5-2024

Identifying Transitions in Plasma with Topological Data Analysis of Noisy Turbulence

Julius Kiewel
William & Mary

Identifying Transitions in Plasma with Topological Data Analysis of Noisy Turbulence

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science with Honors in
Physics from the College of William and Mary in Virginia,

by

Julius F. Kiewel

Accepted for Honors

____
Advisor: Prof. Saskia Mordijck, Physics

Sarah Day
____
Advisor: Prof. Sarah Day, Mathematics

David
____
Prof. David Armstrong, Physics

Williamsburg, Virginia
May 8, 2024

Contents

List of Figures

Acknowledgments

I would like to thank all of the people who advised me on this Thesis for the many hours and meetings put in. A special thanks go to Professor Saskia Mordijck and Professor Sarah Day for meeting with me in all different circumstances, and indeed for giving me the opportunities I have received. Another thank you goes to Sage Stanish for setting the foundations of this project, and giving me the tools to continue working on it. Finally I would like to thank the members of the William & Mary plasma physics lab, who gave me advice and suggestions throughout, especially to Nick, Gwen, and Griffin, who have been with me every step of my physics journey.

Abstract

Cross-field transport and heat loss in a magnetically confined plasma is determined by turbulence driven by perpendicular (to the magnetic field) pressure gradients. The heat losses from turbulence can make it difficult to maintain the energy density required to reach and maintain the conditions necessary for fusion. Self-organization of turbulence into intermediate scale so-called zonal flows can reduce the radial heat losses, however identifying when the transition occurs and any precursors to the transition is still a challenge. Topological Data Analysis (TDA) is a mathematical method which is used to extract topological features from point cloud and digital data to develop a methodology to identify the transition from turbulent dominant to zonal flow dominant behavior. When expanding this approach to experimental observations, certain topological methods are susceptible to noise, which can appear as small scale topological features and crowd out legitimate topology. We explore techniques to mitigate the effects of noise in the use of TDA on plasma data, and to demonstrate methodology that is able to identify transitions despite a high noise-to-signal ratio. In this thesis, we will focus on developing mathematical models to test the efficacy of different smoothing algorithms on reestablishing topological structure lost in modeled noisy data, as well as show that it is possible capture the transition to self-organized flows in the presence of a high noise-to-signal ratio without first using processing to approximate the pre-noise image. Finally, we apply the methodology to experimental image data to capture turbulence transitions.

Chapter 1

Introduction

1.1 Motivation

Turbulence is one of the most persistent problems in classical physics, and is of importance to an incredible number of self-organizing physical processes, encompassing systems such as the Jet Stream and Jupiter's Bands. One developing area of increased interest is the role of turbulence in magnetically confined plasmas, which are relevant to modern efforts in developing energy-generating fusion devices. The most promising versions of such devices use magnetically confined plasmas, where small instabilities can cause turbulence, thereby enhancing heat transport to the plasma edge and outside of magnetic confinement [1]. This can make it difficult to maintain the energy density required to reach and maintain fusion conditions. As a result, identifying when turbulence self-organizes into flows which can quench turbulent heat losses is an important transition to identify, especially as Magnetic Confinement Devices seek to overcome the barrier between low to high (L-H) efficiency confinement mode transition. Current transition diagnostics include Fourier Transform Analysis and require arbitrarily chosen values for the percentage of energy in zonal flows, collective motion of energy or particles within a system, with respect to the total kinetic energy in the turbulent system. These procedures make automation challenging.

1.2 Topological Data Analysis and Noise

A new methodology, specifically Computational Homology, is promising for automatically identifying the transition regimes in plasma simulations. In a previous thesis, Stanish [2] established that the number of topological features in an image has a strong correlation with the turbulent regime identified by Fourier analysis of zonal flows.

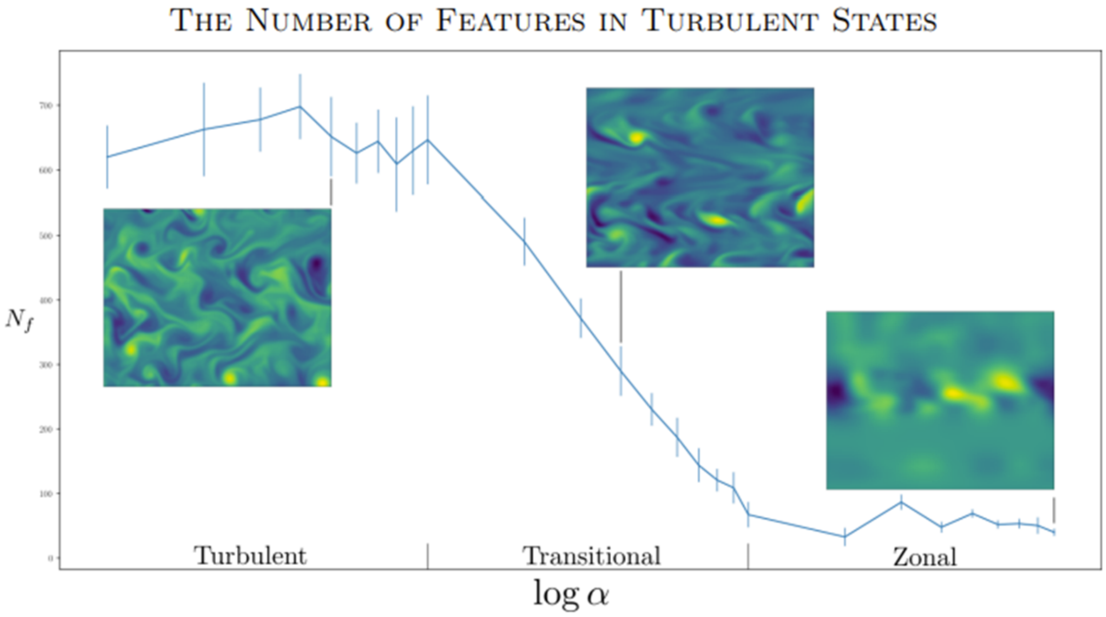

Figure 1.1: Number of features as a function of adiabaticity (α)
Curve demonstrates how topological characteristics (number of features) match with turbulence and zonal regimes. This transition is driven by varying a parameter (adiabaticity) in plasmas. Figure from Stanish [2].

Figure 1.1 shows the number of features in an image (a metric derived from Topological Data Analysis) as a function of the logarithm of adiabaticity, a parameter used to drive different flow regimes within 2-dimensional magneto-hydrodynamic simulations. A clear transition can be identified within the simulated images.

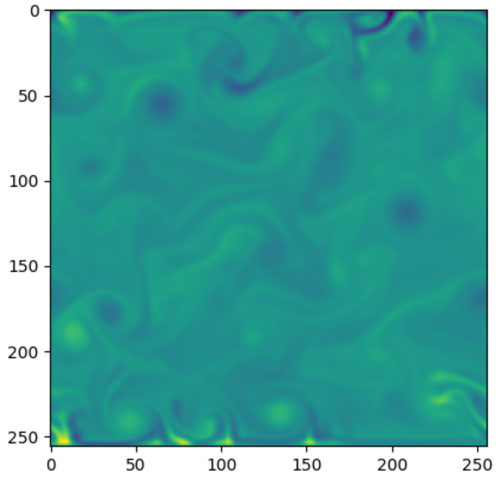

Figure 1.2: Noiseless Simulation Density heatmap.

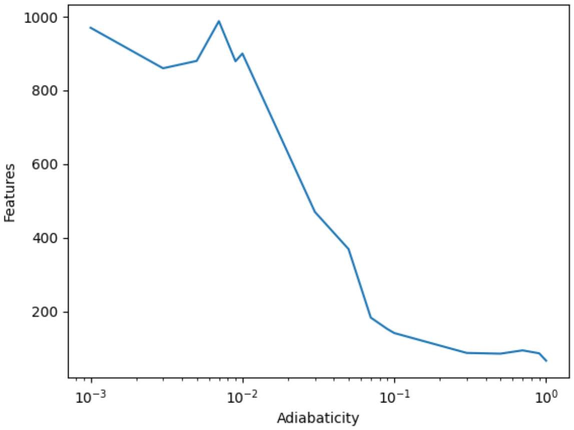

Figure 1.3: Clear transition with no applied noise.

The number of features demonstrates a clear indication of transition occurring within an image. Figure 1.2 is an example of one frame, corresponding to a single point in Fig. 1.3

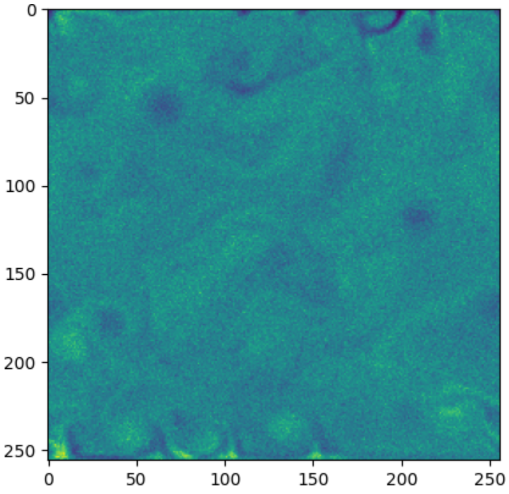

Figure 1.4: Heatmap with applied Poisson noise.

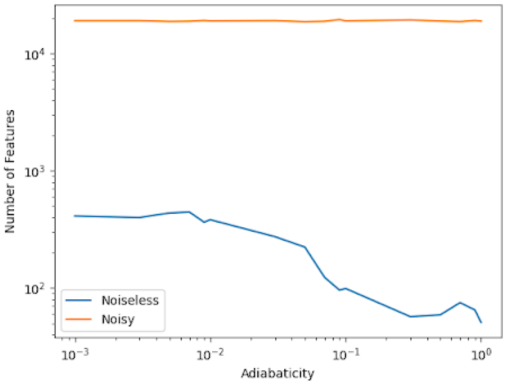

Figure 1.5: Transition with applied noise (orange) compared to noiseless case (blue).

However, one challenge in the application of this methodology is the introduction of noise within real data sets (Fig. 1.4), specifically to some common types of camera noise, like salt and pepper, Gaussian, or Poisson noise. This runs the risk of losing and obscuring relevant data, as can be seen in Fig. 1.5 . Figure 1.5 shows the same transition without noise at the very bottom of the diagram, and we see that the addition of noise can obscure the topological signal, so with the addition of noise to the image, we instead get the curve at the top of this figure instead. As a result, new methods need to be developed and made resilient against noise in order for identification to be possible in real life experimental and control systems settings.

1.3 Goals

These new methods require an understanding of the limits of topology in the presence of image processing algorithms. The goal of this thesis is to investigate the efficacy of different smoothing algorithms for 2D image and video data from plasmas and analyze the benefits and shortfalls of different post-processing algorithms in approximating pre-noise images for the purpose of measuring topology while maintaining topological integrity. The next goal is to demonstrate that alternative topological metrics exist which may better be able to demonstrate transitions in the presence of noise. The final goal is to be able to demonstrate the ability to identify transitions of real-world noisy camera data, both by applying smoothing algorithms, and by using methodologies that can qualitatively identify transitions without first approximating a pre-noise image.

Chapter 2

Topology Theory

Topological Data Analysis (TDA) [3] is a methodology gaining popularity in applied mathematics for its ability to extract topological and geometric information from point cloud (discrete set of points in Cartesian space) and other digital image data. This allows measurement of coarse topological properties in images, and has also demonstrated promise in being a metric for identifying transitions in two-dimensional plasma data.

2.1 Betti Features

Betti numbers represent the connectivity of sets of points, lines, and their n-dimensional counterparts in higher dimensions. In the topological space $\mathbb{X}$, $H_k(\mathbb{X})$ are defined as the k-th homology groups for $k \in \mathbb{Z}$. These groups are an algebraic description of the structure of k-dimensional holes. Their associated Betti numbers $\beta_k = \dim H_k(\mathbb{X})$, count the number of features of a given dimension. For example, for $\mathbb{X} \subset \mathbb{R}^2$, β_0 represents the number of connected components of $\mathbb{X}$ and β_1 represents the number of 1-dimensional holes in $\mathbb{X}$ [4].

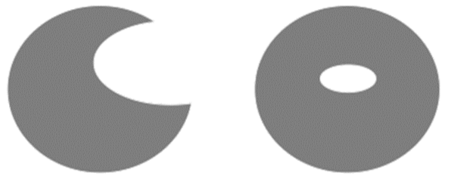

Figure 2.1: Two sets with gray area representing a set with $\beta_0 = 1$.
$\beta_1 = 0$ (left), and $\beta_1 = 1$ (right)

The shapes in Fig 2.1 represent two different sets, where the grey area represents points included in the set. Both consist of one connected set, that is the one 0-dimensional Betti number is $\beta_0 = 1$. The set to the right of the image additionally has a one dimensional hole and $\beta_1 = 1$, while the set on the left has no 1-dimensional holes so $\beta_1 = 0$.

2.2 Persistent Homology - Sublevel Sets

One way to measure simple binary Betti features for continuous functions is to decompose grayscale data into a series of nested sets and then measure the appearance and disappearance of topological features while moving through the set using a tool called *persistent homology*. These nested sets are called a *sublevel set filtration*. Figure 2.2 demonstrates a filtration of a the given 1-dimensional function by increasing the value of a threshold. As the threshold increases, topological changes occur in the sublevel sets. As the threshold continues to increase in this series, passing local minima creates new features. At a local maximum, features merge. By convention, the feature that emerged more recently in the sublevel set is considered to have died.

This simple 1-dimensional example can be extrapolated to more dimensions. Defining a grayscale image as $f : P \to \{0, 1, ..., 255\}$, where $f(x)$ gives the value of pixel x, taking the threshold, for value t, the sublevel set $f_t := \{x \in P | f(x) \leq t\}$ including all grayscale values below the threshold, produces a series of binary images $f_t \subseteq f_{t+1}$. This is the sublevel set filtration for $t = 0, 1, ...255$. The homological critical values in this set, represented in this 1-D case by maxima and minima in the function are

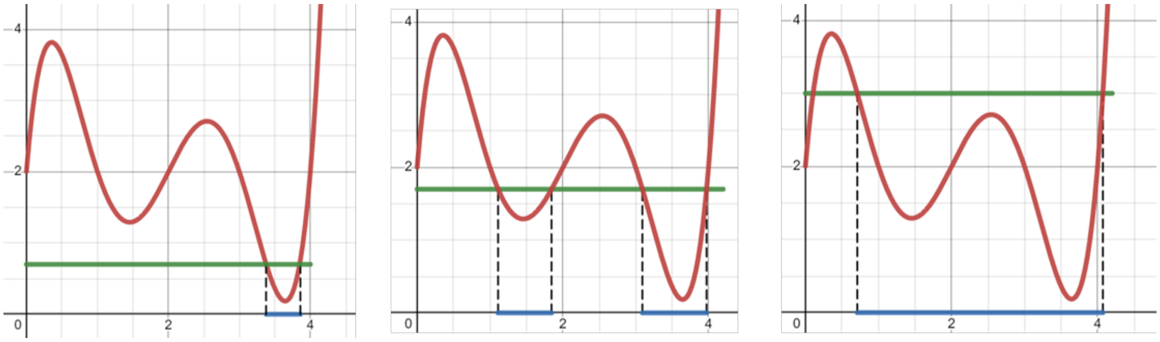

Figure 2.2: Sublevel set thresholding
A threshold (green) moving through a function causes the emergence of features (blue) within the set. Homological critical points are points where these features (blue) are created or destroyed

places where homology groups change. Informally, these critical values are defined as either a “Birth” coordinate or “Death” coordinate for a given feature, depending on whether the critical value represents a point where a new Betti feature first emerges in the set (birth), or the merging of two previously existing features occurs, where by convention the shorter lived feature is considered to have “died”. A feature’s *lifespan* is defined as the difference between an individual feature’s birth and death values in the set.

2.3 Persistence Diagrams

Plotting all “Birth” versus “Death” coordinates yields a *persistence diagram* that represents the topology of an image.

In order to reinforce intuition of a persistence diagram’s connection to image geometry, note that the two outlying points on the left of the persistence diagram correspond to the dark points at the top of the image to the right in Fig. 2.3. Additional information is contained in the spread of different Betti features along the diagonal. During the process of thresholding, connected components are likely to

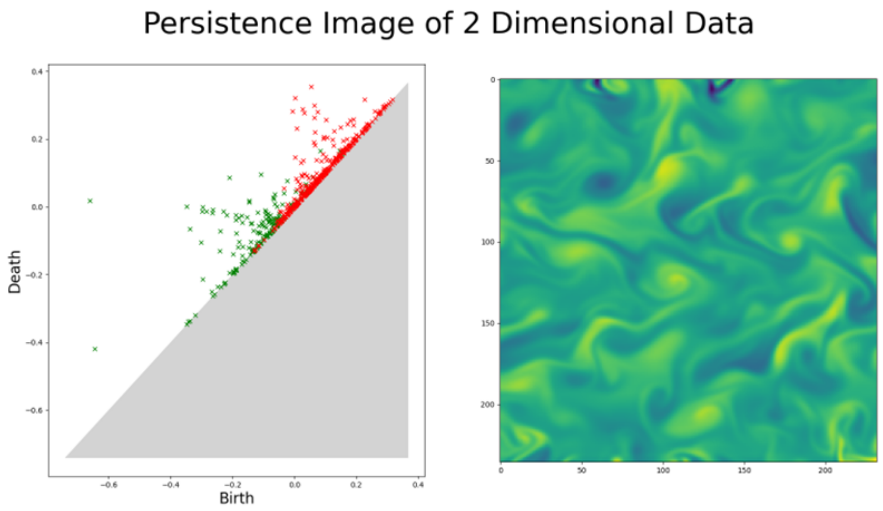

Figure 2.3: Persistence Diagram (left) corresponding to 2-D image (right). On left, β_0 features represented in green, β_1 features represented in red.

appear or be birthed before loops are, leading to the clustering among persistence points corresponding to the same Betti features.

2.4 Wasserstein Distance

The Wasserstein Distance is is a metric measuring the distance between distributions, in this case the distance between two persistence diagrams. It represents the "optimal transport plan" between two sets of points through a minimization of the sum of distances [5]. It can be conceptualized as the minimum total distance required to move one set of points in a distribution to a separate set of points. This minimization defines a scalar value that represents the topological similarity of two data sets. For the purposes of this project, the Wasserstein distance is taken separately for each level of homology. The Euclidean norm of these distances yields an overall Wasserstein persistence difference. Any mention of the Wasserstein distance in this thesis uses this version of the metric.

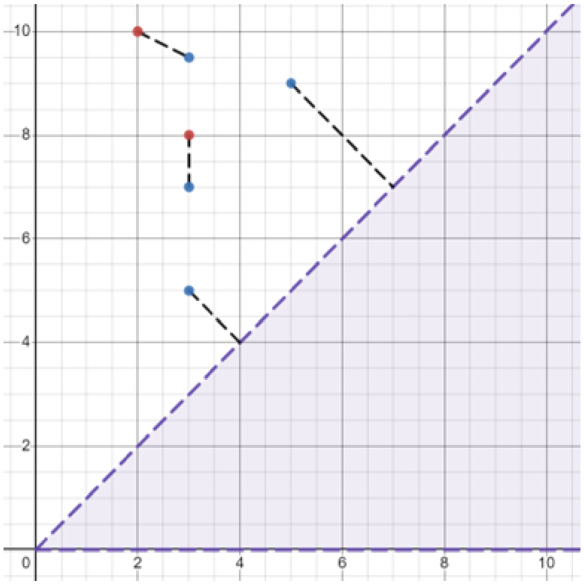

Figure 2.4: Wasserstein Distance
Minimized distance between two separate sets of points (red and blue) is represented by black dotted lines. By convention, points with no corresponding partners in the blue set are matched to a point on the diagonal. The total length of black segments is the scalar Wasserstein Distance.

2.5 Bottleneck Distance

The *bottleneck distance* is a different metric which is calculated with an alternative pairing to the Wasserstein distance. The scalar value of this metric is the longest necessary edge length required in a perfect matching, or one where all points are covered. Computationally, this is used more often in topological analysis than the Wasserstein distance because it trades sensitivity for computational time. However, for the purpose of this project, this lower sensitivity becomes an important advantage. Because the Bottleneck distance only includes the longest edge length, the introduction of a significant amount of short-lived features spread along the diagonal, as would be expected in most common types of noise, does not have a significant effect on the Bottleneck distance. These metrics are ways to quantify the changes we see within physical systems.

Chapter 3

Physical Systems and Turbulence

Topology has applications in a wide range of physical systems which have associated point cloud data. Topology is used to investigate systems from ice core scans to atmospheric currents [3]. There are many physical properties of plasmas, and turbulence in particular, that make magnetohydrodynamic systems interesting for the application of topological data analysis. Specifically there are properties of simulations which are conducive to investigation using TDA, as well as characteristics of turbulence that make TDA a worthwhile subject of investigation.

3.1 Magnetohydrodynamic Simulation

Magnetohydrodynamics (MHD) is the combination of hydrodynamics and electrodynamics which are used to model a large number of electrically conductive fluids, and is good for modelling the large-scale magnetic behavior of plasmas. Specifically, MHD is most applicable when plasmas are highly collisional, have low resistivity, and the time scales and length scales are much greater than the ion gyration time and Larmor radius. This makes MHD an area of interest for research of Driftwave Turbulence and large scale dynamics within future magnetic confinement fusion devices [6].

3.2 Modified Hasegawa Wakatani

The Modified Hasegawa Wakatani MHD model simulates electron density in plasmas [7] and is used to produce the 2-dimensional grayscale turbulence images used in this thesis (an example of which can be seen in Fig. 1.2). The Modified Hasegawa Wakatani model is a pair of coupled differential equations that, along with continuity equations and boundary equations describe the motion of electrons in a free electron gas and is used to approximate the physical setting of the edge region of a tokamak plasma [8]. These equations are

$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\phi - n) - \mu \nabla^4 \zeta \tag{3.1}$$

$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\phi - n) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n \tag{3.2}$$

Here, ζ is the vorticity, n is the electron density, ϕ is the charge density, μ is a parameter governing diffusion rates, and $\{a, b\}$ is the canonical Poisson Bracket Operator. In these simulations x and y are in the radial direction within the core of the plasma, with y being in the direction of the magnetic gradient. Some of the assumptions that are used in these simulations are periodic dimensions, cuboid geometry, and cold ions. Parameters of note that were varied at the beginning of the simulations were the density gradient drive $\kappa = -\frac{\partial}{\partial x} \ln(n_o)$, and α or the adiabaticity which governs (in 2D) collisionality, and acts as a parameter which can be adjusted to reach different flow regimes in the simulations. While it is possible to use κ to drive the change in the flow regime, collisionality presents a more convenient parameter, as its end behaviors are well understood. As $\alpha \rightarrow 0$, zonal flows become entirely suppressed, while as $\alpha \rightarrow \infty$, the model becomes fully dominated by zonal flows. Numata et al. [9] identify distinct dynamic flow regimes in the model, turbulent, transitional, and zonal. By considering the fraction of energy contained within zonal flows in an image $E_z := \frac{1}{2}\int(\frac{\partial\langle\phi\rangle}{\partial x})^2$ to total kinetic energy $E_k := \frac{1}{2}\int(\nabla\phi)^2$, they were able to determine how dominant zonal flows were in a given system. Their results determined that a system is zonal if $\frac{E_k}{E_z} > 0.9$, turbulent if $\frac{E_k}{E_z} < 0.2$, and transitional in between, i.e. $0.2 < \frac{E_k}{E_z} < 0.9$. These results are compared to the results of topological metrics within this thesis.

3.3 Turbulence Scales

Turbulence additionally has several characteristics which lend themselves especially well to topological data analysis. Specifically, turbulence is composed of vortices of different scales. The largest vortices are driven by the mean flow of the medium, and are a result of the geometry and collisionality of the fluid in question. These vortices drive smaller vortices in the inertial subrange, where they are no longer dependent on the geometry of the fluid, but instead exhibit energy decay to smaller and smaller scales. At the smallest feature scales, the viscosity of the fluid dissipates the energy contained in these vortices. This is why a high number of features is characteristic of systems with lower collisionality (and therefore a turbulent flow), where the dissipation scale occurs at lower energies, allowing for a larger inertial subrange [10]. Topological data analysis contains a variety of tools which can quantify the characteristics of turbulence which are visually intuitive, yet mathematically difficult to calculate. This will be most relevant as we move from simple models to more complete plasma simulations.

Chapter 4

Application to Models and Simulation

Topological Data Analysis has broad applicability in all sorts of point cloud and image data. In this chapter, single feature models demonstrate the limits of smoothing algorithms in approximating pre-noise topology. Then, alternative topological metrics to the number of features are examined in the presence of noise in order to determine if there are noise-stable alternatives to previously used metrics.

4.1 Noise in a Single Feature Model

This Thesis examines the effects of real life noise on point cloud data. To represent common noise expected in camera data, this project uses a Poisson distribution. This works well for gray-scale data, but runs into problems in binary images, or images where pixels can have one of two discrete values. A Gaussian distribution is used instead for binary images, with a mean value set as halfway between the binary values. In order for this approximation to remain relevant, binary images are defined as having values of either $\begin{cases} p_{m,n} = 0 \\ p_{m,n} = 255 \end{cases}$ for pixel (m, n) in the pixel grid. For grayscale images (represented by colored heatmap images for readability), Poisson noise is applied, where random values from the distribution are generated for each pixel based on the intensity of said pixel. The topology of a noisy image and its pre-noise image are then compared topologically by using the modified Wasserstein distance and Bottleneck distance metric. Turbulence is defined by the energy cascade, where energy is generally transferred from large spatial scale features to smaller scale features until those features reach dissipation scales. This means that features of all spatial scales are represented [10]. This is difficult for standard smoothing algorithms like a Gaussian blur to deal with, as smaller features can easily be smoothed out and important information can be lost. Therefore, it is important to identify the limits of algorithms at the extremes of feature scales. Two different single feature models provide the simplest cases to examine the behavior of topological metrics at logical extremes.

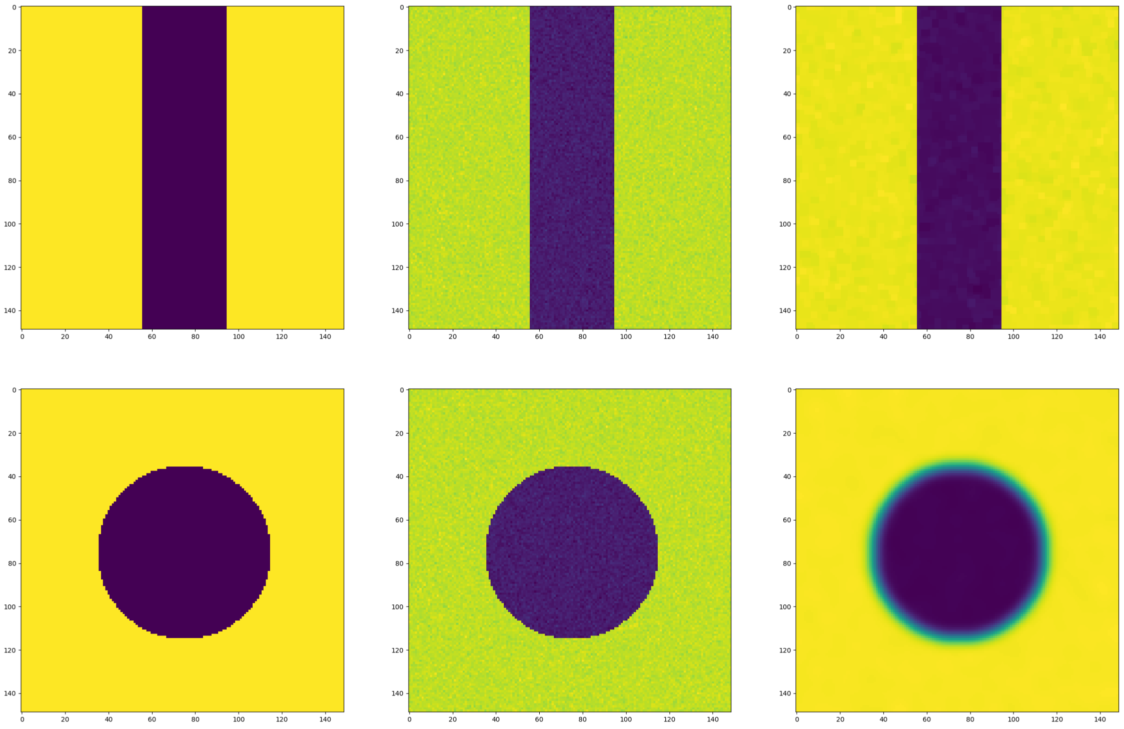

Figure 4.1: Stripe and Circle Single Feature Models
Each model from left to right has a
1) Noiseless image 2) Noisy image 3) Post-Process image.

A binary model with a single stripe is the topologically simplest single feature model. The circular model adds complexity in the form of vertex-to-vertex connections between pixels, which represent a problem for cubical homology within grid point cloud data. Thus the circular model becomes more relevant to real world data at the expense of some topological simplicity.

4.1.1 Processing Algorithms- Opening/Closing

Opening and *Closing* are morphological dual operations that use repetitions of erosion and dilation to attempt to restore noisy images. In opening, the input image is first eroded then dilated, while in closing these processes are reversed. Erosion is the process where each pixel is assigned the minimum pixel value of the set of pixels in its spatial "neighborhood" or kernel. For dilation, the maximum value is assigned instead [11]. Opening and closing an image is sometimes effective in removing small

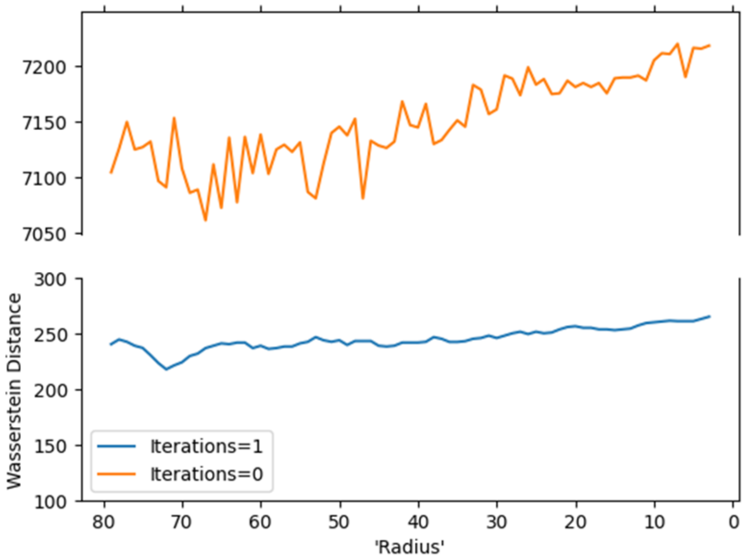

Figure 4.2: Opening and Closing
The curve at the top of this figure (Iterations=0) represents the Wasserstein distance between persistence diagrams of a pre-noise and post-noise image. The bottom curve (Iterations=1) represents the same relationship, but with processing applied to the post-noise image. In both cases the curve is feature scale independent.

scale pixel-wise noise and is often used in that application. In this thesis, a kernel is a 3×3 box of pixels. The single stripe model in Fig. 4.1 is shown in three states: noiseless, with noise applied, and then with noise and opening/closing applied. The Wasserstein distance can be utilized as a metric to determine the topological similarity between the first and third images in this series. A perfect topological match between two images would correspond to a Wasserstein distance of zero (note that two very different images can be topologically identical).

In Fig 4.2, opening then closing removes orders of magnitude from the Wasserstein distance demonstrating a reduction in the level of noise in the image. The relative consistency of the line as the scale ("radius") of the stripe is varied, implies that the spatial scale of relevant features has a minimal impact on the applicability of this metric. However, while this metric is effective for targeting noise over topology, the final effect of the smoothing may not be strong enough to reduce the effects of noise on topological data for an image. However, plotting the Wasserstein distance while varying the feature scale demonstrates that opening/closing is effective at removing pixel-wise additive noise independent of feature scale.

4.1.2 Processing Algorithms- Gaussian Blur

A Gaussian blur, as used in the circle model of Fig 4.1, is widely used to reduce image noise at the expense of image detail. Each pixel's new value is set to a weighted average of the values in the pixel's neighborhood, in this case with weight determined by a 2-dimensional normal Gaussian distribution,

$$f = \frac{1}{(2\pi\sigma)^2} e^{-\frac{(x-\bar{x})^T(x-\bar{x})}{2\sigma}} \tag{4.1}$$

The width of this weighted Gaussian is determined by σ (sigma), and the vector x represents the position of the pixel being assigned a new value, with $\bar{x}$ being the value of said pixel [12]. This also implies that by choosing too large a value for σ, it is possible to wash out genuine topology and not just noise artifacts.

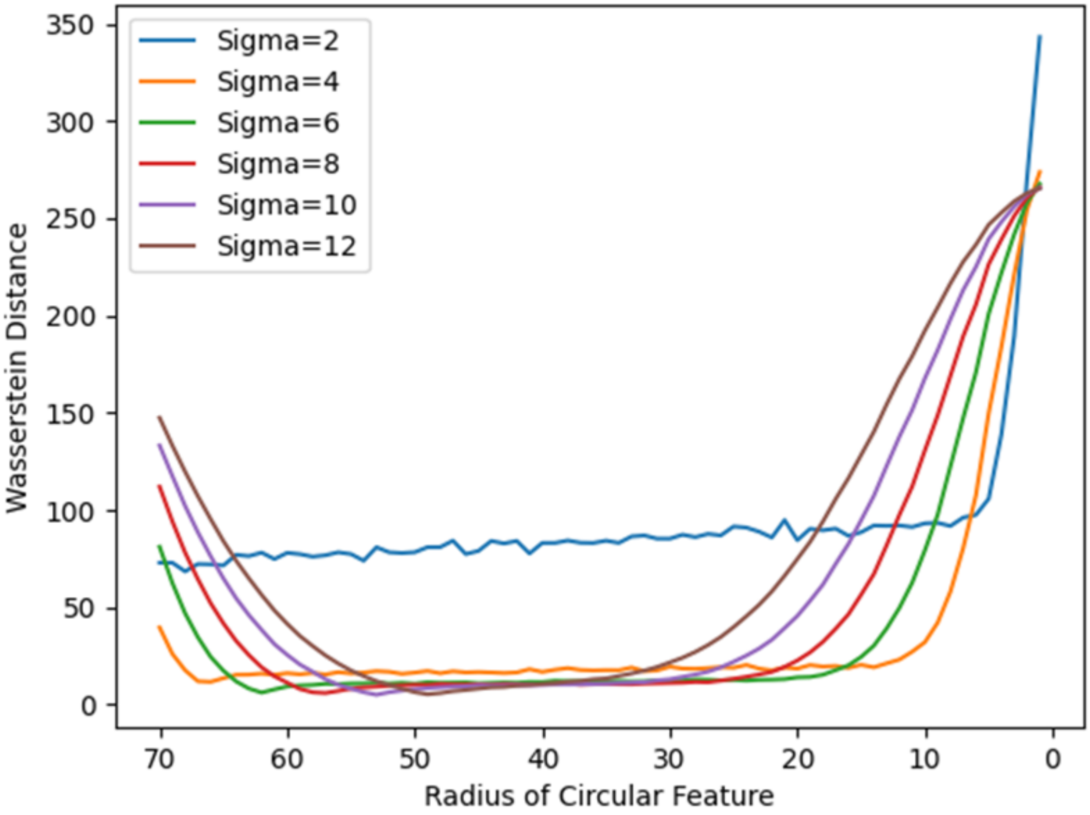

Figure 4.3: Gaussian Filter
Each curve represents how well a Gaussian filter can approximate the pre-noise topology of a noisy image at different feature sizes. Each curve represents a different blur strength based on the σ of the Gaussian distribution.

This embeds inherent limits into the utility of a Gaussian blur, as the blur will be less effective when the scale of noise approaches the scale of the smallest features in an image. Figure 4.3 demonstrates the scale dependence of the effectiveness of a Gaussian blur. In the figure, this limit appears for both small and large values for the radius of the feature, since as the feature scale grows dominant, the background and feature can be viewed as having flipped places. Additionally, as σ is increased, the range of effectiveness for a Gaussian filter decreases, as larger features are able to be smoothed out. However, a higher σ is also associated with a lower Wasserstein distance within this range where a blur is most effective, and thus a better approximation of a noiseless image.

4.2 Application to MHD Simulations

Single feature models give good approximations of the way that topological data may react to the influence of noise. The next important step is to examine how more complex magneto-hydrodynamic simulations react to the addition of noise. By varying the collisionality (α) parameter in the modified Hasegawa-Wakatani simulation module of BOUT++, it is possible to drive different simulation outcomes. For the purposes of this thesis, simulations are run for 4000 timesteps to allow for the effects of input parameters to propagate and yield the desired end behavior. The effect of collisionality (α) on the state can be seen in Figs. 1.1 and 1.3. This set of generated plasma images is used as the basis for further investigation.

4.2.1 Gaussian Filter

The transition between turbulent and zonal flows is characterized by changes in the number of topological features, but also visually by the spatial scale of features. As a result of this variance, images react differently to a Gaussian blur depending on the regime. This presents an alternative methodology for the characterization of images by flow type, as can be seen in Fig. 4.4. By using the Wasserstein Distance to compare, topologically, an image before and after a Gaussian blur is applied, differ-

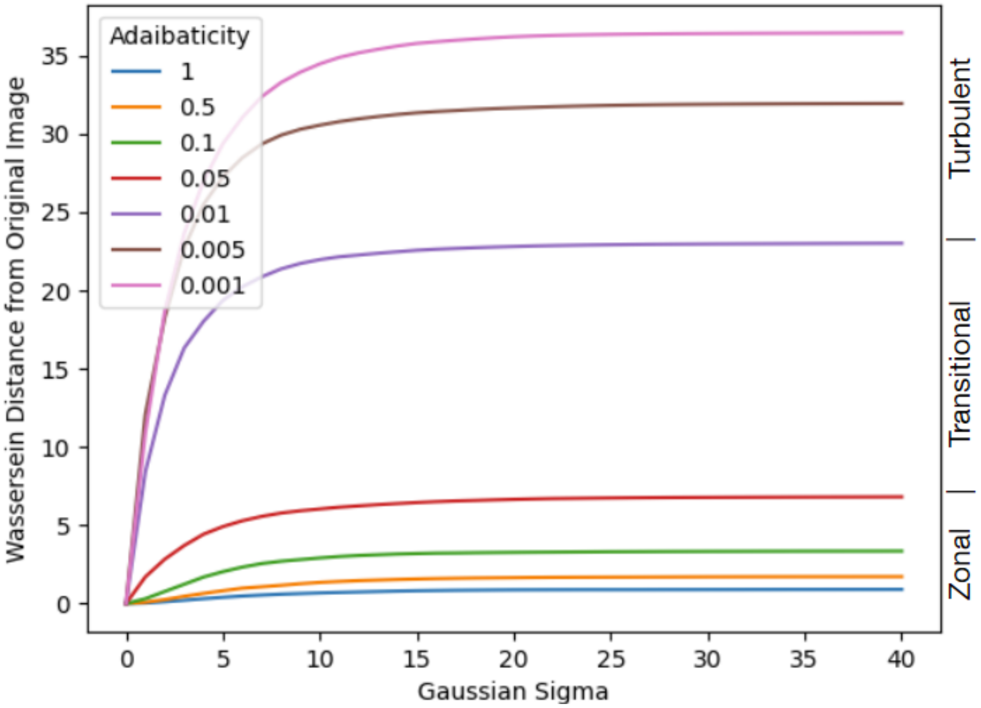

Figure 4.4: Effect of Gaussian Blur
Wasserstein distance plotted against σ (for different values of adiabaticity) reacts differently to a Gaussian filter based on the adiabaticity. This visible separation lines up with the different states seen in Fig. 1.1, as can be seen on the labels on the right.

ent flow regimes (characterized by initializing parameter adiabaticity) seem to lose topological data at different rates. The information loss caused by a Gaussian filter differs not only based on the flow regime, but also in that the reactions of different images represent an alternative to categorize those regimes with a metric other than the Number of Features. Intuitively, as $\lim_{\sigma \to \infty}$, the image approaches homogeneity. This approaches the Wasserstein distance becoming an equivalent metric to the total lifespan of persistent pairs. This is because the set of persistent pairs for a homogeneous image is the null set, and therefore this Wasserstein distance is equivalent to minimizing all points to the diagonal.

4.2.2 Noise Stability of Filter Methodology

In real-world applications this methodology, however, is still unstable in the presence of noise. Figure 4.5 is a simplification of the end behavior of Fig. 4.4, showing the Wasserstein distance between an image and its homogenous average ($\lim_{\sigma\to\infty}$).

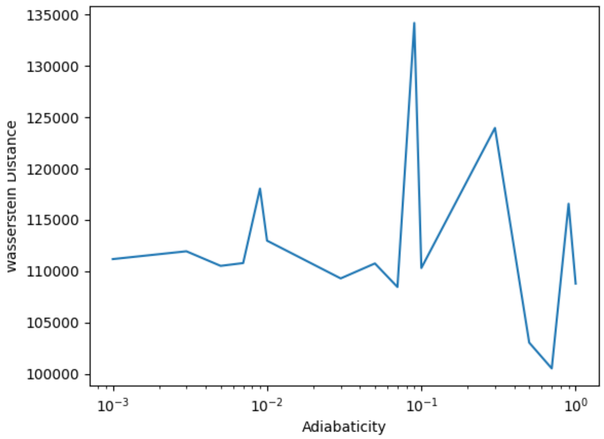

Figure 4.5: Gaussian Blur Only
Curve represents the Wasserstein distance between a noisy image and its homogeneous average as a function of adiabaticity.

Specifically in Fig. 4.5, these images have standard Poisson noise applied to them. Ideally, this would mirror the transition seen in Fig. 1.1. However, noise dominates and this metric shows promise in differentiating flow types. The Wasserstein distance in this case is susceptible to the noise introduced by an application of Poisson noise. Because the Wasserstein distance depends on a distance minimization between persistent pairs and, in this case, the diagonal, the Wasserstein distance is significantly dependent on the number of features. This, as established previously, is unstable in the presence of even minor noise.

4.2.3 Bottleneck Dependence

Fortunately, while the number of features is noise unstable and difficult to use in this context, the stability of Wasserstein distances is more complex. More specifically, the Bottleneck distance- a special case of the Wasserstein distance- is stable under the introduction of noise. The *Cohen-Steiner stability theorem* for topology states that in the presence of noise $\epsilon_{m,n} \in \mathbb{R}$ with $|\epsilon_{m,n}| < \epsilon$, the maximum Bottleneck distance between two persistence diagrams P and $P + \epsilon$ is at most the scale of ϵ [4]. This result implies that the bottleneck distance is a potential noise-stable alternative to a full Wasserstein Distance.

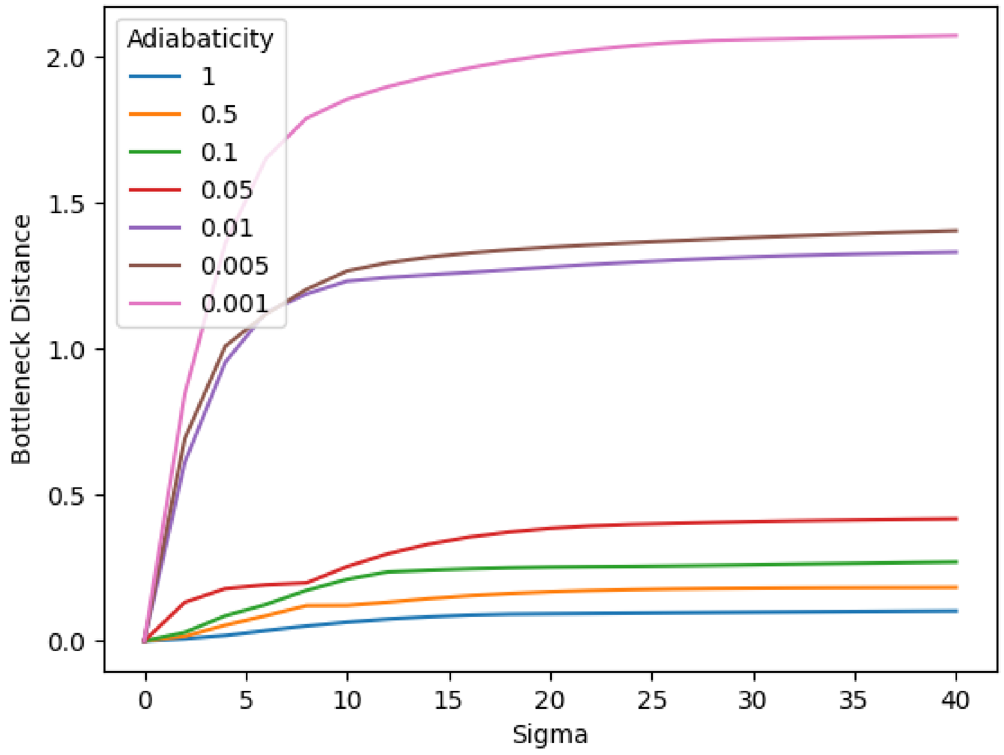

Figure 4.6: Bottleneck Distance Only
Applying the Bottleneck distance in place of the Wasserstein distance from Fig. 4.4, separation still occurs, and the end behavior remains similar.

Figure 4.6 implies that the Bottleneck distance contains the same information as the Wasserstein distance with a loss of sensitivity at some levels. However, the general end behavior of each image remains the same.

4.2.4 Maximum Lifespan

Applying a Gaussian filter with $\lim_{\sigma\to\infty}$ approaches a fully homogeneous image, which has no non-infinite persistence pairs. Therefore, as the filter approaches this limit, all persistence points in a set are paired to a point on the diagonal. Therefore this bottleneck distance is equivalent to measuring the maximum lifespan an image.

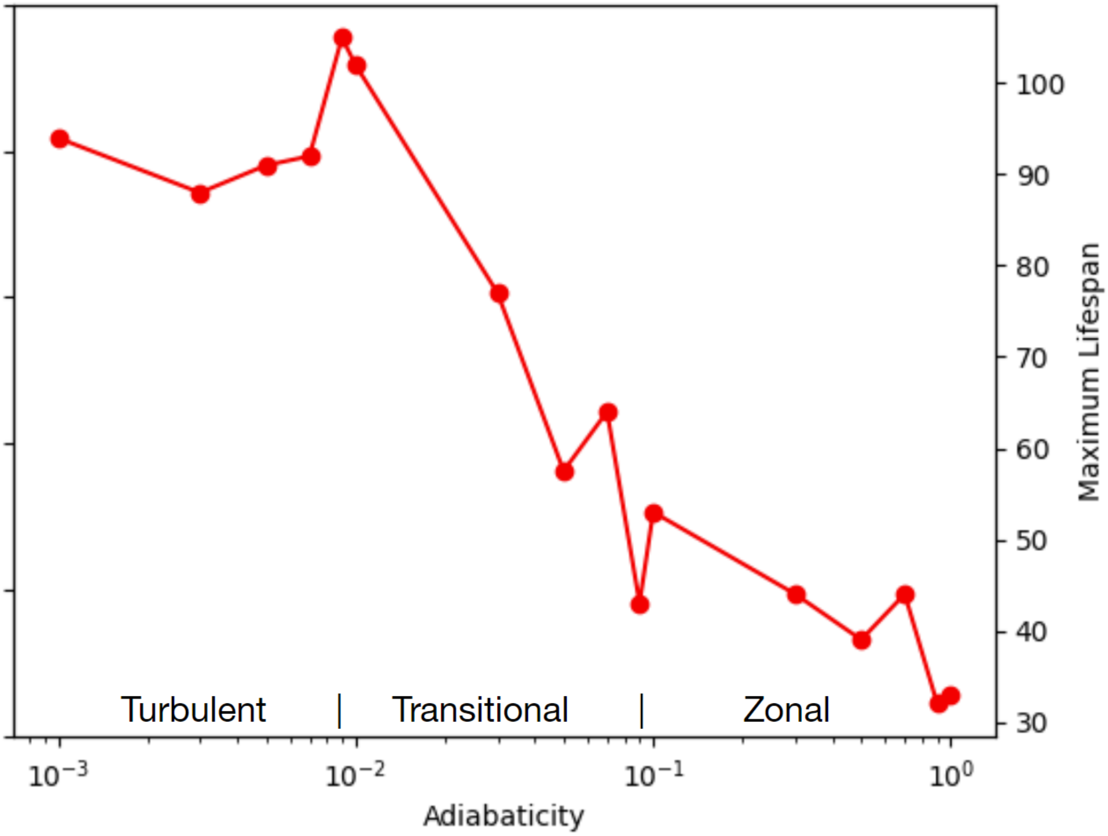

Figure 4.7: Transition Captured with maximum lifespan
Plot of the maximum lifespan within the topology of each image across a transition. This curve demonstrates the same trend as seen with the number of features $\langle N_f \rangle$.

And indeed, replacing the Wasserstein distance in Fig. 4.5 with the maximum lifespan, the expected transition reemerges in Fig. 4.7.

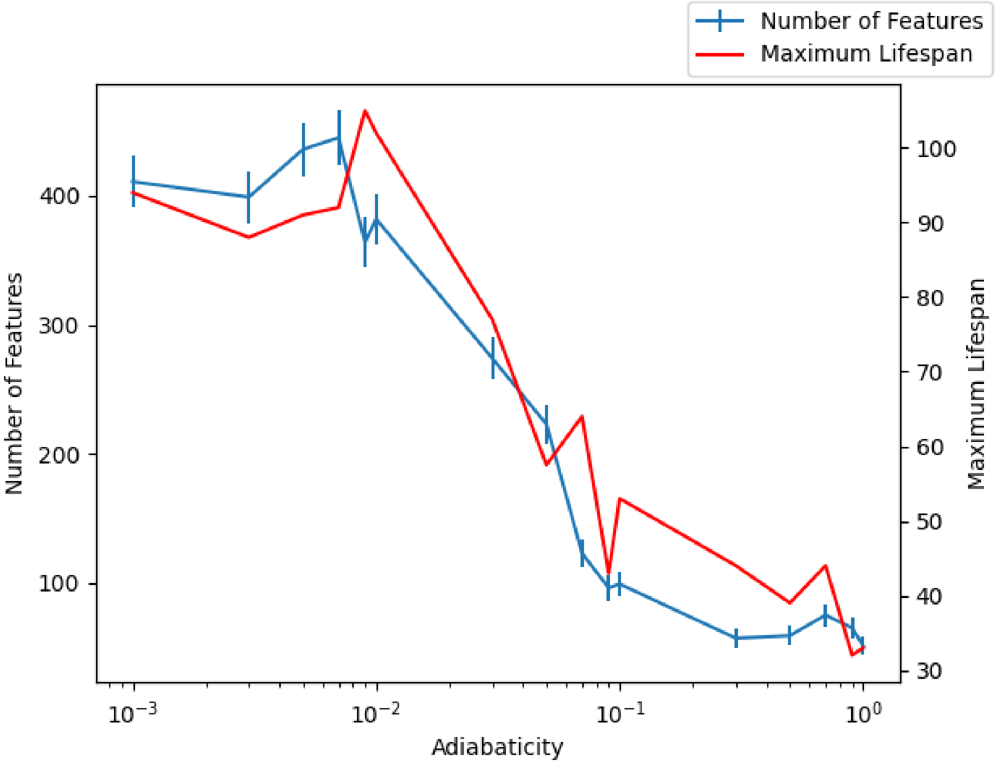

Figure 4.8: Maximum lifespan vs number of features
Both the number of topological features (blue) and the maximum lifespan (red) of topology in an image follow the same trend across a transition.

Comparing the transition captured by Number of Feature methodology vs maximum lifespan methodology (Fig. 4.8) demonstrates that maximum lifespan has potential as a noise-stable alternative to the number of features.

4.2.5 Noisy Maximum Lifespan

This metric is significantly more stable under the influence of noise than the number of features. Because this is essentially a bottleneck distance, the stability theorem applies. Figure 4.9 shows how the bottleneck distance metric reacts to noise, and compares that with a noiseless transition. While this introduces a shift in the diagram, the transition is still identifiable, and critical points where the transition occurs are also still preserved.

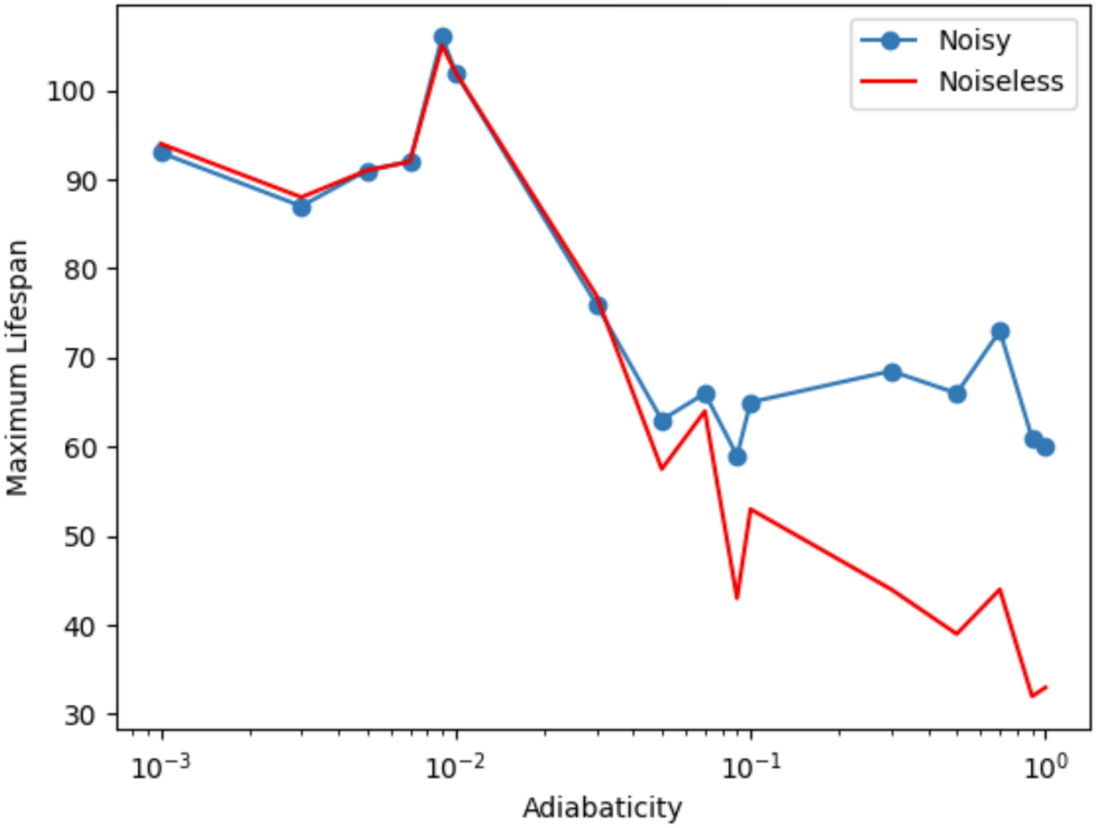

Figure 4.9: Maximum Lifespan Stability
The maximum lifespan under the influence of noise is compared to the maximum lifespan in the noiseless case. The relation is stable under the effects of background noise, and flow regimes can still be identified. Compare to Fig. 1.5

While there is some loss of information, the general structure of the dependence is preserved. This metric is, of course, limited by the stability theorem, requiring that the noise smaller than the signal under investigation. However, comparing the results of Figs. 1.5 and 4.9, this is significantly more stable to introduced noise. In the zonal regime the difference is largest since the bottleneck distance is less stable when the distance is smaller. Conceptually, this can be imagined as the zonal flows appearing more “turbulent” as a result of the type of noise applied.

Chapter 5

Testing TDA on experimental fast camera data

The LArge Plamsa Device (LAPD), located at UCLA, is a cylindrical magnetized plasma device which produces a plasma with a length of 17 meters and a diameter of 60 centimeters. It is ionized using a barium-oxide coated nickel cathode that causes a slow drift of the plasma in the ion diamagnetic direction. Biased obstacles inserted into the core plasma is shown to achieve more continuous control of crossfield flow. In the absence of these biased limiters, the plasma will continue to rotate slowly in the ion diamagnetic drift direction, but biasing the limiters placed in the device with respect to the cathode drives flow in the electron diamagnetic drift direction instead. Importantly for the purposes of this project, during this process, the initial flow is reduced and brought to near zero flow and flow-shear, and is ultimately reversed. This represents a transition state between two flow states [13].

In the video under consideration for the purpose of this section of this thesis, taken from one end of the LAPD tube, records the electron density of a plasma. In this video, a clear transition occurs in the vicinity of frame 50. The goal is to be able to identify when this transition occurs topologically, and whether there is a topological difference between pre and post transitional states.

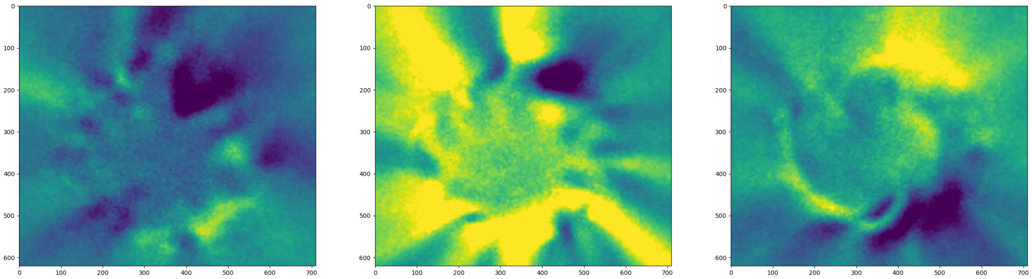

Figure 5.1: Frames from LAPD Video
From left to right, grayscale frames from before (frame 36), during (frame 48) and after (frame 60) transition in shearing turbulence video captured by fast camera at LAPD.

5.1 Number of Features

The number of features within each frame of the LAPD video is an insufficient metric for identifying transitions.

Figure 5.2 plots the number of features in each frame over time. The curve in this figure demonstrates that we are working with a type of noise which does not resemble the shot noise which was previously used to model expected camera noise. Specifically, this introduces an unexpected noise signal on a 12 frame cycle. However, on a frame-by-frame basis, this noise appears to be both small in scale and additive, even if it is not consistent over the course of the video. Under these conditions, the dual morphological operations of opening and closing would be able to help reduce this

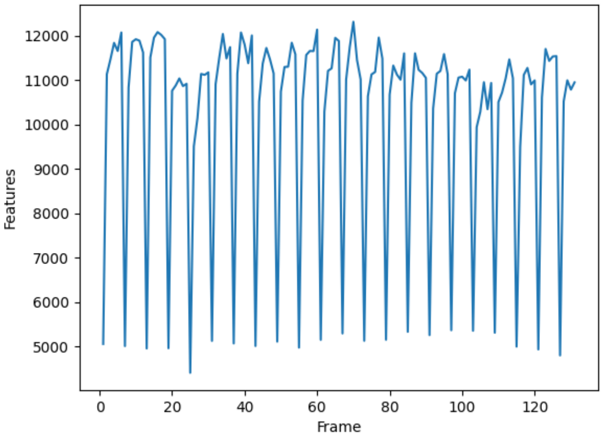

Figure 5.2: Pre-process LAPD data
Effect of noise on number of topological features per frame. Noise

noise. Alternatively, since the maximum scale of this applied noise is small relative to the feature scale, the stability theorem for Bottleneck distances means the maximum lifespan metric may also be applicable.

5.2 Opening/Closing

Opening and Closing are a reasonable image processing methodology for approximating pre-noise images from noisy data with small scale noise. Figure 5.3 demonstrates that when opening/closing is applied to each frame in the LAPD camera data, the number of features lose a lot of their fluctuations. This methodology does indicate

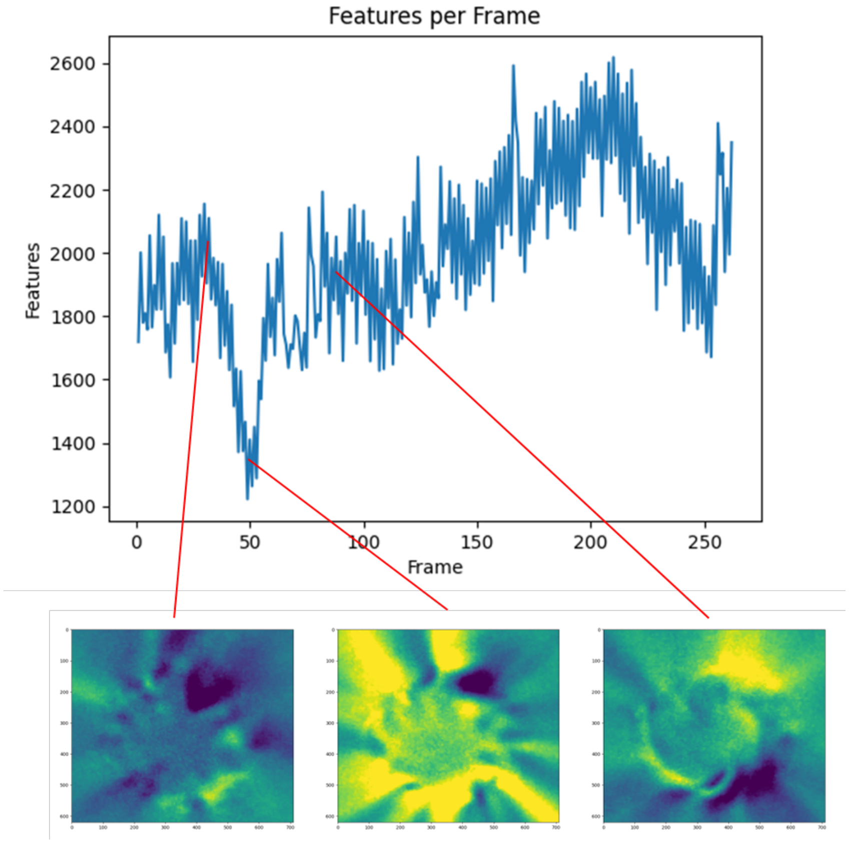

Figure 5.3: Number of Features with Opening/Closing
Opening/Closing operations remove “spikes” seen in Fig. 5.2 to allow the visualization of the topological change around frame 50.

a transition occurring near frame 50, therefore demonstrating that opening/closing in the presence of pixel-wise noise preserves topological structures of interest for this system, and allows the extraction of topological information from previously noisy data.

5.3 Maximum Lifespan

The maximum lifespan in each frame of this video also characterizes the frames where the expected transition occurs. Plotting the maximum lifespan of features in each image once again allows the emergence of a significant signal around frame 50.

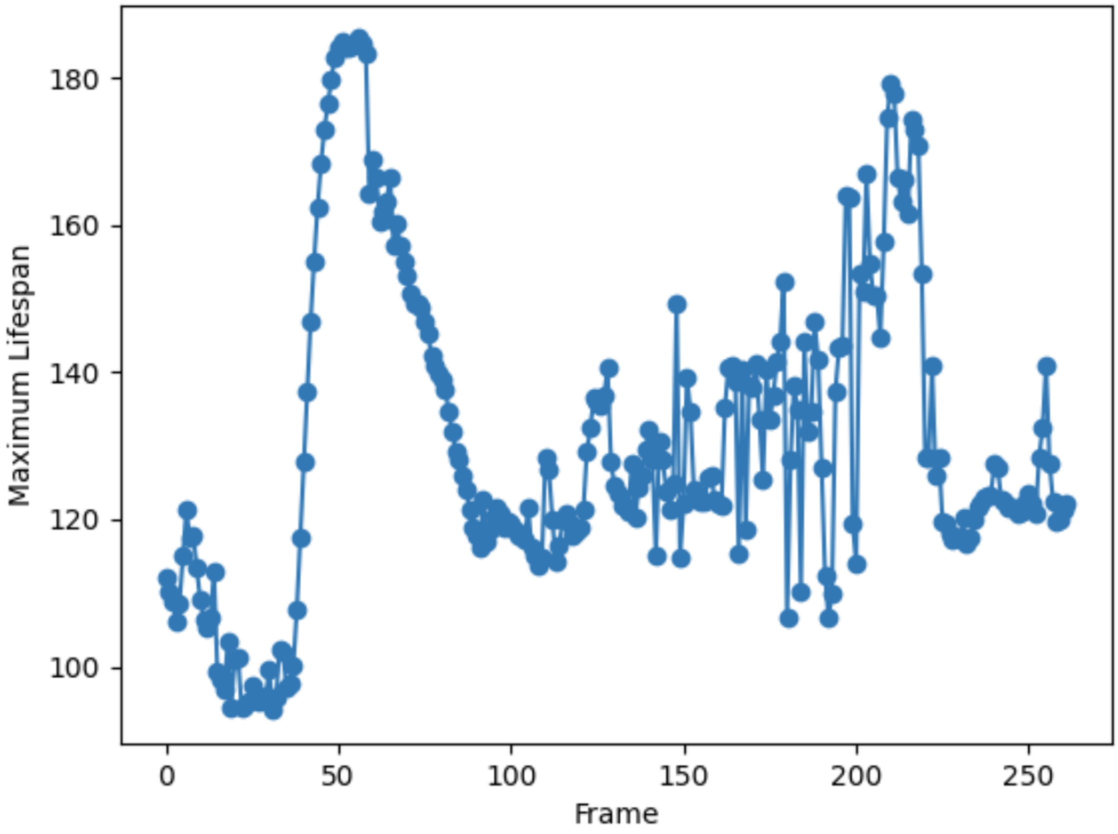

Figure 5.4: Real World Lifespan Diagnostic
The maximum lifespan of each feature demonstrates interesting topological change occurring around frame 50, as expected based on the known transition.

5.4 Interpretation of Results

While both of these methods have established interesting topological changes around frame 50, following expectations of when a transition occurs within the LAPD video, there appears to be another topological feature of interest. Specifically, both of these topological metrics appear to show another type of slower topological change occurring in the range of frame 200. Considering the maximum lifespan as a special case of the Bottleneck distance, which is fully independent of the number of features, the maximum lifespan and number of features are fully decoupled metrics. This implies that the shared shift in topology which is observed at frame 200 is likely physical in origin, or is at least inherent to the camera data itself. This may or may not imply some more subtle transition occurring at this point, and more research is required to identify what that change may be, or if this topologically has any implications at all.

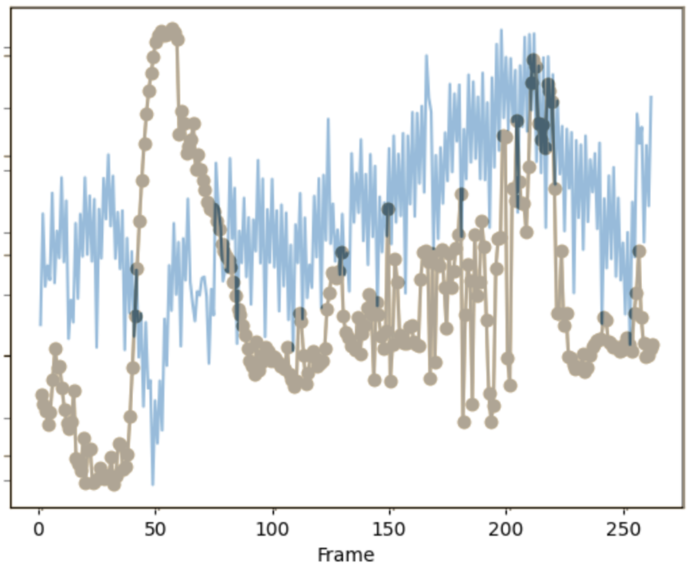

Figure 5.5: Overlapping results from post-processing number of features (fig. 5.3) and lifespan (fig. 5.4)
Both demonstrate a topological shift during known transition at around 50 frames, with evidence of interesting behavior around frame 200.

Chapter 6

Conclusion and Outlook

Motivated by the goal of utilizing topology as a metric for identifying flow states within real life and noisy camera data, this thesis examines several physical systems of interest. The effects of noise on metrics which measure changes in topological behavior were measured in all three of these applications: single feature models, magneto-hydrodynamic simulation, and real world fast camera data.

Building on the results of previous work which determined that the homology of patterns give a reasonable description of when transitions occur, this thesis demonstrates how these metrics are susceptible to common real-world noise. This demonstrates that there are alternative homology methods that provide noise-resilient quantitative descriptions of transition, which match with previous results to identify transitional regimes. Specifically, by characterizing the effect of a Gaussian blur on the Wasserstein distance of differing regimes, continuous improvements to methodologies demonstrated that the maximum lifespan of features within a topological image is a viable and potentially noise stable way to quantify transition.

To test this potential noise-stable methodology, these metrics were applied to real world fast camera data from LAPD, which demonstrated that the maximum lifespan was indeed more stable to noise than the number of features, and demonstrated a clear transition in the camera data, and even potentially captured additional dynamics which could be the subject of future investigation. However, since the maximum lifespan is a modification of the bottleneck distance, this metric trades sensitivity for resilience against noise. Therefore there are still concerns that this metric may not be stable under different noise conditions, or that the methodology may not be delicate enough to capture more subtle transitions. This again could be the direction that future investigation takes.

Topology is a field where real world applications are just being applied to match the power of computational and mathematical discoveries in the field. Topology presents new opportunities for working towards both a greater understanding of turbulent systems, as well as potential applications of control systems and data analysis in the future.

Appendix A

Computer Programs

A.1 Simulation Lifespan Diagnostic Code

The following python code is used to produce figures like figure 4.9:

```
import MethodsMaster as gen
import matplotlib.pyplot as plt
import numpy as np
import random
import xarray as xr
import scipy as sp
import skvideo.io
import skvideo.utils
import skvideo.datasets
from scipy import fftpack, ndimage
plt.rc('font', size=10)

gridar2=[1,0.9,0.7,0.5,0.3,0.1,0.09,0.07,0.05,0.03,0.01,0.009,0.007,0.005,0.003,0.001]
#gridar2=[1,0.5,0.1,0.05,0.01,0.005,0.001]
gridar=gridar2[::-1]

fig=plt.figure()
y=[]
for i in gridar:
grid=np.loadtxt(f"./Simulation/ImRepoComp/BOUT_{i}.txt");grid=gen.normalize(grid)
g=gen.gaussblur(grid,100)
y.append(gen.BottleneckCompare(grid,g))
plt.plot(gridar,y)
plt.xlabel("Adiabaticity")
plt.xscale("log")
plt.ylabel("Maximum Lifespan")

print("CLEARLINE_____________________________________CLEARLINE",end="\x1b[1K\r")
save_name=input("Save with name: ")
fig.savefig(f"./{save_name}.png")
plt.close()
```

A.2 Real World Bottleneck Diagnostic

The following is the Python code that produces figures like figure 5.4:

```
import MethodsMaster as gen
import matplotlib.pyplot as plt
import numpy as np
import random
import xarray as xr
import scipy as sp
import skvideo.io
import skvideo.utils
import skvideo.datasets
from scipy import fftpack, ndimage

db=skvideo.io.vread(f"./shear_example.avi")[:int(524/2),20:640,60:770,0]

x=[];y=[]
for i in range(len(db)):
grid=db[i];grid=gen.convertimg(grid)
g1=gen.gaussblur(grid,100)
x.append(i)
y.append(gen.BottleneckCompare(grid,g1))

fig=plt.figure()
plt.plot(x,y,"o-")
plt.xlabel("Frame")
plt.ylabel("Bottleneck Distance")

print("CLEARLINE_____________________________________CLEARLINE",end="\x1b[1K\r")
save_name=input("Save with name: ")
fig.savefig(f"./{save_name}.png")
plt.close()
```

A.3 Methods Code

The following is the Python code that contains most of the general methods used in the production of graphics and analyzing data for this project. It is imported for all other code samples as **gen**:

```
/***********************************************************/

import numpy as np
import xarray as xr
from PIL import Image, ImageFilter
import random
import scipy as sp
import scipy.ndimage
import gudhi as gd
import gudhi.wasserstein as gdwas
import matplotlib.pyplot as plt
import matplotlib.animation as animation
```

```
from matplotlib.animation import FuncAnimation
#import cv2

#Only used for testing purposes
def openthedata():
  test=input("Simulation Info Desired (n, phi, vort): ")
  data=xr.open_dataset(f"Simulation/BOUT.dmp.0.nc")
  #info=data["info"].values
  data=data[test].values[len(data[test]["t"])-1,2:len(data[test]["x"])-2,0,:]
  return data

#Additive Gaussian Noise
def noisegauss(tempdata,sigmamod):
  returnable=np.zeros((int(len(tempdata[:,0])),int(len(tempdata[0,:]))))
  m=((np.amax(tempdata))*0.05)*sigmamod
  for i in range(int(len(tempdata[:,0]))):
    for j in range(int(len(tempdata[0,:]))):
      returnable[i][j]=tempdata[i,j]+random.gauss(tempdata[i,j],m)
  #print(f"Gaussian Noise Added, sigma={m}")
  return returnable

#True Poisson Noise
def noisepoisson(tempdata):
  returnable=np.random.poisson(tempdata)
  return returnable

#"Additive" poisson noise
def noisepoisson2(tempdata,lamb):
  returnable=np.zeros((int(len(tempdata[:,0])),int(len(tempdata[0,:]))))
  for i in range(len(tempdata)):
    for j in range(len(tempdata[0])):
      returnable[i][j]=tempdata[i][j]+np.random.poisson(lamb)
  return returnable

#normalizes out magnetic gradient along forst axis (useful for BOUT++ output)
def normalize(tempdata):
  returnable=np.zeros((int(len(tempdata[:,0])),int(len(tempdata[0,:]))))
  for i in range(len(tempdata[:,0])):
    returnable[i]=tempdata[i,:]-np.mean(tempdata[i,:])
  return returnable

#Creates an animation of input 3D array
def visualize(viddata):
  def AnimationFuncion(frame):
    dap=viddata[frame]
    plotte.set_array(dap)
    #print(frame)
    return plotte
  dap=viddata[-1]
  #Setting first frame(which is last frame to set colorbar to reasonable scale)
  Figure=plt.figure()
  plotte=plt.imshow(dap)
  plt.colorbar(plotte)
  #Animation drawn and saved
  anim=FuncAnimation(Figure,AnimationFuncion,frames=len(viddata),interval=200)
  FFwriter = animation.FFMpegWriter()
  output_path=input("Save animation with name: ")
  anim.save(f'{output_path}.mp4')
  plt.close()

def genperscalc(img1):
  ccimg1=gd.PeriodicCubicalComplex(top_dimensional_cells=img1,periodic_dimensions=[False,True])
  ccimg1.compute_persistence()
  pimg1=ccimg1.persistence()
  return pimg1

#Wasserstein distance between two images
##euclidean distance between individual sets of betti numbers
def wasserbetti(temppersistence):
  b0=[];y0=[];b1=[];y1=[]
  for i in range(len(temppersistence)):
    if temppersistence[i][0]==0:
      b0.append(temppersistence[i][1][0])
      y0.append(temppersistence[i][1][1])
    elif temppersistence[i][0]==1:
      b1.append(temppersistence[i][1][0])
      y1.append(temppersistence[i][1][1])
  return np.vstack((b0,y0)).T,np.vstack((b1,y1)).T
def WassersteinCompare(img1,img2):
  pimg1=genperscalc(img1);pimg2=genperscalc(img2)
  cimg10,cimg11=wasserbetti(pimg1);cimg20,cimg21=wasserbetti(pimg2)
  bimg1=gdwas.wasserstein_distance(cimg20,cimg10,order=1.,internal_p=2.)
  bimg2=gdwas.wasserstein_distance(cimg21,cimg11,order=1.,internal_p=2.)
  return np.sqrt(bimg1**2+bimg2**2)

def BottleneckCompare(img1,img2):
  pimg1=genperscalc(img1);pimg2=genperscalc(img2)
  cimg1=wasserbetti2(pimg1);cimg2=wasserbetti2(pimg2)
  return gd.bottleneck_distance(cimg1,cimg2)

#Standard wasserstein distance between two images
def wasserbetti2(temppersistence):
  b=[];y=[]
  for i in range(len(temppersistence)):
    b.append(temppersistence[i][1][0])
    y.append(temppersistence[i][1][1])
  return np.vstack((b,y)).T
def WassersteinCompare2(img1,img2):
  pimg1=genperscalc(img1);pimg2=genperscalc(img2)
  cimg1=wasserbetti2(pimg1);cimg2=wasserbetti2(pimg2)
  return gdwas.wasserstein_distance(cimg2,cimg1,order=1.,internal_p=2.)
```

```
def get_betti_lifespan(p):
  b=[];y=[]
  for i in range(len(p)):
    b.append(p[i][1][0])
    if p[i][1][1] != 'inf':
      y.append(p[i][1][1])
  return b,y
def lifespanvals(img1):
  pimg1=genperscalc(img1)
  b,y=get_betti_lifespan(pimg1)
  spans=[]
  for i in range(len(b)):
    if y[i]-b[i] != float("inf"):
      spans.append(y[i]-b[i])
  return spans

#Eucliduan distance between two images
def eucdistance(img1,img2):#Need to be the exact same scale
  returnable=0
  for i in range(len(img1)):
    for j in range(len(img1[0])):
      returnable+=(int(img1[i][j])-int(img2[i][j]))**2
  return np.sqrt(returnable)

#Retrieve persistence diagram for an image
def get_betti_pers(p):
  b0=[];y1=[];b1=[];y2=[];b2=[];y3=[]
  for i in range(len(p)):
    if p[i][0]==0:
      b0.append(p[i][1][0])
      y1.append(p[i][1][1])
    elif p[i][0]==1:
      b1.append(p[i][1][0])
      y2.append(p[i][1][1])
  return(b0,y1,b1,y2)
def persinfo(tempdata):
  p=genperscalc(tempdata)
  b0,y1,b1,y2 =get_betti_pers(p)
  lims = [np.amin(b0)-0.05,np.amax(b1)+0.05]
  fig=plt.figure()
  plt.xlabel("Birth",fontsize=20)
  plt.ylabel("Death",fontsize=20)
  plt.plot(b0,y1,'gx',label="Betti 0 Features")
  plt.plot(b1,y2,'rx',label="Betti 1 Features")
  plt.legend()
  plt.fill_between(lims,lims,y2=lims[0],color='#d3d3d3')
  return fig

#Returns number of topological features in an image
def numfeat(tempdata):
  cc=gd.PeriodicCubicalComplex(top_dimensional_cells=tempdata)
  cc.compute_persistence()
  p=cc.persistence()
  return len(p)

#Converts image from pixel data to greyscale image object
def convertimg(tempdata):
  tempdata=tempdata+abs(np.amin(tempdata))
  tempmax=np.amax(tempdata)
  for i in range(len(tempdata)):
    for j in range(len(tempdata[0])):
      tempdata[i][j]=int((tempdata[i][j]/tempmax)*255)
  tempdata=tempdata.astype(np.uint8)
  temp_image=Image.fromarray(tempdata)
  temp_image.convert("L")
  #print("Return type is now Image")
  return temp_image

#mode blur smoothing algorithm
def modeblur(tempdata,kern):
  tempdata=convertimg(tempdata)
  returnable=tempdata.filter(ImageFilter.ModeFilter(size=kern))
  return np.array(returnable)

#Gaussian blur smoothing algorithm
def gaussblur(tempdata,sigma):
  return sp.ndimage.gaussian_filter(tempdata,sigma)

#Median blur smoothing algorithm
def medianblur(tempdata,kern):
  return sp.ndimage.median_filter(tempdata,size=kern)

#Opening/Closing smoothing algorithm
def imopen(tempdata,d):
  return sp.ndimage.grey_opening(tempdata,size=(3,3))#structure=np.ones((3,3)))
def imclose(tempdata,d):
  return sp.ndimage.grey_closing(tempdata,size=(3,3))#structure=np.ones((3,3)))
def opencloseblur(tempdata,itera):
  d=(len(tempdata),len(tempdata[0]))
  for i in range(itera):
    tempdata=imopen(tempdata,d)
    tempdata=imclose(tempdata,d)
  return tempdata

#Normalize an image with added noise
def noisenormalize(img1,img2):# 1 has no noise 2 is noisy
  return img2-abs(np.average(img1)-np.average(img2))
```

Bibliography

[1] Garbet, X. (2006). Introduction to turbulent transport in fusion plasmas. *Comptes Rendus Physique*, 7(6), 573-583.

[2] Stanish Sage, "Counting Holes in Physical Systems - Applications of Computational Homology to Systems in Physics -" (2022). Undergraduate Honors Theses. William & Mary. Paper 1785. https://scholarworks.wm.edu/honorstheses/1785

[3] Adler, R. J., Bobrowski, O., & Weinberger, S. (2013). Crackle: The persistent homology of noise. *arXiv preprint arXiv:1301.1466.*

[4] Cohen-Steiner, D., Edelsbrunner, H., & Harer, J. (2005, June). Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry* (pp. 263-271).

[5] Dlotko, P. (2020). Cubical complex. In GUDHI User and Reference Manual (3.1.0). Retrieved from https://gudhi.inria.fr/doc/3.1.0/group__cubical__complex.html

[6] Wesson, J. A. (1978). Hydromagnetic stability of tokamaks. *Nuclear Fusion, 18*(1), 87.

[7] Dudson, Dudson, B. D., Umansky, M. V., Xu, X. Q., Snyder, P. B., Wilson, H. R. (2009). BOUT++: A framework for parallel plasma fluid simulations. *Computer Physics Communications, 180*(9), 1467-1480.

[8] Hasegawa, A., & Wakatani, M. (1983). Plasma edge turbulence. *Physical Review Letters, 50*(9), 682.

[9] Numata, R., Ball, R., & Dewar, R. L. (2007). Bifurcation in electrostatic resistive drift wave turbulence. *Physics of Plasmas, 14*(10).

[10] Falkovich, G., Sreenivasan, K. R. (2006). Lessons from hydrodynamic turbulence. *Physics Today, 59*(4), 43-49.

[11] Heijmans, H. J., & Roerdink, J. (Eds.). (1998). *Mathematical morphology and its applications to image and signal processing* (Vol. 12). Springer Science & Business Media.

[12] Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (1993). *Probability and statistics for engineers and scientists* (Vol. 5, pp. 326-332). New York: Macmillan.

[13] Schaffner, D. A., Carter, T. A., Rossi, G. D., Guice, D. S., Maggs, J. E., Vincena, S., & Friedman, B. (2012). Modification of turbulent transport with continuous variation of flow shear in the large plasma device. *arXiv preprint arXiv:1205.3325.*