a real event.  $p$  is moved to the intersection (middle). The triangulation is updated with a local flip22 (right) and two new real triangles emerge. Not pictured is the modification of the imaginary triangles. b) The next event detected is an imaginary event.  $p$  is moved to the intersection (middle). The triangulation is updated with another flip22 (right).

Below are the details of the algorithm moving a single point to along its full trajectory in 3D. I make note of any changes or unnecessary steps needed for the same 2D implementation.

**Input:** The following inputs are required by the algorithm:

1. Beginning and ending position of the moving point  $p$ .

2. Locations of all other stationary points.
3. List of tetrahedra making up DT, denoted  $T$ .

**Output:** E-tec tracks the evolution of the band, as we will describe below, and outputs:

1. The DT at the moving point's final position and corresponding updated tetrahedra list  $T$ .
2. The time-ordered list of events required to maintain the DT. Event details include:
  - (a) event times along the trajectory (parameterized from 0 to 1),
  - (b) the real or imaginary tetrahedra,
  - (c) the flips being performed,
  - (d) the edges that are added/deleted from the triangulation as a result of a flip23/flip32. (Or, the edges added and deleted in a flip 22 in 2D.)

**Data structures:** The following data structures are maintained as a function of time:

1.  $T$ , the ( $n$  by 4) list of  $n$  tetrahedra in the triangulation.
2. For each different moving point  $p$ :
  - (a)  $\text{star}(p)$ , which is the subset of  $T$  containing  $p$ ,
  - (b) the faces in  $\text{link}(p)$ , found from  $\text{star}(p)$ . (Or, the edges in  $\text{link}(p)$  in 2D.)

**Steps:** Here, the key steps taken to move  $p$  and correct the DT are outlined.

1. The parametric equation for the trajectory of  $p$ , denoted  $\vec{p}$ , is used.  $\vec{p}$  is at initial position at time  $t = 0$  and final position at  $t = 1$ .
2.  $\text{Star}(p)$  and  $\text{link}(p)$  are extracted from  $T$ .
3. All real tetrahedra  $\tau_i$  of  $p$  are found from the faces in  $\text{link}(p)$ . Invalid  $\tau_i$  lying inside  $\text{star}(p)$  are disregarded.
4. All imaginary tetrahedra  $\sigma_i$  of  $p$  are found from the ears of  $\text{link}(p)$ . Invalid  $\sigma_i$  are thrown out according to the convexity test in Fig. 4.17.
5.  $\vec{p}$  is substituted into the equation of each circumsphere of  $\tau_i$  and  $\sigma_i$ . Each substitution yields a quadratic equation that has no real solution if there is no intersection, one solution if  $\vec{p}$  is tangent to the sphere, and two solutions if  $\vec{p}$  intersects the sphere.

- (a) Solutions satisfying  $0 < t < 1$  indicate that  $\vec{p}$  intersects a circumsphere along its trajectory. All other real solutions are thrown away, as these intersections occur either before the initial position of  $\vec{p}$  ( $t < 0$ ) or after the final position ( $t > 1$ ).
- 6. If no real solutions satisfying  $0 < t < 1$  are found,  $p$  is moved to its final position and the algorithm is complete. Otherwise, proceed to steps 7 - 8.
- 7. The tetrahedron having closest circumsphere intersection with  $\vec{p}$  is extracted, indicated by the smallest solution satisfying  $0 < t < 1$ .  $p$  is advanced along its trajectory to the point of intersection.
  - (a) If the tetrahedron is real ( $\tau$ ), the adjacent tetrahedron in  $\text{star}(p)$  is extracted. The appropriate flip and update to  $T$  depends on the convexity of these two.
    - i. A flip23 is performed if the two tetrahedra are convex. The convex configuration of and the check for convexity are illustrated in Fig. 4.20a.
    - ii. A flip32 is performed if the two tetrahedra are not convex. Non-convexity indicates that a third tetrahedron adjacent to the two is present in the triangulation [134, 135]. This three tetrahedra, non-convex configuration and corresponding check are illustrated in Fig. 4.20b.
  - (b) If the tetrahedron is imaginary ( $\sigma$ ), the appropriate flip and update to  $T$  depends on whether  $\sigma$  is a 2-ear or 3-ear.
    - i. A flip23 is performed if  $\sigma$  is a 2-ear.
    - ii. A flip32 is performed if  $\sigma$  is a 3-ear.
- 8. With the position of  $p$  and the valid list of DT tetrahedron  $T$  updated, proceed back to step 1.

## 4.4 Topological Entropy Calculation

To compute topological entropy in two and three dimensions, the Delaunay maintenance in Sect. 4.3 is performed separately from the Dual E-tec weight updates (found in Sect. 4.2) applied to a rubber band or sheet. The advantage here is that once the

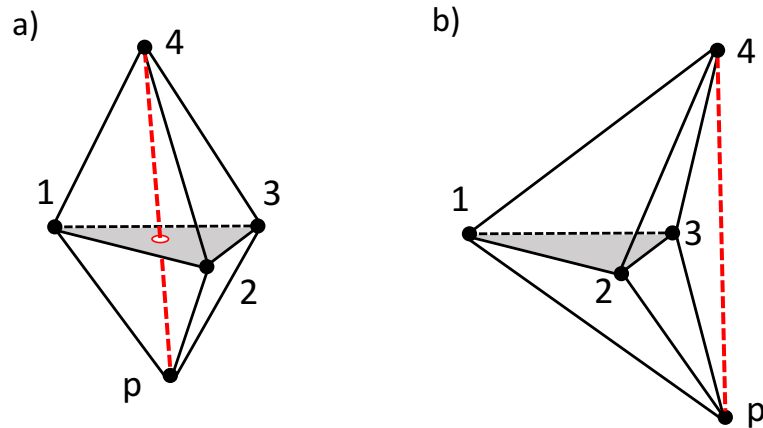


Figure 4.20: **Real Event Convexity Test:** A real event occurs between  $p$  and real tetrahedron  $(1, 2, 3, 4)$ . a) A flip 23 is performed if the line connecting  $p$  and 4 passes through the shaded triangular face of the other three points. b) If the line connecting  $p$  and 4 does not pass through the shaded triangular face, a third tetrahedron  $(p, 2, 3, 4)$  exists in the triangulation of the five points. A flip 32 is then performed.

motion of the points is captured by the DT update events and saved, this information can be applied independently to any number of rubber manifold configurations amongst the initial positions. An overview of the procedure is as follows:

1. Obtain unique list of Delaunay triangulation maintenance events (done once).
  - (a) Initialize the set of points  $S$  with a DT.
  - (b) Evolve  $S$  for the entirety of its motion and at each event, return the points involved, the type of flip, and the edge that is created or annihilated (or modified in 2D).
2. Evolve any rubber manifold configuration using list of events (can be done any number of times).
  - (a) Initialize a rubber band or sheet that runs through the initial DT of  $S$ . Bands anchored at initial edges of the DT are tested in 2D. In 3D, tested rubber manifolds include sheets wrapped around triangular faces and sheets cutting through entire planes of the initial DT.
  - (b) Dual E-tec weight update rules are applied to the edge weights in correspondence with each flip23 or flip32 required to maintain the DT.



- (c) Topological entropy  $h_{top}$  is found by fitting to the log-scaled sum of edge weights vs. time.

Though ensemble motion will be different for the DT-maintained cases, we find some success in verifying that the dynamics are being accurately captured. First, a 2D Dual E-tec topological entropy comparison of both the chaotic lid-driven cavity flow [3, 4, 68, 69] used in Ch. 2 and its DT-maintaining form is presented. The period-driving parameter  $\tau_f = 0.96$ , which yields three period-3 islands that force a true entropy of  $h_{po3} = 0.9624$ , is used for all testing. A convergence test and discussion of results follow. Once verified in 2D, two test cases are examined in 3D. These results excitingly represent the first topological entropy calculations in 3D from trajectory data alone.

#### 4.4.1 2D Dual E-tec Results Using Delaunay Triangulation Maintenance

While the Dual E-tec scheme is shown to be consistent with the original E-tec scheme for the same motion of points in Fig. 4.9, I now show consistency between Dual E-tec weight updates extracted from both the original motion of points and the DT-preserving motion described in Sect. 4.3. Figure 4.21 displays their applications to a number of initial bands. As a control, a band anchored between two distant points is initialized, the re-triangulation events corresponding to point-edge collisions in the original chaotic lid-driven cavity flow dynamics are recorded, and the band is subsequently evolved forward (in red). For the DT-preserving dynamics, events are recorded, then a band is placed around every single edge in the initial, unconstrained DT. The DT-preserving events then operate on each band (dashed). Band edge weights were updated at every flip<sup>22</sup>. The maximal stretching attained (in blue) over all initial bands is consistent with the stretching done by the point-edge collisions of the original dynamics.

Next, DT-maintained re-triangulation events are recorded for ensembles of increasing size for three times the length of time  $T$  as Fig. 4.21. For each ensemble, bands are placed around all edges in the initial triangulation and evolved using the events.

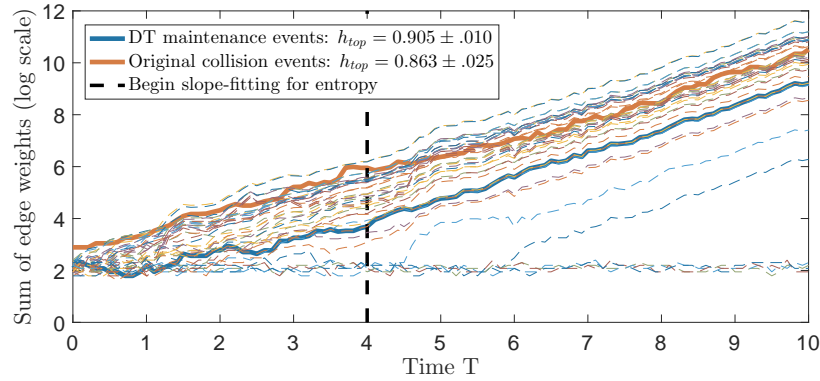


Figure 4.21: **Original Motion vs. Motion Maintaining a Delaunay Triangulation:** An ensemble of 12 trajectories from the chaotic lid-driven cavity flow is used. The true entropy is  $h_{po3} = 0.9624$ . Red line represents the stretching of a single band under the original flow evolution. All other lines represent stretching of single bands under the evolution of the flow maintaining a Delaunay triangulation. Bold blue line represents the initial band yielding the most stretching after the time  $T = 4$ .

The band undergoing maximal stretching at later times is the extracted topological entropy  $h_{top}$ . Convergence to the true, exact entropy is shown in Fig. 4.22.

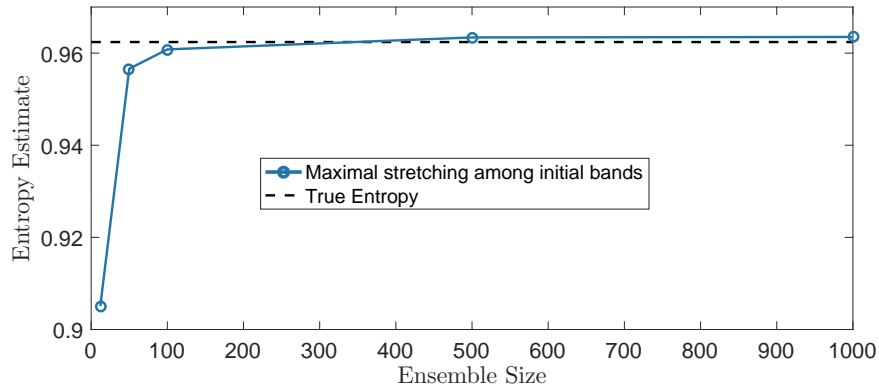


Figure 4.22: **Entropy Convergence:** Topological entropy  $h_{top}$  results using Delaunay triangulation-maintaining event operators as a function of ensemble size. Dashed line is the true entropy result for the original motion of this flow.

Interestingly, the motion maintaining a DT yields slightly higher topological entropy calculations than expected for larger ensemble sizes. This may not be too surprising of a result, as only single points are moved at a time, resulting in motion that is different between time steps from the original motion. Evidence of intrinsically

different motion is found in Fig. 4.23. The number of re-triangulation events occurring per time step increases with ensemble size at a faster rate than when trajectories evolve forward in their natural way. This increase in the number of events, as well as the subsequent increase in the number of edge modifications and flip22 operators, could be one reasons why the entropy is larger.

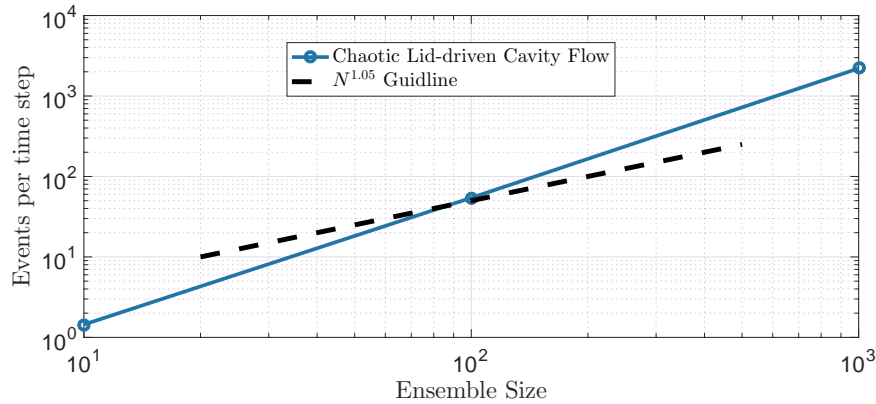


Figure 4.23: **Events per Period:** Average number of re-triangulation events per period required to maintain a Delaunay triangulation. The  $N^{1.05}$  guideline shows the event scaling for the original flow.

Generally, the DT-maintaining motion moving only a single point at a time may yield intrinsically different dynamics than the original motion. It is not the preferred method for evolving a triangulation forward in 2D, points may be re-triangulated differently based on the order in which trajectories are evolved forward in time. However, we keep in mind that the main advantages to this new method for re-triangulating reveal themselves in 3D.

#### 4.4.2 3D Dual E-tec Results Using Delaunay Triangulation Maintenance

Here is an appropriate time to remind the reader of the primary reason why a Delaunay triangulation is maintained as points evolve forward in time: edge-edge collisions will not occur in 3D under this constraint. The re-triangulation ambiguities that come with edge-edge events are bypassed, leaving only a series of flip23 and flip32

operators that modify the triangulation with either an addition or deletion of an edge.

To help verify Dual E-tec in 3D, the same chaotic lid-driven cavity flow used in the above 2D analysis is raised to three dimensions. The same flow parameter value that results in the exact topological entropy  $h_{po3} = 0.9624$  is used. Full trajectories are gathered in two dimensions from a uniform distribution of initial conditions. Only this time, each trajectory is raised to a uniformly random  $z$ -coordinate for the duration of its run. Trajectories are then contained in the rectangular box  $0 < x < 6$ ,  $-1 < y < 1$ , and  $0 < z < 1$ . Three sample trajectories can be seen in Fig. 4.24. The aim in choosing essentially the same flow as previous tests is that the 2D E-tec convergence in both the number and length of trajectories is well-understood. This gives a solid reference for which to compare the 3D results with.

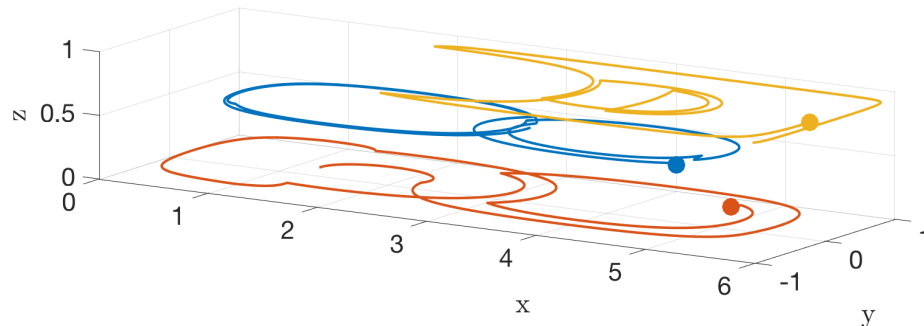


Figure 4.24: **3D Chaotic Lid-driven Cavity Flow:** Three trajectories belonging to the same chaotic flow from previous sections are used. Here, each individual point is uniformly distributed along  $0 < z < 1$  for the entirety of its trajectory. Projecting all trajectories down to the  $z = 0$  plane gives the original 2D dynamics.

Ensembles of size 20, 50, and 100 trajectories are advected forward fifty periods. (Note that any trajectories in the period-3 islands will return back to their starting positions after three periods, ensuring plenty of mixing in 2D.) Eight additional auxiliary points are positioned as boundaries outside the reach of any ensemble trajectory. Similar to the 2D algorithm, the stationary auxiliary points allow us to bypass any issues that arise when re-triangulating on the boundary. 50,280; 239,994; and 732,268 re-triangulation events are recorded for three ensembles run for fifty periods, each of

which include the flip operation and the edge to be added or subtracted. The total number of edges present in the triangulation for each ensemble stays consistent, as seen in Fig. 4.25, indicating the flip23 and flip32 operators are evenly distributed.

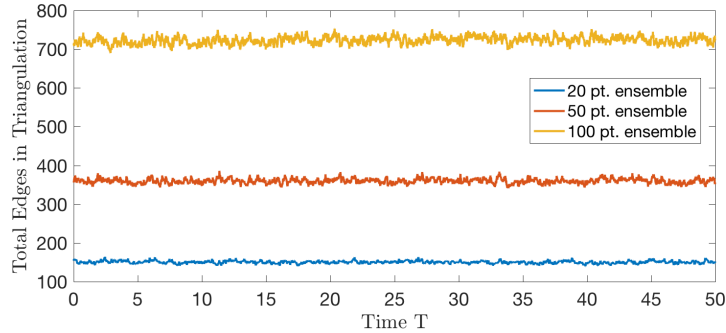


Figure 4.25: **Total Number of Edges in the Triangulation:** Total edges are plotted as a function of time. Increases or decreases to the edge totals result from the flip23 and flip 32 operations.

The list of events is recorded for each ensemble only once and can be used to evolve forward any number of rubber sheets. Sheets are encoded into the initial triangulation using non-zero edge weights, with the integer value of the weight corresponding to the number of times the edge passes through the sheet. While E-tec in 2D was typically initialized with bands anchored at two points, sheets anchored around the three points of a triangular face are troublesome for this model flow due to the  $\dot{z} = 0$  we've introduced by raising each trajectory to the  $z$ -axis. We find that sheets whose anchors do not simultaneously span much of the  $z$ -axis *and* cover a large area when projected down to two dimensions do not generally grow exponentially in time. Rather, this problem is alleviated by considering rubber sheets along planes can still be stretched under the advection of the flow. These planar sheets are not anchored by points, but instead can be thought of as being tied off at infinity. As illustrated in Fig. 4.26, these planar sheets are easily encoded into the triangulation.

Next, re-triangulation events for the three ensemble sizes are applied to varying initial sheet conditions. The simplest configuration is a single planar sheet through  $x = 3$ . However, this plane passes through one of the period-3 islands (now thought of as period-3 rods or cylinders when raised to three dimensions). Better entropy

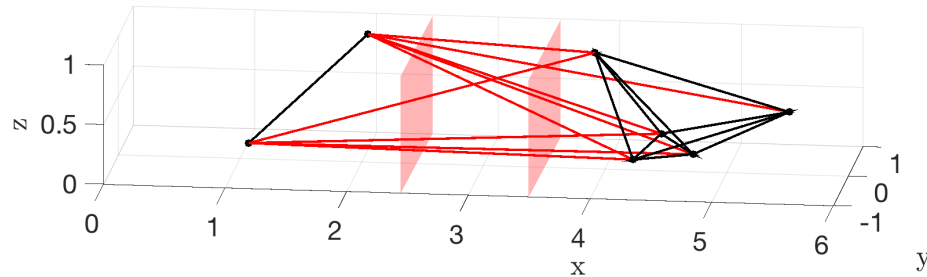


Figure 4.26: **Sheet Initialization:** An initial triangulation of points and the Dual E-tec edge weight representation of two rubber sheets along the planes  $x = 2.5$  and  $x = 3.5$ . Weighted edges of the triangulation are shown in red. Integer weight corresponds to the number of times an edge passes through a sheet.

results are found using two sheets at  $x = 2.5$  and  $x = 3.5$  between the rods. This configuration is depicted in Fig. 4.26. The logarithmic sum of edges weights and the resulting  $h_{top}$  estimation are revealed in Fig. 4.27a. To avoid initial transience,  $T = 5$  is chosen as the time to begin the slope-fitting. Though the results show exponential growth in the sum of edge weights over time and overall growth in  $h_{top}$  in the number of trajectories, the method is well short of the true topological entropy  $h_{po3} = 0.9624$ .

Initial sheets along the  $y = 0$  plane are next initialized and stretched by the same ensembles' re-triangulation events. Figure 4.27b shows a slower growth in the entropy compared to Fig. 4.27a when using the largest sized ensemble (100 trajectories, in yellow). This is possibly due to all three island rods intersecting with the plane, as the rods are pulling and dragging the sheet instead of intersecting and pushing it. However, this is simply the author's own hypothesis.

As a last calculation, one hundred sheets are evenly distributed along the planes through  $y = \alpha, \alpha \in (-1, 1)$ . This is done in hopes of sampling more space disconnected from the islands. (A similar calculation is done using  $x = \alpha, \alpha \in (0, 6)$  with results similar to Fig. 4.27a.) Initializing this setup is once again simple, as only the integer-valued weights along triangulation edges intersecting these sheets increases. Subsequent edge weight sums and resulting  $h_{top}$  estimations are show in Fig. 4.27c. The 100-trajectory ensemble entropy result is improved from the single  $y = 0$  case in

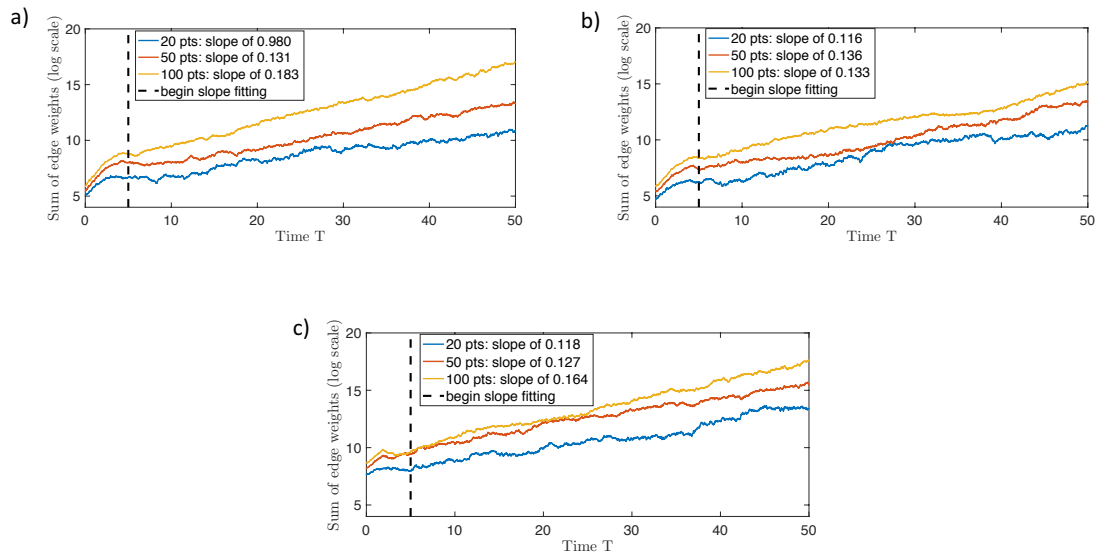


Figure 4.27: **Topological Entropy Results for Three Initial Sheet Configurations:** Three sets of initial sheets are each tested on the 20 pt., 50 pt., and 100 pt. ensembles: a) two planes at  $x = 2.5, 3.5$ , b) one plane through  $y = 0$ , and c) one hundred planes evenly spaced up and down the  $y$ -axis. Planes in a) are chosen such that no plane crosses through the period-2 islands (or rods, in 3D). Topological entropy is stated in inset boxes and is given by the slope of the best-fit line through the natural log of edge weight sums as a function of time. Slope and line fitting begins at  $T = 5$ .

Fig. 4.27b, though initial configuration of two planes through  $x = 2.5$  and  $x = 3.5$  result in the highest estimate of  $h_{top}$  (Fig. 4.27a).

One sign of validation in our entropy calculations is the fact that sum of edge weights is consistent over time once early transience is accounted for. To further verify the entropy calculations, it is shown that the exponential growth of edge weight sums is not simply a product of a growing number of edges transitioning from having zero weight. Figure 4.28 displays consistency between the ratio of weighted edges and all edges after  $T = 5$ , indicating that sheet growth is resulting from an exponential increase in the integer values of the weights. This figure not only justifies the beginning best-fit time of  $T = 5$ , but it also brings to light an interesting phenomenon regarding the weight proliferation at early times; the weighted edge ratio peaks near  $T = 2.5$ . This peak is left unexplained in this manuscript.

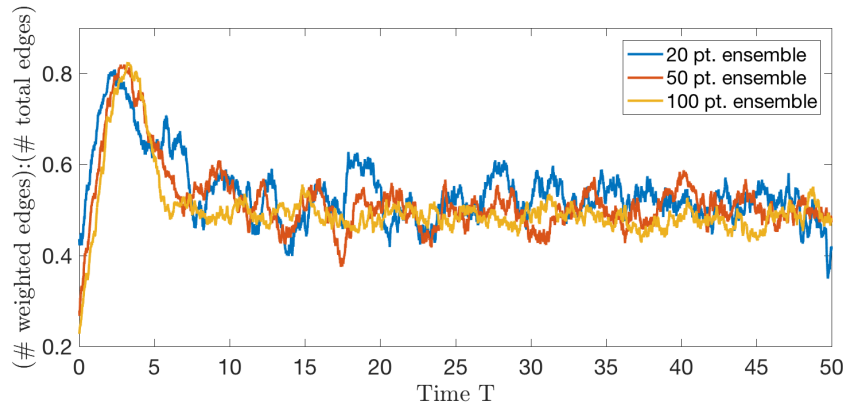


Figure 4.28: **Weighted Edge Ratios:** As a function of time, the ratio between the number of weighted edges to the number of total edges in the triangulation.

### 4.4.3 3D Dual E-tec Results Using Stirring Rods of the Chaotic Lid-driven Cavity Flow

As opposed to a random sampling of trajectories, E-tec is previously shown to converge more quickly to the correct value of  $h_{top}$  if trajectories are chosen from coherent sets that topologically force certain dynamics. In 2D, the lid-driven cavity flow used to study chaotic advection [3, 4, 68, 69] contains three period-3 islands that braid around one another, topologically forcing the flow to have a minimum entropy. In 3D, we extend these islands vertically in three dimensions as cylinders. Not only will these cylinders existing above the islands resemble stirring rods, the braiding of these rods should topologically force the same entropy lower bound of  $h_{po3}$  that the islands force in 2D. To sample from the stirring rods, trajectories existing in the period-3 islands are acquired and raised to a uniformly distributed  $z$ -coordinate value. (This the exact process performed in Sect. 4.4.2.) Figure 4.29 displays the stirring rods and the 2D island braiding motion superimposed over a freeze-frame of the trajectories used.

Twenty trajectories in each stirring rod are chosen and a Delaunay triangulation is maintained while the ensemble is evolved for  $T = 50$  periods. Though the ensemble is smaller in size than the largest test case above, the higher density of points results in an order of magnitude more re-triangulation events. The same three initial sheet



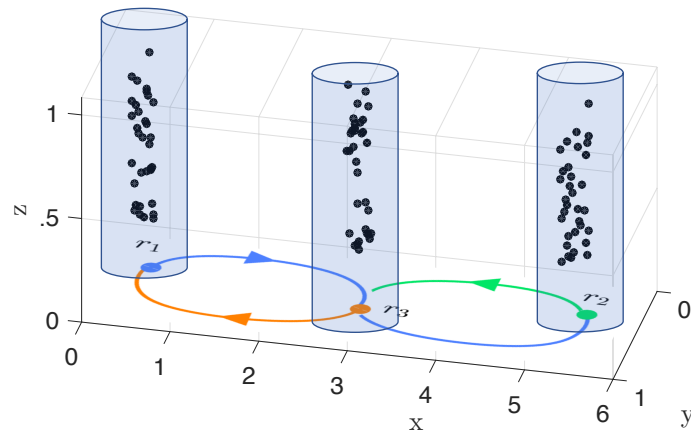


Figure 4.29: **Chaotic Lid-driven Cavity Flow in 3D:** Cartoon of the 2D period-3 island motion is displayed in the  $z = 0$  plane. Raising these islands in three dimensions results in three cylinders that resemble stirring rods. Trajectories in black are sampled from the islands and raised to some uniformly distributed  $z$ -coordinate.

configurations used in previous tests are used: two planes through  $x = 2.5$  and  $x = 3.5$  initially between the rods, one plane along  $y = 0$ , and one hundred planes evenly spaced along  $y = \alpha$ , where  $\alpha \in (-1, 1)$ . Unfortunately, the edge weight totals do not grow as fast as previous tests. Pictured in Fig. 4.30 (using the same  $y$ -axis scaling as the results in Fig. 4.27), no initial configuration yields exponential growth for any extended period of time.

#### 4.4.4 3D Dual E-tec's Loss of Information

The numerical results in Sects. 4.4.2 and 4.4.2 are underwhelming, as the best result yields a topological entropy lower bound of 0.183. This is only 19% of the known lower bound of  $h_{po3} = 0.9624$ . Comparatively, a small ensemble of only 10 points in 2D attained roughly 85% of the same lower bound. One may intuitively be led to believe that larger ensembles of points are needed to further test the convergence and robustness of 3D Dual E-tec. However, upon realization that an important consideration regarding the encoding of surfaces has been overlooked, we believe that updating surfaces using the Dual E-tec rules alone is insufficient and will not result in convergence to the true entropy.

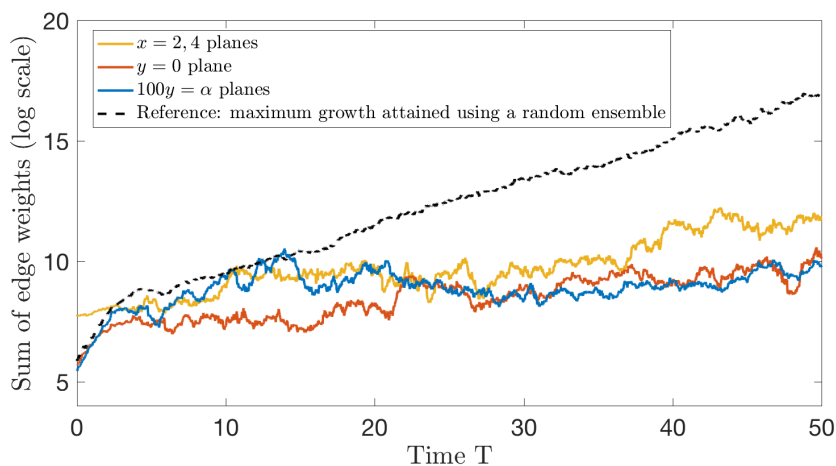


Figure 4.30: **Stirring Rod Entropy Results:** Logarithmic sum of edge weights as a function of time for the three initial sheet configurations. For reference, in black is the best result obtained using random ensembles.

The Dual E-tec rules are losing information about the stretching of manifolds, particularly when a flip32 is performed and the interior edge is lost. This results in not only the expected loss of edge weight described in subsections above, but in a more troublesome ambiguity in the type of surface being tracked. The most succinct example of this is drawn out in Fig. 4.31. Here, two surfaces are being recorded in the triangulation: a long cylindrical surface passing through two points and the familiar rubber sheets wrapped around the same two points. In Fig. 4.31a, the cylinder encases the interior edge shared by the three tetrahedra. Figure 4.31b shows the same cylinder after a flip32 update. Notice that the weight representation remains the same: 6 individual weights of one on each of the edges incident to the top and bottom points. Next, two rubber sheets are encoded in the original three tetrahedra configuration in Fig. 4.31c. Once a flip32 is performed in Fig. 4.31d, the weights have changed accordingly, though the weight representation of the two sheets in this two tetrahedra configuration is the same as the representation of the cylinder using the same two tetrahedra.

In short, information is lost when flipping from three tetrahedra to two and we

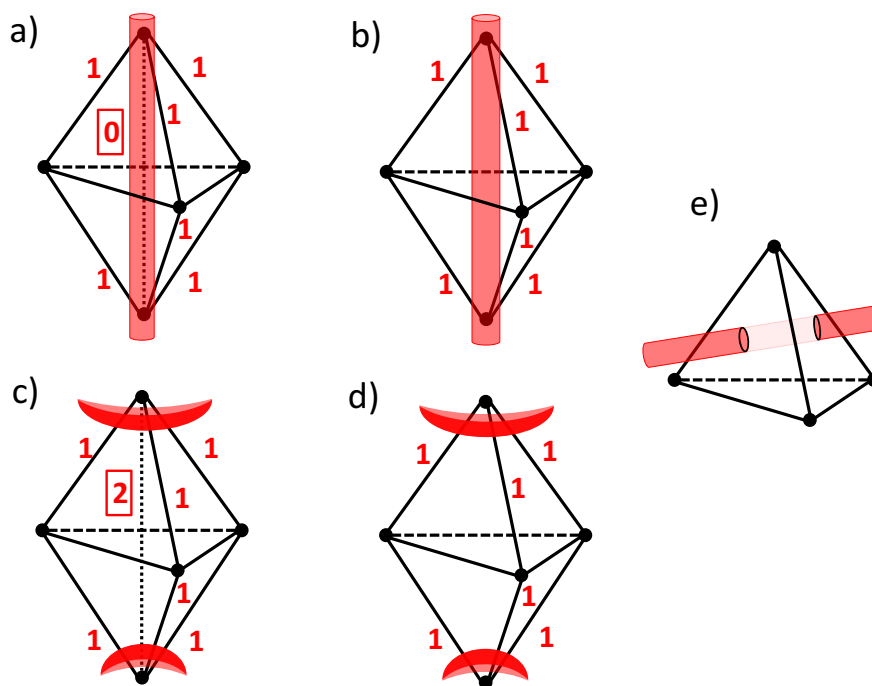


Figure 4.31: **Dual E-tec Loss of Information:** A cylindrical surface and corresponding Dual E-tec weight representations through a three-tetrahedra configuration in a) and after a flip<sub>32</sub> event in b). Likewise, two sheets passing through the same tetrahedra configurations in c) and d). While the two-tetrahedra configurations have differing weights in a) and c) (boxed for emphasis), all edges have the same weight after the flipping events in b) and d). In e) is a cylindrical surface passing through a tetrahedron while intersecting no edges. It has zero edge weight and thus cannot be encoded into the triangulation.

cannot distinguish between two different surfaces. This effectively means that our current method will “pinch off” any local bits of the manifold that look like this, resulting in disconnected sheets that don’t pick up as much growth. Worse yet, this opens up a whole other set of scenarios in which surfaces cannot be properly encoded. For instance, a cylindrical surface may enter a tetrahedron through one face and exit through another without crossing any edges. This scenario is depicted in Fig. 4.31e.

Since the cylindrical surfaces can be deformed into arbitrarily narrow tubes, we will refer to them as “strings” for the remainder of this manuscript. To help the Dual E-tec scheme properly keep track of the lost information, we aim to track the strings

through the faces in which they pass through. For instance, the string in Fig. 4.31b could be properly recorded with the addition of two weights assigned to the middle interior face in which it passes through, one weight for the top tetrahedra and one for the bottom. Likewise, the string in Fig. 4.31e would be easily recorded using two nonzero face weights for each of the two front faces it passes through.

Ideally, a coupling between the 2D sheet surfaces and the cylindrical surfaces could be developed to simultaneously encode two-dimensional rubber sheets and the string structures that arise. Such a scheme conceivably requires much work and is outside the scope of this manuscript. Instead, we take a simpler approach: motivated by the fact that one-dimensional edges of the triangulation are objects in 3D to the one-dimensional strings, we use the edges of a triangulation to stretch and fold strings as the triangulation evolves in time.

## 4.5 Stretching Strings in 3D Using E-tec

Here, we present a modified version of E-tec similar to the Dual E-tec scheme. Similar to the Dual method, once the flipping events necessary for preserving a Delaunay triangulation among moving points are recorded, this information can be applied independently to any number of initial string configurations. In this section, we outline our technique for encoding strings in a triangulation and define the edge weight update rules. These update rules will be applied to each flip23 and flip32 event in order to evolve the string in tandem with the moving points and edges of the triangulation.

### 4.5.1 Encoding the Strings

Since we use triangulation edges as stirrers, we want to encode one-dimensional rubber strings using new weights attached to each edge. We achieve this by assigning a weight to each edge corresponding to the number of times a piece of the string wraps around said edge. For an example, we follow Fig. 4.32. Here, edge (a, d) has two pieces of the string wrapped around it and is thus assigned a weight of 2. Likewise, edge (c,

d) is assigned a weight of 1 for the single piece of string it anchors. Another way of interpreting these weights is that they define how pieces of string puncture and wind around adjacent triangular faces. In the case of edge (d, c), the string punctures faces (b, c, d) and (a, c, d). We simply choose the common edge these two faces share as the edge to assign the appropriate weight.

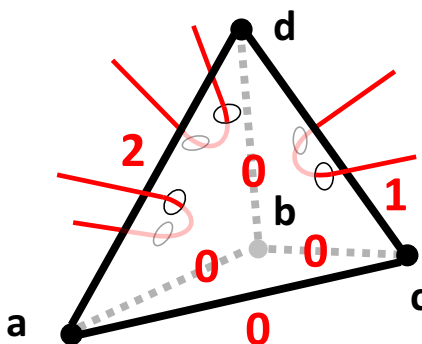


Figure 4.32: **Encoding Strings in a Single Tetrahedron:** Pieces of a rubber string (in red) are wrapped around edges (a, d) and (c, d). The string is represented with nonzero edge weights of 2 and 1 corresponding to the number of times the string wraps around each edge. All other edges have zero weight.

In total, each single tetrahedron has a total of six weights, one for each edge, to describe how the string winds through it. To encode the string in the many neighboring tetrahedra in a 3D triangulation, each *individual* tetrahedron will carry six weights. Adjacent tetrahedra must necessarily have the same number of string pieces puncturing the shared face, though this will be verified in the edge weight update rule derivation below.

We next take steps to simplify the update rules for the string edge weights. These rules will be applied at each application of a flip23 or flip32 event. For a flip23 transformation from two to three tetrahedra, the rules take in as input the 12 edge weights attached to the two tetrahedra configuration and outputs the 18 edge weights consistent with the three tetrahedra configuration and new edge. Likewise, a flip32 update transitions from the 18 edge weights to 12 as three tetrahedra transition to two.

To help visualize the string encoding and to help simplify the weight update rules, we use the topological diagrams of neighboring two and three tetrahedra configurations in Fig. 4.33. In the diagrams, tetrahedra are represented using full black points. Each black point is connected to four open circles representing the four faces of the tetrahedron. There are six edge weights attached to the top tetrahedron in Fig. 4.33a. They are 1-2, 1-3, 2-3, 1- $T$ , 2- $T$ , and 3- $T$ , corresponding to the shared edge between the listed faces. (And similarly for the bottom tetrahedron.) A nonzero weight along edge 1-3 for example indicates a string piece wraps around the edge shared by faces 1 and 3. To further show the ease with which strings may be represented, assume a more complicated scenario: a single piece of string punctures face 1 of the top tetrahedron, passes through the middle face  $T$ , and exits the structure through face 1' on the bottom. This is easily represented using two weights of 1 assigned to edges 1- $T$  and 1'- $T$ .

String representation is more complicated for the three tetrahedra configuration in Fig. 4.33b. For  $k = 1, 2, 3$ , one  $k$ - $k'$  edge is found on each of the three tetrahedra while each interior  $u_k$ - $u_{k+1}$  edge is shared among two of the three tetrahedra. For example, a string entering the structure through face 1 and exiting through face 3 can be represented by weights along 1- $u_2$  and 3- $u_2$  if it does not wind around the interior edge, by weights along 1- $u_3$ ,  $u_3$ - $u_1$ , and 3- $u_1$  if it winds around the interior edge once, or by 1- $u_3$ , 3- $u_1$ , and many  $u_3$ - $u_1$ ,  $u_1$ - $u_2$ , and  $u_2$ - $u_3$  if it winds around the interior edge more than once.

We define below the rules for updating the edge weights representing the string. For each flipping event, edge weights need to be shifted around the new configurations and faces in a way that is consistent with how the actual string would interact with the new or deleted edge. Of note, we assume a minimal topological complexity in the new configuration that is compatible with the weights before the flip. This minimal complexity assures we arrive at a lower bound on the growth of the string, yielding a lower bound on the topological entropy.

For the update rules, we use the following notation:

- $k$  and  $k'$  are used for the integer-named faces, where  $k = 1, 2$ , or  $3$ . Each set of  $k$  and  $k'$  belong to one of the two tetrahedra.

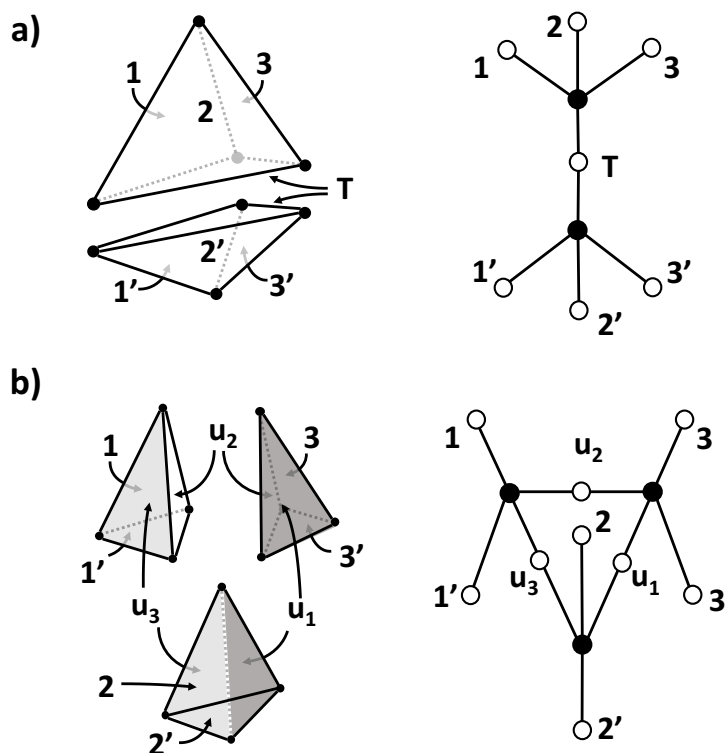


Figure 4.33: **Topological Diagram of Neighboring Tetrahedra:** a) On the left is the two tetrahedra configuration with labeled faces and shared face  $T$ . On the right is the simplified topological diagram of this configuration. Here, open holes represent the faces and the two full points represent the two tetrahedra. b) On the left is the three tetrahedra configuration that share a single edge running vertical. Exterior faces have integer names and are the same as a). The interior faces adjacent to the shared edge are labeled  $u_1, u_2$ , and  $u_3$ . (Faces  $u_2$  and  $u_3$  are colored for clarity.)

- $T$  is still used for the shared face in the two tetrahedra configuration,
- $u_k$ ,  $k = 1, 2, 3$ , is used for the shared interior faces of the three tetrahedra configuration,
- to express edges, we define  $k_m = k - 1$  and  $k_p = k + 1$ , defined cyclically.

The last bullet is particularly useful, as the event updates loop through  $k = 1, 2, 3$  and the current value of  $k$  acts as a pivot. For  $k = 1$ , we have  $k_m = 3$  and  $k_p = 2$ .

Similarly, for  $k = 3$ ,  $k_m = 2$  and  $k_p = 1$ .

### 4.5.2 Flip23 Event Update

We begin with the more straightforward of the two updates. The addition of the interior edge does not add much complexity, as strings connecting outer faces now simply may pass through one of the new interior  $u_k$  faces. The reader is encouraged to view Fig. 4.34. This figure will be referenced throughout the steps below.

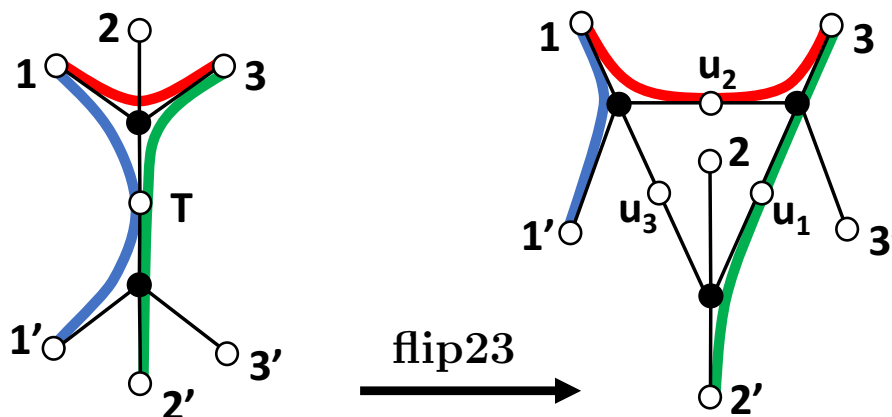


Figure 4.34: **Flip23 Update Rule Example:** On the left are three colored pieces of string entwined in the two tetrahedra configuration. A flip23 occurs and the string pieces are mapped to the new configuration. The update rules moves pieces in the following order: red piece is mapped from 1-3 to 1- $u_2$  and 3- $u_2$ , the blue piece is mapped from 1- $T$  and 1'- $T$  to 1-1', and the green piece is mapped from 3- $T$  and 2'- $T$  to 3- $u_1$  and 2' -  $u_1$ .

**Input:** The flip23 update requires 12 edge weights as input:

1. Six edge weights from the top tetrahedron,  $k-k_p$  and  $k-T$ .
2. Six edge weights from the bottom tetrahedron,  $k'-k'_p$  and  $k'-T$ .

**Output:** The flip23 gives 18 edge weights as output:

1. Three  $k-k_p$  weights for the lateral sides of the three tetrahedra structure.
2. Six  $k-u_{k_m}$  and  $k-u_{k_p}$  weights connecting the outer  $k$  faces with the interior  $u_k$  faces.



3. Six  $k'-u_{k_m}$  and  $k'-u_{k_p}$  weights.
4. Three  $u_k-u_{k_p}$  weights connecting the interior faces.

**Steps:** The steps are given below

1. For  $k = 1, 2, 3$ , string pieces passing through neighboring faces and not puncturing  $T$  will now pass through one of the shared faces  $u_k$ . (Follow the red string piece in Fig. 4.34.) :

$$k-u_{k_p} = k_m-u_{k_p} = k-k_m,$$

$$k'-u_{k'_p} = k'_m-u_{k'_p} = k'-k'_m.$$

2. For  $k = 1, 2, 3$ , string pieces puncturing faces  $k$  and  $k'$  and passing through face  $T$  are pushed to  $k-k'$ . (The blue string in Fig. 4.34.) :

$$m = \min(k-T, k'-T),$$

$$k-k' = m,$$

$$k-T \rightarrow k-T - m,$$

$$k'-T \rightarrow k'-T - m.$$

3. The only string pieces remaining are those passing through  $T$  and puncturing  $k$  and  $k'_m$ . These are pushed to  $k-k'_m$  and forced to pass through an interior  $u_k$  face. (The green string in Fig. 4.34.) :

$$m_1 = \min(k-T, k'_p-T), \quad m_2 = \min(k-T, k'_m-T),$$

$$k-u_{k_p} \rightarrow k-u_{k_p} + m_2,$$

$$k-u_{k_m} \rightarrow k-u_{k_m} + m_1,$$

$$k'_m-u_{k_p} \rightarrow k'_m-u_{k_p} + m_2,$$

$$k'_p-u_{k_m} \rightarrow k'_p-u_{k_m} + m_1,$$

### 4.5.3 Flip32 Event Update

Next we define the more complicated flip32 update in which an edge is removed and the points transition to two tetrahedra. In the previous Dual E-tec scheme, much information of the surface structures was lost when an edge was removed. This is not the case here with the strings; the string can still be uniquely defined by the edge weights upon removal of an edge. When the internal edge is indeed removed, the string will simply return taut around the nine triangulation edges on the exterior of the structure.

Two cases arise in the flip32 update and depend whether or not any of the three interior edge weights  $u_k-u_{k_p}$  are nonzero. If all are zero, the update is similar to the flip23 case above, with string piece puncturing faces  $k$  and  $k_p$  mapping to  $k-k_p$  and pieces puncturing  $k$  and  $k'$  (or  $k$  and  $k'_p$ ) being forced through new face  $T$ . If any  $u_k-u_{k_p}$  are nonzero, this means string pieces are wrapped around the interior edge. The update rule takes care in peeling off these pieces from the interior edge and combining them with appropriate edge weights to preserve the string structure. Once again, Fig. 4.35 illustrates the steps and the placement of string pieces.

**Input:** The flip32 update requires 18 edge weights as input:

1. Three  $k-k_p$  weights for the lateral sides of the three tetrahedra structure.
2. Six  $k-u_{k_m}$  and  $k-u_{k_p}$  weights connecting the outer  $k$  faces with the interior  $u_k$  faces.
3. Six  $k'-u_{k_m}$  and  $k'-u_{k_p}$  weights.
4. Three  $u_k-u_{k_p}$  weights connecting the interior faces.

**Output:** The flip32 gives 12 edge weights as output:

1. Six edge weights from the top tetrahedron,  $k-k_p$  and  $k-T$ .
2. Six edge weights from the bottom tetrahedron,  $k'-k'_p$  and  $k'-T$ .

**Steps for Case I:** If all  $u_k-u_{k_p}$  are zero, no piece of the string returns taut once the edge is removed. The steps below give a formula for transitioning to this configuration. (Case I is visualized in Fig. 4.35b to c. One simply needs to track the mapping of each colored string.)

1. For  $k = 1, 2, 3$ , weights for neighboring  $k$  and  $k'$  faces are forced through face

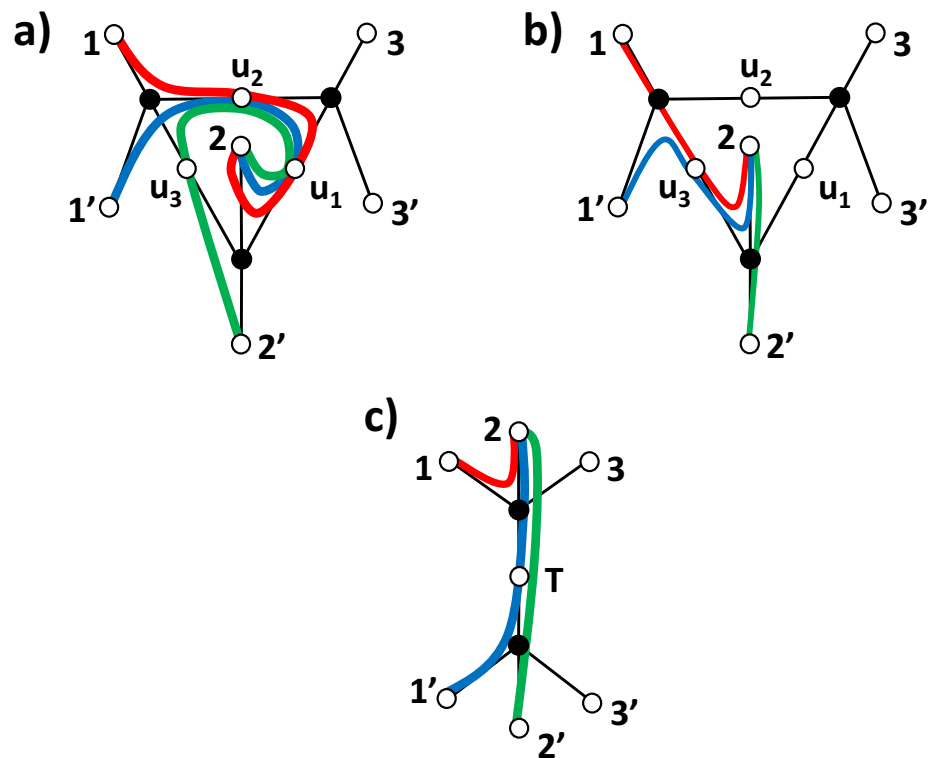


Figure 4.35: **Flip32 Update Rule Example**: a) Three colored pieces of string entwined in the three tetrahedra configuration. b) The update rule moves pieces in the following order: the red piece is mapped to  $1-u_3$  and  $2-u_3$ , the blue piece is mapped to  $1'-u_3$  and  $2-u_3$ , and the green piece is mapped to  $2-2'$ . c) The transition to the two tetrahedra configuration occurs. The red piece is mapped to  $1-2$ , the blue piece is mapped to  $1'-T$  and  $2-T$ , and finally the green piece is mapped to  $2'-T$  and  $2-T$ .

$T$ . (Follow the red string piece in Fig. 4.35b and c.)

$$k-T = k-k',$$

$$k'-T = k-k',$$

$$k-k' \rightarrow 0.$$

2. For  $k = 1, 2, 3$ , string pieces puncturing either  $k$  and  $k_m$  are moved to  $k-k_m$ . (The blue string in Fig. 4.35b and c.)

$$m = \min(k-u_{k_p}, k_m-u_{k_p}),$$

$$k-k_m \rightarrow k-k_m + m,$$

$$k-u_{k_p} \rightarrow k-u_{k_p} - m,$$

$$k_m-u_{k_p} \rightarrow k_m-u_{k_p} - m.$$

Repeat this step with primed faces swapped for unprimed faces.

3. The only string pieces remaining are those puncturing  $k$  and  $k'_m$  that will pass through  $T$  in the new configuration. (The green string in Fig. 4.35b and c.) For  $k = 1, 2, 3$ :

$$m = \min(k-u_{k_p}, k'_m-u_{k_p}),$$

$$k-T \rightarrow k-T + m,$$

$$k'_m-T \rightarrow k'_m-T + m,$$

$$k-u_{k_p} \rightarrow k-u_{k_p} - m_1,$$

$$k'_m-u_{k_p} \rightarrow k'_m-u_{k_p} - m_1.$$

Repeat this step with primed faces swapped for unprimed faces.

**Steps for Case II:** The steps are given below if any of  $u_k-u_{k_p}$  are nonzero. The process is visualized in the transition from Fig. 4.35a to b.

1. Use  $u_k-u_{k_p}$  to convert as many  $k-u_{k_p}$  and  $k_p-u_k$  segments into  $k-k_p$  segments. (The red string transition from Fig. 4.35a to b.) For  $k = 1, 2, 3$ :

$$m_1 = \min(k-u_{k_p}, k_p-u_k) \quad m_2 = \min(k'-u_{k_p}, k-u_{k_p}).$$

(a) If  $u_k-u_{k_p} < m_1 + m_2$ :

- The strings in  $u_k-u_{k_p}$  will need to be apportioned between the top and bottom tetrahedra. This is not uniquely specified, though a logical way to do it is proportionately. Therefore, define  $a_1$  and  $a_2$  by

$$\begin{aligned}
a_1 &= \frac{m_1}{m_1 + m_2} u_k - u_{k_p}, \\
a_2 &= \frac{m_2}{m_1 + m_2} u_k - u_{k_p},
\end{aligned} \tag{4.20}$$

where one of  $a_1$  or  $a_2$  is rounded *up* to the nearest integer and the other is rounded *down*.

- We update using  $m_1 = a_1$  and  $m_2 = a_2$ .
- If  $u_k - u_{k_p} \geq m_1 + m_2$ ,  $m_1$  and  $m_2$  remain unchanged.

Once this condition is accounted for, we have:

$$\begin{aligned}
k - u_{k_m} &\rightarrow k - u_{k_m} + m_1, \\
k' - u_{k_m} &\rightarrow k' - u_{k_m} + m_2, \\
k_p - u_{k_m} &\rightarrow k_p - u_{k_m} + m_1, \\
k'_p - u_{k_m} &\rightarrow k'_p - u_{k_m} + m_2, \\
k - u_{k_p} &\rightarrow k - u_{k_p} - m_1, \\
k' - u_{k_p} &\rightarrow k' - u_{k_p} - m_2, \\
k_p - u_k &\rightarrow k_p - u_k - m_1, \\
k'_p - u_k &\rightarrow k'_p - u_k - m_2, \\
u_k - u_{k_p} &\rightarrow u_k - u_{k_p} - (m_1 + m_2).
\end{aligned}$$

This step is repeated on loop until  $u_k - u_{k_p}$  remains unchanged after applying this step to the three values of  $k$ .

2. This step is similar to the above step, only this time swapping any connecting primed integer faces with unprimed ones. We use  $u_k - u_{k_p}$  to convert as many  $k - u_{k_p}$  and  $k'_p - u_k$  segments into  $k - k'_p$  segments. (The blue string transition from Fig. 4.35a to b.) For  $k = 1, 2, 3$ :

$$m_1 = \min(k-u_{k_p}, k'_p-u_k) \quad m_2 = \min(k'-u_{k_p}, k_p-u_k).$$

- (a) If  $u_k-u_{k_p} < m_1+m_2$ , once again proportion the strings by applying Eq. 4.20, rounding accordingly, and updating  $m_1$  and  $m_2$ .

Once this condition is accounted for, we have:

$$\begin{aligned} k-u_{k_m} &\rightarrow k-u_{k_m} + m_1, \\ k'-u_{k_m} &\rightarrow k'-u_{k_m} + m_2, \\ k'_p-u_k &\rightarrow k'_p-u_k + m_1, \\ k_p-u_k &\rightarrow k_p-u_k + m_2, \\ u_k-u_{k_p} &\rightarrow u_k-u_{k_p} - (m_1 + m_2), \\ k-u_{k_p} &\rightarrow k-u_{k_p} - m_1, \\ k'-u_{k_p} &\rightarrow k'-u_{k_p} - m_2, \\ k'_p-u_k &\rightarrow k'_p-u_k - m_1, \\ k_p-u_k &\rightarrow k_p-u_k - m_2. \end{aligned}$$

This step is again repeated on loop until all  $u_k-u_{k_p}$  remain unchanged after applying this step to the three values of  $k$ .

3. At this point, if there is a nonzero  $u_k-u_{k_p}$  left, the string punctures a face  $k$ , wraps around the middle edge, then leaves through the  $k'$  face below  $k$ . This means the weights  $k-u_{k_m}$ ,  $k-u_{k_p}$ ,  $k'-u_{k_m}$ , and  $k'-u_{k_m}$  must be zero except for a single value of  $k$ . Furthermore,  $u_k-u_{k_p} = u_k-u_{k_m}$ . (The green string transition from Fig. 4.35a to b.)

For this single value of  $k$  only:

$$m_1 = \min(k_p-u_k, k'_p-u_{k_m}) \quad m_2 = \min(k'_p-u_k, k_p-u_{k_m}).$$

- (a) If  $u_k-u_{k_p} < m_1 + m_2$ , once again apply Eq. 4.20, round  $a_1$  and  $a_2$ , and

update  $m_1$  and  $m_2$ .

After updating the  $m$  values, we have:

$$k_p - k'_p \rightarrow k_p - k'_p + (m_1 + m_2),$$

$$u_k - u_{k_p} \rightarrow u_k - u_{k_p} - (m_1 + m_2),$$

$$u_{k_p} - u_{k_m} \rightarrow u_{k_p} - u_{k_m} - (m_1 + m_2),$$

$$k_p - u_k \rightarrow k_p - u_k - m_1,$$

$$k'_p - u_k \rightarrow k'_p - u_k - m_2,$$

$$k'_p - u_{k_m} \rightarrow k'_p - u_{k_m} - m_1,$$

$$k_p - u_{k_m} \rightarrow k_p - u_{k_m} - m_2,$$

4. At this point, the only portion of string leftover is that which has not been pushed aside to the outer edges. That means if remaining nonzero  $u_k - u_{k_p}$  still exist, then a portion of the string enters and exits through the same face. Additionally, this piece of the string may wind around the middle edge any number of times. Rather than allowing the string to return taut along some other tetrahedron's edges, we take three steps in order to pin the string to the edge opposite the face it punctures.

- Redundant windings around the interior edge are eliminated by updating with:

$$m = \min(u_k - u_{k_p}, u_{k_p} - u_{k_m}, u_{k_m} - u_k),$$

$$u_k - u_{k_p} \rightarrow u_k - u_{k_p} - m \quad \text{for all } k.$$

- The string weight puncturing face  $k$  or  $k'$  twice is pinned to the opposite edge  $k_p - k_m$  or  $k'_p - k'_m$ . This amounts to shifting the weight to this opposite edge.
- The neighboring tetrahedra must be updated with the newly pinned edge

weight. The weight added in the step above is simply added to all tetrahedra sharing this edge.

5. Lastly, we apply the steps in case 1, as all interior weights are zero.

#### 4.5.4 String Stretching Results

We apply one-dimensional string stretching application of E-tec on the familiar chaotic lid-driven cavity flow used in Ch. 2 and raised to three dimensions in Sect. 4.4. For a proof of concept, six special trajectories are selected from the flow. We use a single pair of slightly perturbed trajectories in each of the three islands. Next, the island pairs are slightly perturbed in their  $z$ -coordinate value such that one lies above  $z = 0$  and the other below. The main advantage is that since the stirring rods (the islands raised to 3D) stay isolated, the edges between each pair of points never disappear from the triangulation. These long-lived edges make for the most efficient stirrers as there is no chance for the string to “diffuse” through the more densely-populated 3D rods with disappearing and reappearing edges.

The flip23 and flip32 events are recorded for the Delaunay triangulation maintenance of the evolving six points. Since the pairs of trajectories are placed slightly above and below the  $z$ -axis, we choose an initial string placed along the  $z = 0$  plane and between two of the raised islands. The exponential growth of the edge weights representing the string is displayed in Fig. 4.36. Fitting to  $T \in [30, 50]$ , these special trajectories yield an E-tec result equal to  $h_{top} = 0.9619 \pm 0.0012$ . This result is consistent with  $h_{po3} = 0.9624$ , the flow entropy guaranteed by the braiding of the island rods.

To generalize results, trajectory ensembles are randomly seeded and sampled from the entire space in which the flow lives and from within the island rods. The E-tec results from these two types of ensembles are shown in Fig. 4.37. The string growth obtained from 60 island rod trajectories, 20 in each island, attains only  $\approx 60\%$  of the entropy lower bound. This is in stark contrast to the result directly above in Fig. 4.36 which was obtained using one-tenth the number of trajectories. To make sense of this issue, we note that all though the small ensemble only had three triangulation edges



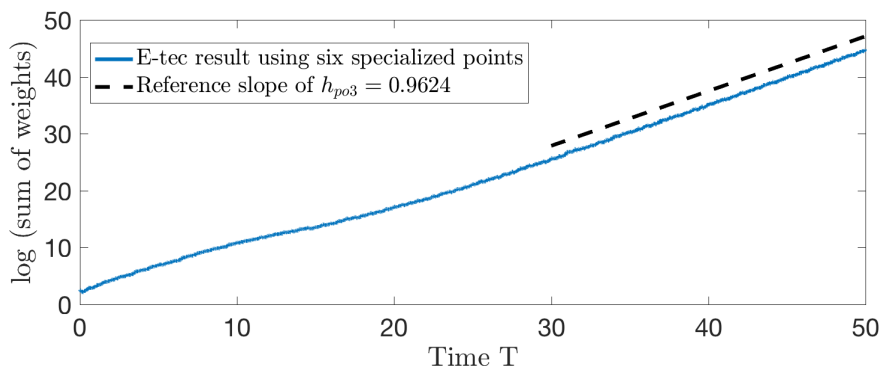


Figure 4.36: **Proof of Concept:** The growth of a rubber string stirred by the edges of an evolving triangulation of six trajectories specially curated from the chaotic lid-driven cavity flow coherent islands. For reference is a line with slope  $h_{po3} = 0.9624$ , the exact flow entropy ultimately produced by the braiding of the islands.

performing the stirring, these edges never disappeared after a flip32 event. Since the edges living in the rods all disappear at some time for the larger ensemble, we believe the band is “diffusing” through the rods.

The same diffusion issue may be responsible for the lack of growth in the larger random ensembles. Since these sets are nested (each ensemble is contained in all larger ones), we had reason to believe the growth rate would increase with ensemble size. This is not the case, as the 50 point ensemble surprisingly yields the best result and the largest ensemble yields the worst. This leads the authors to believe that higher densities of points result in many more flipping events and disappearing edges, which in turn gives the string more opportunity to diffuse through the tetrahedra structures.

## 4.6 Future Considerations

### 4.6.1 Coupling 2D Sheet Stretching and Cylindrical Surface Tracking

Both the sheet stretching and string stretching methods fail to converge to the true topological entropy as a function of ensemble size. We propose a coupling of these

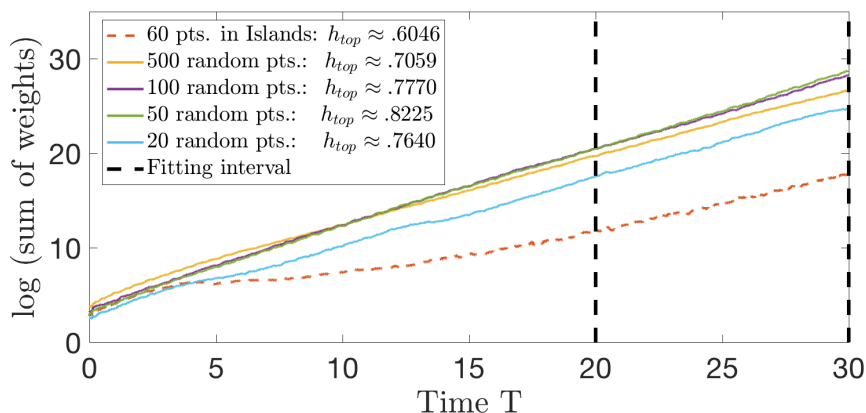


Figure 4.37: **String Stretching from Random Trajectories:** The dashed orange line shows string growth using 20 random trajectories in each of the three islands. the other results come from nested sets of trajectories that are uniformly and randomly distributed among the space in which the flow is contained.

methods as the next step to further E-tec along. The path forward could involve a foundation sheet stretching Dual E-tec rules, augmented with rules derived to encode strings in the triangulation. This is conceivably enough to keep track of the cylindrical pieces of manifold that can be topologically deformed into the long, narrow strings. To further simplify, an additional set of weights along each triangular face may be sufficient in tracking when the cylindrical strings pass through a face.

## 4.6.2 Parallelizing the Re-triangulation Process

There remains work to be done in verifying that E-tec will converge to the correct entropy in 3D. Certainly, it is expected that many more trajectories are needed to attain good results. The curse of dimensionality guarantees this; phase-space volume grows exponentially with dimension, resulting in exponentially more trajectories required to maintain sufficient densities. Future iterations of E-tec can certainly be optimized with respect to runtime in order to adequately handle larger ensembles.

We are motivated by the fact that Delaunay triangulation maintenance consumes a majority (99.9%) of E-tec runtime. However, by taking advantage of the locality of each individual re-triangulation event, we discuss a process in which re-triangulating

may be heavily parallelized. The idea is to group points together in disconnected groups, process these groups together in parallel, then process any remaining points. Figure 4.38 depicts this procedure in 2D. In each quadrant, the non-convex boundary of all edges contained in the quadrant is found and shaded. Points interior to these shaded regions are processed in parallel since all adjacent points and incident edges are contained in the closures of shaded regions. Any point that transitions to the exterior of a shaded region will be placed in one final grouping with points on the shaded region boundaries. This final grouping is not processed in parallel due to adjacent points residing in other quadrants.

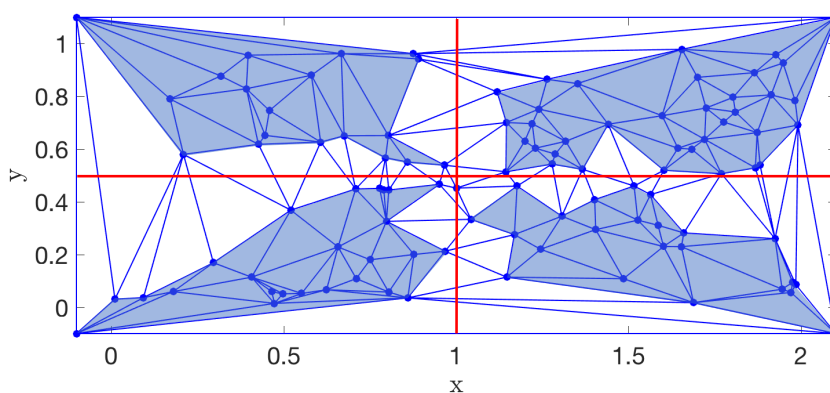


Figure 4.38: **Parallelizing Procedure:** A Delaunay triangulation in 2D (containing four auxiliary boundary points) is split up into four quadrants. In this depiction, four sets of trajectories interior to the four shaded regions will be processed in parallel. After, to complete the time step, remaining trajectories (those initially on the shaded region boundaries or those exiting their boundary regions) will be processed.

The choice of four regions/groupings for parallelizing above is chosen for convenience. The size and number of regions is arbitrary and variation in runtime performance will surely depend on choices for both. In 3D, this idea extends rather intuitively. Phase space will be partitioned by planes into smaller 3D regions. Points interior to faces contained in each region (the 3D analogy to the shaded regions) will be processed first in parallel, followed by all remaining trajectories on the boundary faces and that exit each region.

## 4.7 Conclusion

Presented in this chapter are three major advancements to the original 2D E-tec algorithm described in Ch. 2. First, the Dual E-tec scheme is developed as a different, more flexible way to encode a rubber manifold stretched across points. It is shown to be consistent with its counterpart in estimating a planar flow’s topological entropy. Most importantly, Dual E-tec allows the user to implement any re-triangulation scheme to compute topological entropy.

Secondly, implementation of a three dimensional E-tec algorithm is presented. This represents the first topological entropy calculation in 3D that requires only flow trajectories and no knowledge of governing equations. We demonstrate that individual 3D topological entropy estimations converge in the length of trajectories. We provide some evidence that estimations will grow as ensemble sizes are increased, though we detail a fundamental oversight in this technique: tube-like structures are not uniquely defined and crucial information about the manifolds is lost upon deletion of an edge.

Third, instead of evolving 2D sheets as they intersect with points, we stretch and fold 1D strings using triangulation edges as stirrers. We derive the edge weight update rules for the flipping events and provide a proof of concept using only six trajectories and long-lived triangulation edges. This method is demonstrated to converge to a topological entropy value in time, though convergence as a function of ensemble size is alluding the authors.

## 4.8 Acknowledgments

The author wishes to acknowledge support from the National Science Foundation under grant number #1808926 titled “Self-mixing Active Fluids.” The author also gratefully acknowledges Spencer A. Smith for insightful discussions and derivation of the Dual E-tec weight update rules in both 2D and 3D.

# Chapter 5

## Conclusions and Future Direction

### 5.1 Conclusion

Lagrangian-based approaches to fluid dynamics, those focused on how different particle trajectories explore a fluid and separate from each other, remain invaluable tools for understanding chaotic dynamics. These trajectory-based analyses continue to aid in identifying phase space structures that govern fluid mechanics. When trajectory data is limited or difficult to acquire, topological Lagrangian-based methods for quantifying fluid complexity need to be appreciated for their potential in accommodating sparse data. Thus, because higher-dimensional analysis requires a prohibitively large number of trajectories, there is a critical need for new topological approaches to Lagrangian-based methods.

This dissertation outlines my contributions to the field of chaotic fluid dynamics and topological entropy calculation. In particular, one question motivated this research throughout: how to best study three dimensional and higher chaotic flows when restricted to sparse trajectory data? Chapter 2, begins to address this question through the introduction and development of the Ensemble-based Topological Entropy Calculation, or E-tec, in two dimensions. The algorithm requires only a potentially-sparse set of system trajectories as input to stretch and fold material curves to compute a flow's topological entropy. It is shown to converge to the true

entropy in both the length and number of trajectories, and furthermore, of all techniques requiring only trajectory data, it is shown to attain the most favorable runtime scaling.

In Chapter 3, E-tec theory meets practice as it is applied to experimental data obtained from an active matter microbiological fluid. The system’s topological entropy is shown to bound experimentally-gathered Lyapunov exponents in what is the first study of chaotic mixing and active fluids. Next, E-tec considerations on unbounded flows are explored, followed by a proposed strategy for detecting coherent sets.

In Chapter 4, E-tec is scaled up to three dimensions. A new E-tec scheme for encoding rubber manifolds in a triangulation, is devised, one that allows for any re-triangulation scheme to be used. As a result, the first 3D topological entropy calculation requiring only system trajectories is made. E-tec convergence in the length of trajectories is shown and we find evidence of topological entropy converge in ensemble size.

E-tec represents a significant new tool for the fields of topological mixing and chaotic fluid dynamics, though many additional questions are raised. To conclude this manuscript, future considerations and strategies are addressed.

## 5.2 Future Work and Considerations

### 5.2.1 Three Dimensional Considerations

Necessary for showing E-tec convergence in 3D is adapting the algorithm to better handle large ensembles of trajectories. Chief among these runtime considerations is the parallelization technique outlined in Ch/ 4. Though promising, parallelization remains unimplemented and unexplored. Questions remain regarding how runtime will scale with the parallelizing process, particularly with the number of separate regions used. Additionally, one might consider if this technique can further be enhanced by optimizing the grouping of points; at the beginning of each time step, does the cost of calling a clustering algorithm to find nearby groups of points offset any potential gains in re-triangulation efficiency?

Another question raised in 3D is how calculations are affected by the choice of initial condition. The final results displayed in Fig. 4.27 vary greatly depending on the initial plane. Missing is an ergodic argument, to argue that all initial rubber manifolds will eventually start growing at the same rate, much like how we saw different initial band choices in 2D all grow at the same rate after initial transience. It remains an unanswered question if robustness to the choice of initial sheet can be obtained in 3D.

### 5.2.2 E-tec in Higher Dimensions

An important avenue in E-tec advancement regards its generalization to higher dimensions. Much like how a long, thin rubber band wrapped around 1-dimensional edges is used to capture stretching and folding in 2D, or how the entanglement of flat, rubber sheets stretched along 2-dimensional faces gives insight to a 3D flow's complexity, we wish to consider a  $(d - 1)$ -dimensional rubber manifold stretched around a collection of initial points in a  $d$ -dimensional phase space. As these points evolve in time, they carry the manifold along with them, stretching and folding it. However, difficulties arose in 3D that surely lead insight to potential difficulties in higher dimensions; much like how the edge-edge collisions proved difficult to handle, we envision ambiguities in re-triangulating after any collision that involves two simplices in which one is not a point (edge-edge, edge-face, etc) in higher dimensions.

With the advent of 2D Dual E-tec allowing the freedom to choose any re-triangulate procedure, two research avenue exist: 1) can the Dual E-tec weight updates be generalized to higher dimensions? and 2) does there exist a re-triangulation scheme in higher dimensions that eliminates the need to track collisions between simplices with dimension one or higher? Exploring an algorithm for maintaining a higher dimensional Delaunay re-triangulation seems like the most plausible place to start answering the second question, though an answer to the first remains unexplored. No work is completed for Dual E-tec update rules for edge-edge collisions or collisions involving a point and higher dimensional simplices.

# Appendix A

## Expansion Entropy

### A.0.1 Introduction

Despite the rich history in the field of nonlinear dynamics and chaos, the exact definition of the term chaos is still very much up for debate. While many proposed definitions of chaos hold up against the commonly held intuitive notions of orbit complexity and sensitive dependence on initial conditions, there exists not a single definition that allows for a straightforward numerical computation for verifying chaos that applies generally to *any* type of dynamical system. Disorder and unpredictability are landmark consequences of chaos in a dynamical system, so it makes sense that entropy, often thought of as the measure of increasing disorder in a physical system evolving through time [144], is an indispensable tool for defining the existence of chaos in a dynamical system. The existence of positive topological entropy, the measurement of the exponential growth rate of orbits restricted to an invariant set [49, 145], or positive metric entropy, the highly probabilistic approach of measuring the loss of information over time [49, 146], are suitable definitions under certain conditions or mappings, though these methods, along with many others, lack overall generality and many require knowledge of the velocity field [50].

We investigate a new term entitled expansion entropy which is a generality to the current leading entropy definitions. Expansion entropy measures the difference between the exponential growth rate of volume expansion in an arbitrary region  $S$



and the exponential rate at which volume escapes from  $S$ . This is a sampling method and is computed by contrasting the singular values of the evolving tangent map (or Jacobian matrix) of uniformly distributed initial conditions in  $S$  with the rate at which the trajectories leave  $S$ . The existence of a straightforward and "easy-to-implement" test for chaos in a general region could have a large impact in wide array of applications such as parameter control and identifying regions where chaotic mixing is present (note that  $S$  need *not* be invariant, a necessary criteria for the computation of topological entropy).

I will begin with an introduction and discussion of this alternative entropy definition in Sect. A.0.2, followed by a discussion on the numerical calculation adopted by Hunt and Ott in Sect. A.0.3. Next, I present two things of significance that arose from my work – first, the algorithm is applied to continuous flow for the first time in Sect. A.0.4, something not addressed in the paper, and secondly, I present my published result of an upper bound for topological entropy of a chaotic spherical vortex in Sect. A.0.5. My expansion entropy result helps verify the Homotopic Lobe Dynamics [2, 6, 7] calculation of topological entropy in Smith, et al. [147].

## A.0.2 Expansion Entropy Theory

We consider a positive volume set  $S$  (which need not be invariant) and a smooth dynamical system on a finite-dimensional manifold  $M$  defined by the mapping  $f_{(t',t)}(x) : M \rightarrow M$  satisfying  $x' = f_{(t',t)}(x)$ , where  $f$  maps  $x$  at time  $t$  to  $x'$  at time  $t'$ . To define the expansion entropy of the system, the notion of local volume expansion needs to be considered. Given an  $m$  by  $n$  matrix  $A$ , the singular values of  $A$  describe the amount of stretching done to a unit ball in orthogonal directions. Then, letting  $G(A)$  to be the product of singular values greater than one,  $G$  is applied to the derivative map  $Df_{(t',t)}$ . Upon application,  $G(Df_{(t',t)}(x))$  then defines the ratio of local volume growth of the map  $f_{(t',t)}$ . Finally, using the notation adopted by Hunt and Ott, denote  $\mu(S)$  to be the volume of the set  $S$  and  $S_{t't} \subset S$  to be the set of  $x \in S$  that stays in  $S$  for all time  $t'' \in [t, t']$ . Expansion entropy is defined as

$$H_0(f, S) = \lim_{t' \rightarrow \infty} \frac{\ln(E_{t',t}(f, S))}{t' - t}, \quad (\text{A.1})$$

where the

$$E_{t',t}(f, S) = \frac{1}{\mu(S)} \int_{S_{t',t}} G(Df_{(t',t)}(x)) d\mu(x), \quad (\text{A.2})$$

term quantifies the stretching.

### A.0.3 Numerical Computation

We begin with a continuous dynamical system  $\dot{\vec{x}} = f(\vec{x})$ . To numerically compute Eq. A.1, Eq. A.2 is expressed in a more numerical, discrete way. A large number of initial conditions  $x_i$  in a general region  $S$  is uniformly distributed. We then, for large  $T$ , evolve the trajectory  $f_{(T,0)}(x_i)$  of each initial condition  $x_i$  forward along with the corresponding tangent map  $Df_{(T,0)}(x_i)$ , disregarding any trajectory that does not remain in  $S$ . At discrete times  $T$ , I compute

$$\hat{E}_T(f, S) = \frac{1}{N} \sum_{i=1}^N G'(Df_{(T,0)}(x_i)) \quad (\text{A.3})$$

where the prime symbol denotes the computation of only those trajectories remaining in  $S$  for all times up to  $T$ . The estimated expansion entropy is the slope of  $\ln \hat{E}_T(f, S)$  plotted as a function of  $T$ . A positive slope, for large  $N$ , then indicates the existence of chaos in the general region  $S$

For continuous flows, the tangent map  $D$  and how it is integrated are derived. Given an  $n$ -dimensional ODE

$$\dot{\vec{x}} = f(\vec{x}, t), \quad (\text{A.4})$$

where  $\vec{x} \in \mathbb{R}^n$ , a flow solution  $F_T(\vec{v})$  is assumed, which is the solution set of Eq. (A.4) with initial condition  $\vec{v}$  after  $T$  units of time. In other words,  $F_T(\vec{v})$  is the point at which the orbit starting at initial condition  $\vec{v}$  arrives after time  $T$ . Plugging the flow solution into the original ODE yields

$$\frac{d}{dt}(F_T(\vec{v})) = f(F_T(\vec{v})), \quad (\text{A.5})$$

which, after differentiating with respect to the initial condition  $\vec{v}$ , the variational equation

$$\frac{d}{dt}(DF_t(\vec{v})) = Df(F_t(\vec{v})) \cdot DF_t(\vec{v}). \quad (\text{A.6})$$

Here,  $D$  represents the derivative taken with respect to the initial conditions. Lastly, defining  $J_t = DF_t(\vec{v})$  as the derivative of the orbit starting at  $\vec{v}$  at time  $t$  and  $A(t)$  as the matrix of partial derivatives of  $f(\vec{x}, t)$  with respect to  $\vec{x}$ , the variational equation [148] is alternatively expressed as

$$\dot{J}_t = A(t) \cdot J(t). \quad (\text{A.7})$$

Intuitively,  $\dot{J}_T$  measures the evolution of small perturbations of the initial condition  $\vec{v}$  at time  $T$ . We note that for the numerical computation of expansion entropy, the derivative mapping  $Df_{(t',t)}(\vec{x})$  is simply the  $\dot{J}_{t'-t}$  after  $t' - t$  units of time. But since the matrix  $A(T)$  typically depends on the position at time  $T$ ,  $J_T$  must be solved alongside the original ODE  $\dot{\vec{x}} = f(\vec{x}, t)$  and calculate the singular values of  $J_T$  at discrete times.

#### A.0.4 Expansion Entropy of the Chaotic Lorenz System

The analysis begins with verifying the topological entropy of a continuous flow. Continuous flows were not addressed in the original publication by Hunt and Ott [50]. Only discrete mappings were used which greatly simplified the computation of the derivative map  $Df$  to that of simple composition of Jacobian matrices.

The Lorenz System of ODEs is used here to verify the algorithm works for continuous flows. The three dimensional, nonlinear, deterministic system is given by

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= (\rho - z)x - y. \\ \dot{z} &= xy - \beta z \end{aligned} \quad (\text{A.8})$$

Fixing  $\beta = \frac{8}{3}$  and  $\sigma = 10$ , there exists three fixed points for  $\rho > 1$ , one at the origin and two at  $(x, y, z) = (\pm\sqrt{\beta(\rho - 1)}, \pm\sqrt{\beta(\rho - 1)}, \rho - 1)$ . For the value  $\rho = 28$ , the

origin is a saddle and the two other fixed points are attractors, resulting in a chaotic system [149].

In computing  $Df_T(\vec{x}) = J_T$ , some slight analytical work is in order.  $J_T$  must be solved for numerically alongside the three-dimensional Lorenz system. The matrix form of Eq. A.8 given by:

$$A(T) = \begin{bmatrix} -\sigma & \sigma & 0 \\ -(\rho - (z(T))) & -1 & -x(T) \\ y(T) & x(T) & -\beta \end{bmatrix}. \quad (\text{A.9})$$

Then, denoting

$$J_T = \begin{bmatrix} \dot{j}_{xx} & \dot{j}_{xy} & \dot{j}_{xz} \\ \dot{j}_{yx} & \dot{j}_{yy} & \dot{j}_{yz} \\ \dot{j}_{zx} & \dot{j}_{zy} & \dot{j}_{zz} \end{bmatrix},$$

we arrive at the Lorenz System's nine term variational equation

$$\dot{J}_T = \begin{bmatrix} -\sigma j_{xx} + \sigma j_{yx} & -\sigma j_{xy} + \sigma j_{yy} & -\sigma j_{xz} + \sigma j_{yz} \\ (\rho - z(T))j_{xx} - j_{yx} - x(T)j_{zx} & (\rho - z(T))j_{xy} - j_{yy} - x(T)j_{zy} & (\rho - z(T))j_{xz} - j_{yz} - x(T)j_{zz} \\ y(T)j_{xx} + x(T)j_{yx} - \beta j_{zx} & y(T)j_{xy} + x(T)j_{yy} - \beta j_{zy} & y(T)j_{xz} + x(T)j_{yz} - \beta j_{zz} \end{bmatrix}. \quad (\text{A.10})$$

containing twelve dependent equations. Integration of the flow  $F(\vec{x}, t)$  and coupled derivatives  $J_t$  is done simultaneously and reported at equally spaced times  $T$  using 'ODE45,' an adaptive, Runge-Kutta 4 solver in the MATLAB standard ODE library. Expansion entropy requires the calculation of the singular values of  $J_t$  at the reported times  $T$ .

Using the Taken's Method [150], which extracts topological entropy from the fractal dimension estimation of time series data, Newhouse and Pignataro previously estimated the topological entropy for our target parameters  $\sigma = 10, \rho = 28$ , and  $\beta = \frac{8}{3}$  to be between the values 0.8947 and 0.9075 [46]. In the expansion entropy paper, Hunt and Ott state that for an autonomous system on a compact manifold, topological entropy and expansion entropy are equal for  $C^\infty$  maps [150, 151]. Seeing as how the Lorenz System calculated in a finite restraining region fits these criteria, our numerical estimate of the expansion entropy should then approach Newhouse and

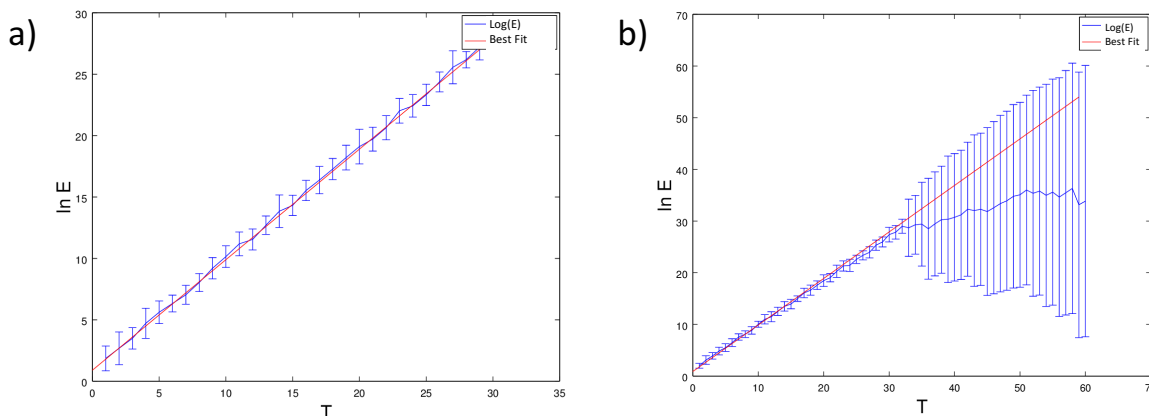


Figure A.1: Expansion entropy results for the chaotic Lorenz system. Plotted in blue is the computation of  $\ln \hat{E}_t$  versus  $T$  using 20 samples of  $N = 100$  uniformly distributed initial conditions. The expansion entropy estimate is the slope of the best-fit red line. a) Best-fit line slope is  $0.901 \pm .011$ . Here, the restraining region  $S$  is large enough such such that no trajectory leaves  $S$ . b) Best-fit line is fitted until time  $T = 30$ . The slope is  $0.884 \pm .020$ . A smaller restraining region  $S$  is used. Expelled trajectories are left out of the sum.

Pignataro's topological entropy estimation of  $h_t \in (0.8947, 0.9075)$  for a large number of initial conditions.

The expansion entropy of the chaotic Lorenz system is computed using two different regions  $S$ , one which is large enough region such that  $S$  is invariant (Fig. A.1 a), and a second region small enough such that trajectories escape and  $S$  shrinks (Fig. A.1 b). Using only  $N = 20000$  initial conditions parsed into 20 samples for both regions, the expansion entropy estimate is with error of the Newhouse calculation. Of course, using a larger  $N$  will lead to better statistics and a tighter bound. In their paper, Hunt and Ott use  $N$  on the order of  $10^6$  or  $10^7$ , though more points are necessary if trajectories escape from  $S$  rapidly.

### A.0.5 Published Result: Expansion Entropy as an Upper Bound for Topological Entropy

Lastly, I present an estimation of the expansion entropy of a modified version of Hill's spherical vortex [152, 153]. Hill's vortex is a well-known solution to Euler's equations for in inviscid incompressible fluid, represented by the Stokes stream function

$$\phi_0(r, \theta) = \begin{cases} (1/2) U(1 - a^3/r^3)r^2 \sin^2 \theta = \phi_{0+}(r, \theta) & (r > a) \\ -(3/4) U(1 - r^2/a^3)r^2 \sin^2 \theta = \phi_{0-}(r, \theta) & (r < a). \end{cases} \quad (\text{A.11})$$

To induce mixing, the Hill's vortex is modified by a sequence of time-periodic adjustments to the flow. We find a lower bound on the topological entropy of  $\ln(2.1106) = 0.7469$  by using the homotopic lobe dynamics technique [147], which uses the stretching and folding of stable and unstable manifolds to build a symbolic representation of the dynamics [7]. The "symbolic entropy" of the resulting tangle is extracted and acts as an estimation to the topological entropy. Due to the exponentially increasing complexity of the tangle, the above lower bound was found by taking only four iterates of the tangle. As the information given by the tangle increases with higher iterates, the "symbolic entropy" will converge from below on the full topological entropy.

To get a sense of how well the homotopic lobe dynamics technique is estimating the entropy the above lower bound of  $h_t = 0.7469$  is compared to the expansion entropy computation. Figure A.2 reveals the expansion entropy result using 1.5 million randomly seeded trajectories. The resulting data is parsed in two different ways. Though it remains unclear which parsing method is more accurate, the true topological entropy lies between the minimum and maximum of two results: 0.9824 and 1.0554. This confirms that the symbolic entropy calculation is a strict lower bound.

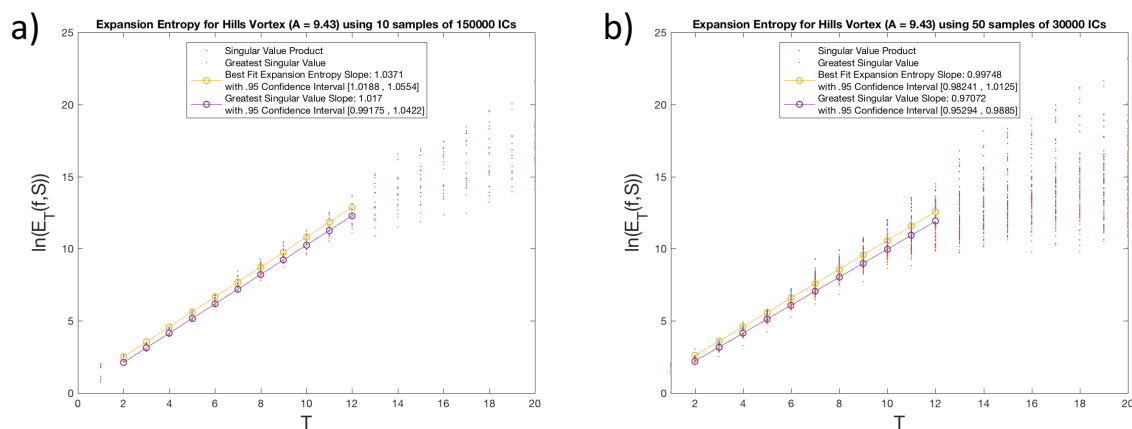


Figure A.2: We choose  $N = 1.5 \times 10^6$  points distributed randomly and a ball of radius 1.5 units centered at the origin as the restraining region  $S$ . We plot the exponential growth of both the product of singular values greater than one and the single largest singular value a) Data is parsed into 10 different samples of 150000 points. We estimate expansion entropy to be between 1.0188 and 1.0554. b) Same data is parsed into 50 samples of 30000 points, yielding an estimate of .98241 and 1.0125.

# Appendix B

## 1D Interval Stretching

We attempt to quantify chaos in 1D discrete dynamical systems by means of *interval stretching* that results from orbit collisions. This method acts as a sort-of precursor to the 2D E-tec algorithm, using 1D orbit data projected as a time series (1D positioning as a function of time). Initial partitions between orbits are labeled as intervals and they are stretched and folded over one another as the orbits evolve in time.

Iterate  $n$  denotes the  $n$ -th mapping of a discrete function. As illustrated in Fig. B.1, subintervals  $I_n^j$  are ordered in time by iterate  $n$ . At each time step  $n$ , the superscript  $j = 1, 2, \dots$  indexes through the descending order of orbit positions on the  $y$ -axis. The given ordering of subintervals  $I_n^j$  with respect to  $j$  does not depend on the previous iterate's subintervals  $I_{n-1}^j$ . For each iterate, the full interval is denoted  $I_n = \bigcup_{j=1}^{m-1} I_n^j$  and has length  $|I_n| = (\max f^n(x_0) - \min f^n(x_0))$ , representing the total length of the range of orbits at iterate  $n$ .

The subinterval evolution at each iterate is the key to quantifying the stretching of a times series of 1D data. In Fig. B.1, the three orbits  $b, g$ , and  $r$  create two subintervals at each iterate. As orbits cross, subintervals at iterate  $n - 1$  map to a union of one or more subintervals at the next iterate  $n$ . Procedurally-speaking, I count the number of times a subinterval is represented after a forward evolution, compute the sum of the lengths of these subintervals, and normalize over the iterate's full interval length  $|I_n|$ . We will denote this ratio of interval lengths at the  $n^{\text{th}}$  iterate as  $l_n$ . This, in essence, allows us to sort of quantify the stretching due to the crossing



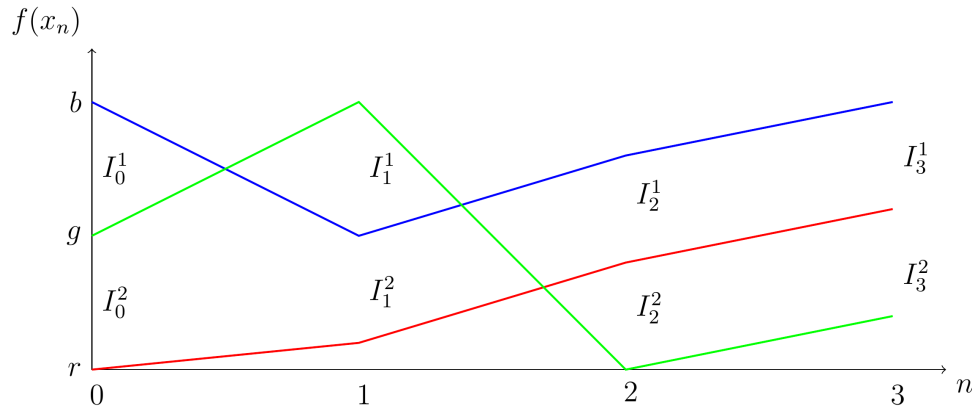


Figure B.1: Time series plot of three orbits  $b, g$ , and  $r$ . Please note, the superscript ordering of subintervals given up and down at a specific orbit  $n$  are independent of previous orderings.

of 1D orbit in a way that mirrors the 2D braiding work of Jean-Luc Thiffeault and Mario Budivsic [5, 42, 56].

I illustrate my method by finding these interval stretching ratios  $l_n$  in the example drawn in Fig. B.1,:

**Iterate  $n = 1$ :** The top subinterval at  $n = 0$ ,  $I_0^1$ , is mapped to the top subinterval interval  $I_1^1$ . The interval is flipped, but no stretching occurs, as  $I_0^1 \rightarrow I_1^1$ . Stretching occurs as the bottom green and red subinterval  $I_0^2$  is mapped to  $I_1^1 \cup I_1^2$ . Therefore, accounting for the crossing of the blue and green orbits, our stretched interval length ratio is now

$$l_1 = \frac{2|I_0^1| + |I_0^2|}{|I_1^1|}.$$

**Iterate  $n = 2$ :** We are now looking at how the subintervals at  $n = 1$  map to subintervals at  $n = 2$ . The green and blue subinterval  $I_1^1$  is now mapped to  $I_2^1 \cup I_2^2$  while the blue and red subinterval  $I_1^2$  is mapped to the single subinterval  $I_2^1$ . The stretched interval length ratio is then

$$l_2 = \frac{2|I_1^1| + |I_1^2|}{|I_2^1|}.$$

**Iterate  $n = 3$ :** At iterate  $n = 3$ , the no orbits cross and the ordering does not

change.  $I_2^1$  is mapped to  $I_3^1$  and  $I_2^2$  is mapped to  $I_3^2$ . No stretching occurs and the stretched interval length ratio is

$$l_3 = \frac{|I_2^1| + |I_2^2|}{|I_3|}.$$

The idea is to simply track the forward subinterval mapping at each iterate of the time series, keeping track of the orbit crossings and resulting increase in subinterval representation. To quantify stretching that results from these orbit intersections, I proceed in a manner similar to expansion entropy: a initial conditions are uniformly distributed and iterated forward while the stretched interval length ratio are  $l_n$  are computed for each iterate  $n = 0, 1, 2, \dots$ . The natural logarithm of the cumulative product of these ratios

$$E(t) = \ln \left( \prod_{j=0}^t l_t \right) \tag{B.1}$$

is plotted as a function of time. The "orbit collision/ interval exchange" entropy-like term, which we shall call  $h_{stretch}$ , will then be the slope of the best-fit line through Eq. B.1. Preliminary findings lead us to evidence that suggest  $h_{stretch}$  approximates topological entropy  $h_t$ . Let us begin with some examples.

## B.1 Chaotic Logistic Map

We begin with the logistic map, given by  $x_{n+1} = \sigma x_n(1-x_n)$ . For the parameter  $\sigma = 4$ , the logistic map has topological entropy  $h_t = \ln(2) \approx 0.69315$ . Interval stretching entropy  $h_{stretch}$  results are shown in Fig. B.2. The solid red lines plot Eq. B.1 as a function of iterate and represent the growth of the subinterval mappings. At each iterate, the orbits are resorted in descending order before computing the next iterate. It there tracks only the growth from one interval to the next. Blue line computations are done without any sorting of the subintervals and therefore represent only the growth of the initial subintervals.

With agreeance between  $h_{stretch}$  and the exact entropy value of  $\ln(2)$  for the logistic map with maximally chaotic parameter  $r = 4$ , I test my idea out on a full range of

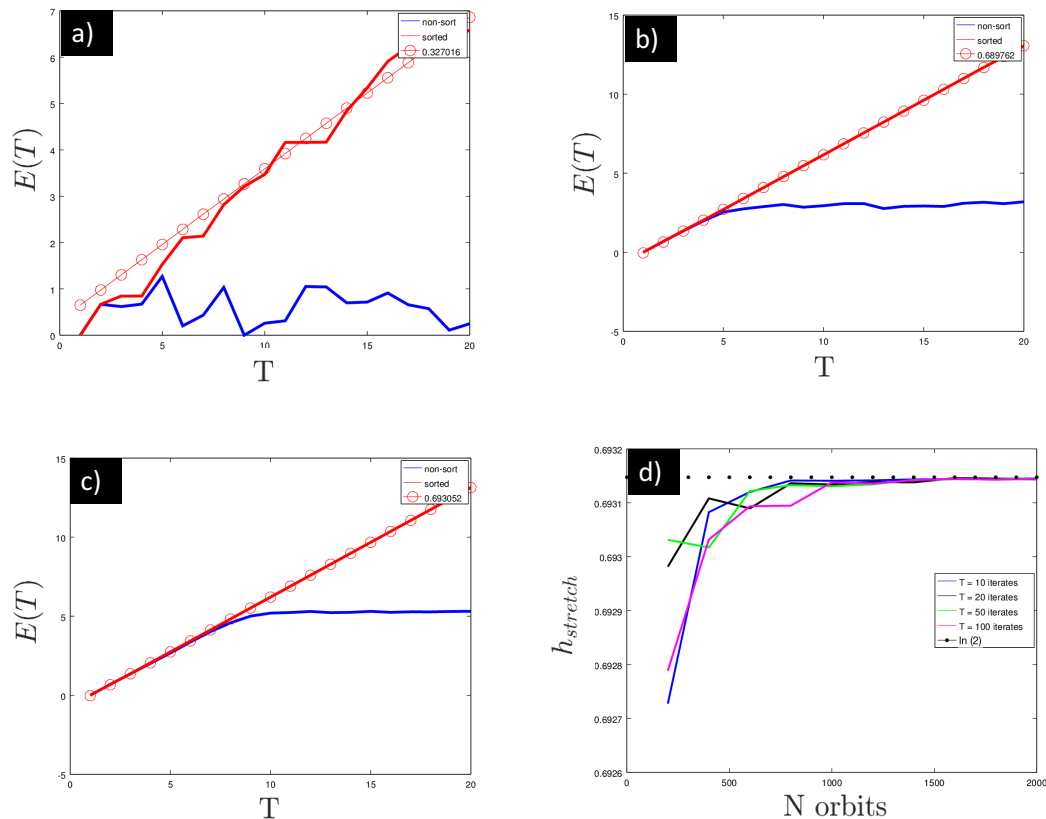


Figure B.2: Interval stretching entropy estimate for the chaotic logistic map with parameter  $\sigma = 4$ . Entropy is given as best-fit line through the solid data. a)  $h_{stretch} \approx 0.32702$  for  $N = 5$  orbits. b)  $h_{stretch} \approx 0.68976$  for  $N = 50$  orbits. c)  $h_{stretch} \approx 0.69305$  for  $N = 500$  orbits. d) Entropy results as a function of the number of orbits used. Correct entropy of  $\log(2)$  is shown for reference. Colored lines represent the iterate  $T$  at which the best-fit line is taken.

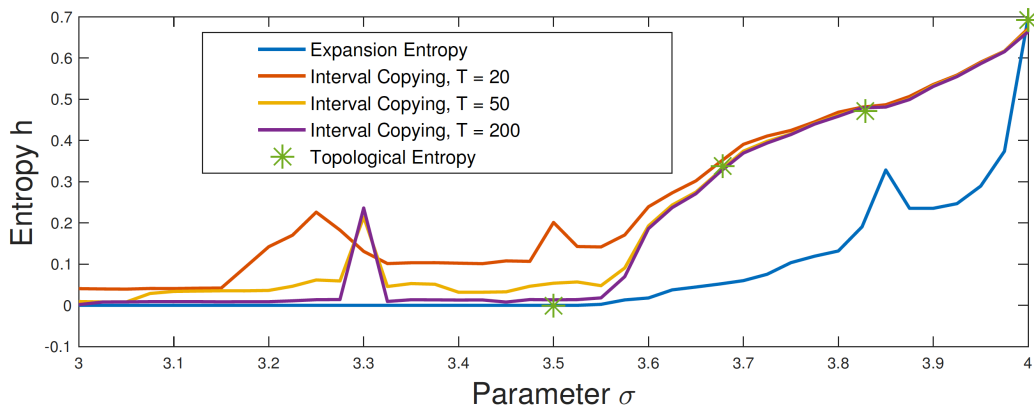


Figure B.3: Interval stretching (or copying) entropy  $h_{stretch}$  of the logistic map for a range of  $\sigma$  values. 1000 orbits are used for each computation of  $h_{stretch}$ , with different ending fit times  $T$  shown in differing colors. The expansion entropy for the same parameter values are shown. Exact topological entropy values are shown in green markers.

$\sigma$  values, using 1000 uniformly distributed orbits for each. I see good agreeance in Fig. B.3 between my method (in red, yellow, and purple) and the exact topological entropy values (in green) for non-chaotic parameter  $\sigma = 3.5$  and all chaotic parameters  $\sigma > 3.5$ . (The actual onset of chaos occurs at  $\sigma \approx 3.57$ .) the topological entropy is 0 for all  $\sigma < 3.5$ . However, my interval stretching method gives a false positive for many of these values. An explanation as to why is not given here and remains unexplored.

## B.2 Chaotic Gauss Map

Lastly, I analyze the Gauss Map, given by

$$x_{n+1} = \frac{1}{x_n} - \text{floor}\left(\frac{1}{x_n}\right) \quad \text{or} \quad x_{n+1} = \frac{1}{x_n} \bmod (1), \quad x_0 \in (0, 1]. \quad (\text{B.2})$$

Plotted in Fig. B.4, this map takes the fractional part of the reciprocal of the previous state to get the next state. This map is interestingly shown to have finite metric

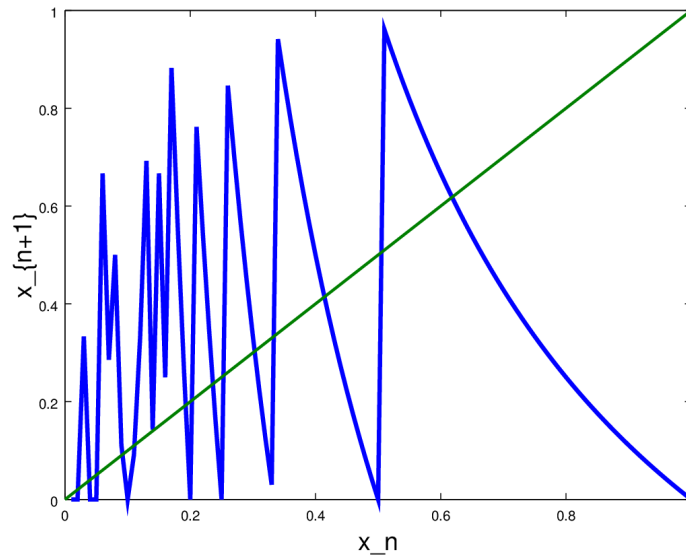


Figure B.4: Graph of discrete Gauss Map with fixed point line as reference.

entropy  $h_{\text{metric}} = \frac{\pi^2}{6 \ln(2)}$  [154] but infinite topological entropy [155]. Detailed in Fig. B.5 b), I compute  $h_{\text{stretch}}$  as a function of the number of orbits  $N$ . In contrast to the logistic map's convergence of  $h_{\text{stretch}}$  to  $\ln(2)$  for large  $N$ , the Gauss Map entropy appears to be growing logarithmically as  $N \rightarrow \infty$ , indicating  $h_{\text{stretch}} = \infty$  for large  $N$ .

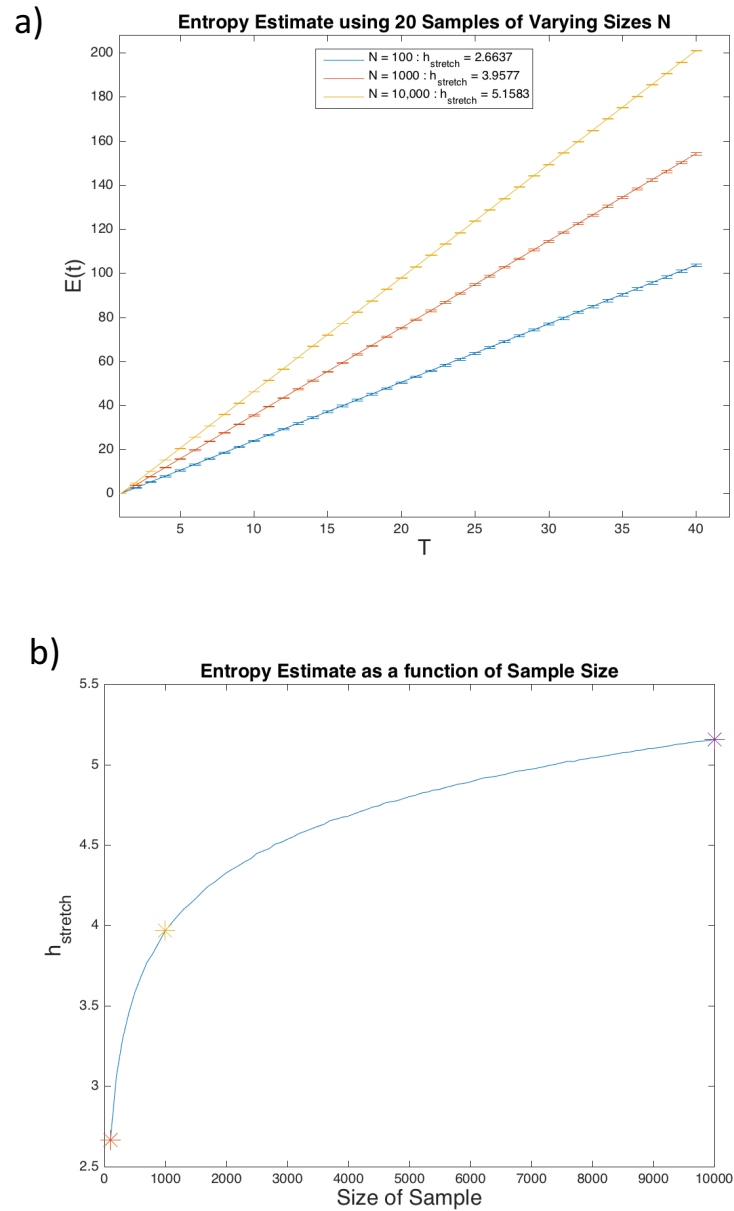


Figure B.5: a) Expansion Entropy results using 20 samples of  $N = 100, 1000,$  and  $10,000$  orbits. b) Interval stretching entropy  $h_{stretch}$  as a function of sample size  $N$ . Expansion Entropy results from a) are shown starred.

# Bibliography

- [1] Philip L Boyland, Hassan Aref, and Mark A Stremler. Topological fluid mechanics of stirring. *Journal of Fluid Mechanics*, 403:277–304, 2000.
- [2] Sulimon Sattari, Qianting Chen, and Kevin A Mitchell. Using heteroclinic orbits to quantify topological entropy in fluid flows. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(3):033112, 2016.
- [3] Piyush Grover, Shane D Ross, Mark A Stremler, and Pankaj Kumar. Topological chaos, braiding and bifurcation of almost-cyclic sets. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):043135, 2012.
- [4] Mark A Stremler, Shane D Ross, Piyush Grover, and Pankaj Kumar. Topological chaos and periodic braiding of almost-cyclic sets. *Physical Review Letters*, 106(11):114101, 2011.
- [5] Jean-Luc Thiffeault. braidlab: a software package. *arXiv preprint arXiv:1410.0849*, 2014.
- [6] Kevin A Mitchell. The topology of nested homoclinic and heteroclinic tangles. *Physica D: Nonlinear Phenomena*, 238(7):737–763, 2009.
- [7] Kevin A Mitchell. Partitioning two-dimensional mixed phase spaces. *Physica D: Nonlinear Phenomena*, 241(20):1718–1734, 2012.
- [8] Amanda J Tan, Eric Roberts, Spencer A Smith, Ulyses Alvarado Olvera, Jorge Arteaga, Sam Fortini, Kevin A Mitchell, and Linda S Hirst. Topological chaos in active nematics. *Nature Physics*, 15(10):1033–1039, 2019.

- [9] Charles Perrings. Resilience in the dynamics of economy-environment systems. *Environmental and Resource Economics*, 11(3-4):503–520, 1998.
- [10] V Křivan. Construction of population growth equations in the presence of viability constraints. *Journal of Mathematical Biology*, 29(4):379–387, 1991.
- [11] Francis Heylighen. Mediator evolution: a general scenario for the origin of dynamical hierarchies. *Worldviews, Science and Us. (Singapore: World Scientific)*, 2(3), 2006.
- [12] Xianning Liu and Lansun Chen. Complex dynamics of holling type ii lotka–volterra predator–prey system with impulsive perturbations on the predator. *Chaos, Solitons & Fractals*, 16(2):311–320, 2003.
- [13] Martin Bohner, Meng Fan, and Jimin Zhang. Existence of periodic solutions in predator–prey and competition dynamic systems. *Nonlinear Analysis: Real World Applications*, 7(5):1193–1204, 2006.
- [14] Pei-Chann Chang, Chen-Hao Liu, Jun-Lin Lin, Chin-Yuan Fan, and Celeste SP Ng. A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, 36(3):6889–6898, 2009.
- [15] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):2513–2522, 2005.
- [16] Donatello Materassi and Giacomo Innocenti. Topological identification in networks of dynamical systems. *IEEE Transactions on Automatic Control*, 55(8):1860–1871, 2010.
- [17] ML Shuler, S Leung, and CC Dick. A mathematical model for the growth of a single bacterial cell. *Annals of the New York Academy of Sciences*, 326(1):35–52, 1979.
- [18] Kathryn G Link, Michael T Stobb, Jorge Di Paola, Keith B Neeves, Aaron L Fogelson, Suzanne S Sindi, and Karin Leiderman. A local and global sensitivity



- analysis of a mathematical model of coagulation and platelet deposition under flow. *PloS One*, 13(7):e0200917, 2018.
- [19] Philip Holmes. Poincaré, celestial mechanics, dynamical-systems theory and chaos. *Physics Reports*, 193(3):137–163, 1990.
- [20] Henri Poincaré. Sur le problème des trois corps et les équations de la dynamique. *Acta Mathematica*, 13(1):A3–A270, 1890.
- [21] Henri Poincaré. *The three-body problem and the equations of dynamics: Poincaré’s foundational work on dynamical systems theory*, volume 443. Springer, 2017.
- [22] Ralph Abraham and Yoshisuke Ueda. *The chaos avant-garde: Memories of the early days of chaos theory*, volume 39. World scientific, 2000.
- [23] Otto E Rössler. Chaos, hyperchaos and the double-perspective. *The Chaos Avant-garde: Memoirs Of The Early Days Of Chaos Theory*, 39:209, 2001.
- [24] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.
- [25] James Gleick. *Chaos: Making a new science*. Open Road Media, 2011.
- [26] Vladimir G Ivancevic and Tijana T Ivancevic. *Complex nonlinearity: chaos, phase transitions, topology change and path integrals*. Springer Science & Business Media, 2008.
- [27] Hassan Aref. The development of chaotic advection. *Physics of Fluids*, 14(4):1315–1325, 2002.
- [28] Julio M Ottino and JM Ottino. *The kinematics of mixing: stretching, chaos, and transport*, volume 3. Cambridge university press, 1989.
- [29] Hassan Aref, John R Blake, Marko Budišić, Silvana SS Cardoso, Julyan HE Cartwright, Herman JH Clercx, Kamal El Omari, Ulrike Feudel, Ramin

- Golestanian, Emmanuelle Guillard, et al. Frontiers of chaotic advection. *Reviews of Modern Physics*, 89(2):025007, 2017.
- [30] María J Olascoaga and George Haller. Forecasting sudden changes in environmental pollution patterns. *Proceedings of the National Academy of Sciences*, 109(13):4738–4743, 2012.
- [31] Gary Froyland, Christian Horenkamp, Vincent Rossi, Naratip Santitissadeekorn, and Alex Sen Gupta. Three-dimensional characterization and tracking of an Agulhas ring. *Ocean Modelling*, 52:69–75, 2012.
- [32] Chia-Yen Lee, Chin-Lung Chang, Yao-Nan Wang, and Lung-Ming Fu. Microfluidic mixing: a review. *International Journal of Molecular Sciences*, 12(5):3263–3287, 2011.
- [33] Tim Sanchez. *Self-Organization in Active Cytoskeletal Mixtures: Cilia-like Beating of Microtubule Bundles and Spontaneous Bulk Mixing*. Brandeis University, 2012.
- [34] Matthew D Finn and Jean-Luc Thiffeault. Topological optimization of rod-stirring devices. *SIAM Review*, 53(4):723–743, 2011.
- [35] Jean-Luc Thiffeault. The mathematics of taffy pullers. *The Mathematical Intelligencer*, 40(1):26–35, 2018.
- [36] Yushi Chu, Qingliu Hu, Yindong Zhang, Zhigang Gao, Zaijin Fang, Lu Liu, Qiqi Yan, Yinyao Liu, Shiyu Sun, Gang-Ding Peng, et al. Topological engineering of photoluminescence properties of bismuth- or erbium-doped phosphosilicate glass of arbitrary  $P_2O_5$  to  $SiO_2$  ratio. *Advanced Optical Materials*, 6(13):1800024, 2018.
- [37] Ruiqiang Ding and Jianping Li. Nonlinear finite-time lyapunov exponent and predictability. *Physics Letters A*, 364(5):396–400, 2007.

- [38] J-P Eckmann and David Ruelle. Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems. In *Turbulence, Strange Attractors, and Chaos*, pages 447–449. World Scientific, 1995.
- [39] Gary Froyland and Kathrin Padberg-Gehle. Finite-time entropy: A probabilistic approach for measuring nonlinear stretching. *Physica D: Nonlinear Phenomena*, 241(19):1612–1628, 2012.
- [40] Hassan Arbabi and Igor Mezić. Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator. *arXiv preprint arXiv:1611.06664*, 2016.
- [41] Michael R Allshouse and Thomas Peacock. Lagrangian based methods for coherent structure detection. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097617, 2015.
- [42] Marko Budišić and Jean-Luc Thiffeault. Finite-time braiding exponents. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(8):087407, 2015.
- [43] Simon Candelaresi, David Ian Pontin, and Gunnar Hornig. Quantifying the tangling of trajectories using the topological entropy. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(9):093102, 2017.
- [44] Yosef Yomdin. Volume growth and entropy. *Israel Journal of Mathematics*, 57(3):285–300, 1987.
- [45] Sheldon E Newhouse. Entropy and volume. *Ergodic Theory and Dynamical Systems*, 8(8):283–299, 1988.
- [46] Sheldon Newhouse and Thea Pignataro. On the estimation of topological entropy. *Journal of Statistical Physics*, 72(5):1331–1351, 1993.
- [47] Sheldon E Newhouse. Continuity properties of entropy. *Annals of Mathematics*, 129(1):215–235, 1989.

- [48] Rufus Bowen. Periodic points and measures for axiom A diffeomorphisms. *Transactions of the American Mathematical Society*, 154:377–397, 1971.
- [49] Lai-Sang Young. Entropy in dynamical systems. *Entropy*, 313, 2003.
- [50] Brian R Hunt and Edward Ott. Defining chaos. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097618, 2015.
- [51] Philip Boyland. Topological methods in surface dynamics. *Topology and its Applications*, 58(3):223–298, 1994.
- [52] A Vikhansky. Chaotic advection of finite-size bodies in a cavity flow. *Physics of Fluids*, 15(7):1830–1836, 2003.
- [53] Emil Artin. Theory of braids. *Annals of Mathematics*, pages 101–126, 1947.
- [54] Albert Fathi, François Laudenbach, and Valentin Poénaru. *Thurston’s Work on Surfaces (MN-48)*, volume 48. Princeton University Press, 2012.
- [55] Jean-Luc Thiffeault. Measuring topological chaos. *Physical Review Letters*, 94(8):084502, 2005.
- [56] Jean-Luc Thiffeault. Braids of entangled particle trajectories. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):017516, 2010.
- [57] Matthew D Finn and Jean-Luc Thiffeault. Topological entropy of braids on the torus. *SIAM Journal on Applied Dynamical Systems*, 6(1):79–98, 2007.
- [58] Matthew D Finn, Jean-Luc Thiffeault, and Emmanuelle Guillard. Topological chaos in spatially periodic mixers. *Physica D: Nonlinear Phenomena*, 221(1):92–100, 2006.
- [59] Ivan Alekseevich Dynnikov. On a Yang-Baxter map and the Dehornoy ordering. *Russian Mathematical Surveys*, 57(3):592–594, 2002.
- [60] James R Ledwell, Andrew J Watson, and Clifford S Law. Evidence for slow mixing across the pycnocline from an open-ocean tracer-release experiment. *Nature*, 364(6439):701–703, 1993.

- [61] Douglas H Kelley and Nicholas T Ouellette. Separating stretching from folding in fluid mixing. *Nature Physics*, 7(6):477, 2011.
- [62] Nicholas T Ouellette and Jerry P Gollub. Dynamic topology in spatiotemporal chaos. *Physics of Fluids*, 20(6):064104, 2008.
- [63] Michael R Allshouse and Jean-Luc Thiffeault. Detecting coherent structures using braids. *Physica D: Nonlinear Phenomena*, 241(2):95–105, 2012.
- [64] Marc Lefranc. Reflections from the fourth dimension. *Topology and Dynamics of Chaos: In Celebration of Robert Gilmore’s 70th Birthday*, 84:205, 2013.
- [65] Marc Lefranc. Alternative determinism principle for topological analysis of chaos. *Physical Review E*, 74(3):035202, 2006.
- [66] Marc Lefranc, Pierre-Emmanuel Morant, and Michel Nizette. Topological characterization of deterministic chaos: enforcing orientation preservation. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1865):559–567, 2008.
- [67] Csaba D Toth, Joseph O’Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry*. Chapman and Hall/CRC, 2017.
- [68] Pradeep Rao, Andrew Dugleby, and Mark A Stremler. Mixing analysis in a lid-driven cavity flow at finite reynolds numbers. *Journal of Fluids Engineering*, 134(4):041203, 2012.
- [69] VV Meleshko and AM Gomilko. Infinite systems for a biharmonic problem in a rectangle: further discussion. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2043):807–819, 2004.
- [70] Philip Boyland, Mark Stremler, and Hassan Aref. Topological fluid mechanics of point vortex motions. *Physica D: Nonlinear Phenomena*, 175(1):69–95, 2003.
- [71] Mladen Bestvina and Michael Handel. Train tracks and automorphisms of free groups. *Annals of Mathematics*, 135(1):1–51, 1992.

- [72] Jean-Luc Thiffeault and Matthew D Finn. Topology, braids and mixing in fluids. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 364(1849):3251–3266, 2006.
- [73] Emmanuelle Gouillart, Jean-Luc Thiffeault, and Matthew D Finn. Topological mixing with ghost rods. *Physical Review E*, 73(3):036311, 2006.
- [74] Tien-Yien Li and James A Yorke. Period three implies chaos. *The American Mathematical Monthly*, 82(10):985–992, 1975.
- [75] OM Sharkovsky. Coexistence of the cycles of a continuous mapping of the line into itself. *Ukrainskij Matematicheskij Zhurnal*, 16(01):61–71, 1964.
- [76] Sarah E Tumas and Jean-Luc Thiffeault. Estimating topological entropy from the motion of stirring rods. *Procedia IUTAM*, 7:117–126, 2013.
- [77] Zhiping You, Eric J Kostelich, and James A Yorke. Calculating stable and unstable manifolds. *International Journal of Bifurcation and Chaos*, 1(03):605–623, 1991.
- [78] John Toner and Yuhai Tu. Long-range order in a two-dimensional dynamical XY model: how birds fly together. *Physical Review Letters*, 75(23):4326, 1995.
- [79] Yael Katz, Kolbjørn Tunstrøm, Christos C Ioannou, Cristián Huepe, and Iain D Couzin. Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725, 2011.
- [80] Kyogo Kawaguchi, Ryoichiro Kageyama, and Masaki Sano. Topological defects control collective dynamics in neural progenitor cell cultures. *Nature*, 545(7654):327, 2017.
- [81] Thuan Beng Saw, Amin Doostmohammadi, Vincent Nier, Leyla Kocgozlu, Sumesh Thampi, Yusuke Toyama, Philippe Marcq, Chwee Teck Lim, Julia M Yeomans, and Benoit Ladoux. Topological defects in epithelia govern cell death and extrusion. *Nature*, 544(7649):212, 2017.

- [82] Andrey Sokolov, Igor S Aranson, John O Kessler, and Raymond E Goldstein. Concentration dependence of the collective dynamics of swimming bacteria. *Physical Review Letters*, 98(15):158102, 2007.
- [83] Henricus H Wensink, Jörn Dunkel, Sebastian Heidenreich, Knut Drescher, Raymond E Goldstein, Hartmut Löwen, and Julia M Yeomans. Meso-scale turbulence in living fluids. *Proceedings of the National Academy of Sciences*, 109(36):14308–14313, 2012.
- [84] Henry Hess, John Clemmens, Christian Brunner, Robert Doot, Sheila Luna, Karl-Heinz Ernst, and Viola Vogel. Molecular self-assembly of nanowires and nanospools using active transport. *Nano Letters*, 5(4):629–633, 2005.
- [85] Tim Sanchez, Daniel TN Chen, Stephen J DeCamp, Michael Heymann, and Zvonimir Dogic. Spontaneous motion in hierarchically assembled active matter. *Nature*, 491(7424):431, 2012.
- [86] Wen Lu, Michael Winding, Margot Lakonishok, Jill Wildonger, and Vladimir I Gelfand. Microtubule–microtubule sliding by kinesin-1 is essential for normal cytoplasmic streaming in drosophila oocytes. *Proceedings of the National Academy of Sciences*, 113(34):E4995–E5004, 2016.
- [87] Luca Giomi. Geometry and topology of turbulence in active nematics. *Physical Review X*, 5(3):031003, 2015.
- [88] Gil Henkin, Stephen J DeCamp, Daniel TN Chen, Tim Sanchez, and Zvonimir Dogic. Tunable dynamics of microtubule-based active isotropic gels. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2029):20140142, 2014.
- [89] Stephen J DeCamp, Gabriel S Redner, Aparna Baskaran, Michael F Hagan, and Zvonimir Dogic. Orientational order of motile defects in active nematics. *Nature Materials*, 14(11):1110, 2015.

- [90] Pau Guillamat, Jordi Ignés-Mullol, and Francesc Sagués. Patterning active materials with addressable soft interfaces. *arXiv preprint arXiv:1511.03880*, 2015.
- [91] Jeremy R Knowles. Enzyme-catalyzed phosphoryl transfer reactions. *Annual Review of Biochemistry*, 49(1):877–919, 1980.
- [92] John C Crocker and David G Grier. Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science*, 179(1):298–310, 1996.
- [93] Yongxiang Gao and Maria L Kilfoi. Accurate detection and complete tracking of large populations of features in three dimensions. *Optics Express*, 17(6):4685–4704, 2009.
- [94] David L Valentine, Igor Mezić, Senka Maćešić, Nelida Črnjarić-Žic, Stefan Ivić, Patrick J Hogan, Vladimir A Fonoberov, and Sophie Loire. Dynamic autoinoculation and the microbial ecology of a deep water hydrocarbon irruption. *Proceedings of the National Academy of Sciences*, 109(50):20286–20291, 2012.
- [95] Alireza Hadjighasem and George Haller. Geodesic transport barriers in Jupiter’s atmosphere: A video-based analysis. *SIAM Review*, 58(1):69–89, 2016.
- [96] George Haller and FJ Beron-Vera. Coherent Lagrangian vortices: The black holes of turbulence. *Journal of Fluid Mechanics*, 731, 2013.
- [97] Sulimon Sattari and Kevin A Mitchell. Using periodic orbits to compute chaotic transport rates between resonance zones. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(11):113104, 2017.
- [98] Víctor J García-Garrido, Jezabel Curbelo, Ana M Mancho, Stephen Wiggins, and Carlos R Mechoso. The application of Lagrangian descriptors to 3D vector fields. *Regular and Chaotic Dynamics*, 23(5):551–568, 2018.



- [99] Daniel Blazeovski and George Haller. Hyperbolic and elliptic transport barriers in three-dimensional unsteady flows. *Physica D: Nonlinear Phenomena*, 273:46–62, 2014.
- [100] Michał Branicki and Stephen Wiggins. An adaptive method for computing invariant manifolds in non-autonomous, three-dimensional dynamical systems. *Physica D: Nonlinear Phenomena*, 238(16):1625–1657, 2009.
- [101] Daniel Mukiibi, Gualtiero Badin, and Nuno Serra. Three-dimensional chaotic advection by mixed layer baroclinic instabilities. *Journal of Physical Oceanography*, 46(5):1509–1529, 2016.
- [102] Blake Rutherford, Gerhard Dangelmayr, and Michael T Montgomery. Lagrangian coherent structures in tropical cyclone intensification. *Atmospheric Chemistry and Physics*, 12(12):5483–5507, 2012.
- [103] Joao H Bettencourt, Cristóbal López, and Emilio Hernández-García. Oceanic three-dimensional lagrangian coherent structures: A study of a mesoscale eddy in the benguela upwelling region. *Ocean Modelling*, 51:73–83, 2012.
- [104] Igor Mezić and Stephen Wiggins. On the integrability and perturbation of three-dimensional fluid flows with symmetry. *Journal of Nonlinear Science*, 4(1):157–194, 1994.
- [105] Jezabel Curbelo, Víctor José García-Garrido, Carlos Roberto Mechoso, Ana Maria Mancho, Stephen Wiggins, and Coumba Niang. Insights into the three-dimensional Lagrangian geometry of the Antarctic polar vortex. *Nonlinear Processes in Geophysics*, 24(3):379–392, 2017.
- [106] Jezabel Curbelo, Carlos R Mechoso, Ana M Mancho, and Stephen Wiggins. Lagrangian study of the final warming in the southern stratosphere during 2002: Part I. the vortex splitting at upper levels. *Climate Dynamics*, pages 1–14, 2018.
- [107] Yimin Luo, Tianyi Yao, Daniel A Beller, Francesca Serra, and Kathleen J Stebe. Deck the walls with anisotropic colloids in nematic liquid crystals. *Langmuir*, 2019.

- [108] Nitin Kumar, Rui Zhang, Juan J de Pablo, and Margaret L Gardel. Tunable structure and dynamics of active liquid crystals. *Science Advances*, 4(10):eaat7779, 2018.
- [109] Hugo Ledoux. The kinetic 3D Voronoi diagram: A tool for simulating environmental processes. In *Advances in 3D Geoinformation Systems*, pages 361–380. Springer, 2008.
- [110] Hugo Ledoux and Christopher M Gold. Modelling three-dimensional geoscientific fields with the Voronoi diagram and its dual. *International Journal of Geographical Information Science*, 22(5):547–574, 2008.
- [111] Yong Xue, Min Sun, and Ainai Ma. On the reconstruction of three-dimensional complex geological objects using Delaunay triangulation. *Future Generation Computer Systems*, 20(7):1227–1234, 2004.
- [112] Mir Abolfazl Mostafavi and Christopher Gold. A global kinetic spatial data structure for a marine simulation. *International Journal of Geographical Information Science*, 18(3):211–227, 2004.
- [113] Mir Abolfazl Mostafavi. Development of a global dynamic data structure. 2003.
- [114] Martin J Fritts, W Patrick Crowley, and Harold Trease. The Free-Lagrange method; proceedings of the first international conference, Hilton Head Island, SC, March 4-6, 1985. In *Lecture Notes in Physics, Berlin Springer Verlag*, volume 238, 1985.
- [115] Charles L Lawson. Software for c1 surface interpolation. In *Mathematical Software*, pages 161–194. Elsevier, 1977.
- [116] Barry Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8(2):123–142, 1991.
- [117] Herbert Edelsbrunner and Raimund Seidel. Voronoi diagrams and arrangements. *Discrete & Computational Geometry*, 1(1):25–44, 1986.

- [118] Charles L Lawson. Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3(4):231–246, 1986.
- [119] Adrian Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.
- [120] Rex A Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6(3):343–367, 1991.
- [121] Herbert Edelsbrunner and Nimish R Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15(3):223–241, 1996.
- [122] David A Field. Implementing Watson’s algorithm in three dimensions. In *Proceedings of the second annual symposium on Computational geometry*, pages 246–259. ACM, 1986.
- [123] Arne Maus. Delaunay triangulation and the convex hull ofn points in expected linear time. *BIT Numerical Mathematics*, 24(2):151–163, 1984.
- [124] Ernst P Mücke. A robust implementation for three-dimensional delaunay triangulations. *International Journal of Computational Geometry & Applications*, 8(02):255–276, 1998.
- [125] Masaharu Tanemura, Tohru Ogawa, and Naofumi Ogita. A new algorithm for three-dimensional Voronoi tessellation. *Journal of Computational Physics*, 51(2):191–207, 1983.
- [126] David F Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.
- [127] Kevin Q Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9(5):223–228, 1979.
- [128] Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Dewall: A fast divide and conquer Delaunay triangulation algorithm in ed. *Computer-Aided Design*, 30(5):333–341, 1998.

- [129] Steven Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1-4):153, 1987.
- [130] CL Bajaj and WJ Bouma. Dynamic Voronoi diagrams and Delaunay triangulations. In *Proc. 2nd Canadian Conference on Computational Geometry*, page 273, 1990.
- [131] Keiko Imai, S Sumino, and Hiroshi Imai. Minimax geometric fitting of two corresponding sets of points. In *Proceedings of the fifth annual symposium on Computational geometry*, pages 206–275. ACM, 1989.
- [132] Thomas Roos and Hartmut Noltemeier. Dynamic Voronoi diagrams in motion planning. In *Workshop on Computational Geometry*, pages 227–236. Springer, 1991.
- [133] Thomas Roos. Voronoi diagrams over dynamic scenes. *Discrete Applied Mathematics*, 43(3):243–259, 1993.
- [134] G Albers. Three-dimensional dynamic Voronoi diagrams. *Diploma thesis, University of Würzburg*, 1991.
- [135] Gerhard Albers and Thomas Roos. Voronoi diagrams of moving points in higher dimensional spaces. In *Scandinavian Workshop on Algorithm Theory*, pages 399–409. Springer, 1992.
- [136] Gerhard Albers, Leonidas J Guibas, Joseph SB Mitchell, and Thomas Roos. Voronoi diagrams of moving points. *International Journal of Computational Geometry & Applications*, 8(03):365–379, 1998.
- [137] Marina L Gavrilova and Jon Rokne. Updating the topology of the dynamic Voronoi diagram for spheres in Euclidean d-dimensional space. *Computer Aided Geometric Design*, 20(4):231–242, 2003.
- [138] Tomáš Vornáčka. Delaunay triangulation of moving points. In *Proceedings of the 12th Central European Seminar on Computer Graphics*, pages 67–74, 2008.

- [139] Leonidas Guibas and Daniel Russel. An empirical comparison of techniques for updating Delaunay triangulations. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 170–179. ACM, 2004.
- [140] Leonidas J Guibas, Menelaos I Karavelas, and Daniel Russel. A computational framework for handling motion. In *ALLENEX/ANALC*, pages 129–141, 2004.
- [141] Christopher M Gold. Spatial data structures: the extension from one to two dimensions. In *Mapping and Spatial Modelling for Navigation*, pages 11–39. Springer, 1990.
- [142] Christopher M Gold, Peter R Remmele, and Thomas Roos. Voronoi methods in gis. In *Advanced School on the Algorithmic Foundations of Geographic Information Systems*, pages 21–35. Springer, 1996.
- [143] G Erlebacher. Finite difference operators on unstructured triangular meshes. In *The Free-Lagrange Method*, pages 22–53. Springer, 1985.
- [144] Michael Allaby and M Allaby. Oxford Dictionary of Earth Science, 2008.
- [145] Roy L Adler, Alan G Konheim, and M Harry McAndrew. Topological entropy. *Transactions of the American Mathematical Society*, 114(2):309–319, 1965.
- [146] Ya G Sinai. Metric entropy of dynamical system. *Math. Princeton. Edu*, pages 1–4, 2007.
- [147] Spencer A Smith, Joshua Arenson, Eric Roberts, Suzanne Sindi, and Kevin A Mitchell. Topological chaos in a three-dimensional spherical fluid vortex. *EPL (Europhysics Letters)*, 117(6):60005, 2017.
- [148] Dmitriy V Anosov, S Kh Aranson, Vladimir Igorevic Arnold, IU Bronshtein, Yu S Il'yashenko, and VZ Grines. *Ordinary differential equations and smooth dynamical systems*. Springer-Verlag New York, Inc., 1997.
- [149] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.

- [150] Peter Grassberger and Ir Procaccia. Measuring the strangeness of strange attractors. In *The Theory of Chaotic Attractors*, pages 170–189. Springer, 2004.
- [151] OS Kozlovski. An integral formula for topological entropy of  $c^{\text{inf}}$  maps. *Ergodic Theory and Dynamical Systems*, 18(2):405–424, 1998.
- [152] Micaiah John Muller Hill. On a spherical vortex. *Proceedings of the Royal Society of London*, 55(331-335):219–224, 1894.
- [153] HK Moffatt and DW Moore. The response of Hill’s spherical vortex to a small axisymmetric disturbance. *Journal of Fluid Mechanics*, 87(4):749–760, 1978.
- [154] Ricardo Mané. *Ergodic theory and differentiable dynamics*, volume 8. Springer Science & Business Media, 2012.
- [155] Robert M Corless, Gregory W Frank, and J Graham. Chaos and continued fractions. *Physica D: Nonlinear Phenomena*, 46(2):241–253, 1990.