Topological visualisation techniques to enhance understanding of lattice QCD simulations

Thomas, Dean P.

Topological visualisation techniques to enhance understanding of lattice QCD simulations

Dean P. Thomas

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

2018

Abstract

Lattice QCD is an established field of research in theoretical physics in the pursuit of understanding the strong nuclear force. Despite a long history dating back several decades [1] much of the existing literature relating lattice QCD and computer science focus on the technical aspects of computations. A survey of the use of visualisation techniques in lattice QCD initially returned relatively few results. However, a few examples were found that utilised established volume rendering techniques — showing potential for the use of visualisation in the domain.

This thesis presents several contributions based upon topological visualisation techniques with the objective of stimulating interest and discussion within the lattice QCD community. We feel the use of topology, in conjunction with display of observables using volume rendering techniques, presents a compelling case for inclusion of visualisation in the lattice QCD computation and analysis pipeline. Our decision to use topological visualisation techniques is largely influenced by the need for domain scientists to be able to summarise vast amounts of data using a quantitative approach. Interest in the use of topology as part of the visualisation process was initiated by early meetings with physicists where lattice observables were presented using isosurfaces. Since then regular meetings with active researchers in lattice QCD helped guide the work in reaching this objective.

In order to demonstrate the benefits of visualisation, and in particular approaches that rely on scalar topology, we present a range of case studies in this thesis. These demonstrate the application of various algorithms — beginning with an example of how connected contours, as opposed to isosurfaces, can split a scalar field into unique objects. We show how properties of observables can be computed directly using topological persistence measures and by using the meshes constructed from contour generation algorithms. We also show how more recent advances in multivariate topology can assist in the understanding of evolving lattice phenomena. Multivariate topology also presents a unique opportunity for studying the relationship between several fields existing on the lattice. The thesis concludes with the presentation of a custom tool developed specifically to serve the needs of lattice QCD.

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date ....

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

Acknowledgements

I would like to give special thanks to my supervisors Rita Borgo, Simon Hands, and Bob Laramee for their support and guidance throughout my Ph.D studies. In particular I am very appreciative of the countless hours spent discussing elements of theoretical physics in depth with Simon; Rita for her advice, knowledge, and enthusiasm for research and software engineering; and Bob for his encouragement at the beginning of my Ph.D and additional assistance in the writing up phase of my research. I would like to thank the College of Science at Swansea University for their sponsorship of my research studies, without which I would not have been able to pursue this work.

Additionally, I wish to extend my gratitude to the staff and research students in the Computer Science department at Swansea. Their friendship, humour, and intellectual conversation really helped motivate me in my research on a daily basis. I would also like to thank Dr.Hamish Carr for his advice and feedback on aspects of my research throughout my studies.

Finally, I would like to thank my family and friends. Without their continual support I would never have been able to finish this thesis.

List of supporting literature

Throughout the progression of this research a number of presentations and publications have been used to promote the use of computational topology in lattice QCD. The following details journal articles and conferences relating to this thesis.

Journal articles & conference proceedings

Thomas, D. P., Borgo, R., Carr, H., & Hands, S. J. (2017). *Joint Contour Net analysis of lattice QCD data* — This was presented at *Topology-Based methods in Visualization (TopoInVis) 2017*, as a result we were invited to submit an extended version of the paper for inclusion in a special issue of Topology-Based Methods in Visualization (*currently in review*). URL `https://arxiv.org/abs/1703.02488`.

Thomas, D. P., Borgo, R., Laramee, R. S., & Hands, S. J. (2017). *QCDVis: a tool for the visualisation of Quantum Chromodynamics (QCD) data.* In Computers & Graphics, 67, 115126. *DOI: 10.1016/j.cag.2017.07.006* — This paper was originally presented at the 2017 *Spring Conference on Computer Graphics (SCCG)*, it was later selected for publication in a special issue of the Computers and Graphics journal. URL: `https://dl.acm.org/citation.cfm?id=3138075`.

Thomas, D. P., Borgo, R., & Hands, S. J. (2016). *Topological Visualisation techniques for the understanding of Lattice Quantum Chromodynamics (LQCD) simulations.* In EGUK Computer Graphics & Visual Computing Conference. Eurographics UK. *DOI: 10.2312/cgvc.20161298* — This was presented at the 2016 *Computer Graphics & Visual Computing (CGVC)* conference where it won an award for “best full paper”. URL: `https://diglib.eg.org/handle/10.2312/cgvc20161298`.

Peer Reviewed Extended abstracts

Thomas, D. P., Borgo, R., & Hands, S. J. (2015). *QCDVis: a tool for the visualization of Lattice Quantum Chromodynamics data.* EGUK Computer Graphics & Visual Computing Conference (CGVC). Eurographics UK.

Thomas, D. P., Borgo, R., & Hands, S. J. (2014). *Visualization of Lattice Quantum Chromodynamics (LQCD) data with varying chemical potential.* Eurographics Conference on Visualization (EuroVis). Eurographics.

Academic posters

Thomas, D. P., Borgo, R., & Hands, S. J. (2016). *Topological visualisation techniques to enhance understanding of lattice QCD simulations.* XQCD: The 14th International workshop on QCD in eXtreme conditions.

Thomas, D. P., Borgo, R., & Hands, S. J. (2015). *QCDVis: a tool for the visualisation of Lattice QCD data.* Fifth Annual DiRAC Science Day.

Thomas, D. P., Borgo, R., & Hands, S. J. (2014). *Visualization of Lattice Quantum Chromodynamics (LQCD) data with varying chemical potential.* Eurographics Conference on Visualization (EuroVis). Eurographics.

Table of Contents

Chapter 1

Introduction

Contents

1.1 Volume visualisation

Volume visualisation is an established field of research in Computer Science with a core objective of using 3D graphics techniques to present volumetric data in a form which guides and informs scientific interpretation. Early volume visualisations arose from the need to display medical data, typically acquired from Computation Tomography (CT) or Magnetic Resonance Imaging (MRI) scans, in an easily navigable format. Many of the initial advances in volume visualisation were driven by the needs of medical professionals; however, over time the use of volume visualisation spread to multiple fields of science and engineering.

Volume visualisation can be split into two distinct methods of presenting the data. Direct volume rendering maps each sample in a scalar field to colour and opacity values through the use of transfer functions of varying complexity. A second approach, the focus of this thesis, is the use of indirect volume rendering.

Scientific visualisation often uses indirect volume rendering to translate discrete data into visual representations through the creation of an approximate model of the data with a particular set of parameters. The Marching Cubes algorithm [2] presented a significant step forward in computing objects as three dimensional surfaces that could be rendered and manipulated as objects in 3D space. Indirect volume rendering techniques allow the user to explore data as the scalar value is varied through the use of isosurfaces.

An important step forward in indirect volume rendering was the ability to segment data into connected components, referred to as contours, allowing discrete objects to be seeded from a single sample point in the data [3]. Topology-led methods, such as the contour tree [4], provide compact data structures to encapsulate and represent contours of scalar data as the function value varies. Beyond allowing the efficient computation of meshes for visualisation, connectivity graphs can also be presented as a method of summarising the topology of a data set. Information regarding topological structure can be relayed back to the user permitting further insights into the data [5].

Many of the techniques used in topology driven visualisation can be generalised to multivariate and high dimensional scenarios. Multivariate topology can enables scientists in various domains to understand correlations between different fields defined upon common sampling points. Several approaches exist including Jacobi Sets [6], graph comparison [7, 8, 9], Reeb spaces [10], and range tessellation [11].

1.2 Quantum Chromodynamics

Gravity and electromagnetism are two fundamental forces which we directly observe and experience. Physicists however identify four fundamental forces in nature: electromagnetic force, gravitational force, weak interaction, and strong interaction. The weak and strong interaction forces work at the atomic level: the strong interaction holds the nucleus together and the weak interaction is responsible for radioactive beta-decay.

The strong force, along with electromagnetism and weak interaction, underpin all of nuclear physics and the vast array of phenomena from the microscopic to the astrophysical level which we all experience. With respect to analytical frameworks the gravitational force is governed by Einstein's theory of general relativity while the strong interaction, electromagnetic force, and weak interaction are described by quantum field theory — theoretical frameworks for constructing quantum mechanical models of subatomic particles in particle physics.

The most systematic way of calculating the strong interactions of QCD is a computational approach known as *lattice gauge theory* or *lattice QCD*. Space-time is discretised so that field variables are formulated on a four dimensional hypercubic lattice. Gluon fields are described by link variables emerging from the sites (specified by 4 integers) in one of four directions.

Each link variable is an element of the Special Unitary group of matrices $\mathsf{SU}(N_c)$ where N_c represents the number of colours in the theory. True QCD sets $N_c = 3$; however, a simplified version of QCD sets $N_c = 2$ in order to allow the freedom to vary certain simulation parameters. Various lattice fields exist as path-ordered products representing a closed traversal of link variables from each site on the lattice.

1.3 Research question

The purpose of this thesis is to investigate lattice QCD datasets through the use of topological visualisation techniques. More specifically we ask how the topology based techniques and data structures can be used to observe the effect of varying the thermodynamic control parameter chemical potential μ. Altering this input whilst generating the data is thought to induce a bias in the number of quarks and anti-quarks present on the lattice.

The ability to modify chemical potential is a unique feature of the lattice QCD configurations generated by our work group at Swansea University. Most lattice groups work using fixed chemical potential, instead varying the temperature control parameter. The end result is that we are able to simulate extreme environments, such as those existing at the core of neutron stars.

Whilst we intend to use topological visualisation techniques to investigate our specific data sets, the approaches used in this thesis could be applied to the data generated by other lattice groups.

1.4 Overview and contributions

In this section we introduce each chapter of the thesis and provide a summary of the research contributions.

Chapter 2 presents an overview of the visualisation theory and algorithms used in this thesis. This includes an overview of indirect volume visualisation techniques, beginning with a comprehensive review of the development of Marching Cubes algorithms. We use this as a foundation for adopting a topology-led approach in creating mathematical models for the visualisation process. Topological visualisation techniques are then discussed in detail beginning with a review of algorithms working on scalar fields defined with a single variable and then generalised to multivariate and temporal scalar data. The survey is completed with a discussion of methods of presenting data in higher dimensions.

Chapter 3 gives a detailed introduction to Quantum Chromodynamics — the modern theory of the strong nuclear force. This models interactions between elementary quark particles as they form strongly-bound composites called *hadrons* by the exchange of gluons. The most familiar examples are the protons and neutrons found in atomic nuclei. We provide an introduction to the theory of Lattice Quantum Chromodynamics (lattice QCD), a discrete version of the theory formulated in terms of field variables on a four dimensional space-time lattice. Existing lattice QCD visualisation projects are reviewed as part of the overview and are used to demonstrate the current state-of-the-art in lattice QCD literature.

Chapter 4 presents a summary of the scalar fields studied in lattice Quantum Chromodynamics using existing general purpose visualisation software. Initially, these techniques were used to validate that the lattice model was correctly configured by presenting the data to physicists. This led to domain experts asking what could visualisation tell us that existing statistical physics techniques could not? Furthermore, what visualisation techniques may help interpret the effects of modifying chemical potential μ?

Chapter 5 details the progression from direct volume renders and isosurfacing towards a topology based approach. In this chapter we present lattice observables using several techniques with feedback gathered from the continual input from domain experts. We show that in using topological visualisation methods we are able to question basic properties of lattice configurations, such as the number of objects and the distribution of objects by isovalue. The model based approaches used in this chapter also enable us to question what happens to the structure of individual objects when varying parameters of the configurations. Also discussed are some the challenges encountered whilst working with lattice QCD data and the resolutions to these problems.

Chapter 6 takes the form of a case study where we demonstrate how topological visualisations can be used to identify and isolate the core observables of lattice QCD. Output from the algorithms enables us to examine the consequences of domain specific noise reduction techniques as they are iteratively applied to the data. We ask what happens to the “physical” properties of the core observables as the process proceeds — summarising the process using quantitative approaches.

Chapter 7 provides a second case study where we consider the use of multivariate topological techniques on temporal lattice fields. This opens up the possibility of comparison of spatial-temporal slices to capture evolving phenomena in the data by investigating the lifespans of four dimensional “pseudo-particles”. We are able to question what multivariate topology can tell us about these observables as they fade into and out of existence on the lattice. Principally we ask how the geometric structure of the objects change as time progresses.

Chapter 8 shows an alternative use of multivariate topology where it is used in to study the relationship between different fields existing on the lattice. This allows us to ask how core lattice QCD observables appear in various bivariate field combinations — and crucially if there are any potential variations in the data that may be affected by thermodynamic control parameters. We also ask if multivariate topology is able to detect and identify other objects that are typically ignored in lattice QCD. We also investigate how the multivariate topology can be captured in a simplified form for analysis purposes.

Chapter 9 is a final case study that builds upon the knowledge built from earlier chapters. The main objective of this case study is to apply topological visualisation techniques as part of a traditional domain workflow in order to enhance the analytical process. This enables us to question how multivariate topology can capture the effects of varying temperature T and chemical potential μ. We also use the techniques to help evaluate the *cooling process* in greater detail — asking if topology can assist physicists in selecting the optimal level of noise reduction to perform an investigation. Due to the uniqueness of using topology in lattice QCD we present our interpretations as a potential benchmark for future studies.

Chapter 10 reflects upon the framework built in collaboration with domain scientists over the course of this work. We ask how knowledge of existing research processes can help to design a bespoke tool to explore lattice QCD data. Many new methods of visualising and analysing the data were introduced to physicists over the course of this work — we ask how effective these techniques are by performing user studies. Finally, we question how the inclusion of the new topological visualisation features might lead to physicists asking new questions of their data that were previously beyond their reach.

Chapter 11 brings the thesis to a conclusion where we summarise what has been learned in the process of applying topological visualisation techniques to lattice Quantum Chromodynamics data sets. We also suggest some future directions for related research.

Figure 1.1 provides an overview of the relationship between the chapters of the thesis. Chapters 2 and 3 provide an introduction to the main thesis text and give the relevant background information required. However, due to the amount of new terminology presented in these chapters a glossary is provided in Appendix A.

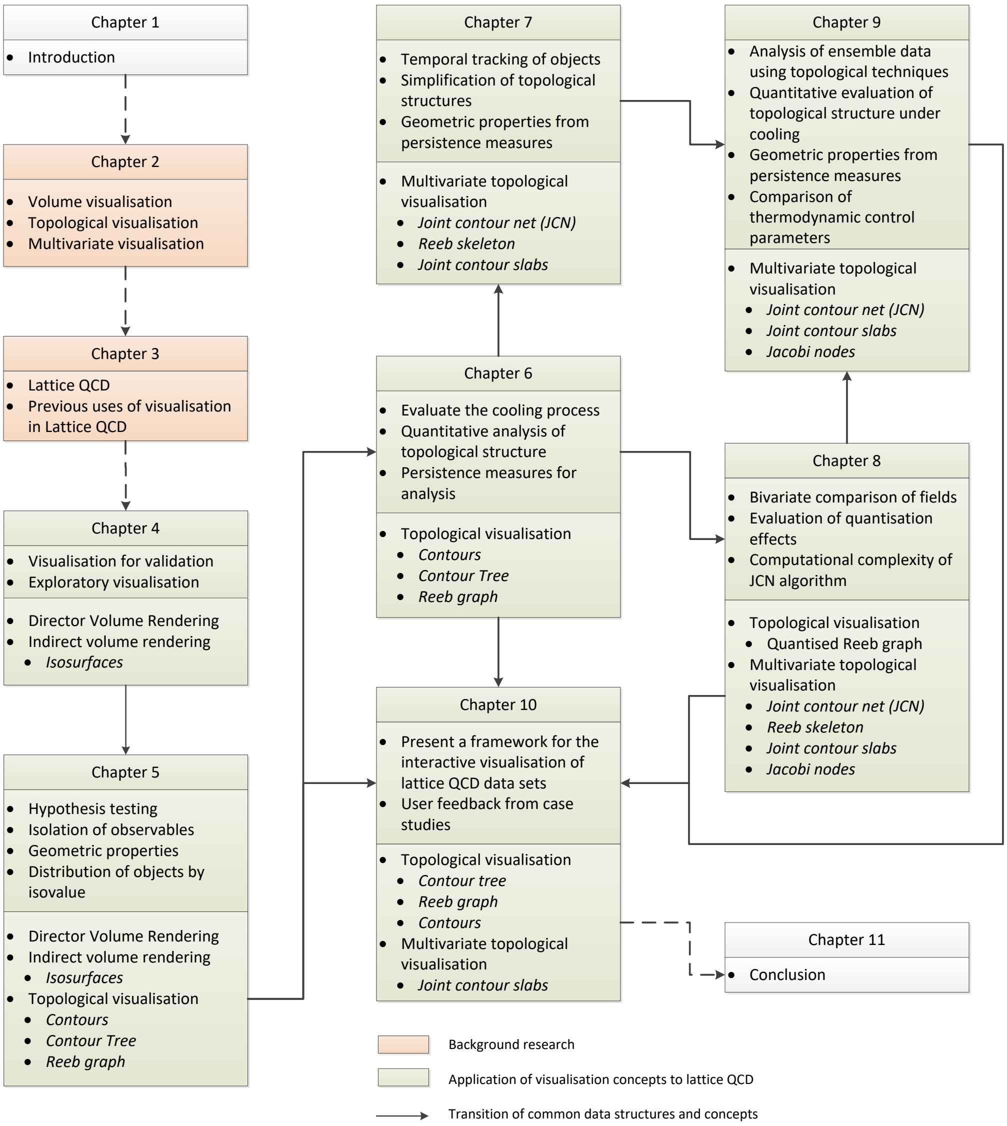

Chapter 1
Introduction
Chapter 2
Volume visualisation
Topological visualisation
Multivariate visualisation
Chapter 3
Lattice QCD
Previous uses of visualisation in Lattice QCD
Chapter 4
Visualisation for validation
Exploratory visualisation
Director Volume Rendering
Indirect volume rendering
Isosurfaces
Chapter 5
Hypothesis testing
Isolation of observables
Geometric properties
Distribution of objects by isovalue
Director Volume Rendering
Indirect volume rendering
Isosurfaces
Topological visualisation
Contours
Contour Tree
Reeb graph
Chapter 6
Evaluate the cooling process
Quantitative analysis of topological structure
Persistence measures for analysis
Topological visualisation
Contours
Contour Tree
Reeb graph
Chapter 7
Temporal tracking of objects
Simplification of topological structures
Geometric properties from persistence measures
Multivariate topological visualisation
Joint contour net (JCN)
Reeb skeleton
Joint contour slabs
Chapter 8
Bivariate comparison of fields
Evaluation of quantisation effects
Computational complexity of JCN algorithm
Topological visualisation
Quantised Reeb graph
Multivariate topological visualisation
Joint contour net (JCN)
Reeb skeleton
Joint contour slabs
Jacobi nodes
Chapter 9
Analysis of ensemble data using topological techniques
Quantitative evaluation of topological structure under cooling
Geometric properties from persistence measures
Comparison of thermodynamic control parameters
Multivariate topological visualisation
Joint contour net (JCN)
Joint contour slabs
Jacobi nodes
Chapter 10
Present a framework for the interactive visualisation of lattice QCD data sets
User feedback from case studies
Topological visualisation
Contour tree
Reeb graph
Contours
Multivariate topological visualisation
Joint contour slabs
Chapter 11
Conclusion
Background research
Application of visualisation concepts to lattice QCD
Transition of common data structures and concepts

Figure 1.1: A representation of the transition of concepts and techniques between various chapters in the thesis. Orange cells representing background research chapters give a brief overview of the topics explored. Those representing applied visualisation chapters (green) give a brief overview of the research intentions (top) and techniques used (bottom). Many visualisation techniques and data structures acquired in earlier chapters help to inform other phases of the research, this is indicated by an arrow between chapters.

1.5 Abstracts of published work

Below we present abstracts for published papers related to this thesis. A supporting list of work carried out during the production of this thesis, including links to the full papers, can also be found on my webpage at: `http://cs.swansea.ac.uk/~csdthomas`.

Topological Visualisation techniques for the understanding of Lattice Quantum Chromodynamics (LQCD) simulations. [12] The use of topology for visualisation applications has become increasingly popular due to its ability to summarise data at a high level. Criticalities in scalar field data are used by visualisation methods such as the Reeb graph and contour trees to present topological structure in simple graph based formats. These techniques can be used to segment the input field, recognising the boundaries between multiple objects, allowing whole contour meshes to be seeded as separate objects. In this paper we demonstrate the use of topology based techniques when applied to theoretical physics data generated from Quantum Chromodynamics simulations, which due to its structure complicates their use. We also discuss how the output of algorithms involved in topological visualisation can be used by physicists to further their understanding of Quantum Chromodynamics.

URL: `https://diglib.eg.org/handle/10.2312/cgvc20161298`
DOI: `10.2312/cgvc.20161298`

QCDVis: a tool for the visualisation of Quantum Chromodynamics (QCD) Data. [13] Quantum chromodynamics, most commonly referred to as QCD, is a relativistic quantum field theory for the strong interaction between subatomic particles called quarks and gluons. The most systematic way of calculating the strong interactions of QCD is a computational approach known as lattice gauge theory or lattice QCD. Space-time is discretised so that field variables are formulated on the sites and links of a four dimensional hypercubic lattice. This technique enables the gluon field to be represented using 3×3 complex matrices in four space-time dimensions. Importance sampling techniques can then be exploited to calculate physics observables as functions of the fields, averaged over a statistically-generated and suitably weighted ensemble of field configurations. In this paper we present a framework developed to visually assist scientists in the analysis of multidimensional properties and emerging phenomena within QCD ensemble simulations. Core to the framework is the use of topology-driven visualisation techniques which enable the user to segment the data into unique objects, calculate properties of individual objects present on the lattice, and validate features detected using statistical measures. The framework enables holistic analysis to validate existing hypothesis against novel visual cues with the intent of supporting and steering scientists in the analysis and decision making process. Use of the framework has lead to new studies into the effect that variation of thermodynamic control parameters has on the topological structure of lattice fields.

URL: `https://dl.acm.org/citation.cfm?id=3138075`
DOI: `10.1016/j.cag.2017.07.006`

Joint Contour Net analysis of lattice QCD data [in review] Lattice Quantum Chromodynamics (QCD) is an approach used by theoretical physicists to model the strong nuclear force. This works at the sub-nuclear scale to bind quarks together into hadrons including the proton and neutron. One of the long term goals in lattice QCD is to produce a phase diagram of QCD matter as thermodynamic control parameters temperature and baryon chemical potential are varied. The ability to predict critical points in the phase diagram, known as phase transitions, is one of the on-going challenges faced by domain scientists. In this work we consider how multivariate topological visualisation techniques can be applied to simulation data to help domain scientists predict the location of phase transitions. In the process it is intended that applying these techniques to lattice QCD will strengthen the interpretation of output from multivariate topological algorithms, including the joint contour net. Lattice QCD presents an interesting opportunity for using these techniques as it offers a rich array of interacting scalar fields for analysis; however, it also presents unique challenges due to its reliance on quantum mechanics to interpret the data.

URL: `https://arxiv.org/abs/1703.02488`

Chapter 2

Background

Contents

In this chapter we review advances in scientific visualisation that we later use throughout this thesis to present and analyse lattice Quantum Chromodynamics data. In Section 2.1 we present a brief overview of volume visualisation — where we also discuss the differences between direct and indirect volume rendering techniques. Section 2.2 gives an in depth review of the Marching Cubes algorithm which is commonly used to generate meshes for indirect volume rendering. We introduce topological visualisation techniques in Section 2.3 where we discuss the motivations and advantages of using the process. In Section 2.4 we discuss how the topological methods can be extended to analyse data with multiple variables. Finally, in Section 2.5 we discuss how many of the algorithms presented in this chapter can be generalised to data exceeding 3 spatial dimensions.

2.1 Volume visualisation

Early research into the rendering of volumetric data suitable for projection onto a flat surface was focused upon the use of medical data captured from X-rays, Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scans [14, 15, 2]. However, volumetric data rendering is now applied to a huge range of scientific domains including theoretical physics, which often works with simulated data. The field is ever evolving, spawning new methods for imaging the data, whilst also taking advantage of significant improvements in computer hardware.

[redacted]

Image taken from Gouraud [21].

Figure 2.1: Voxel representations of a canine left ventricle [15]. Left: with basic shading. Right: with Gouraud shading.

There are two basic types of volume visualization; direct volume rendering and indirect volume rendering.

2.1.1 Direct volume visualization

A direct volume render (DVR) projects the source data on screen using established methods such as ray tracing. Maximum Intensity Projection [14, 16], commonly abbreviated to MIP, is an established simple algorithm for performing DVR. The technique is used extensively in medical imaging, for tasks such as highlighting blood vessels, by projecting rays through a medium to find the maximum sample. An opacity transfer function can be used to make it possible to peel away individual parts of the model; for example decreasing the opacity of a flesh layer can allow the skeleton to become visible in medical scans.

However, in its simplest form the algorithm doesn't give an overall perception of depth. This can be addressed using techniques including Local Maximum Intensity Projection [17] which gives geometric feedback via shading. This approach also provides an increase in speed by allowing rays to terminate as soon as a threshold value is encountered. Other improvements to MIP allowed Csebfalvi et al. [18] to provide an interactive interface using a technique called Fast Maximum Intensity Projection. Mora and Ebert [19] provides a Low Complexity MIP algorithm using tree structures to reduce the computational complexity by bypassing regions of the volume with low intensity.

Surface boundaries do not need to be computed directly when using DVR [20], meaning construction of a model is not required prior to rendering. Additionally, the ability to defer rendering of parts of the image obscured by an opaque surface decreases rendering time and also enables the direct computation of shadows. However, depending on the size of a data set, the ability to create images in real-time can be severely impacted upon by processing speed.

2.1.2 Indirect volume visualisation

Indirect volume rendering creates a mathematical model of the data prior to rendering. Most commonly this is achieved by creating a number of isosurfaces that represent layers of the volume at a common function value, sometimes known as an isovalue. However, the model created is only an approximation of the true data and research is ongoing into creating increasingly accurate methods.

Marching cubes algorithms [2], covered in detail in Section 2.2, are commonly used to create isosurfaces. Alternatively, Herman and Lui [15] used density data collected using X-rays to display human organs, using voxels to represent points in 3D space on a regular grid. Voxels are small enough in size to be able to be stacked in formation to produce detailed representations of the desired organ. However, when projected on screen they can be made up of multiple pixels. An example of a canine left ventricle rendered using voxels is given in Figure 2.1. The authors suggest that the same procedure could be achieved using triangular polygons but with the processing power present at the time they felt this would be too time consuming and there would be a noticeable lack of precision. However, there are certain benefits in using uniformly ordered cubes including that surface normals, required for shading, are simple to compute relative to other graphical methods present at the time such as Bezier patches. Culling of cubes that are not visible to the user is also a simple process in comparison to triangular polygons of non-uniform orientation. These advantages gave the algorithm a performance boost on the hardware available at the time.

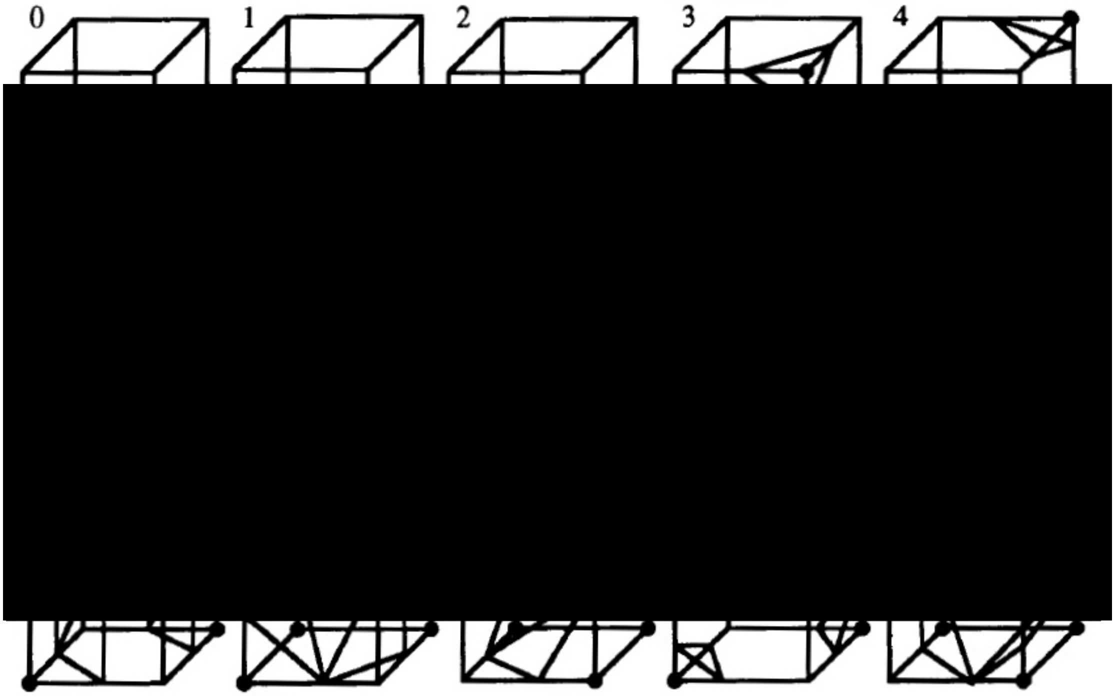
2.2 Marching cubes algorithms

Marching cubes (MC) is an established algorithm used to generate isosurfaces from fields of scalar values and has become one of the most popular methods for completing this task. A large body of literature has grown around the topic and this has seen the release of survey papers focused on the development and improvement of the algorithm. A 2006 survey paper by Newman and Yi [22] attributed one-hundred and ninety-five papers directly to development and improvement of the algorithm. The survey paper showed that the original MC method had been modified for use in various situations using varying grid structures and volume subdivision methods. A related approach known as Marching tetrahedra (MT) is discussed in Sec. 2.2.4.

2.2.1 Motivation and implementation

The original MC algorithm [2] was proposed in 1987 as a method for displaying 3D images of medical data. It was intended to work with data from CT and MRI scanners, turning it into a mathematical model that is simpler to work with and manipulate. Existing methods of dealing with such data were to work with a large collection of 2D slices, requiring a great deal of training for the operator to be able to understand the data correctly. Lorensen and Cline [2], argued that algorithms existing at the time were unable to produce models of high enough detail and often introduced unwanted rendering artefacts.

MC algorithms build up a three-dimensional mesh of triangular elements by dividing the input volume into a number of cubic cells. A divide and conquer approach is deployed to



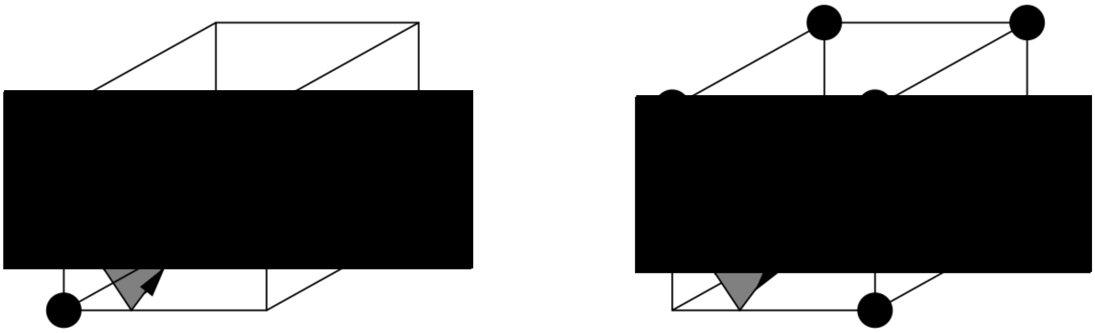
[redacted]

Image adapted from Lorensen [2]

Figure 2.2: Lookup table of vertex configurations from the original marching cubes algorithm.

determine the way in which an isosurface intersects each of cell by considering the function at each vertex of the cube. Each of the eight vertices of a cube are considered as Boolean values; a logical one assigned for a vertex that is above the surface, and a zero for those below. In total there are $2^8 = 256$ possible ways the cube can be intersected by the surface. However, due to rotational and reflective symmetries this is reduced to fifteen possible configurations that form a vertex configuration lookup table (Figure 2.2). The placement of the vertices of the triangles are controlled by a linear interpolation of the surface value at the point where the edge intersects it. Output from the algorithm is a set of vertices defining each of the triangles obtained from the lookup table. From this a set of normals vectors can also be calculated that can be used to apply shading to the model.

Complementary cases existing in the reduced MC lookup tables can lead to the production of incorrectly oriented triangles (Fig. 2.3). This occurs when a configuration produces outputs with opposing windings and presents a problem for most rendering frameworks. The ordering of vertices is often used to determine the front or back of a surface; hence, it is most common to notice this as rendering artefacts when surface opacity is applied. A property known as *coherency* can be used to guard against ambiguities — this states that edges common to two adjoining triangles should appear twice in the mesh (in opposite directions). Wallin [23] initially implemented the coherency check as a post-construction step, but it can also be checked as part of the surface construction [24].



[redacted]

[redacted]

Image taken from Jones [24].

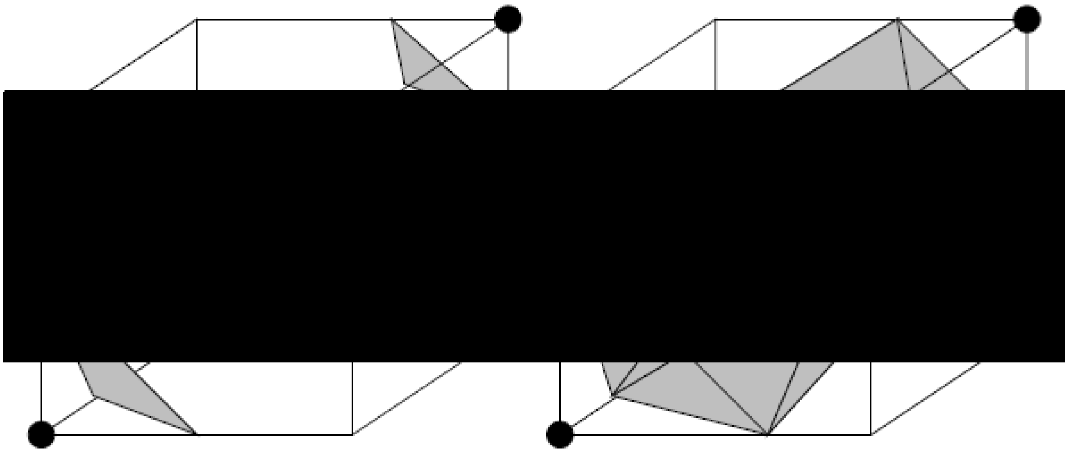
Figure 2.3: Triangles with opposing windings produced by complimentary vertex configurations.

2.2.2 Surface irregularities and improvements

The MC algorithm was able to develop 3D polygonal models from sets of scalar values; however, the quality of the meshes was far from perfect. One particularly problematic aspect of the algorithm was that it created isosurfaces that were prone to holes. This was a point first brought to light in a response letter to the Computer Graphics journal shortly after the algorithm was published [25]. A later modification by Nielson and Hamman [26] attempted to address this issue using a technique named the *Asymptotic Decider*. The MC algorithm did not cope well with triangulation methods in ambiguous configurations of the lookup table. An ambiguous configuration is defined as a situation where a face contains four vertices with the same isovalue. Of the original fifteen configurations present within the MC lookup table, five configurations are prone to generating ambiguous faces. Further investigation of the problem revealed that in total there were an additional twenty-six variations on those already present. Whilst it is conceded that the ambiguities are infrequently present within real world data-sets, they believe their method of inserting further branches into the lookup table was able to address the issue. Additionally, this is achieved without a noticeable increase in running time.

Chernyaev's [27] attempt to alleviate topological inconsistencies in marching cubes was *Marching Cubes 33 (MC33)*, named so because the modification provided a lookup table of thirty-three configurations. Specifically, fixing internal ambiguities was the main focus of this algorithm and it was largely similar in implementation to the asymptotic decider. An example of an internal ambiguity is the case where opposing vertices are marked as in or out of the surface but it is not possible to tell if they are connected using internal triangles (Figure 2.4). Using a bilinear interpolation across a plane parallel to a face Chernyaev suggested a possible solution to the problem, describing the corrected ambiguous configurations.

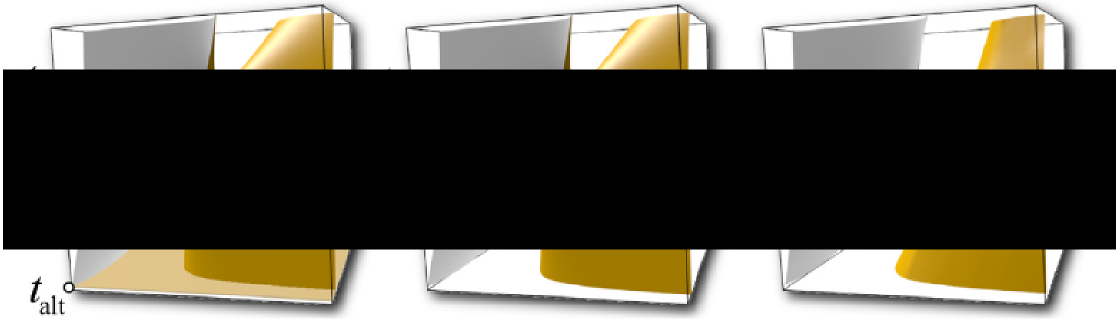
Lewiner et al. [28] showed that the MC33 algorithm did not create topologically correct surfaces in all situations. Their algorithm extended MC33 by testing for additional internal ambiguous situations whilst also removing redundant tests in the early algorithm. The resultant



[redacted]

Image taken from Chernyaev [27]

Figure 2.4: Internal ambiguities present in configuration 4 of marching cubes as identified by Chernyaev.



[redacted]

Image taken from Custodio et al.[30]

Figure 2.5: Errors identified in configuration 6. Two surfaces are separated (right) but should be connected by a tunnel (left). Custodio et al. perform an additional test at t_{max} to check the presence of a connection (centre).

algorithm was claimed to be topologically correct by removing cracks and topological inconsistencies. Further topological irregularities were highlighted in the MC33 algorithm by Etiene et al. [29] with rectifications proposed in an algorithm known as *Corrected-MC33 (C-MC33)* by Custodio et al. [30]. The authors of C-MC33 acknowledge that other MC implementations are able to produce the desired topologically correct isosurfaces, however they felt that as the MC33 algorithm is still widely used it was important to fix the problems. The result of the corrected algorithm is that ambiguities within configurations 10, 12 and 13 are solved. Other improvements include the rectification of missing tunnels between surfaces, as shown in Figure 2.5.

Alternative approaches to MC33 have been proposed to correct topological irregularities using additional triangles within configurations that had ambiguities. Whilst the addition of adding polygons to the mesh impacted on raw processing speed, Montani et al. [31] felt that the added simplicity of avoiding branching in ambiguous configurations counteracted this. In practice when compared to reference functions and using real datasets, little computational overhead became evident when compared to the MC original algorithm. An extension to both the original marching cubes algorithm and the asymptotic decider was later released by Nielson [32], that reverted back to a branching method. This further solved the ambiguities that were liable to cause holes in isosurfaces by extending the linear interpolation of values and later bilinear interpolation of faces with a trilinear interpolation within the interior of voxels. Using several test cases for each of the look-up table configurations Nielson was able to prove that topologically sound surfaces could be generated using this method. This was later confirmed as being topologically correct by Carr [33], who described the method for completing the trilinear interpolation process as dividing the existing cubes into smaller blocks with simpler topology. However, the trilinear interpolation method used showed differences in outputs to those proposed by Montani et al. [31].

2.2.3 Limitations

Whilst well suited for approximating isosurfaces on smooth functions, MC derived algorithms struggle to accurately capture features such as sharp edges and corners. Poorly shaped triangles within the mesh of a surface is one issue that arises from MC on sharply edged models. Kobbelt et al. [34] presented a modified MC algorithm known as *Extended Marching Cubes*. This technique is heavily influenced by standard MC style algorithms but with additional sampling points generated at 'sharp features'. Sharp features are identified by considering the tangent to the contour at existing sampling points.

The *Dual Marching Cube* algorithm [35] attempts to produce smoother surfaces by using quad patches. The method used for construction is similar to the original algorithm; however, a different set of configurations are required to create a quad-patch, or MC-patch, mesh. The final MC-Dual surface is a geometric dual of the MC-patch surface; hence, it can be computed directly without an intermediate surface being produced. An added advantage to the algorithm is that the normals for generated surfaces produce cleaner features when used for lighting calculations as the polygons have more slight changes over the surface as a whole.

Typically additional sample points are used to try to better capture sharp edges, resulting in more triangles in the output mesh. Bhattacharya and Wenger [36] took an alternative approach, using cube merging to reduce the number of triangles around difficult features. The modified algorithm was not guaranteed to form a manifold surface, but was seen as a step towards a generalised algorithm capable of handling data traditionally unsuited to Marching Cubes algorithms.

For extremely complicated data sets featuring huge numbers of sampling points the isosurfacing algorithm, particularly in its serial form, can become extremely slow to compute and display. *Progressive isocontouring* [37] is a technique that can be deployed to compute isosurfaces at ever increasing levels of accuracy. By computing the isosurface at increasingly fine resolutions it is possible to maintain an acceptable responsiveness for the user when interacting

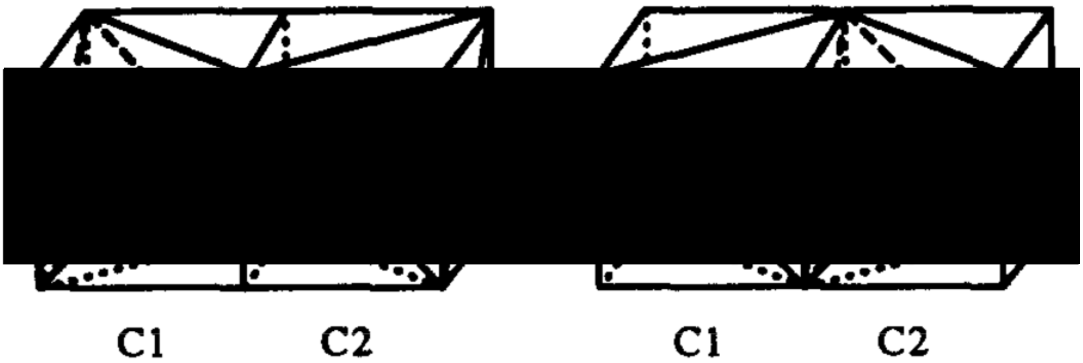

[redacted]

Image taken from Zhou et al. [39].

Figure 2.6: To avoid ambiguities in the computed surfaces marching tetrahedra requires alternating the orientation of the decomposition.

the data. Once a region of interest has been identified it is possible to iteratively increase the resolution to diverge towards the best representation.

2.2.4 Marching tetrahedra

Marching tetrahedra (MT) provides an alternative approach to surface generation that splits each cube into tetrahedra for processing. Using tetrahedral cells instead of cubes allows the ambiguities present in MC to be addressed using a non-branching approach. The lookup table for tetrahedral cells contains 16 possible configurations which unlike MC are free of potential ambiguity. Additionally, MT algorithms can provide triangles with an improved structure in the resultant mesh leading to higher quality rendering results [38].

Many implementations split each cubic cell into five tetrahedra, producing four identical tetrahedra surrounding one larger regular tetrahedron [40, 41, 39]. Using 5 tetrahedra per cell introduces a directional bias, resulting in the need to alternate the orientation of the subdivision between neighbouring cells (Fig. 2.6). The effect can be observed in the output as ambiguities depending on the ordering of the cells during processing. Zhou et al. [39] proposed to address this by using cubic interpolation, rather than bilinear interpolation, along the edges of the tetrahedra to compute the true intersection of the surface. However, this resulted in a more complicated lookup table of 59 entries, required to the handle potential for multiple edge crossings by a single surface. Other approaches to MT decompose each cubic cell into more than five tetrahedra, removing the need to alternate orientation between cells or using more complex lookup tables. Chan and Purisima [42] use a body centred approach, adding an additional vertex at the centre of the cube, resulting in 12 well proportioned, congruent tetrahedral cells for the computation. However, a limitation of these approaches are that they create additional cells for processing.

The subdivision of cells from cubes to tetrahedra, regardless of decomposition scheme, typically raises the number of triangles in the output mesh. This means that generally there are additional rendering and storage overheads associated with MT in comparison to MC. The need to reduce the number of triangles can be handled whilst computing the surface or by post processing the output. Gueziec and Hummel [41] limit the complexity by merging mesh triangles sharing a common vertex. Treece et al. [38] use vertex clustering during surface construction to simplify the input whilst preserving the topology of the data. Applying these methods can vastly reduce the number of triangles in the mesh, seeing reductions of up to 70% in comparison to regular MT and 40% when compared to MC [38]. Furthermore, the merging process can help to create less degenerate triangles, helping to improve the quality of the rendered output [41].

2.2.5 Generalisation to multivariate data

MC algorithms are applicable to scalar fields where the data at each point is a single variable. Besides univariate scalar fields it is also possible to visualise fields where each point is associated with vector or tensor data. These variations are established area of study in their own right and beyond the scope of this thesis. Besides this the scientific community is becoming increasingly interested in multivariate scalar fields.

For this purpose the concept of the isosurface generalises to that of the fibre surface in multi-field data sets. Previous solutions presented the data using direct volume visualisation; however, using indirect volume rendering techniques is also beneficial. A MC based algorithm was presented by Carr et al. [43] that allowed the capturing of fibres for bi-variate data sets. This allows the boundaries of intersecting isosurfaces to be captured as surfaces constructed from multiple fibres, the multi-field equivalent of contours. The captured fibre surfaces are geometrical in nature, rather than topological, and doesn't take connectivity into account.

The *fibre surface* generalises the concept of the isosurface to *multivariate* data sets. The algorithm, presented by Carr et al. [43], is similar in execution to its univariate counterpart equivalent, the MC algorithm [2]. Until recently multi-field visualisation has largely been limited to direct volume techniques (DVR); however, this approach limits its use as an analytical technique. Fibre surfaces utilise indirect volume techniques to create geometric models of the multi-field, in addition to generating visual representations this allows geometric properties of the multi-field to be queried.

Physically fibre surfaces can be considered as the bounding surface between two or more fields. The ability to visualise multiple scalars as fibre surfaces is seen as a tool to compliment existing techniques, such as multivariate scatter plots [44], to allow quicker boundary extraction [43]. When compared to similar DVR techniques (Fig. 2.7), such as multi dimensional peak finding [45], it was found that fibre surfaces were quicker to render but had an addition preprocessing overhead. Fibre surfaces represent a simplified method of presenting the data captured using the multivariate topological techniques discussed in Section 2.4.

[redacted]

Image taken from Carr et al. [43].

Figure 2.7: Direct comparison of a fibre surface and a DVR render of an ethane-diol simulation from chemistry. Central image shows a scatter plot of the same data.

2.3 Topology driven visualisation

Topology driven visualisation is used to capture characteristics of a data set using basic topological invariants such as the number of holes or points in a connected region. This information provides a means for obtaining a skeleton of the data which can then be used to construct visualisations in various forms. Use of topological visualisation techniques can see a reduction in the size of a large data set, as only key features need to be retained. By using topological structures to represent a data set intuitive methods can be used for speeding up computations; for example, by allowing many parts of a data set to be rendered in parallel. A further increase in efficiency is provided by the ability to bypass redundant computations — such as the processing of empty cells in MC. Increases in rendering performance comes with the overhead of the need to do a one-time preprocessing of the data.

Many topology based algorithms rely on abstract graph based structures for storage of data. In many situations the graph structures can be viewed directly to form an overview of the data in an easily perceived format. For those with little prior knowledge of topological visualisation techniques it can be hard to form an initial understanding of the data using such formats. In order to better understand the data it is common to directly link the graph visualisation, either statically or dynamically, with a rendered view of the data. However, difficulty in understanding topological data representations can be further complicated when a data set is large in size. Hence, other approaches are sometimes used that try to take advantage of human perception, such as topological landscapes. In this thesis the individual simulation datasets we display are relatively small; therefore, we tend to use graph and surface based representations throughout.

Section 2.2 described algorithms for computing isosurfaces, otherwise known as level sets, to join vertices in a scalar field sharing a common function height. During the construction of an isosurface no topological information is used; hence, an isosurface is unable to distinguish individual connected regions. Morse theory enables the analysis of the topology by considering the gradient and second derivatives of the level set at each vertex in the scalar field. Through the use of tetrahedral mesh cells (for 3D scalar fields) we can approximate the isosurface as a piecewise linear function between sampling points. However, at the boundary between cells this approach invalidates a key condition of Morse theory which requires a function to have derivatives defined at all sampling points. Work by Edelsbrunner et al. [46] and Bremer et al. [47] provides a mechanism for handling these limitations, allowing us to consider the data as a Morse function.

Morse theory provides a method for computing the topology of scalar data as we sweep through the isovalue range. In this section we review the algorithms and structures used in topological visualisation, beginning with an overview of the key terminology used.

Simplex The simplex is a generalisation of the triangle or tetrahedron to n dimensions. Properties of simplexes in dimensions 0 (a point) through to 4 (a Pentatope) are given in Table 2.1). Computational topology often refers to meshes as simplicial complexes — a set of n simplexes glued together at their boundaries.

Betti numbers Allow us to categorise the topology of a mesh existing in n dimensions by examining the connectivity of its structure as a number of n dimensional simplexes. Each dimension, from 0 to $n-1$, has its own Betti number β_n associated with it to represent the quantity of cuts required through a mesh in order to separate it into two parts. We can consider these numbers as representing the number of holes in the mesh in n dimensions (Table 2.2). Most topological visualisation algorithms consider only the 0^{th} dimensional Betti number, used to represent connectivity. However, it is possible to augment the graphs with higher order Betti numbers to provide feedback on changes in the mesh such as the morphing from a sphere to a torus (Table 2.3). In this work we are primarily only interested in the zeroth dimensional Betti number β_0.

Genus This is a way of expressing the number of holes in a surface. In three dimensions, a sphere is genus 0, and torus genus 1. Surfaces with additional holes are commonly referred to as an n-torus.

Critical point The sampling points in a scalar field $f(x)$, at which the function value is defined, can be categorised into two types. Critical points have a local gradient $f'(x) = 0$ and represent positions where the topology of the level set changes. These relate to local extrema or saddle points in the data; the second derivative $f''(x)$ allows us to classify if the critical point is a local minima $f''(x) > 0$ or a local maxima $f''(x) < 0$. In the case where a critical point is a local minima all neighbouring vertices will have a higher function value, whilst a local maxima means that all surrounding vertices are lower in value. Regular points have a non-zero gradient and have no overall effect on the connectivity the level set.

Contour The boundary of connected regions of a level set, also known as the 0-dimensional homology group, are known as contours. Each individual contour is an element in the level set that represents a distinct topological object within the isosurface.

Topological persistence The most simplistic way of defining topological persistence is as a quantitative measure of importance of each contour. In literature topological persistence is often known as geometric measures [48] due to the fact it relates to measures such as volume or surface area. It is possible to compute persistence values directly from the meshes representing each contour by performing a piecewise sum of all cells in a connected region. Alternatively, the persistence can be approximated by counting the number of regular vertices making up a connected region. Persistence measures are often used as a method of eliminating noise from the topological structure by iteratively removing the contour with smallest persistence value.

Name	Dimension (n)	Common name	Vertices	Edges	Faces	Cells
0-simplex	0	Point	1	-	-	-
1-simplex	1	Line-segment	2	1	-	-
2-simplex	2	Triangle	3	3	1	-
3-simplex	3	Tetrahedron	4	6	4	1
4-simplex	4	Pentatope (or 5-cell)	5	10	10	5

Table 2.1: Classification of simplexes in dimensions zero to four.

Most of the techniques and structures discussed in this section can be generalised for use in any number of dimensions — a list of general terms is provided in Table 2.4.

2.3.1 Merge trees

One of the fundamental data structures in computational topology are merge trees, commonly used to describe two similar concepts. Merge trees track connected level sets within the data as the isovalue is swept across the scalar range. The *join tree* captures connected sub-level sets, or regions, of the data below a specific isovalue threshold. Similarly the *split tree* captures connected super-level sets.

Merge trees present a simple method for capturing topological information as the arcs in a tree-like structure. This can then be used to compute zero-dimensional persistence measures representing the number of connected components in a region [49]. Merge trees are often used as a computation step in the construction of more complex topological data structures such

Betti number	Associated simplex	Property
β_0	Point	Connected components
β_1	Line-segment	Holes
β_2	Triangle	Voids

Table 2.2: Betti numbers β_0, β_1, and β_2 and associated concepts of connectivity.

Topological concept	β_0	β_1	β_2
Sphere	1	0	1
Torus	1	2	1

Table 2.3: Betti numbers used to describe the sphere and torus.

Name	Dimension (n)	Common name
0-manifold	0	Point
1-manifold	1	Polyline
2-manifold	2	Polygon
3-manifold	3	Polyhedron
4-manifold	4	4-Polytope

Table 2.4: Classification of manifolds in dimensions zero to four

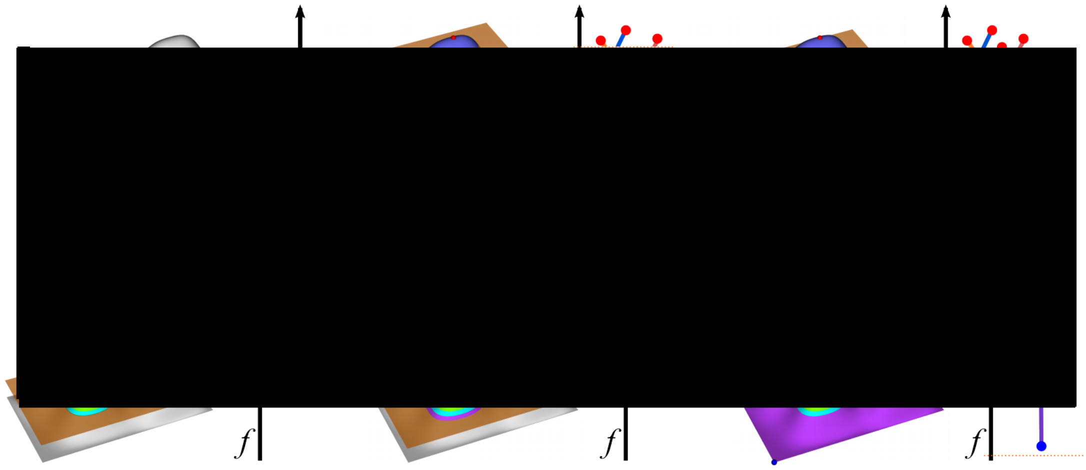
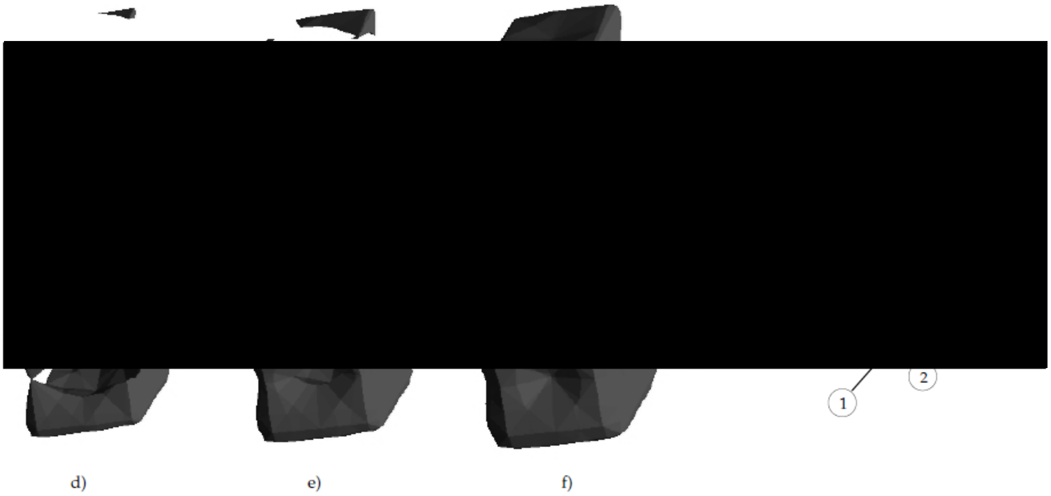

[redacted]

Image taken from Landge et al. [50].

Figure 2.8: A visual representation of the merge tree algorithm for the function in the top-left corner. The merge tree captures each of the local maxima as the function height is reduced (moving left-to-right, top-to-bottom). A real world analogy is to think of the isovalue as the water level in a valley that gradually drains to reveal the peaks of multiple hills.

as the contour tree, described in detail in Sec. 2.3.2. Alternatively, merge trees can be used directly for analysis and feature extraction in complex data sets [50].

A visual metaphor for understanding the concept of merge trees is given in Fig. 2.8. To capture the merge tree, in this case the join tree of the grey function (top-left), the isovalue is gradually decreased from the global maxima. In the top-centre image this has revealed three local maxima, each denoted as a red critical point. In the top right image a fourth local maxima has been revealed (the green contour) and the red and blue structures have merged, this is marked by a node in the graph. In the bottom left image, the pink and green surfaces have merged to form the cyan region. The bottom centre image sees the small intervals represented by the cyan and yellow surfaces merged into one large purple region. If the isovalue continues to be reduced it is possible to observe no further changes in the topology until reaching the global minima represented by the blue graph node.



[redacted]

Image taken from Carr [51]

Figure 2.9: Left: A varying surface representing the value of a function as it is changed. Right: contour tree representation of the surface as the value is changed

2.3.2 The contour tree

The contour tree, introduced by Boyell and Ruston [52] uses concepts from Morse theory [53, 54] to capture changes in the topology of scalar field at critical points in the data. The tree structure allows the tracking of splits and joins in the topology as the isovalue is varied by tracing a path through the graph. An example is given in Figure 2.9 which demonstrates how individual contours in the data are captured by the contour tree using a one-to-one correspondence with edges. Each leaf in the contour tree represents an individual local extrema and critical vertices represent a change in the connectivity of the level set. Typically, the contour tree does not contain information regarding changes in topological genus; however, this information can be added to the output [55].
We can formally describe the structure of the contour tree as follows:

- Vertices or *supernodes*
  - Leaf nodes correspond to:
    - A local maxima where a contour is created.
    - A local minima where a contour is destroyed.
  - Interior nodes correspond to:
    - A saddle point where one more contours are created as a contour splits into two or more disjoint contours.
    - A saddle point where one or more contours are destroyed as two or more disjoint contours merge into one.

- Edges or *superarcs*
  - Represent a single contour between the supernode where it is created and the supernode where it is destroyed.

In addition to optimizations of contour tree computation, research is focused on approaches for storing the contour tree hierarchy, allowing it to be traversed in an efficient manner. A number of related approaches have been proposed including Kd-trees, segment trees, and interval trees. These formats prove problematic as they can require a large amount of storage and are difficult to navigate. An alternative method is proposed in [56] using the observation that an isosurface is the level set of a continuous function in 3D space; therefore, a whole contour can be traced from a single element. Vertices where contours can be built from are given the name seeds; algorithms for computing seeds relate to work in image processing that require similar storage facilities. An entire contour can be computed for the specified desired isovalue from the seed that is bounded by the two supernodes representing the critical points of the contour. Results showed that by using this method the number of seeds required for representation were in most cases significantly reduced. However, often the overall storage size of the contour tree was increased in comparison to other methods.

Carr et al. [51] investigated the use of contour trees in higher dimensional data sets, whilst also improving upon the algorithm proposed in [4]. They also introduced the concept of *augmented contour trees*, an extension that added non-branching vertices at non-critical points in the data to provide additional values for isosurfaces to be seeded from. This feature was built into a GUI allowing the user to identify regions of interest, using colour coding and to distinguish them as directed by the user [57]. Carr and Snoweink [58] use the contour tree as the underlying data structure to generate object meshes using path seeds. The use of the contour tree can be extended to finding paths between two points given condition clauses, such as a minimum or maximum values, on a function defining a landscape [59].

Chiang et al. [60] introduce a modified contour tree construction algorithm that improves processing time by only sorting *critical vertices* in the merge tree construction stage. This vastly increases processing speeds in very large data sets. However, as observed by the authors, critical vertices are difficult to identify in data with dimensionality of four or more. Further increases in speed are offered by storing multiple seeds for the monotone path construction algorithm [61] used to generate surfaces. This results in an increased storage overhead but with the advantage that a surface does not require complete re-extraction each time the isovalue is changed. Recently Gueunet et al. [62] presented a multithreaded approach to computing the augmented contour tree using a shared memory approach. Initial evaluations of the "contour forest" algorithm showed quicker computations achievable using the approach; however, at present the extent of the speed up is limited by load imbalance and redundant computations.

For a given technique of subdividing the input domain the output of the contour tree algorithm is fixed. However, in the case that a different technique is used to define the neighbourhood of sampling points the location of the critical point may change trivially. For example, changing the method of defining neighbours during split / merge tree computations can slightly alter the location of critical points. The underlying structure of the tree, the number of supernodes and superarcs, remains fixed for a given input with only the geometric locations of supernodes changing. However, we always use the same fixed neighbour function throughout the thesis to maintain data integrity.

2.3.3 The Reeb Graph

Using the contour tree comes with the limitation that the input must be defined on a *simply connected domain* that is free of loops. Limitations in the merge tree computation phase of the contour tree algorithm prevent it from correctly handling data without a boundary. However, the Reeb graph [63], a generalisation of the contour tree, can be used to compute the topology of scalar data in such situations [64]. As with the contour tree, the Reeb graph can be applied to models of any dimension provided it is represented on a simplicial mesh [65]. The Reeb graph has also been used to assist in the design of transfer functions in volume visualisation [66], by assigning opacity based upon how many objects were obscured by nested surfaces and their proximity to the edge of a scalar field.

The first use of the Reeb graph for encoding topological features for visualisation purposes was by Shinagawa et al. [67] where the structure was used as a way of representing objects obtained from computerized-tomography (CT) sources [68]. An online algorithm is given by Pascucci et al. [65] that allowed the Reeb graph of a function to be computed using a streaming approach with the output continually updated as additional data points are added. Efficiency of the algorithm is optimal for two dimensional inputs and the streaming nature of the algorithm limits peak memory usage. However, the $O(n^2)$ complexity of the algorithm, where n is the number of triangles, means it performs less favourably for higher dimensional inputs.

A key feature of many Reeb graph algorithms is that a *2-skeleton* of the input volume is used to define the relationship between vertices, edges and triangles. The same restriction applies to scalar fields in n dimensions; hence, provided a triangulation of the data is available the algorithm will be able to compute a correct n dimensional Reeb graph. Reeb graph algorithms can largely be split into two groups; those that are *sweep based*, requiring the maintenance of level sets, and those that use a *split and compute* method to collapse the problem to that of a *contour tree* computation. A third alternative was given by Harvey et al. [69] that randomly collapsed triangles for arbitrary 2-skeletons to improve the running time to $O(m \log m)$.

The sensitivity of Reeb graph computations to the topological genus of the input domain was first demonstrated by McLaughlin [64] in the case of a non-simply connected domain, such as that representing a sphere or torus. Additionally, the number of loops present in the data as a result of the Morse function can further extend the complexity of the algorithm. In order to address this undesired effect the Reeb graph computation can be reduced to that of the simpler contour tree algorithm. This approach was first used by Tierny et al. [70] where they performed “loop-surgery” by making symbolic cuts during the *merge tree* step of the algorithm. The symbolic cuts could then be stitched back together to retrieve a topologically correct Reeb graph of the data. An alternative approach, capable of generalising to higher dimensional inputs, was later proposed by Doraiswamy and Natarajan [71] that explicitly maintained level sets, eliminating the need for a loop-surgery preprocessing step.

More recently Doraiswamy and Natarajan [72] offered an improved algorithm for constructing the Reeb graph of n-dimensional scalar fields, reverting to the split and compute contour tree technique, as originally proposed in [70]. A further optimised algorithm [73], using a variation of the split and compute technique, used the join tree of the data to identify potential loops. An important modification upon the technique developed by Tierny et al. [70] was the segmentation of domain into multiple loop free contour trees, instead of a single contour tree, thus enabling multiple regions of the input to be computed in parallel.

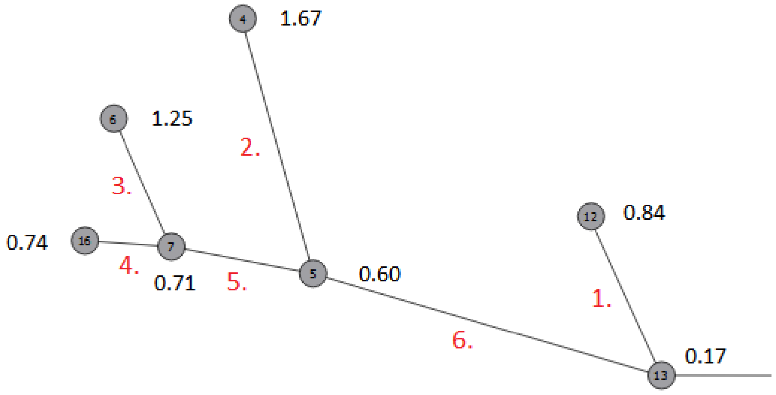
Output from Reeb graph algorithms remains fixed provided the simplicial subdivision defined upon the input domain does not change. As is the case with the contour tree, variations can slightly alter the location of the critical vertices without changing the overall structure of the graph. In this thesis we use a fixed simplicial subdivision for each case study — this is given in the implementation details of each chapter.

2.3.4 Seeded contours

Segmented contours, as opposed to isosurfaces, can be grown using seed values extracted from the topology to generate triangular meshes. Wyvill et al. [74] generate contour surfaces as a two-stage process; first, the core region is flood filled by evaluating neighbouring cells for inclusion in the level set. In the second stage boundaries are calculated using look-up tables similar in form to those of the MC algorithm [75]. Carr and Snoeyink [58] use the contour tree as the underlying data structure to store and generate object meshes. Rather than using *minimal seed sets* as used by van Kreveld et al. [3] to generate contours, an alternative method was deployed using *path seeds*. Contour trees are a reliable source of seed cells for propagation algorithms similar to those of Wyvill et al. [74]; hence, the contour tree can be used at runtime to generate surfaces for each superarc at a given isovalue.

De Berg et al. [59] made the observation that an isosurface is the level set of a continuous function in 3D space suggesting that a whole contour could be traced starting from a single element — a seed. An entire contour can be computed for the desired isovalue from the seed that is bounded by two super-nodes at the critical points of the contour. Carr et al. [51] investigated the use of contour trees in higher dimensional datasets, whilst also improving upon the algorithm proposed by Tarasov and Vyalyi [4]. They introduced the concept of augmented contour trees, an extension that added non-branching vertices to provide values for isosurface seeding. Separate isosurfaces can be generated for each non-critical value identified in the contour tree, separately colour coded and manipulated by the end user. This enables users to identify regions of interest and distinguish them accordingly [57].

The *flexible isosurface* [76] is a technique for combining many of the features discussed in this section with the concepts of *topological persistence* (see Section 2.3.5) into a single interface. Simplification methods, as discussed in [48], can also be applied to the data so as to display isosurfaces in a meaningful way. This approach allows each contour to be hidden or displayed according to the users preferences at runtime. In order to further aid data exploration contours can also remain fixed in view as the global isovalue is varied. Colour can be applied to each surface (or a sub-tree of the main contour tree) to allow assignment of meaning to contours by the user by providing a simple grouping mechanism.



Superarc		Top Supernode		Bottom Supernode	
id	persistence	id	isovalue	id	isovalue
1	0.67	12	0.84	13	0.17
2	1.07	4	1.67	5	0.60
3	0.54	6	1.25	7	0.71
4	0.03	16	0.74	7	0.71
5	0.11	7	0.71	5	0.60
6	0.43	5	0.60	13	0.17

Figure 2.10: An example of topological persistence on the superarcs of the contour tree or Reeb graph. The measure of persistence defined here is the isovalue range — this is given as the difference in the isovalues of the two critical vertices defining the superarc.

2.3.5 Topological persistence and simplification

Different scientific fields of study often define the interesting features and attributes in a domain specific form. However, some features are invariant and can be useful in any field of science. The contour spectrum [5] was introduced as a method of relaying quantitative information about individual contours in scalar data including surface area and volume. Carr et al. [77] directly compare isosurface statistics against raw histograms of scalar data for a number of data sets. Measurements evaluated included the cell intersection count, triangle count and isosurface area. It was found that using these measures a truer distribution of the scalar field could be computed. An improvement was given by Meyer et al. [78] using concepts from geometric measure theory that minimised the effect of noise on the observed distributions. The key to this improvement was introducing a normalisation of the individual contour statistics to the domain average.

Scalar field data often contains noise; when partitioned using a topology sensitive algorithm this manifests in the generation of a large quantity of small objects present at a limited range of isovalues. Methods for reducing noise were first proposed by Edelsbrunner et al. [49] using an iterative process that performed topological simplification by assessing the Betti-numbers of topological objects. The goal of the algorithm is to simplify shape whilst preserving the underlying topological features of data existing on a triangular mesh. Primary uses of the algorithm were predicted to be work in computational geometry; however, the removal of noise in isosurfacing algorithms was another potential application proposed by the authors.

Carr et al. [48] proposed the use of concepts originally discussed as part of the contour spectrum [5], such as enclosed volume and surface area, as an aid for noise removal. Objects are queued in ascending order, according to the user selected measure, and iteratively removed until a terminating level of simplification is achieved. The effect of three different simplification features are compared using X-ray data from a human skull. Use of the isovalue range persistence measure (Fig. 2.10) resulted in the removal of important features such as the eyes. Carr et. al. remark that simplification techniques are sensitive to the source and should be chosen using domain specific knowledge. Additionally, whilst the technique is suited to *n* dimensional data, time variate data requires additional considerations in the simplification process due to connectivity between time steps.

2.3.6 Topology in Direct Volume Rendering

The techniques discussed in Section 2.3.4 model topological objects as meshes, meaning they are well suited to indirect volume visualisation. However, topology based approaches can also be applied to data displayed using direct volume rendering approaches.

A segmentation algorithm such as the contour tree or *volume skeleton tree* is first used to look for boundaries between objects in the scalar field topology. Following this, traditional direct volume rendering techniques can be applied to the data based upon attributes of the identified objects. Takeshima et al. [79] use attributes such as the number of equal valued contours and occlusion to assign levels of opacity to objects. The net effect was that the outermost objects were assigned lower opacities so as to not obscure features centred in the volume. A more flexible approach was used by Weber et al. [80] applying distinct transfer functions to each object, or topological zone, as directed by the user. This customization, implemented via the user interface, allowed grouping of similar features and related components by the user using colour and transparency.

2.3.6.1 Temporal univariate scalar data

Recent interest in topological visualisation research has focused upon the comparison of scalar data through topological methods to assign a degree of similarity to data. This has potential uses in different scientific domains including physics, chemistry, and climate science, which often feature a temporal component. The merge tree, with its simple data-structure, presents an attractive method for computing topological differences between data sets. Beketayev et al. [7] define a distance measure between merge trees with potential applications in a range of scientific disciplines. The algorithm uses a *branch decomposition* approach to deconstruct the merge tree into multiple sub-graphs, each a descending path from a saddle to a leaf vertex. Each branch decomposition can then be scored via an adapted form of the edit distance [81] between two graphs. A recursive algorithm then tests if two merge trees are within an epsilon value to determine if they are classified as similar.

This approach was further extended by Saikia et al. [9] to produce the *extended branch decomposition graph*, a union of all individual sub-trees. This data structure allows quicker comparison between merge trees by computing multiple similarity thresholds in parallel. In addition the extended branch decomposition graph improves upon memory usage by reducing the redundancy found in multiple disjoint branch decompositions. Branch decomposition methods have also be applied to the more complex contour tree to compute similarity and symmetry in scalar topology [82].

The method used by Beketayev et al. [7] had a runtime of $O(n^5)$, where a merge tree contains n nodes, this was lowered to $O((n \log n)^2)$ in related work [9]. Potential downfalls to the optimised algorithm are that instabilities can arise from permuted forms of branch decompositions, this is handled in [7] by considering all possible permutations at the cost of scalability. The optimised algorithm uses the *extended branch decomposition graph*, a union of all possible branch decomposition graphs, to store all branch decompositions in a single tree structure. Two potential uses for the distance measure are suggested; identification of similar structures within a single data set or, for time variate data, repetition between time steps.

Fluid dynamics is also an area of science that can benefit from visualisation of 3D volumes with a time component. The complexity of mixing two fluids is one specific problem that can be better understood using topological methods as the system changes in time [83]. Topological analysis makes it possible to take a slice of the volume at a given time-step and count the number of bubbles present. A secondary use for topological analysis is the ability to trace features throughout the simulation including the volume and centre of gravity of individual bubbles.

2.4 Multivariate topological visualisation

Multivariate topological visualisation, when compared to topological visualisations of a single field, is a more recent research interest in the visualisation community. Due to the relative infancy of the field, Carr et al. [84] highlight the need for sufficiently complex data sets to extensively test the emerging algorithms in the area.

Many of the topological structures and algorithms applicable to a single variable can be generalised to more than one variable being defined at each sampling point. We begin this section by defining the concept of *Jacobi nodes*, we then continue to examine structures used to capture and the display the topology of multiple variables.

Jacobi node These are critical points in the scalar fields where the gradients of the inputs become equal or zero. This demonstrates an alignment of the functions or the existence of local extrema in a shared location. These points are guaranteed to have the same geometric locations, as a pre-requisite for the existing algorithms is that the fields are defined on a common sampling mesh.

2.4.1 The Reeb space

The *Reeb space* is a generalisation of the Reeb graph allowing for multivariate or temporal data. The first discussion of using the Reeb space to compute topological structure of multiple functions is presented by Edelsbrunner et al. [10], where it is suggested that the Reeb space can be modelled mathematically in the form $f : \mathbb{M} \mapsto \mathbb{R}^k$, where $\mathbb{M}$ represents the domain and

f the output of k scalar functions. For the simple case, where $k = 1$, this is directly comparable to the Reeb graph. The Reeb space extends this formulation to situations where $k \geq 2$.

2.4.1.1 The joint contour net

Carr et al. [11] presented the first discrete representation of the Reeb space using the *Joint Contour Net* (JCN). For functions of n variables defined in an $\mathbb{R}^m$ dimensional space the algorithm approximates the Reeb space as a number of multivariate contours named joint contour slabs. These represent connected regions of the domain with respect to the isovalue of multiple functions. In situations where $n \geq m$ the JCN can still be computed; however, the output is not an approximation of the Reeb space but instead a subdivision of the input geometry over n variables. The JCN captures the Reeb space as an undirected graph structure, where vertices represents slabs n isovalue tuples, and edges are used to show adjacency between regions. An example JCN of two scalar functions is presented in Figures 2.11 and 2.12.

In comparison to contour tree algorithms, the JCN is better suited to parallelisation as each joint contour slab is constructed from smaller independent regions called fragments [86]. Existing development of the algorithm has focused largely on implementation as a parallel algorithm. A distributed memory model is used by Duke and Hosseini [87] to construct multiple sub-JCNs in parallel, these are merged into a single output as a final post-processing step. Current results suggest that the merge step is the limiting factor to parallel algorithm speed up; therefore, other parallelisation strategies are under investigation.

In nuclear physics the JCN has previously been used to visualise and analyse scission datasets where it was used to identify the splitting of an atomic nucleus into multiple parts [85]. It was found that the JCN was well suited to capturing this divergent behaviour using proton and neutron density fields as inputs. Figure 2.13 shows the multi-field and JCN prior to the scission event; in Figure 2.14 it is clear to see the two distinct blue regions are captured as two branches of the JCN. This experiment was initially performed at a single temperature [88], but later repeated at multiple temperatures [89] due to its ability to capture the splitting of the compound nucleus as a forking in the multi-field topology. Whilst performing the analysis a number of other events were captured and linked to the scission theory. It was found that "starburst-like" features in the topology likely equated to well formed fragments [88]. A "zippering effect" was picked up at large temperatures (Fig. 2.15); this was determined to represent a spatial connectivity between distinct regions in the multi-field. In terms of the underlying physical phenomena, domain experts interpreted it as a "de-localisation of quasi-particles" at high temperatures.

More recently the JCN was used to visually analyse data from hurricane Isabel [90]. Vertices were used to represent the joint contour slabs by mapping to their barycentric spatial coordinates. An interactive environment was developed that allowed users to relate interactions in the temperature, pressure and precipitation fields to physical phenomena such as rain bands and the eye of the hurricane. The ability to relate properties of the JCN to known physical features helped to increase understanding of how the JCN is able to capture multi-field interactions.

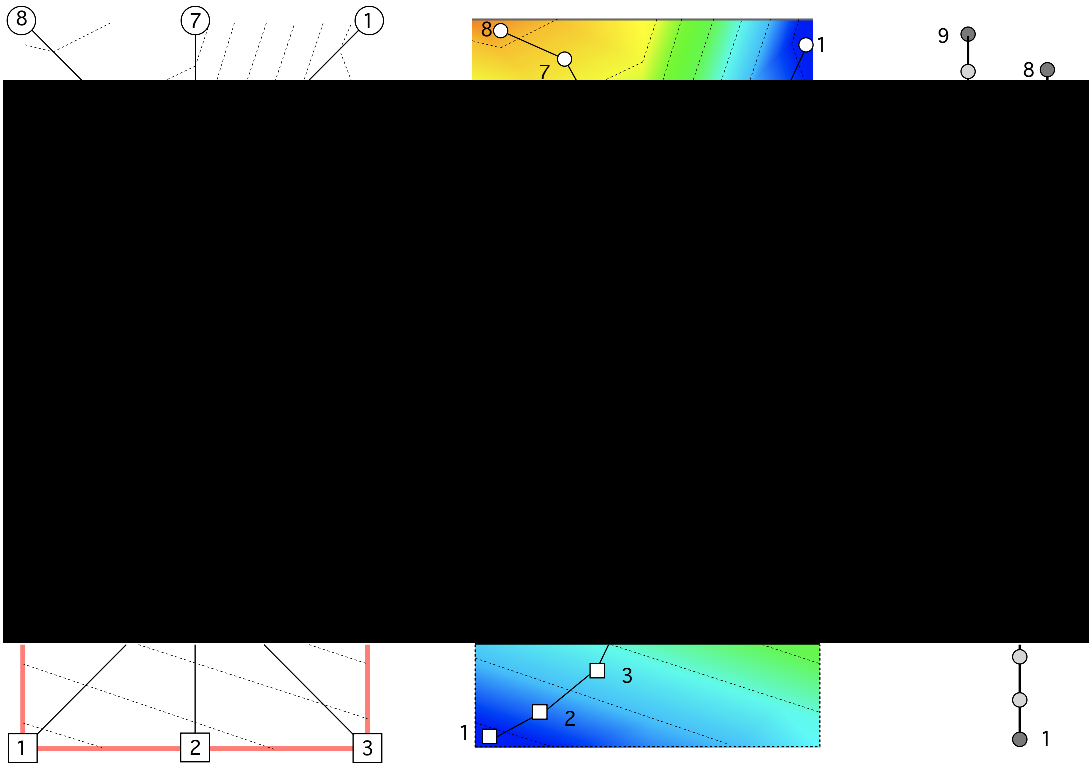

[redacted]

Image taken from Duke et al. [85]

Figure 2.11: Two simple scalar functions defined on a simplicial grid (left) where the dotted lines represent the quantisation intervals. The quantised contour tree for each function (right) is shown mapped to scalar field in the centre.

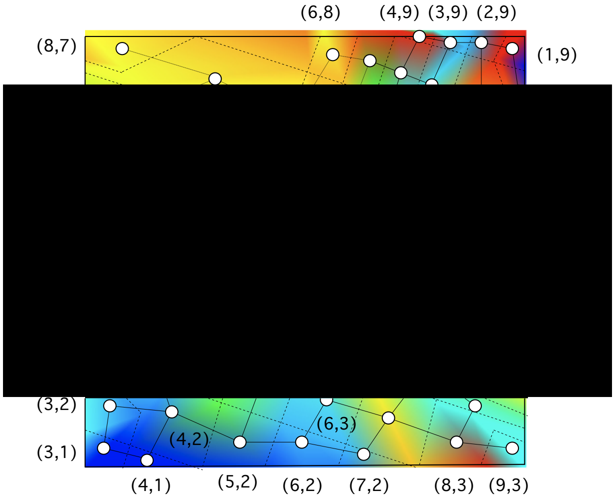
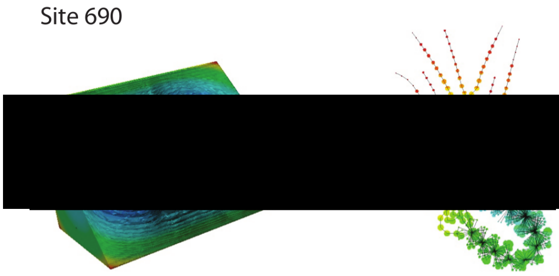

[redacted]

Image taken from Duke et al. [85]

Figure 2.12: The JCN capturing the bivariate topology of the two simple functions shown in Fig. 2.11. The bivariate field is decomposed by overlaying the quantisation intervals of the two input fields (dotted line). A vertex is placed at the barycentre of each region, or joint contour slab, and edges mark adjacency.



[redacted]

Figure 2.13: Nuclear scission slabs and JCN taken before event. JCN shows a single branch in the blue region.

Figure 2.14: Nuclear scission slabs and JCN taken after event. JCN shows a two branches in the blue region.

Images taken from Duke et al. [85]

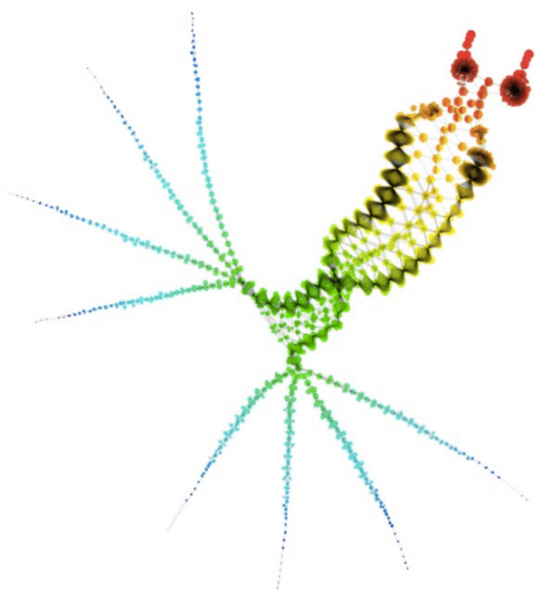


Image taken from Schunck et al. [89].

Figure 2.15: The “zippering” effect witnessed in the JCN.

2.4.2 Related topological structures

In multivariate topology a *Jacobi set* represents the set of critical points generated when one Morse function is restricted to the level set of another. Alternatively, the Jacobi set can be considered as the set of points where there is an alignment of the gradient of two or more Morse functions or the gradient of one function vanishes. The similarity between two or more functions can be evaluated using the resulting Jacobi set using a method defined by Edelsbrunner et al. [91]. Applications of the Jacobi set include use as a feedback loop on simulation parameters or to perform comparison of algorithms. An algorithm was presented by Edelsbrunner and Harer [6] for computing Jacobi sets of multiple Morse functions defined on a triangular mesh of isovalue n-tuples in Euclidean space. This allows multiple independent functions such as pressure, temperature, or wind speed to be used as inputs. Alternatively for temporal data multiple time-steps can be used as input to allow the evaluation of a function with respect to time.

Carr et al. extend the JCN [11] to extract further topological structure from the Reeb space by first evaluating an intermediate structure, the *Multi Dimensional Reeb graph* (MDRG) [92]. This is a hierarchical structure that recursively stores the joint contours of each function (f_1, f_2, $\ldots$, f_n) restricted to the contours of those preceding it. At the top level the MDRG represents the contours of the function f_1 as a Reeb Graph; the second tier relates to the Reeb graph of function f_2 when restricted to the contours of f_1, continuing down to function f_n restricted to the contours of f_1, $\ldots$, f_{n-1}. Besides being used in the extraction of the Jacobi structure and the related *Reeb skeleton* [93], the MDRG represents a convenient structure for extracting the Reeb graphs of each individual function making up the Reeb space.

An additional abstraction, the *Jacobi structure*, related to the mathematical topology concept of *singular fibre-components* are introduced by Chattopadhyay et al. [92] as part of the MDRG extraction algorithm. This represents an improvement over the Jacobi set by providing a method of relating the Jacobi set directly to the Reeb space. The Jacobi structure is able to capture the exact location of topological change and is defined as the projection of the Jacobi set from the domain to the Reeb space. In practice this extends the Jacobi set to also include the “regular sheets” connecting one another in the Reeb space. This means the Jacobi structure is able to capture elements of the topological structure that the Jacobi set is unable to represent. The Jacobi structure is extracted as the set of critical nodes in the Multi Dimensional Reeb Graph (MDRG), itself a structure for storing Reeb space criticalities. Forking in multi-field topology in nuclear scission data is a example behaviour that can be captured by the Jacobi structure [92].

The *Layered Reeb graph* is an alternative approach deployed by Strodthoff and Juttler [94] for presenting the Reeb space of multiple scalar functions. This approach to representing the Reeb space differs from the MDRG [92] by working directly with the Jacobi sets, rather than the more recently proposed Jacobi structure.

2.4.3 Topological persistence and simplification

Persistence in multivariate data sets is more complex to define in comparison to the univariate case. Simplification and persistence metrics can be defined on a number of secondary structures computable from the multi-field topology. The concept of isosurface statistics [77], [78] is extended to multivariate inputs through the use of *Continuous Scatterplots* [44]. These can be defined to show relations between m dimensional inputs with n scalar fields; in the case where $m = 3$ and $n = 1$ the output approximates to the output of Meyer et al. [78].

Multivariate data gives rise to *multi-filtrations* due to their parametrisation by more than one variable; this leads to no definable compact invariants, such as the Betti numbers, existing in multi-fields. Therefore, existing concepts, such as the persistence bar code [95], do not directly generalise to multi-variate domains. However, this does not mean that persistence and simplification cannot be applied, instead other approaches have been suggested. The “rank invariant” is a method for representing persistence in a multi-field by generalising upon the concept of Betti numbers present in univariate topology. For the univariate case, the rank invariant and persistence bar code are the same [96]. The original algorithm used to compute the rank invariant was exponential in time complexity, this was later improved to polynomial time by reformulating the problem as an algebraic geometry problem [97].

The Jacobi set, where the gradient of multiple functions align or have a gradient of zero, can assist in defining persistence measures [91]. When the multi-field is used to represent temporal data this can be used to augment the univariate notion of persistence with a lifetime parameter. This approach was used by Bremer et al. [98] to compute persistence in the context of the Morse-Smale complex. However, when generalised to non-temporal functions defining persistence as a feature of the Jacobi set becomes a non-trivial task [99].

2.4.3.1 The Reeb skeleton

The Reeb skeleton (Fig. 2.16) is a simplified graph structure that takes into account the size of connected components, allowing measures of persistence to be assigned to its arcs. An extended Jacobi set, the *Jacobi Structure*, is used by the Reeb skeleton algorithm to aid multivariate simplification [93]. The Jacobi structure [92] is a promising starting point for simplification due to its ability to separate the Reeb space, as approximated by the JCN, into singular and *regular components*. Just as in the univariate equivalent, the Reeb graph, singular nodes in the Jacobi structure map to topological changes in the multi-field. To exploit this, the Reeb Skeleton extends the concept of the Jacobi structure further, primarily to aid multi-dimensional simplification [93].

The Reeb skeleton is generated as the dual graph of the singular and regular components captured in the Jacobi structure. Visually, this means the Reeb skeleton translates the sheet-like form of the Jacobi structure into a simplified skeletal form. The simplified graph data structure allows measures of persistence to be assigned to arcs of the Reeb skeleton in a similar manner to that of the Reeb graph. Lip pruning techniques, similar to the leaf pruning method of simplification found in univariate topological structures [48] can then be applied to progressively remove noisy features in the multi-field. Example persistence measures that can be applied to the JCN include the accumulated volume of joint contour slabs in a connected region.

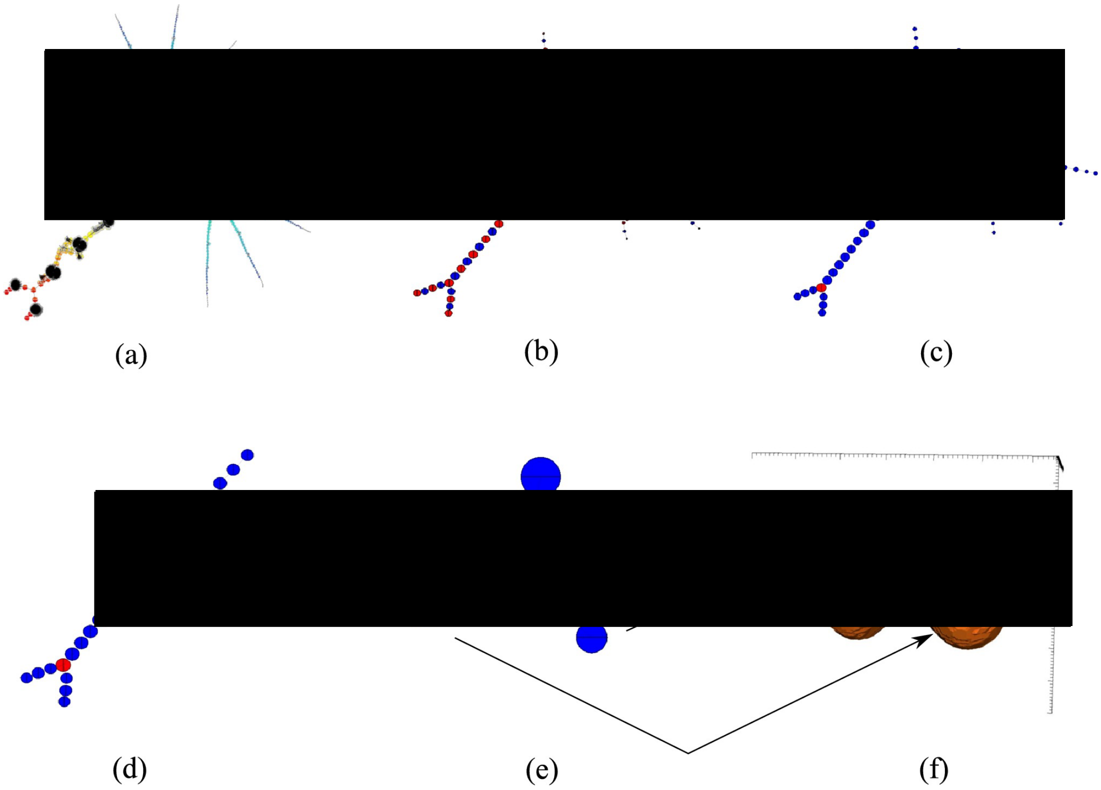

[redacted]
[redacted]

Image taken from Chattopadhyay et al. [93].

Figure 2.16: An example of the Reeb skeleton. The original JCN, from which the Reeb skeleton derives, is shown in (a); (b) shows the full, unsimplified Reeb skeleton; (c) modifies the Reeb skeleton to highlight only critical changes in the multivariate topology (red vertices); (d) shows how performing simplification using the Reeb skeleton removes less significant regions of the topology; (e) after simplification, the Reeb skeleton can be reduced to only key vertices; (f) the arrowed regions relate to the two surfaces.

2.5 Visualisation in higher dimensions

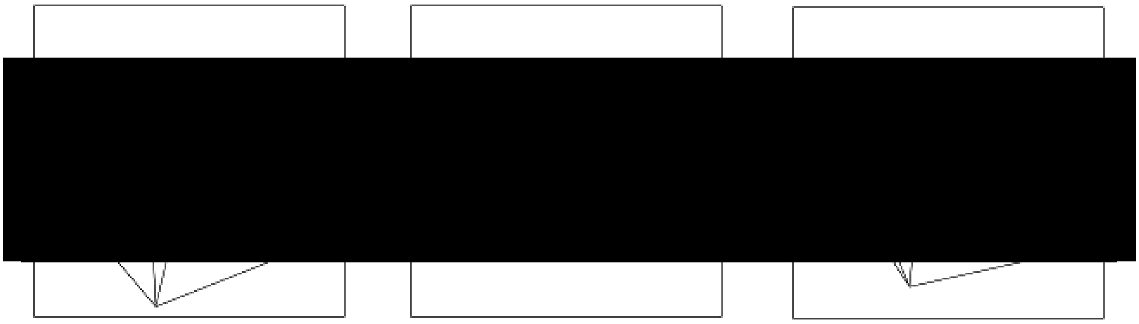
When moving to higher dimensional spaces it is beneficial to generalise the terminology used to describe the geometry. The n-dimensional analogue of the square is the *hypercube*, often shortened to n-cube (see Table 2.5). When working in higher dimensional spaces it is common to perform a simplicial sub-division of the n-cube into n-simplexes, this helps to avoid the ambiguities often associated with MC style algorithms [26].

2.5.1 Projection and perception

Creating easily perceivable and topologically correct three-dimensional models for projection on to flat surfaces, the computer monitor, is a difficult task. Volumes of data become increasingly hard to visualize as their dimensionality increase; for example, by introducing a

Name	Dimension (n)	Common name	Vertices	Edges	Faces	Cells
0-cube	0	Point	1	-	-	-
1-cube	1	Line-segment	2	1	-	-
2-cube	2	Square	4	4	1	-
3-cube	3	Cube	8	12	6	1
4-cube	4	Tesseract	16	32	24	8

Table 2.5: Properties of hypercubes (n-cubes) in dimensions zero to four


[redacted]

Images taken from Hollasch [101]

Figure 2.17: Examples of a wireframe tesseract (4D hypercube). Left: 3D perspective projection. Centre: 3D isometric projection. Right: 3D perspective project from a different view point.

temporal fourth-dimension. One of the major limiting factors is our in ability to perceive things four-dimensionally. A metaphor for trying to understand four-dimensions in a three-dimensional world, is to consider the case of two-dimensional creatures trying to understand a three-dimensional world. This is a thought exercise discussed in [100], which also discusses how a four-dimensional Euclidean representation of space-time relates to real world physics.

Existing projection methods are available that take a four-dimensional objects and display them on a two-dimensional surface, usually in wire-frame form. Quite often the projections are animated to show the object as it rotates on one or more axis; however, this can also be adapted to allow the user to rotate the object through conventional approaches such as mouse interaction. The effect of perspective and isometric projection are presented by Hollasch [101] using tesseracts — the 4D hyper-cube (see Figure 2.17 for examples). In addition, the use of ray-tracing in four-dimensions is explored. Theoretically this can produce understandable images as depth cueing is handled automatically by the algorithm in the creation of shadows. However, the added complexity of a fourth dimension in a looping ray tracing algorithm makes it time consuming and unworkable for real time display. In conclusion, the work states that wire-frame models are the best solution available, particularly when curved surfaces are to be projected, but for a more accurate view it is suggested that ray-tracing is used. Hollasch, goes on to suggest using tessellated tetrahedra to represent the surface, in much the same way that 3D surfaces can be approximated using triangles in marching cubes.

n	Centroid division($2^{n-1} \times n!$)[102]	$n!$
2	4	2
3	24	6
4	192	24
5	1920	120
6	23040	720

Table 2.6: Number of simplexes generated using differing sub-division techniques in n-dimensions

2.5.2 Computing surfaces in higher dimensions

Whilst presenting difficulties with regard to visualisation, it can be beneficial to compute surfaces in higher dimensions. This is especially true in the case of volumetric data with a temporal element; animation can often be used to reconstruct this form of data but that presents its own perceptual issues. Computation of isosurfaces on fields existing in $\mathbb{R}^4$ space can help to improve animations by providing an interpolation between the discrete time-steps, resulting in smoother and easier to perceive animations. Alternatively the high-dimensional topology can be sliced along arbitrary axes to provide a snap-shot with a reduction in dimension (e.g. $\mathbb{R}^4 \mapsto \mathbb{R}^3$).

An unavoidable consequence of upping the dimensionality of the input field is an increase in the complexity of its storage and computation. An early example of computing surfaces in > 3 dimensions is considered by Weigle and Banks [102] using a recursive technique to split n-dimensional cells into n-simplexes. The resulting surfaces exist in four dimensions and are able to be displayed using two techniques; stereographic projection (Sec. 2.5.1) or slicing to reduce dimensionality. It is noted that the centroid division technique used to break cells into simplexes used in this work is suboptimal (see Table 2.6), but can be improved using lookup tables similar to those used in marching cubes. The technique was used in [103] to compute the swept volumes of time-varying data generated from electromagnetic field simulations, allowing them to be displayed as animations.

Extending MC into the fourth dimension, and beyond, was the approach explored by Bhaniramka et al. [104]. At the time of the report, a relatively small amount of work had been conducted in studying isosurfaces beyond the third dimension. It was found that by extending MC into 4 dimensions the look-up table, following removal of symmetrical configurations, required 222 separate configurations. As with the 3D variation of MC, one aspect to be taken into consideration is the dealing of ambiguous configurations; a mathematical proof of correctness is provided to verify that the topological structures generated are valid. The primary use for this algorithm, from the authors point-of-view, would be to present volume data with a time dimension by selecting a three-dimensional slice of the hyper-volume.

2.5.3 Topological landscapes

An alternative approach of viewing contour information, the topological landscape, was proposed by Weber et al. [105] that took the contour tree and used it to build a 3D terrain model.

The purpose of this is to harness the natural ability that humans have in understanding terrain structure and use it to provide an easier to understand model of the underlying data topology. Valleys in the terrain illustrate events where an isosurfaces splits in to two or more parts and peaks represent where two or more isosurfaces merge. Topological landscapes can be applied to contour trees of any number of dimensions; hence, they can provide a useful method for exploring what is happening in high dimensional data sets. The topological landscape methodology was further expanded using a number of different methods for laying out the data, primarily from established 2D visualisation techniques [106]. The intended outcome was that signatures of interesting features would surface across multiple data sets.

2.6 Summary

In this chapter we have introduced the core concepts of volume visualisation using direct and indirect volume rendering techniques. We have shown how isosurfaces can be used to create mathematical models of scalar data, and how use of topology can help to understand the relationship between unique objects in the data. We also discussed the potential ways in which topological visualisation algorithms can be generalised to any number of dimensions and input variables.

In Chapter 3 we introduce the theoretical aspects of Quantum Chromodynamics and explain how we can compute observables from a discrete representation of the theory. We also present current techniques used by physicists to visualise their data.

Chapter 3

Lattice Quantum Chromodynamics

Contents

In this chapter we introduce the theory of Quantum Chromodynamics and explain how it is computed using a discrete formulation of the theory. We begin, in Section 3.1, by introducing the theoretical aspects behind Quantum Chromodynamics and how simulation can be used to investigate how it operates. Section 3.2 explains how the physical theory is implemented in the discrete setting of a space-time lattice. This allows data to be generated using the techniques detailed in Section 3.3, where we also introduce the scale and complexity of the data set. We present existing visualisations techniques used by domain scientists in Section 3.4.

The following terminology is used in this chapter:

Lattice The structure used to model the data consists of *sites* existing at the vertices of a lattice and *links* between neighbouring sites.

Periodic Boundary Condition A key feature of the computations is that they are modelled on lattices without boundaries — this allows all lattice sites in space-time to be treated equivalently. The term periodic boundary condition refers to the concept that the extremes of each axis are connected just like any other neighbouring site.

(Anti-)Instanton This is a four dimensional pseudo-particle which, unlike real-world particles, has a limited lifespan in time. The location of (anti-)instantons in space-time is linked to specific properties of the gluon field.

Cooling This is a process used by physicists to remove noise from the generated data which exists as small fluctuations between neighbouring sites. The cooling process is designed to smooth the data whilst trying to preserve key observables called (anti-)instantons. The process is applied iteratively, with each iteration of the cooling algorithm resulting in a new unique lattice.

3.1 Quantum Chromodynamics (QCD)

Quantum Chromodynamics is the theory used to describe interactions between the sub-nuclear particles that bind together to form hadrons. In its most basic form it describes interactions between quarks as they bind with one another via the exchange of massless gluon particles. Hadrons exist in two states; those consisting of three quarks known as baryons (Fig. 3.2), and quark-antiquark configurations that make up the meson group (Fig. 3.3). Protons and neutrons are part of a subset of the baryon group called nucleons that combine with the electron to create the atoms that form the periodic table of elements. The electron is part of the lepton group and is not constructed from quarks; its effect is beyond the scope of this particular work but other QCD calculations do attempt to model its influence.

Modifications of the thermodynamic control parameters, chemical potential μ and temperature T, directly influence the physical properties of Hadronic matter. Chemical potential

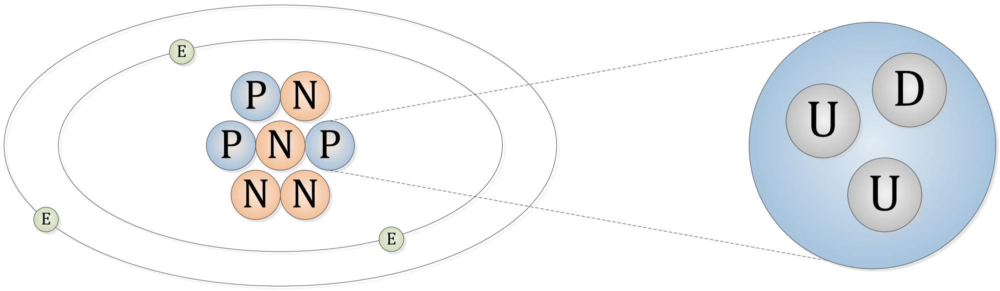


Figure 3.1: The strong nuclear force works at the sub-nuclear scale, examining particles within Hadrons.

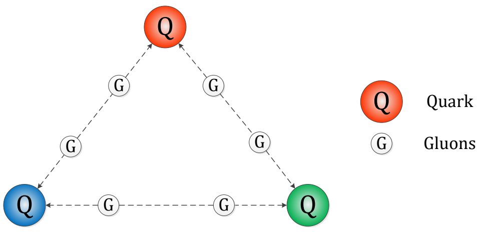


Figure 3.2: A baryon particle (Tri-quark configuration).

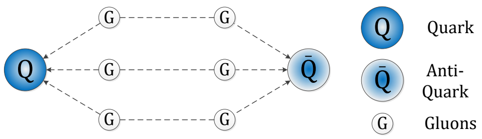


Figure 3.3: A meson particle (Quark-anti-quark configuration).

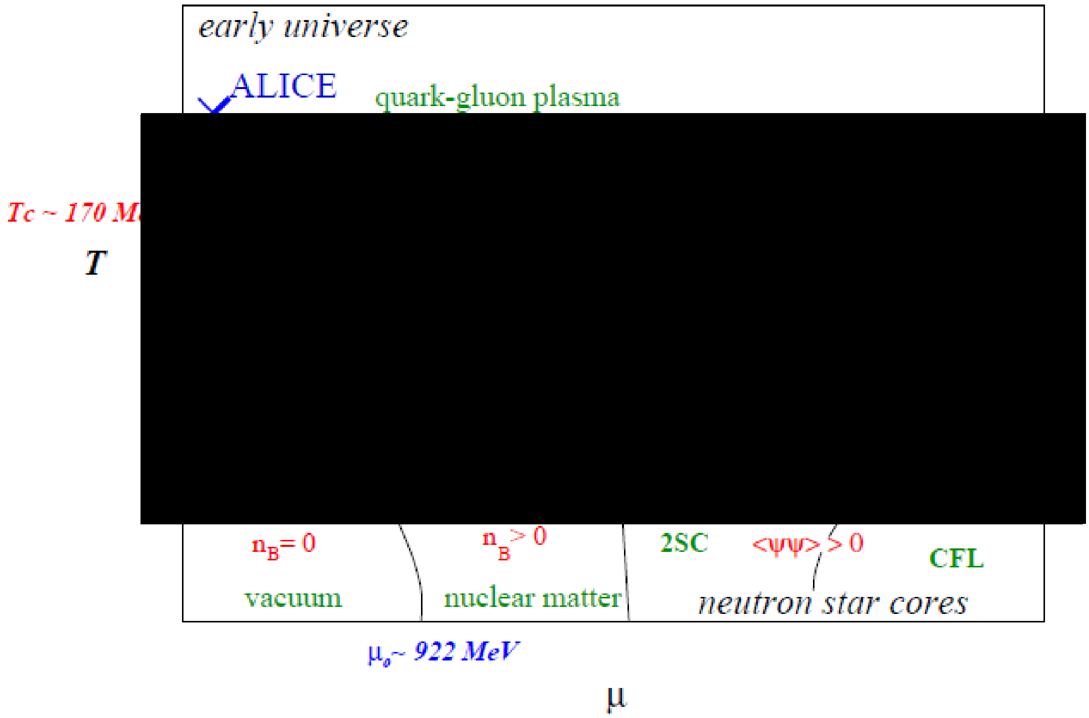

[redacted]

Figure 3.4: The phase diagram for Quantum Chromodynamics, taken from Hands [107].

represents the energy change as either a quark is added or an anti-quark is removed from a system. Hence, it provides a theoretical method of modifying the quark density in much the same way that temperature modifies the average particle energy of a system. A situation known as the “Quark Gluon Plasma” (QGP) is expected when matter is exposed to high temperatures; high chemical potentials lead to a form of matter known as “quark matter”.

Physicists use a phase diagram to capture the behaviour of quarks and gluons and to predict state changes as the two thermodynamic parameters vary. A proposed model given by Hands [107] (Fig. 3.4) suggests that neutron star cores are described by a system with a non-zero chemical potential. Output from real-world experiments, including ALICE, RHIC, and SPS, demonstrates how work in this area informs experiments in nuclear and particle physics. These studies collide atomic nuclei at velocities approaching the speed of light to create regions of very high energy and matter density.

Quarks and gluons appear impossible to observe in an unbound state. A single quark has never been observed and they are always found in *colour* neutral bound states [108], held together by the exchange of gluons. This phenomenon is known as colour confinement; unlike its electromagnetic force equivalent the energy needed to separate two or more bound particles increases indefinitely with distance. At the time of writing, no reliable analytical method can account for this calculation. Hence, in order to test theories of the quark model computer simulations are employed to model quark-gluon interactions. The process of computing Quantum Chromodynamics as discrete models is known as *lattice QCD*; this in itself is a field of continual development, an overview of which follows in Section 3.1.1.

3.1.1 QCD and high performance computing (HPC)

Lattice QCD simulations require a large quantity of computing power, often relying on multi-teraflop systems to perform the calculations. The large quantities of data and iterative processes required to generate simulations has extensive research invested into optimisation of the algorithms. Performance of different architectures varies greatly for this task; hence, studies have been carried out to rank those available. A 1995 study compared several platforms with the Numerical Wind Tunnel (NWT) supercomputer using a FORTRAN77 based simulation, adapted for parallel computation [109]. Despite the NWT machine being originally designed for Computational Fluid Dynamics (CFT), it was found to outperform many of the other super computers available at the time.

An alternative approach to adapting existing hardware was that taken by Italian theoretical physicists, who in 1984 formed their own collaborative group called APE, with the intention of producing supercomputers designed from the ground up for lattice QCD calculations [110]. The APE machines are programmed using conventional ANSI 99 C or TAO, a variation on FORTRAN with function overloading and object orientation additions. Much of the architectural design of the APE systems, of which there were four generations in 2005, is similar to IBM's own Blue Gene supercomputer [111]. It is suggested within the report that a significant amount of the findings from the lattice QCD community had been the result of simulations performed on hardware built by the APE group. Dedicated architectures can outperform mainstream supercomputing systems due to the ability to minimise latency in communication between closely mapped lattice sites. This is achieved by laying out the hardware in a way mapped to the lattice interconnecting structure.

However, the increase in processing power on mainstream desktop systems and workstations lead to an alternative method being developed beginning in 2000. The focus was instead on building a generalised framework for lattice QCD to be performed on a range of machine architectures, building upon the Chroma [112] software from the SciDAC and UKQCD research facilities. The idea was to provide a framework of message passing protocols between different machines, along with operators that could be performed across the lattice. Assembly code was used to provide highly optimised machine specific functions where performance issues were likely to cause bottlenecks. In order to provide cross platform support and web access to data a XML-based software layer was also provided.

Availability of massively parallel computing resources has increased in the past decade and this has seen a shift from earlier supercomputers with relatively few processors. The lattice QCD community has thoroughly researched porting existing algorithms to use MPI and GPU processing, focusing largely on Monte Carlo simulations.

Thermalisation is the process that iteratively sweeps across all sites on the lattice, generating new configuration based upon probabilistic sampling techniques. The ability to run many processes in parallel is most noticeable when dealing with large lattices, vastly improving upon processing time. Cucchieri et al. use the QCDMPI package to generate local lattices, sub-volumes of the main lattice composed of contiguous sites, to effectively evolve smaller hyper volumes on each node [113]. Generation times using this method saw a logarithmic decrease as the number of nodes were increased, with noticeable speed up found even on small hyper volumes.

A later study considers the optimal way to partition lattice sites, minimising messaging between nodes and evenly spreading computational load. Early MPI simulations used simple 1D splits, sites divided in a chequerboard pattern on a single axis, Cucchieri et al. [114] considers lattice splits in one, two and three dimensions. Sites marked as black are distributed to one pool of processor nodes and those marked red to another node pool, giving an even split of the data. High data locality reduces the need to send messages between different processing nodes as this proves to be a major performance bottleneck on a parallel implementation of Monte Carlo algorithm. An interesting finding is that the 1D decomposition methods appeared to be most favourable even when the number of nodes is vastly increased. Data partitioning is further considered when used to perform calculations with large sparse matrices in [115], taking advantage of improvements in processor cache sizes to achieve up to double the previous maximum computation speed. Gauge field compression is used to reduce the memory bandwidth for sending messages between nodes, as only the top two rows of the 3×3 matrix are passed. The bottom row is calculated as conjugates of the top row values using properties of special unitary group matrices.

Adapting large sparse matrix operators to GPUs is considered in [116], which notes that correct memory ordering is required for optimal performance. In GPU processing the major departure from distributed computing is that cores on a graphics card share memory with a number of others as part of a multiprocessor group. Therefore, to achieve optimal performance a multiprocessor group requires adjacent memory addressing. Latency issues introduced by accessing non-adjacent memory locations can result in stalling of a process. When this occurs the processing group may be swapped by the GPU with a new set of processing threads, thus adding extra computation time overheads.

A high performance alternative to using a single graphics card is to use a cluster equipped with multiple GPUs; however, this introduces additional factors to consider. The limiting factor, as with MPI, is bandwidth; multiple graphics cards share the same PCI express bus (PCI-E) for communication with the host CPU. QUDA [117], a CUDA based lattice QCD library, offers memory-saving features including gauge field compression and precision truncation. The use of QUDA is further adapted for situations where multiple GPU-equipped nodes communicate using InifiniBand in [118]. QCD Message Passing (QMP) allows calculations to be performed on hyper-volumes exceeding the size that a single GPU can handle on multiple networked GPUs. Computational throughput is also increased; some computations were able to complete with a linear increase in performance relative to the number of GPUs. However, other calculations suffer a performance drop-off at a critical number of nodes, suggesting that bandwidth problems were still restraining their full potential output. This was confirmed in a later study that compares the output of single and multi-node GPU clusters for the fat-link computation stage of a QCD simulation [119]. This indicated that the performance gains gradually dropped off as additional GPU-equipped nodes were added to the cluster. Gauge field compression techniques are used but there is still evidence of a bottleneck due to bandwidth limitations. It was found that partitioning the data along different planes has little effect on overall calculation speed.

3.2 Lattice structure

Kenneth Wilson was the first physicist to suggest that a discrete lattice could be used to model properties of quark-gluon fields [1]. In the following section we describe how the physical theory of Quantum Chromodynamics is translated into a form that can be modelled on a computer.

The lattice is a hyper-torus in Euclidean space-time, meaning that the three spatial dimensions and the time dimension are treated as equal. Periodic boundary conditions $f(x) = f(x - L_x)$ are used so that all sites in the lattice are translationally invariant — this means it is impossible to consider any site to be on the boundary of the system. As the lattice exists in four dimensions it can be convenient to think of the lattice of being made up of hyper-cubic cells, as shown in Figure 3.5. The scale at which lattice QCD acts is typically in the region of 2 to 4 femtometres (10^{-15}m), with state of the art simulations having a lattice spacing a of 0.02 to 0.04 femtometres.

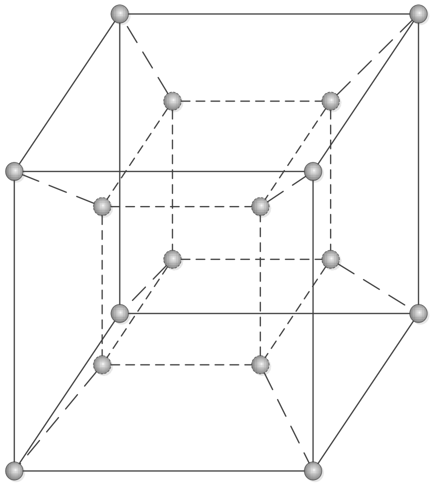


Figure 3.5: Arrangement of lattice sites for one 4D cell (dashed line represents t dimension).

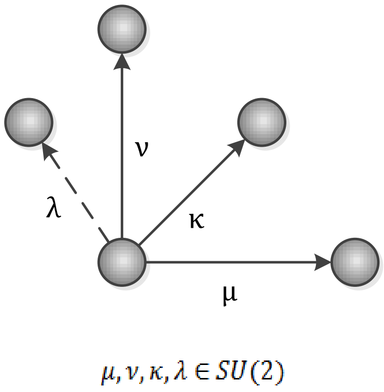


Figure 3.6: Placement of data is on the four lattice links from each site X.

Quarks are located on the lattice at positions with integer indices referred to as *sites*. In the remainder of this work we will label sites on the lattice as X where $X \in \mathbb{Z}^4$. In lattice QCD data is not directly placed on these sites, instead they represent the starting point for computations with a given origin. From each lattice site four *link variables* are used to model the gluon potential in the x, y, z and t directions between two sites. Link variables are represented as $U_\mu(X)$ i.e. the variable defined emerging from the site X in the direction of μ. It is also possible to use the notation $U_\mu^\dagger(X)$ to define the link emerging from site $X + \hat{\mu}$ in the reverse direction (see Fig. 3.10). Where required, additional link variables can also be represented using the Greek symbols ν, ρ, σ where each symbol denotes any space-time direction.

Each link variable is a member of the *special unitary group* of matrices, identified using

Gauge theory	n	Colours
$\mathsf{SU}(2)$	2	red / anti-red, blue / anti-blue
$\mathsf{SU}(3)$	3	red / anti-red, blue / anti-blue, green / anti-green

Table 3.1: Common gauge theory groups and their conventional representation of colour neutrality.

the notation $\mathsf{SU}(n)$. The value of n represents the number of charge colours used in the gauge theory, with true QCD defined with $n = 3$. However, in this work we use a simplified two colour model of the theory using $\mathsf{SU}(2)$ matrices (3.1). Colour is used in this context to parametrise the concept of colour neutrality used to explain confinement (see Table 3.1). One of the primary reasons for using a simplified model is that it allows us the freedom to vary the chemical potential of the system [120].

$$\mathsf{SU}(2) = \left\{ \begin{bmatrix} a & -\overline{b} \\ b & \overline{a} \end{bmatrix} : a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1, \det = 1 \right\} \tag{3.1}$$

$$where\ \overline{x}\ represents\ the\ complex\ conjugate\ (see\ Fig.\ 3.9).$$

Due to the discretisation of placing sites on a lattice, it is possible to calculate paths around sites in space and time. *Wilson loops* are one such observable that can be calculated on the lattice and represent any closed path, typically in a two-dimensional plane [121]. Wilson loops do not necessarily need to be square, neither are they required to be convex, but they must form a closed path. Examples of valid Wilson loops are given in Figs. 3.7 and 3.8. In this work we work with the simplest form of the Wilson Loop, the *plaquette*, which represents a unit Wilson loop in the plane (3.2).

$$U_{\mu\nu} = tr\left(U_\mu U_\nu U_\mu^\dagger U_\nu^\dagger\right) \tag{3.2}$$

$$where\ \mu\ and\ \nu\ represent\ links\ in\ any\ 2\ different\ space-time\ directions.$$

For any Wilson loop at a given location in space and time it is possible to obtain a scalar value referred to as the *Wilson action* (3.3). This value is calculated by multiplying each $\mathsf{SU}(2)$ link variable in path order. When traversing the lattice, we define movements in the reverse direction (Fig. 3.10) using the adjoint form of link variables, defined using as in (3.4). The matrix trace of a Wilson loop is fully gauge-invariant (3.5); we use this property as part of the lattice validation process (see Sec. 4.2.1.4). Gauge invariance means the scalar value remains the same even if the entire lattice is multiplied by a lattice of random unitary matrices defined independently at each site.

$$S_w = -\beta \times \Re\left(tr\left(\sum_{\mu<\nu} U_{\mu\nu}(X)\right)\right) \tag{3.3}$$

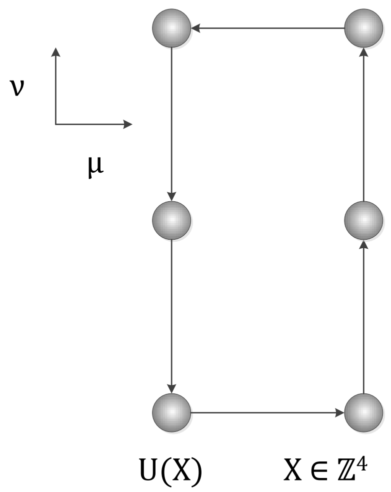


Figure 3.7: A rectangular Wilson loop.

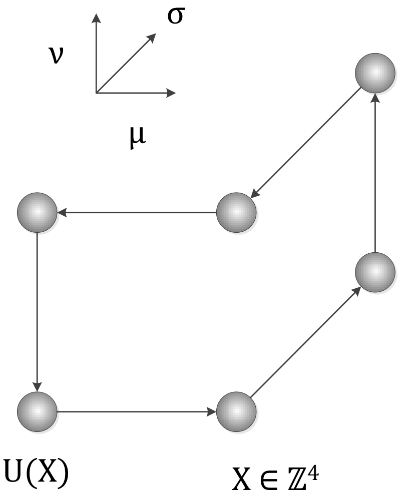


Figure 3.8: A non-planar Wilson loop.

$$U^\dagger = \begin{bmatrix} a - \mathrm{i}b & e - \mathrm{i}f \\ c - \mathrm{i}d & g - \mathrm{i}h \end{bmatrix} \quad \textit{with}\ U = \begin{bmatrix} a + \mathrm{i}b & c + \mathrm{i}d \\ e + \mathrm{i}f & g + \mathrm{i}h \end{bmatrix} \qquad (3.4)$$
$$\textit{where}\ \mathrm{i}^2 = -1$$

$$U_\mu(X) \mapsto \Omega_\mu(X) U_\mu(X) \Omega_\mu^\dagger(X + \hat{\mu}) \quad \textit{with}\ U_\mu, \Omega_\mu \in \mathsf{SU}(2) \qquad (3.5)$$

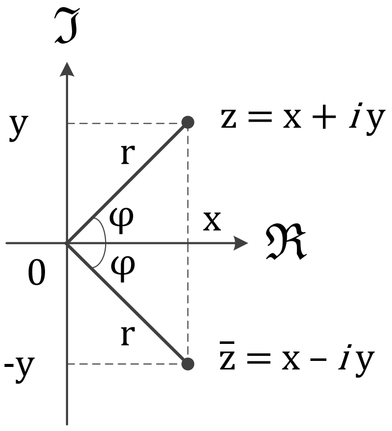


Figure 3.9: The complex conjugate $\bar{z}$, shown here on the complex plane, is a swapping of the sign of the imaginary part of the number.

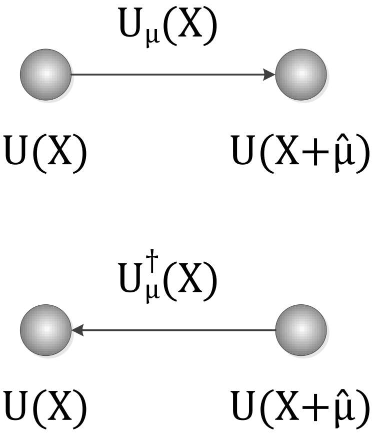


Figure 3.10: It is possible to traverse links in reverse by taking the adjoint of link variables (denoted by the † symbol).

3.3 Data generation

The core data sets used in this project were generated for use in previous lattice QCD studies [122]. Prior to visualisation there are many steps that must be undertaken to take the raw lattice configurations to a form suitable for indirect visualisation techniques. Some of the stages completed as part of this process generate additional data that can be visualised and analysed using the techniques presented in Chapters 6, 7, 8, and 9.

For the purposes of completeness, and to explain why we must work with ensemble averages, we give a brief overview of how lattice QCD data sets are generated using Monte Carlo methods in Sec. 3.3.1. We also make use of existing FORTRAN code to perform cooling of the generated data; a description of the cooling algorithm, along with an explanation of why this is performed is discussed in Sec. 3.3.2. Having introduced all of the necessary details for generating the data used in lattice QCD we discuss how this affects the size and complexity of data in Sec. 3.3.3.

3.3.1 Ensemble generation

Lattice QCD data sets are typically generated using Monte Carlo methods, allowing an ensemble of configurations to be computed and an observable average to be calculated. Algorithms used to generate the ensemble should be chosen in order to explore the configuration space in a efficient but un-biased way. The data analysed in this thesis was computed using a method known as the *Hybrid Monte Carlo algorithm.* Below we give a brief overview of the process by considering multiple aspects of the implementation, for a more complete explanation of the process we refer the reader to Chapter 16 of [123].

3.3.1.1 Markov chains

Use of a Markov chain allows multiple configurations to be generated using an initial set of conditions, or parameters. We can use the Markov chain to consider the state of the system at discrete simulation steps. However, the system state should be independent of all prior states apart from the one immediately proceeding it. From a set of configurations $\{C_{\tau_i}\}$ generated by the Markov chain, the observable average O from N configurations (3.6). By computing additional configurations it is possible to reduce the error to converge on the true value (3.7).

$$\langle O\rangle_N = \frac{1}{N}\sum_{i=1}^{N} O(C_{\tau_i}) + \varepsilon \tag{3.6}$$

where ε represents statistical error

$$\langle O\rangle_{true} = \langle O\rangle_N + \frac{1}{\sqrt{N}} \tag{3.7}$$

3.3.1.2 Metropolis Method

The Metropolis method [124] is applicable to any type of system and allows generation of a balanced set of configurations in lattice QCD. The core concept is that an acceptance constraint is placed upon the transition step within the Markov chain. The transition probability is given as $P_0(C \rightarrow C^{'})$, where C represents the current configuration and $C^{'}$ the proposed update.

Using this approach link variables are updated independently one or more at a time and an updated action $S(C^{'})$ is computed using the new lattice configuration. If the existing action $S(C)$ is lowered by the update, the configuration is immediately accepted. However, an increase will see the update accepted with the probability R determined by (3.8), otherwise it is discarded.

$$\begin{aligned} R &= \min\left\{1, e^{-\Delta S}\right\} \\ \Delta S &= \frac{S(C)}{S(C')} \end{aligned} \tag{3.8}$$

In most lattice QCD situations the number of link variables updated per step is minimised, as large updates result in large jumps in the action. If the number of links updated is too high the possibility of acceptance is reduced and the configuration space is explored slowly. Because of these limitations, newer algorithms were later devised for exploring the configuration space.

3.3.1.3 The Hybrid Monte Carlo algorithm

The Hybrid Monte Carlo algorithm is the current state of the art technique for generating "exact" quark configurations, building upon features of earlier algorithms. The *Hybrid algorithm* [125] represented an improvement in efficiency over previous techniques [126, 127, 128];

however, a systematic error was introduced by the finite time steps in the algorithm. Hence, to minimise the effect of error a Metropolis acceptance test is applied to output from the Hybrid algorithm. This technique was first applied to the Langevin algorithm for simulating fermions on a lattice [129] and later adapted for use in full QCD using the Hybrid algorithm [130, 131].

With these modification the Hybrid Monte Carlo algorithm is defined as the following series of steps:

1. Choose an arbitrary set of fields $\{\phi_i\}$.
2. Choose an initial set of momenta $\{\pi_i\}$ from the Gaussian ensemble.
3. Calculate an updated set of momenta $\tilde{\pi}$ using the Hybrid algorithm approach.
4. Iterate the molecular dynamics chain for a predefined number of time steps. The last step in the molecular dynamics chain is defined as $\{\phi_i^{'}, \pi_i^{'}\}$ obtained using the update step.
5. Using the probability (3.13) accept $\{\phi_i^{'}, \pi_i^{'}\}$ as the new configuration.
6. If the updated configuration $\{\phi_i^{'}, \pi_i^{'}\}$ is not accepted we keep the old configuration $\{\phi_i, \pi_i\}$ and repeat the cycle from step 2. Otherwise, we use the coordinate set $\{\phi_i^{'}\}$ to generate a new configuration.

The Hybrid Monte Carlo algorithm updates an entire configuration in each iteration. Hence, without careful consideration of parameters it can be susceptible to acceptance problems similar to those existing in the Langevin approach. However, even in the worst case scenario, where the acceptance level is too strict, it still represents an optimal traversal of the configuration space without a systematic error. Hybrid Monte Carlo represents a satisfactory approach towards generation of lattice QCD ensembles as the global updates are good for ergodicity decorrelation. Furthermore, it can be tuned to optimal acceptance rate and the lack of a time-step based update approach means the statistical error is virtually alleviated.

$$\phi_i(n+1) = \phi_i(n) + \varepsilon\tilde{\pi}_i(n) \tag{3.9}$$

$$\tilde{\pi}_i(n+1) = \tilde{\pi}_i(n) - \varepsilon\frac{\partial S[\phi]}{\partial\phi_i(n+1)} \tag{3.10}$$

$$\tilde{\pi}_i(n) = \pi_i(n) - \frac{\varepsilon}{2}\frac{\partial S[\phi]}{\partial\phi_i(n)} \tag{3.11}$$

$$\tag{3.12}$$

$$p = \min\left\{1, \frac{e^{-H[\phi^{'},\pi^{'}]}}{e^{-H[\phi,\pi]}}\right\} \tag{3.13}$$

where H is the Hamiltonian

3.3.2 Cooling Algorithm

The output from the Monte Carlo simulations described in Section 3.3.1 features noise in the form of short wavelength quantum fluctuations. The *cooling* algorithm attempts to smooth short wavelength quantum fluctuations, whilst maintaining long range physical interactions.

Instantons are one of the primary observables that physicists studying the QCD phase diagram look for in a lattice QCD simulation. The name comes from the fact that they have a finite extent in the time dimension and are able to appear and disappear, unlike real particles. In order to reveal the structure of *instanton* observables, the effect of quantum fluctuations must be minimised through cooling.

The appearance of instantons can be linked to the breaking of chiral symmetry, indicative of de-confinement. They are a described as being a pseudo-particle and are a method for understanding interactions in four-dimensional volumes. The concept of instantons and their definition as pseudo-particles originate in [132], as non-perturbative solutions to the Yang-Mills equations of motion. Perfect cooling would present a stable "pure" instanton configuration whilst minimising the action.

Instantons are generally able to persist in the data even when subject to these noise reduction techniques [133]; however, over use of the algorithm can destroy the instanton-anti-instanton pairs that are intended to be measured [134]. Due to the discrete nature of the lattice and the fact the size of an instanton can vary, it is possible to shrink objects until they disappear. The effect of annihilation of an observable by over cooling is referred to an object "falling through the lattice" on account of the discrete spacing between lattice sites (a). As the cooling algorithm is implemented as a iterative cycle it is possible to evaluate the overall effect at multiple discrete steps. If an object falls through the lattice a loss of information results in misleading output in the overall topological charge density measure (see Figs. 3.11 and 3.12). It is therefore extremely important to cool the system to a suitable level to observe the instantons. When a configuration is continued to be cooled, the eventual outcome is a trivial vacuum configuration in which all instantons have been destroyed.

3.3.2.1 Process description

In order to perform a step in the cooling algorithm the following procedure is implemented by sweeping across each site in the lattice. First, we compute the Euclidean action originating at the site. In the implementation of cooling algorithm used in this work this takes the form of computing *Wilson plaquettes*. This serves as an initial benchmark of the pre-cooled lattice.

We then begin the process of varying the lattice by choosing a single link $U_\mu(X)$ originating from the site, changing the $\mathsf{SU}(2)$ lattice in such a way as to lower the action at the site. In order to measure the effect of varying this variable we compute the *staple* of link, made up of links belonging to six neighbouring space-time plaquettes (Fig. 3.13). The value of the staple W is a non-unitary matrix obtained by summing the six plaquettes. The staple is then renormalised into the $\mathsf{SU}(2)$ group using (3.14), and an updated action S_E computed for the site (3.15). If we assume that all other links from the site are fixed we can assign the link an updated value that is the inverse of the normalised staple (3.16).

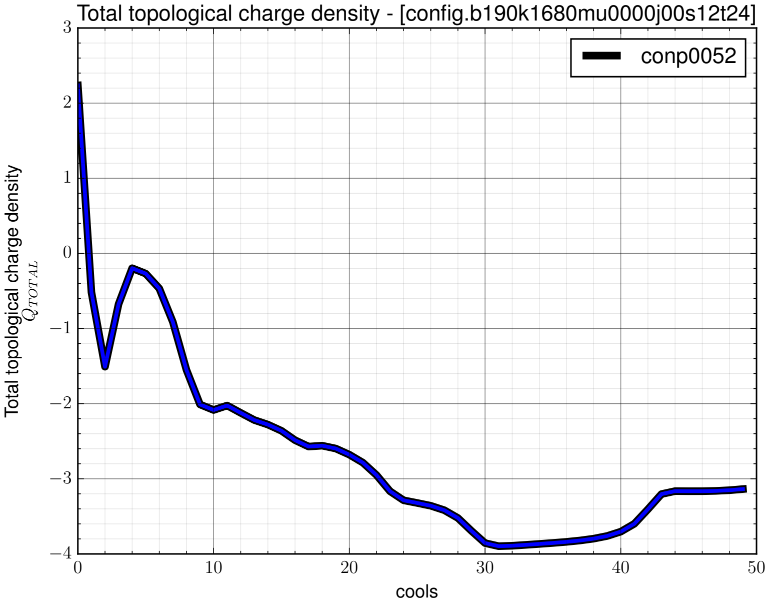


Figure 3.11: An (anti-)instanton being annihilated by over cooling is characterised by an integer change in the total topological charge density.

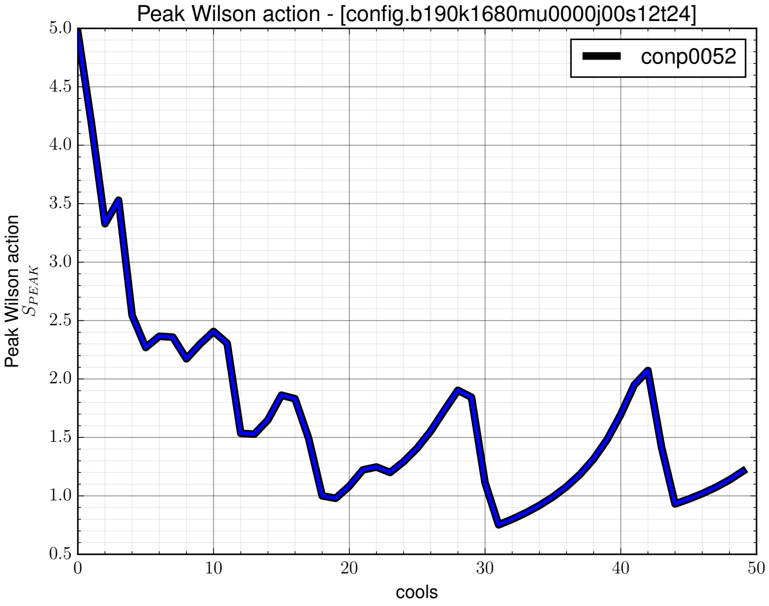


Figure 3.12: Over-cooling also coincides with a large drop in the peak Wilson action of the lattice.

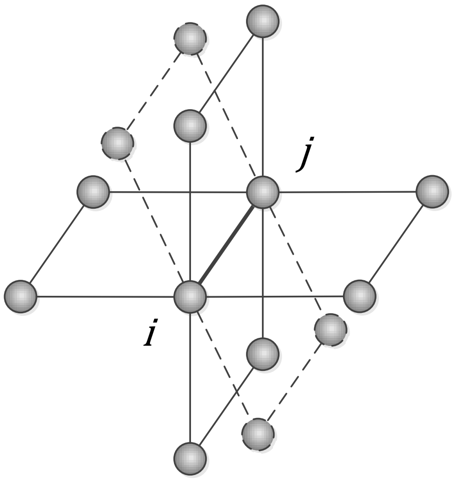


Figure 3.13: The staple can be visualised as the six space-time plaquettes that include the link between sites *i* and *j*.

$$\hat{W} = \frac{1}{\sqrt{\det W}} W \qquad (3.14)$$

$$\textit{where}\ W \notin \mathsf{SU}(2),\ \hat{W} \in \mathsf{SU}(2)$$

$$S_E = -\beta \sqrt{\det W}\, Tr(U\hat{W}) \qquad (3.15)$$

$$U = \hat{W}^{-1} \qquad (3.16)$$

A complete sweep across all sites in the lattice performing the previous steps is known as a cooling iteration. The next step is to evaluate the effect the cooling has had on the lattice. In order to do this we compute the net action and topological charge density for the lattice. A “successful” cooling iteration will see a drop in the net action without a substantial change in the net topological charge density.

3.3.2.2 Implementation

The cooling algorithm utilised in this thesis is a pre-existing implementation written in FORTRAN [134]. However, for visualisation purposes the source code has been modified to output additional information whilst running the cooling algorithm. We believe this can help assist physicists looking for observables on the lattice and takes the form of the estimated position of minima and maxima in the topological charge density and action density fields. We can then display this information alongside the available data sets when using the field generation tool

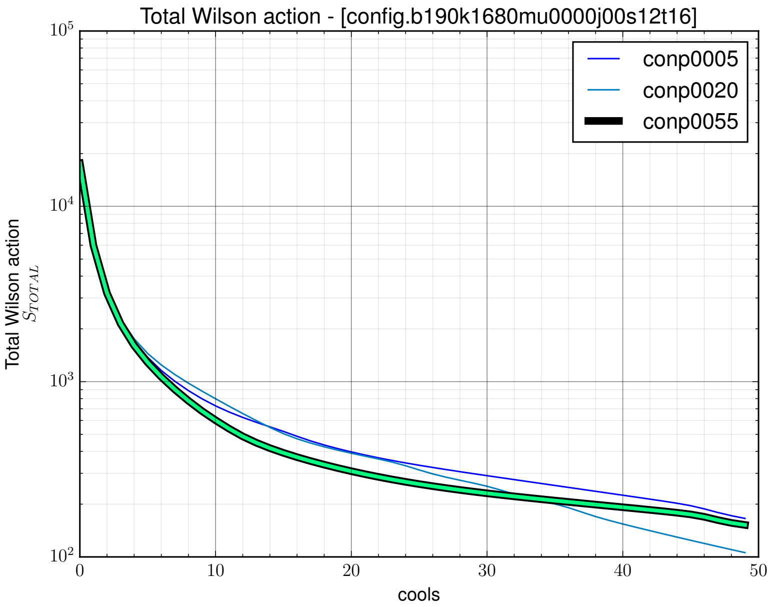


Figure 3.14: The Wilson action drops off throughout the cooling process.

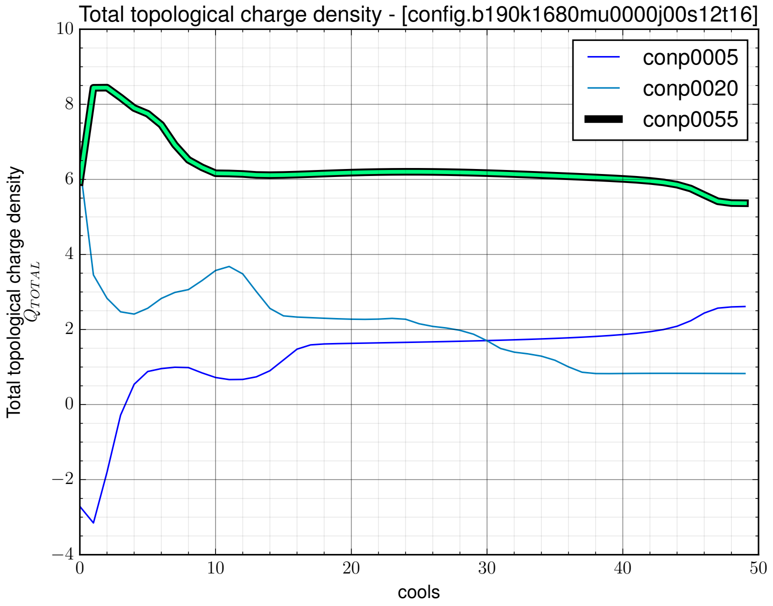


Figure 3.15: Flat regions in the total topological charge density show stable lattice configurations under cooling.

Lattice	Sites	Links	Values	Data size[†]	File size[‡]
$12^3 \times 16$	27,648	110,592	884,736	7,077,888 B	7,077,904 (7.08 MB)
$12^3 \times 24$	41,472	165,888	1,327,104	10,616,832 B	10,616,848 (10.62 MB)
$16^3 \times 8$	32,768	131,072	1,048,576	8,388,608 B	8,388,624 (8.39 MB)
$16^3 \times 32$	131,072	524,288	4,194,304	33,554,432 B	33,554,448 (33.55 MB)

[†] Stored at double precision (8 bytes).

[‡] Each file includes a 4 byte header, 4 byte footer, and a 8 byte random number seed.

Table 3.2: File sizes for uncompressed lattice configurations.

described in Section 10.2.1. One purpose of this data is to visualise the cooling process as the output of individual configurations, assisting in the selection of optimally cooled configurations for visual analysis or ensemble studies. The effect cooling has on the net action density and topological charge density is shown in Figures 3.14 and 3.15.

In a non-discrete formulation of QCD the topological charge density Q is a topological invariant it is only able to take on positive / negative integer values. Configurations with differing topological charge densities are non-homotopic and represent a jumping of the lattice from one state to another. These jumps are what characterise the destruction of instanton observables on the lattice. Hence, plateaus in the topological charge density graph represent stable states under cooling (see Fig. 3.15). As lattice QCD is a discrete version of the theory, in practice plateaus in the topological charge density assume approximate integer values.

The FORTRAN code has also been modified to output the lattice configuration at each iteration of the cooling algorithm. This allows the computation of lattice fields at various points in the cooling chain, allowing visualisation and topological analysis to be used to evaluate the cooled lattice. Physicists are able to specify the desired number of cooling iterations for the lattice; typically this would be in the region of 20-30 cools, as beyond this there lies the possibility to degrade important lattice observables. From the output configurations we can then begin to compute different lattice operators as scalar fields for visualisation (see Section 4.3).

3.3.3 Data complexity

Most lattice QCD computations take place on small four dimensional lattices with the size given in the form $d_s^3 \times d_t$ where d_s expresses the spatial lattice extents and d_t the temporal extent. In this thesis we work with $12^3 \times 16$, $12^3 \times 24$, $16^3 \times 8$, and $16^3 \times 32$ lattices. Despite the relatively small lattice sizes the processes of ensemble generation and cooling contribute to the overall complexity of the data.

Each lattice site consists of four separate link variables, each of these is an $\mathsf{SU}(2)$ matrix. In order to store each link variable 8 floating point values are required, these represent the real and imaginary parts of each matrix element. Gauge compression, which takes advantage of symmetries existing in the $\mathsf{SU}(2)$ matrices, can be used reduce the storage overhead for each link variable by half. File sizes for the lattices used in this work are given in Tables 3.2 and 3.3.

Lattices are rarely considered by physicists in isolation, instead they are analysed as part of a Markov chain of configurations generated from a fixed set of thermodynamic control

Lattice	Sites	Links	Values	Data size†	File size‡
$12^3 \times 16$	27,648	110,592	442,368	3,538,944 B	3,538,960 B (3.54 MB)
$12^3 \times 24$	41,472	165,888	663,552	5,308,416 B	5,308,432 B (5.31 MB)
$16^3 \times 8$	32,768	131,072	524,288	4,194,304 B	4,194,320 B (4.19 MB)
$16^3 \times 32$	131,072	524,288	2,097,152	16,777,216 B	16,777,232 B (16.78 MB)

† Stored at double precision (8 bytes).

‡ Each file includes a 4 byte header, 4 byte footer, and a 8 byte random number seed.

Table 3.3: File sizes for Gauge compressed lattice configurations.

Lattice	Sites	Links	Uncompressed size
$12^3 \times 16$	1,382,400	5,529,600	353,895,200 B (353.90 MB)
$12^3 \times 24$	2,073,600	8,294,400	530,842,400 B (530.84 MB)
$16^3 \times 8$	1,638,400	6,553,600	419,431,200 B(419.43 MB)
$16^3 \times 32$	6,553,600	26,214,400	1,677,722,400 B (1.68 GB)

Table 3.4: Sizes for an entire ensemble of 50 configurations stored in uncompressed format.

Lattice	Sites	Links	Uncompressed size
$12^3 \times 16$	41,472,000	165,888,000	10,616,856,000 B (10.62 GB)
$12^3 \times 24$	62,208,000	248,832,000	15,925,272,000 B (15.92 GB)
$16^3 \times 8$	49,152,000	196,608,000	12,582,936,000 B (12.58 GB)
$16^3 \times 32$	196,608,000	786,432,000	50,331,672,000 B (50.33 GB)

Table 3.5: Sizes for an ensemble of 50 configurations, each cooled for 30 iterations.

parameters. This is referred to as an ensemble of configurations, analysed as a group average, to enable the testing of hypotheses about a particular set of starting parameters. An entire ensemble of 50 configurations can result in file sizes in excess of 1 GB, as shown in Table 3.4. In this work many of the ensembles consist of either 50 or 100 separate lattice configurations.

Cooling is applied as an iterative process to each of the raw lattices in an ensemble, resulting in a new modified lattice for each iteration. In this work many of the configurations are cooled for 30 iterations of the algorithm — producing 30 unique lattices in addition to the lattice in its initial uncooled state. For some studies carried out in this thesis, where we study the effect of cooling on the topology of the data, the number of cooling iterations is extended to 50 cycles. Due to the quantity of data this process tends to create, see Table 3.5, a physicist may typically decide to discard some of the configurations at lower levels of cooling. For the purposes of this work we retain all data to enable a complete assessment of the evolving topology of the lattice to be studied.

In order to test a hypothesis about the effects of thermodynamic control parameters on the lattice each experiment is carried out on multiple ensembles. For each of the four lattice sizes studied in this thesis (referring to the spatio-temporal extents of the lattice) there are multiple ensembles at differing levels of chemical potential μ. The actual number of different chemical

Lattice	Ensembles	Sites†	Links†	Uncompressed size†
$12^3 \times 16$	15	622,080,000	2,488,320,000	159,252,840,000 B (159.25 GB)
$12^3 \times 24$	26	1,617,408,000	6,469,632,000	414,057,072,000 B (414.06 GB)
$16^3 \times 8$	4	196,608,000	786,432,000	50,331,744,000 B (50.33 GB)
$16^3 \times 32$	13	2,555,904,000	10,223,616,000	654,311,736,000 B (654.31 GB)

† Based upon 50 configurations per ensemble each cooled for 30 iterations.

Table 3.6: Data sizes for an ensemble study at multiple levels of chemical potential μ.

Lattice	Values	Size (at double precision)
$12^3 \times 16$	27,648	221,184 B (0.22 MB)
$12^3 \times 24$	41,472	331,776 B (0.33 MB)
$16^3 \times 8$	32,768	262,144 B (0.26 MB)
$16^3 \times 32$	131,072	1,048,576 B (1.04 MB)

Table 3.7: Complexity of a single scalar field defined in $\mathbb{R}^4$.

Lattice	Ensembles	Values	Size (at double precision)
$12^3 \times 16$	15	622,080,000	4,976,640,000 (4.98 GB)
$12^3 \times 24$	26	1617,408,000	12,939,264,000(12. 94 GB)
$16^3 \times 8$	4	196,608,000	1,572,864,000 (1.57 GB)
$16^3 \times 32$	13	2,555,904,000	20,447,232,000 (20.48 GB)

Table 3.8: Complexity of an $\mathbb{R}^4$ scalar field when evaluated for 50 configurations each cooled for 30 iterations.

potentials held for each data set varies greatly between lattices (Tab. 3.6). The variation in the quantities of chemical potential ensembles for each lattice is due to the hypotheses being tested in prior experiments.

The complexity values given so far are for raw lattice configurations — this forms a basis for the scalar field computations in lattice QCD. For each configuration a range of scalar fields can be computed from the raw lattice data. Each of these fields are computed as path ordered multiplications of closed loops from lattice sites in various configuration. These computations are explained in detail in Section 4.3.

Most of the fields studied in this work exist as four dimensional arrays of double precision scalar values. When considered in isolation each of these fields are moderate in size (Tab. 3.7). However, when evaluated as an ensemble study to examine the effects of different thermodynamic control parameters the data size quickly escalates (Tab. 3.8). The Polyakov loop (Sec. 4.3.3) flattens the temporal dimension of the data at each lattice site to produce a three dimensional scalar field (Tables 3.9, 3.10).

Lattice	Values	Size (at double precision)
$12^3 \times 16$	1,728	13,824 B (13.82 MB)
$12^3 \times 24$	1,728	13,824 B (13.82 MB)
$16^3 \times 8$	4,096	32,768 B (32.77 MB)
$16^3 \times 32$	4,096	32,768 B (32.77 MB)

Table 3.9: Complexity of the $\mathbb{R}^3$ Polyakov loop field for a single configuration.

Lattice	Ensembles	Values	Size
$12^3 \times 16$	15	38,880,000	311,040,000 (0.31 GB)
$12^3 \times 24$	26	67,392,000	539,136,000 (0.54 GB)
$16^3 \times 8$	4	24,576,000	196,608,000 (0.20 GB)
$16^3 \times 32$	13	79,872,000	638,976,000 (0.64 GB)

Table 3.10: Complexity of the $\mathbb{R}^3$ Polyakov loop scalar field when evaluated for 50 configurations each cooled for 30 iterations.

3.4 Visualisation in lattice QCD

The traditional use of visualisation in lattice QCD is often limited to dimension independent forms such as line graphs to plot properties of simulations as part of a larger ensemble. Visualisation of a space-time nature is less frequently observed but does occasionally occur in literature on the subject. Surface plots were used by Hands [135] to view 2D slices of data and give a basic understanding of energy distributions in a field.

Application of visualisation of QCD in higher dimensions first appear in the form of three dimensional plots [136] of instantons showing their correlation to other lattice observables (Fig. 3.16). Primarily, these were monopole loops, instantons that persist throughout the entire time domain of a simulation. Prior to visualisation, simulations are subjected to iterative cooling to remove noise whilst keeping the desired observables intact. Visualisation is a useful tool in this case to confirm that monopole loops and instantons are able to co-exist in a simulation,

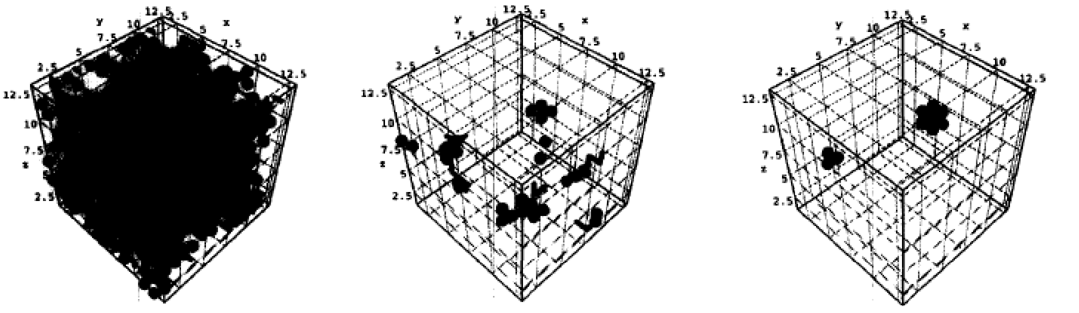


Figure 3.16: Examples of instanton observables after 0, 15 and 30 cooling sweeps by Feurstein et al. [136].

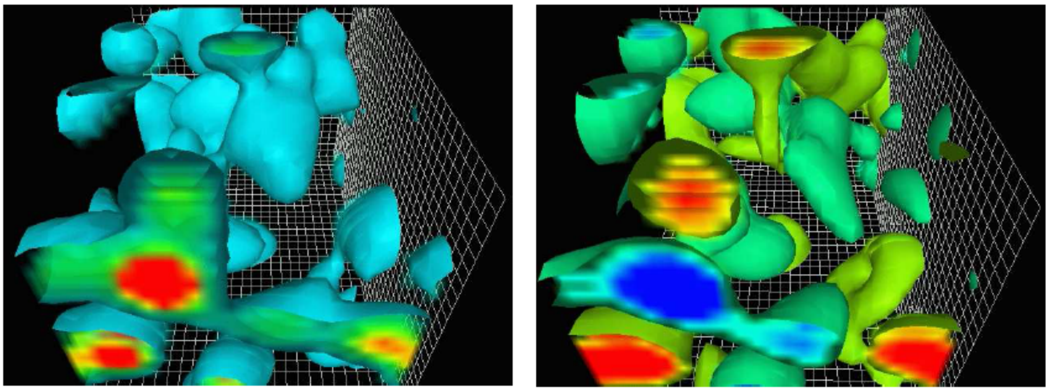


Figure 3.17: The action density (left) and topological charge density (right) after 30 cooling sweeps rendered using isosurfaces by Leinweber [137].

a fact that can help theoretical physicists understanding of the QCD vacuum structure.

A more detailed visualisation, making use of established computer graphics techniques, was performed by Leinweber [137] as a way of conveying to the physics community that visualisation of large data sets would help their understanding of QCD. An area of lattice QCD investigated is the view that instanton-anti-instanton pairs can attract and annihilate one another during cooling phases. Topological charge density and Wilson action measurements are used to show separate spherical (anti-)instantons in the lattice hyper-volume. Using animation it is possible to see that instantons in vicinity of anti-instantons deform and merge (Fig. 3.17). Leinweber explains the need for sensible thresholds when discarding isosurfaces that have action densities below a critical value as their disappearance can be incorrectly interpreted as them falling through the lattice.

More recently DiPierro et al. [138, 139, 140] produced several visual designs as part of a larger project with the aim of unifying various active lattice QCD projects. Expectations of the work were that visualisations could be used to identify errors with the simulation process, along with increasing understanding of lattice QCD. Many existing file structures used within the lattice QCD community could be processed and output using the VTK, this could then be rendered using a number of pre-existing visualisation tools (Figure 3.18), including VisIt [141] and Paraview [142]. In addition, the evolution of instantons over Euclidean time spans could be rendered into movies. A web front end application and scripting interface was also provided that allowed users to set up an online processing chain to analyse simulations on a large scale. Despite the apparent success of the work it would appear that it is no longer maintained and many of the links present with the project literature are inaccessible.

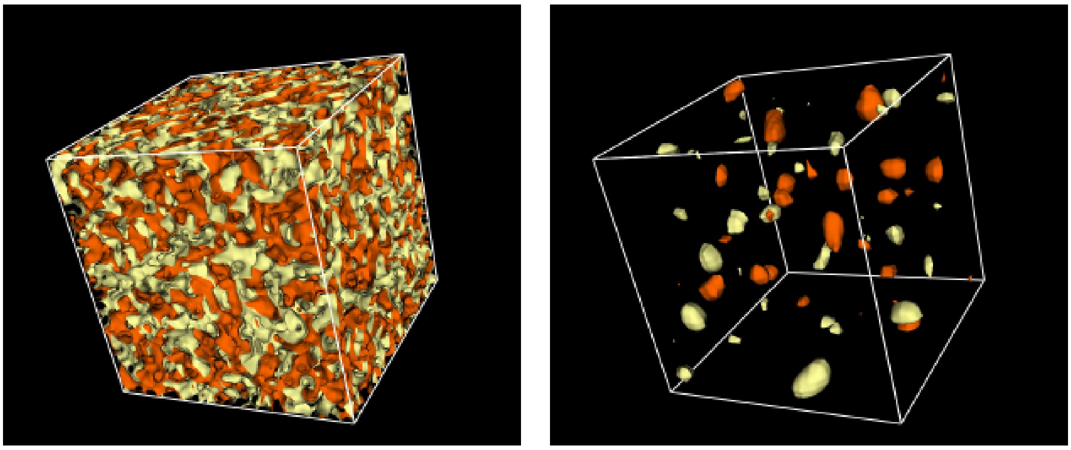


Figure 3.18: Topological charge density at different levels of cooling using techniques presented by Di Pierro et al. [140].

3.5 Summary

In this chapter we have presented an introduction to the quantum field theory used by physicists to model the strong nuclear force. We have also shown how physicists use a discrete formulation of the theory, known as lattice Quantum Chromodynamics, to model a phase diagram of matter under varying thermodynamic control parameters.

We have also reviewed visualisation techniques currently used by domain scientists to view and analyse their data. In Chapter 4 we use these techniques to create an initial set of visualisations, allowing us to validate our own computations on the lattice.

Chapter 4

Computation and visualisation of lattice observables using isosurfaces

Contents

Chapter 3 introduced the theory of lattice QCD, in this chapter we implement a model of the lattice. This proved to be a challenging process, as the data was written by FORTRAN code in an unknown file structure. The model enables us to assign values from pre-computed data sets in order to computer a number of lattice observables. In order to assist in validation of the output we were able to use visualisation techniques, including isosurfacing and DVR, to inspect the output of the algorithms.

In Section 4.1 we describe the process of assigning the raw QCD configurations to the lattice. We discuss the key role of data validation techniques in Section 4.2. Finally, the various lattice fields are introduced in Section 4.3, where we also present initial visualisations of each field.

4.1 Assignment of configuration data to lattice

Prior to visualisation the (un-)cooled lattices must be processed to generate the desired scalar field observables. In this section we focus mainly on the theoretical aspects of the process including parsing of raw configurations, assignment of $\mathsf{SU}(2)$ matrices to the lattice, validation, and generation of field observables. A detailed description of how the user is able to perform this process using a visual application is given in Section 10.2.1.

4.1.1 Parsing of configuration files

Lattice configurations typically come from a number of different sources; in this project we have focused on those generated on a Blue Gene system [122]. The framework described in this thesis has also been applied to data generated by the DiRAC collaboration that outputs its configurations in a similar format. The range of data sources means it is not unusual to receive configurations that are stored in big or little Endian formats. In order to handle all possibilities we allow the user to specify the desired Endianness as part of the file parsing process. This is achieved by doing a byte swap on the input as required; a simplified comparison of integer values in big and little Endian formats is given in Figs. 4.1 and 4.2. Incorrect Endianness of the output lattice values is detectable using the validation process described in Section 4.2.1.

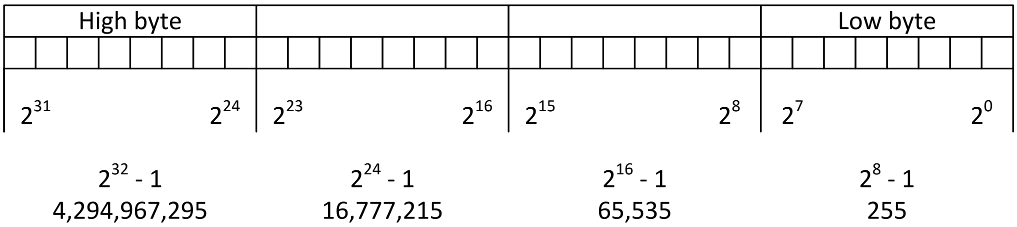


Figure 4.1: A 32-bit unsigned integer stored in Big-Endian format.

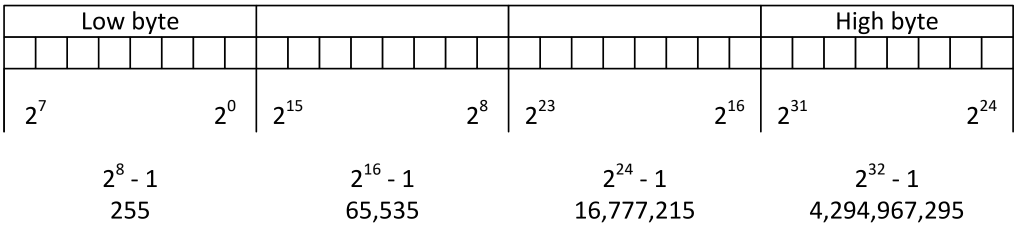


Figure 4.2: A 32-bit unsigned integer stored in Little-Endian format.

Lattice configurations can be saved at varying precisions, depending on the source. In order to understand if a configuration is stored in single or double precision format, calculations can be performed to estimate the expected file size. The number of sites (in four dimensions) used to represent a specific ensemble of configurations is known in advance; hence, we use this to calculate the required number of complex matrix link variables, represented as complex matrices. Many lattice QCD experiments also output their data in *gauge compressed* formats to reduce file size. This relies upon the Hermitian properties of $\mathrm{SU}(2)$ matrices, where a matrix is equivalent to its own conjugate transpose (4.1). Gauge compressed configurations (4.2) are relatively simplistic to identify as the expected file size will be just short of half that of an uncompressed configuration (4.3).

$$\begin{aligned} u_{ij} &= \overline{u_{ji}} \\ U &= \overline{U^T} \end{aligned} \tag{4.1}$$

$$U = \begin{bmatrix} u_{11} & u_{12} \\ -\overline{u_{12}} & \overline{u_{11}} \end{bmatrix} \quad with\, u_{11}, u_{12} \in \mathbb{C} \tag{4.2}$$

$$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \quad with\, u_{11}, u_{12}, u_{21}, u_{22} \in \mathbb{C} \tag{4.3}$$

With variable precision and the compressed nature of a file determined, values are parsed into interim storage arrays ahead of further processing. If the data has been gauge compressed two arrays (U_{11}, U_{12}) are required to represent complex elements of the link variables, otherwise four arrays (U_{11}, U_{12}, U_{21}, U_{22}) are used. Each array encodes the real and imaginary floating point values in an interlaced form.

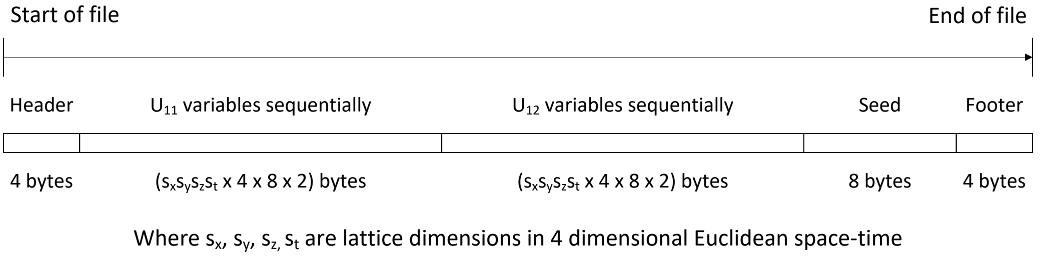

Start of file
End of file
Header
U_{11} variables sequentially
U_{12} variables sequentially
Seed
Footer
4 bytes
($s_x s_y s_z s_t$ x 4 x 8 x 2) bytes
($s_x s_y s_z s_t$ x 4 x 8 x 2) bytes
8 bytes
4 bytes
Where s_x, s_y, s_z, s_t are lattice dimensions in 4 dimensional Euclidean space-time

Figure 4.3: File structure of a gauge compressed $\mathsf{SU}(2)$ lattice.

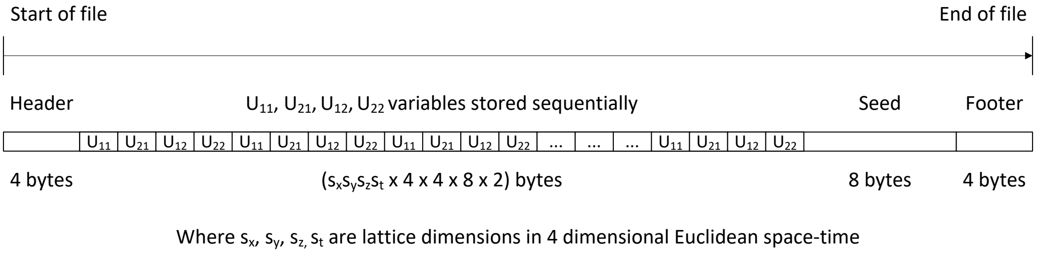

Start of file
End of file
Header
U_{11}, U_{21}, U_{12}, U_{22} variables stored sequentially
Seed
Footer
U_{11} U_{21} U_{12} U_{22} U_{11} U_{21} U_{12} U_{22} U_{11} U_{21} U_{12} U_{22} ... ... U_{11} U_{21} U_{12} U_{22}
4 bytes
($s_x s_y s_z s_t$ x 4 x 4 x 8 x 2) bytes
8 bytes
4 bytes
Where s_x, s_y, s_z, s_t are lattice dimensions in 4 dimensional Euclidean space-time

Figure 4.4: File structure of an uncompressed $\mathsf{SU}(2)$ lattice.

The method used to extract the floating point values from a configuration file varies depending on gauge compression. The FORTRAN file structure for a compressed $\mathsf{SU}(2)$ lattice is given in Fig. 4.3; filling the U_{11} and U_{12} data arrays is a matter of copying the data from the file as a block, applying byte swapping as required. For uncompressed files values for each matrix are stored sequentially in the file, which are looped over and filtered into the arrays U_{11},

U_{12}, U_{21} and U_{22} for assignment to the lattice in the next step. If the file has been correctly parsed it is possible to check that the header and footer bytes match.

Both compressed and uncompressed configurations contain a single 64-bit integer seed value before the file footer. This represents the seed of the random number generator for the Markov chain used in the configuration generation process. For visualisation purposes we do not require this information; however, it can be used by physicists to repeat the ensemble generation process, if required.

4.1.2 Assignment of $\mathsf{SU}(2)$ matrices to lattice structure

With a configuration loaded from the file the values must be assigned as $\mathsf{SU}(2)$ link variables on the lattice. The parsed configuration data exists as either two or four one-dimensional arrays of complex numbers dependent on gauge compression. Assignment of the raw data to the correct lattice links is a complex operation as the site and link ordering in the file cannot be assumed due to the variety of sources the data can originate from. However, it is possible to validate that the lattice links have been correctly assigned using methods given in Section 4.2. In particular, computing average plaquettes for the lattice in each plane can identify if the links are correctly aligned.

link index (d)	space-time axis
0	x
1	y
2	z
3	t

Table 4.1: Mapping of link indices to space-time directions used throughout this work.

The configurations typically used in this thesis are stored in column major order due to their production from FORTRAN code. For the purposes of assigning the data to the lattice the one-dimensional zero-based indices map to a five-tuple of zero-based indices of the form (x,y,z,t,d), where the fifth index (d) represents the link direction as given in Table 4.1. We can therefore directly map an offset in the one-dimensional input array (i) to the relevant link on the lattice using five nested *for* loops; the innermost loop represents the x coordinate and outermost loop the link direction d. Assignment of the input to $\mathsf{SU}(2)$ lattice links is as given in (4.4) for gauge-compressed data or (4.5) for uncompressed configurations.

$$f(x,y,z,t,d) = \begin{bmatrix} u_{11}(i) & u_{12}(i) \\ -\overline{u_{12}(i)} & \overline{u_{11}(i)} \end{bmatrix} \tag{4.4}$$

$$f(x,y,z,t,d) = \begin{bmatrix} u_{11}(i) & u_{12}(i) \\ u_{21}(i) & u_{22}(i) \end{bmatrix} \tag{4.5}$$

$$with\ i\ =\ \{x \in \mathbb{Z} \mid 0 \leq x < L_x L_y L_z L_t \times 4\},\ d\ = \{x \in \mathbb{Z} \mid 0 \leq x \leq 3\}$$

4.2 Data validation

Due to the complexity of the data being handled in this thesis, validation is key to correct representation of the observables. We therefore handle the validation procedure by combining methods from the underlying physics, and from computer science, as detailed in the following section. It is also possible to check periodically that the data conforms to known truths of lattice configurations.

4.2.1 Preparation of simulated configurations

Parsing of a QCD configuration requires successfully passing a number of requirements in order to be considered valid. This can be broken down into a number of steps (see Figure 4.5), as detailed below:

4.2.1.1 Check number of values

It is possible to validate that the correct number of values exist in a configuration file by examining the file structure, as shown in Figures 4.3 and 4.4. The most likely cause for an invalid number of values being present in a file is variations in the lattice extents, or the use of gauge compression.

4.2.1.2 Integrity check of values

Endianness of data can be determined by assessing each value read into the interim storage arrays. All values, real and imaginary, must fall into the range ±1; if this is not the case it is an indication that the Endianness is incorrect and a byte swap can be performed.

A further check calculates that the sum of the squares of each array element (4.6) or (4.7) correctly sums to 1. This confirms that values are in the correct sequence for constructing SU(2) matrices and that assignment of the lattice link variables can begin. Failure at this stage of processing is an indication of an offset occurring in the byte stream used to read in the file.

$$\Re(u_{11})^2 + \Im(u_{11})^2 + \Re(u_{12})^2 + \Im(u_{12})^2 = 1 \tag{4.6}$$

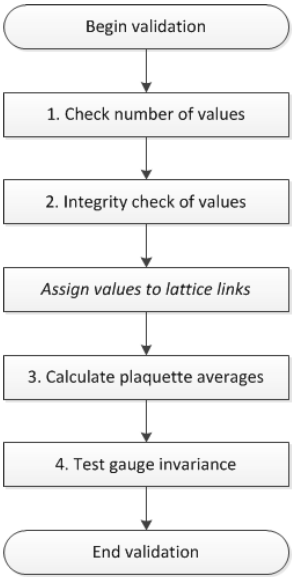

Begin validation
1. Check number of values
2. Integrity check of values
Assign values to lattice links
3. Calculate plaquette averages
4. Test gauge invariance
End validation

Figure 4.5: The process used to determine if a configuration has been correctly parsed.

$$\begin{aligned}&\Re(u_{11})^2+\Im(u_{11})^2+\Re(u_{12})^2+\Im(u_{12})^2\\&+\Re(u_{21})^2+\Im(u_{21})^2+\Re(u_{22})^2+\Im(u_{22})^2=2\end{aligned}\tag{4.7}$$

4.2.1.3 Check plaquette averages

With data assigned to the links of the lattice it is necessary to confirm that all links are correctly orientated. This is determined by calculating the plaquette (4.11) average in each plane where μ and ν are determined by the direction of lattice traversal (Fig. 4.6). Expected values are determined for the averages using statistical mechanics methods discussed in [1]. In this instance all possible plaquette combinations for a site (in 4 dimensions) are calculated and averaged in each plane.

4.2.1.4 Test gauge invariance

A final check that the lattice is correctly configured is to test it for *gauge invariance*. This requires the generation of a second lattice of random $\mathsf{SU}(2)$ site variables $\Omega(X)$, with the same dimensionality as the data set. Each site on the lattice configuration is transformed $U_\mu(X) \mapsto \Omega_\mu(X)U_\mu(X)\Omega^\dagger_\mu(X+\hat{\mu})$. Average plaquettes are then recomputed and compared to the original data. A correctly stored and orientated lattice will return the same average plaquette values following the transformation.

4.2.2 Conformation to $\mathsf{SU}(n)$ group structure

Certain properties of the $\mathsf{SU}(2)$ group can be used to ensure variables remain valid whilst performing calculations. These assertions are used as pre-condition and post-condition checks during calculations. The design by contract approach helps to document the code whilst catching errors at an early stage. A simple test is multiplication of the matrix by its *adjoint* (3.4), for a valid configuration this should return the identity matrix (4.8).

$$U^{\dagger}U = UU^{\dagger} = \mathbb{1}_2 \quad with\, U, U^{\dagger} \in \mathsf{SU}(2) \tag{4.8}$$

$$U^{-1}U = UU^{-1} = \mathbb{1}_2 \tag{4.9}$$

$$U^{\dagger} = U^{-1} \text{ and } \det U = 1 \iff U \in \mathsf{SU}(2) \tag{4.10}$$

A more comprehensive check involves substituting the inverse matrix for the adjoint in the above test. Matrices in the $\mathsf{SU}(2)$ group have a determinant of 1, therefore an inverse matrix is always defined. Calculating the inverted form of a complex matrix is computationally expensive, hence we use this method sparingly. The first step is conversion of the matrix in $\mathbb{C}^2$ to its $\mathbb{R}^4$ equivalent to allow computation of the $\mathbb{R}^4$ *cofactor matrix*, itself an expensive operation. This is transposed and inverted before conversion back to a matrix in $\mathbb{C}^2$. If the Special Unitary condition has not been breached, multiplication by the inverted matrix will also result in the identity matrix (4.9).

The special unitary group of matrices specifies that the adjoint and inverse forms of a matrix are equivalent (4.10). Therefore, for general run-time tests we can use the adjoint matrix for quick tests that $\mathsf{SU}(2)$ group conditions are upheld.

4.2.3 Validation of field calculations

Field calculations, such as the generation of the topological charge density field $q(X)$, can be further evaluated against the output of the cooling algorithm. Various measures are performed on the data as part of the cooling algorithm including the calculation of the lattice sites with the highest and lowest topological charge density — giving a prediction of the location of any potential (anti-)instantons.

4.3 Computation and visualisation of lattice fields

Lattice QCD presents a rich array of observables; however, in this work we are primarily interested in scalar fields. The cooling program has been modified to output the state of the lattice after each iteration to allow inspection of these fields. Changes between cooling iterations in the lattice fields should be a smooth process, any large changes in behaviour potentially indicate the data has been incorrectly parsed. Additionally, this method can be used to validate the cooling algorithm and to learn more about effects that cooling have on the various observables.

The visualisations shown in the remainder of this chapter were generated using VisIt [141], a tool previously used in lattice QCD (see Section 3.4). Isosurfaces were produced by resampling the data to make it cell centred and applying the *Pseudo colour* operator with customised colour tables. The data was not upscaled in accordance to the wishes of the physicists who felt this may give a false representation of the data. Where direct volume renders are presented, they are generated using splatting with customised colour tables. An opacity transfer function is used that favours interesting regions of the data in the context of the domain [143] at the extremes of the field range.

4.3.1 Plaquette fields

The plaquette is a Wilson loop in its simplest form forming a closed unit loop in any two dimensional plane. From each site on the lattice it is possible to define a total of six plaquettes (Fig. 4.6) originating from a site $U(X)$. Three of these exist purely in spatial planes (XY, XZ and YZ) and are labelled as the *space-like plaquettes*. The remaining three plaquettes have an element from the Euclidean time dimension (XT, YT and ZT) and are denoted as *time-like plaquettes*.

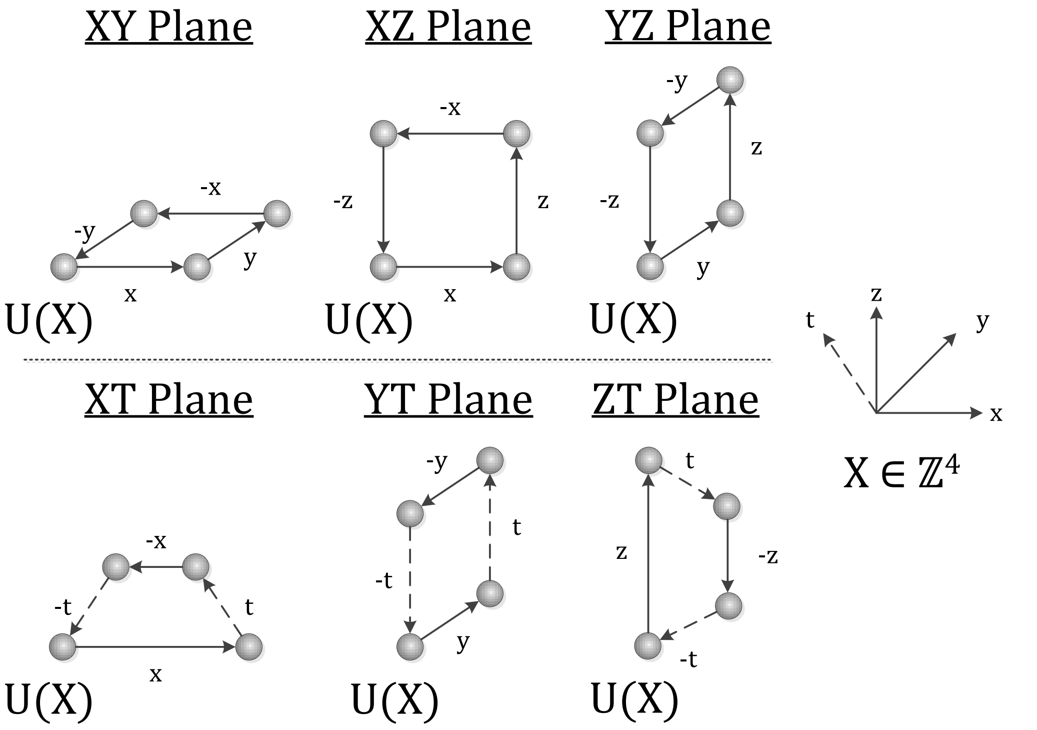


Figure 4.6: Top row: The three space-like plaquettes on the lattice. Bottom row: the three-time like plaquettes on the lattice.

Due to the Euclidean nature of the lattice it is possible to treat the space-like and time-like plaquettes as equivalent for calculation purposes (Fig. 4.7), where the place holder variables μ and ν are used to represent any two unique space-time directions. Plaquettes are computed by multiplying the links in path order (4.11), returning an SU(2) matrix at the lattice site in

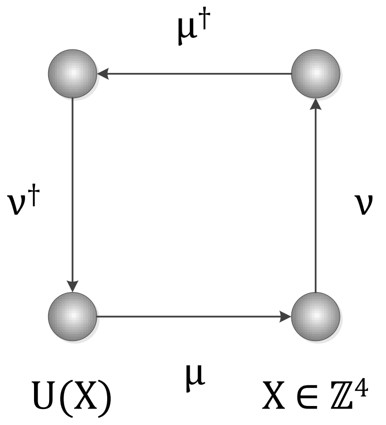


Figure 4.7: We can define an arbitrary plaquette in two dimensions from a given starting point $U(X)$.

the given plane. The value is reduced to a scalar by taking the matrix trace, which for $\mathsf{SU}(2)$ matrices has a zero imaginary part; hence, it can be discarded. Finally, the value is divided by two in order to produce an output in the range $[-1.0,+1.0]$ (4.12). This value can then be used to compute the average plaquette value in each plane for validation purposes (see Section 4.2.1.3) or used for visualisation.

By sweeping across the entire lattice we can compute the space-like and time-like plaquettes as a field of scalars. For the purposes of visualisation the space-like and time-like plaquette averages are computed as two distinct fields. We do this by computing the three relevant plaquettes (see Fig. 4.6), reducing each to a scalar using (4.12), and computing the average of the three input planes. By defining an additional place holder link (ρ) we can summarise the calculation required at each site as (4.13). These two fields are related to another lattice observable, the topological charge density, discussed in detail in Section 4.3.2.

$$U_{\mu\nu}(X) = U_\mu(X)U_\nu(X+\hat{\mu})U_\mu^\dagger(X+\hat{\nu})U_\nu^\dagger(X) \tag{4.11}$$
$$\textit{where } \mu,\nu = \{0,1,2,3 \mid \mu \neq \nu\},\ X \in \mathbb{Z}^4,\ U_{\mu\nu}(X) \in \mathsf{SU}(2)$$

$$S_{\mu\nu}(X) = \frac{1}{2}\Re(tr(U_{\mu\nu}(X))) \tag{4.12}$$
$$\textit{where } S_{\mu\nu}(X) \in [-1.0,+1.0]$$

$$f(X) = \frac{S_{\mu\nu}(X)+S_{\mu\rho}(X)+S_{\nu\rho}(X)}{3} \tag{4.13}$$
$$\textit{where } f(X)\ \in [-1.0,+1.0],\ \mu,\nu,\rho = \{0,1,2,3 \mid \mu \neq \nu \neq \rho\}$$

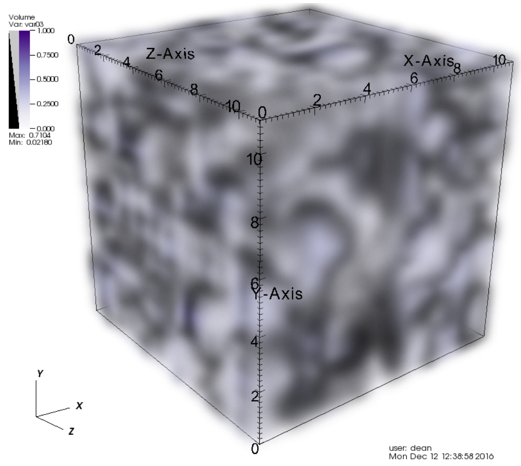


Figure 4.8: A slice of the uncooled space-like plaquette field rendered using DVR.

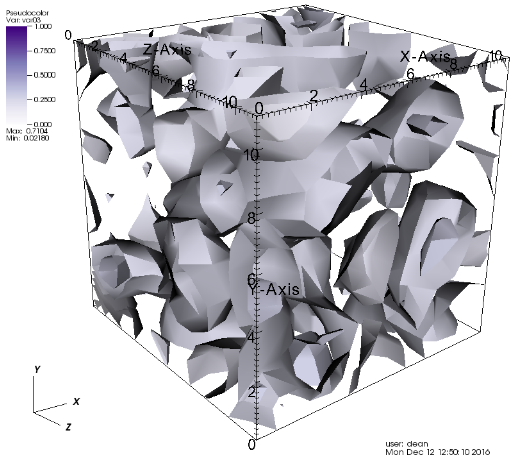


Figure 4.9: A slice of the uncooled space-like plaquette field rendered using nested isosurfaces

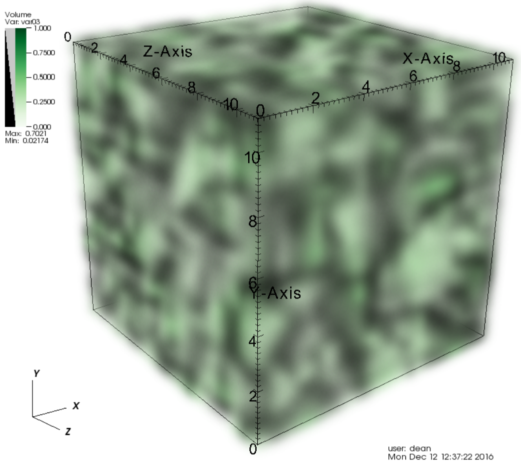


Figure 4.10: A slice of the uncooled time-like plaquette field rendered using DVR.

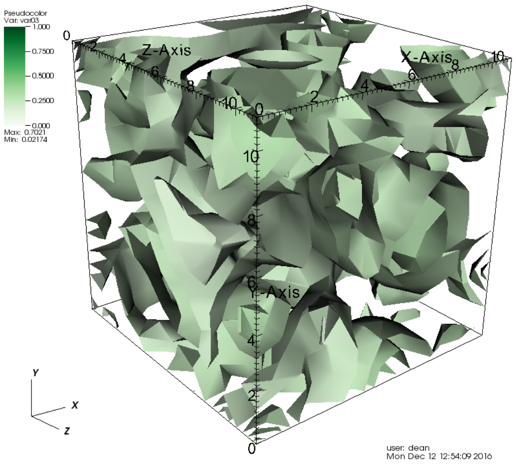


Figure 4.11: A slice of the uncooled time-like plaquette field rendered using nested isosurfaces.

Figures 4.8 and 4.10 show the uncooled plaquette fields using direct volume rendering techniques. The isovalues in both fields have a theoretical range of $[-1.0, 1.0]$ as they are derived from $\mathrm{SU}(2)$ matrices, but in practice always fall into the range $[0.0, 1.0]$. It is possible to see in the direct volume rendered images that the plaquette fields feature regions of similar isovalues separated by near-zero regions. Under cooling these regions become better defined and distinct blob-like objects are observed in the plaquette fields. If the same data is rendered using isosurfaces the structure of the plaquette field becomes clearer, as demonstrated by Figures 4.9 and 4.11.

These visualisations actually show the magnitude of two vector fields representing the magnetic field B and electric field E on the lattice. The magnetic field at lattice sites $B(X)$ can be represented by assigning the space-like planes (4.14) to an unsigned vector (4.15). Likewise, we can define the electric field $E(X)$ as an unsigned vector (4.17) of the time-like plaquettes (4.16). However, for the purposes of this work we deal directly with the scalar values provided by the plaquette fields.

Under normal conditions the two fields look similar; however, by visualising the fields physicists are able to ask new questions of the plaquette fields. One such question is how the space-like plaquette field and time-like plaquette field structure begin to differ as thermodynamic parameters, such as chemical potential μ, are varied. We use the mathematical models generated for indirect volume rendering to begin to answer those questions.

$$\begin{aligned} B_x = U_{\nu\rho}(X) &= \frac{1}{2}\Re(tr(U_\nu(X)U_\rho(X+\hat{\nu})U^\dagger_\nu(X+\hat{\rho})U^\dagger_\rho(X))) \\ B_y = U_{\mu\rho}(X) &= \frac{1}{2}\Re(tr(U_\mu(X)U_\rho(X+\hat{\mu})U^\dagger_\mu(X+\hat{\rho})U^\dagger_\rho(X))) \\ B_z = U_{\mu\nu}(X) &= \frac{1}{2}\Re(tr(U_\mu(X)U_\nu(X+\hat{\mu})U^\dagger_\mu(X+\hat{\nu})U^\dagger_\nu(X))) \\ &\text{where } \mu\ = 0,\ \nu = 1,\ \rho = 2,\ X \in \mathbb{Z}^4 \end{aligned} \tag{4.14}$$

$$B(X) = \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \tag{4.15}$$

$$\begin{aligned} E_x = U_{\mu\sigma}(X) &= \frac{1}{2}\Re(tr(U_\mu(X)U_\sigma(X+\hat{\mu})U^\dagger_\mu(X+\hat{\sigma})U^\dagger_\sigma(X))) \\ E_y = U_{\nu\sigma}(X) &= \frac{1}{2}\Re(tr(U_\nu(X)U_\sigma(X+\hat{\nu})U^\dagger_\nu(X+\hat{\sigma})U^\dagger_\sigma(X))) \\ E_z = U_{\rho\sigma}(X) &= \frac{1}{2}\Re(tr(U_\rho(X)U_\sigma(X+\hat{\rho})U^\dagger_\rho(X+\hat{\sigma})U^\dagger_\sigma(X))) \\ &\text{where } \mu = 0,\ \nu = 1,\ \rho = 2,\ \sigma = 3,\ X \in \mathbb{Z}^4 \end{aligned} \tag{4.16}$$

$$E(X) = \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} \tag{4.17}$$

4.3.2 Topological charge density field

Instantons represent recognised topological objects that appear in the gluon field in relation to other phenomena on the lattice. Fluctuations in the action density is able to reach critical levels of concentration at specific time intervals that correlated (anti-)instantons becoming visible in the output [144]. Instantons carry a topological charge, anti-instantons carry an equal and opposite charge. A calculable observable from a 4D hyper-volume is the overall topological charge, computed as the algebraic sum of instantons and anti-instantons [145]. Compelling evidence for the existence of (anti-)instantons, along with a more detailed discussion of their role in Hadronic physics can be found in [146].

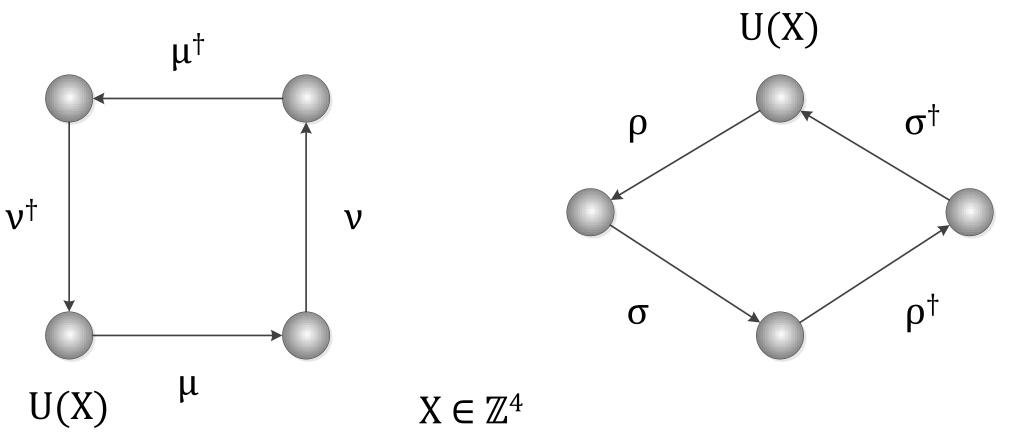


Figure 4.12: It is possible to define two orthogonal plaquettes from a site ($U(X)$) on the lattice, these are commonly referred to as F and $\bar{F}$ (*F-dual*).

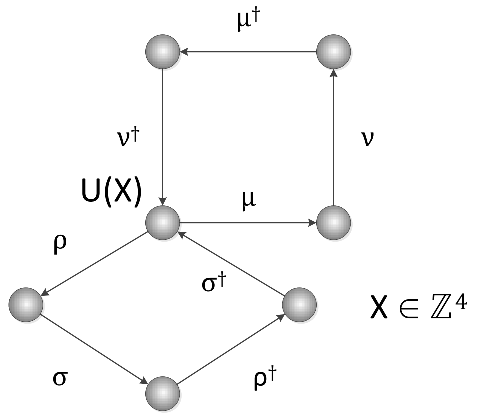


Figure 4.13: The topological charge density contribution of a site involves the multiplication of all possible permutations of orthogonal plaquettes from a lattice site ($U(X)$). The dagger symbol on the links represent a movement in the reverse direction, represented by an adjoint matrix of a link variable.

$$q(X) = \frac{1}{32\pi^2} F_{\mu\nu}\bar{F}_{\mu\nu} \tag{4.18}$$

$$\begin{gathered} F_{\mu\nu}\bar{F}_{\mu\nu} = \varepsilon_{\mu\nu\rho\sigma}F_{\mu\nu}F_{\rho\sigma} \\ where\ \mu,\nu,\rho,\sigma = \{0,1,2,3\} \end{gathered} \tag{4.19}$$

$$\varepsilon_{\mu\nu\sigma\rho} = \begin{cases} +1 & if\ (\mu,\nu,\sigma,\rho)\ is\ an\ even\ permutation\ of\ (0,1,2,3) \\ -1 & if\ (\mu,\nu,\sigma,\rho)\ is\ an\ odd\ permutation\ of\ (0,1,2,3) \\ 0 & otherwise \end{cases} \tag{4.20}$$

The definition of topological charge density at a site $q(X)$ that we use is a modified form of that given by Peskin [147] that provides a definite parity [148], given in (4.18, 4.19, 4.20). Figures 4.12 and 4.13 represent this process visually from a given lattice site origin $U(X)$. By computing the topological charge density at all sites on the lattice a scalar field is generated, that potentially allows the visualisation of (anti-)instantons as local field minima and maxima.

Computation of the topological charge density is an expensive operation requiring several matrix multiplications. For each site in four dimensional space we first define six intermediate $SU(2)$ matrices, the *difference plaquettes*, by computing the difference between a plaquette and its conjugate transpose (4.21). The topological charge density is then obtained by multiplying difference plaquettes in opposing planes and summing according to the anti-symmetrical four-tensor (ε) defined in (4.20). A scalar value is created by taking the real part of the matrix trace at each site and multiplying by two, summarised by (4.22).

$$D_{\mu\nu}(X) = (U_\mu(X)U_\nu(X+\hat{\mu})U^\dagger_\mu(X+\hat{\nu})U^\dagger_\nu(X)) - (U_\mu(X)U_\nu(X+\hat{\mu})U^\dagger_\mu(X+\hat{\nu})U^\dagger_\nu(X))^\dagger \tag{4.21}$$

$$where\ \mu,\nu = \{0,1,2,3 \mid \mu \neq \nu\},\ X \in \mathbb{Z}^4$$

$$f(X) = 2 \times \Re(Tr(D_{\mu\nu}(X)D_{\rho\sigma}(X) - D_{\mu\rho}(X)D_{\nu\sigma}(X) + D_{\mu\sigma}(X)D_{\nu\rho}(X))) \tag{4.22}$$

$$where\ \mu,\nu,\rho,\sigma = \{0,1,2,3 \mid \mu \neq \nu \neq \rho \neq \sigma\},\ X \in \mathbb{Z}^4$$

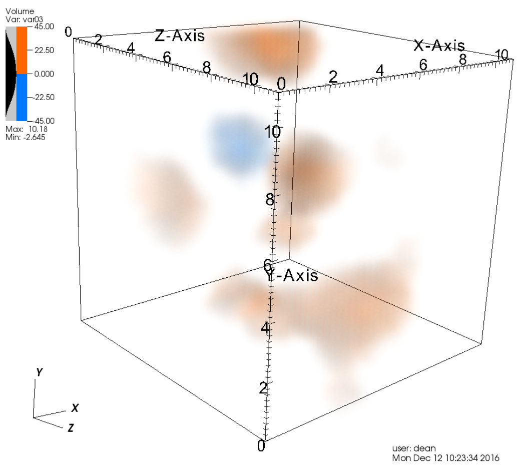


Figure 4.14: A time slice of topological charge density data, cooled for 10 iterations and visualised using DVR. Chemical potential in this ensemble is set to ($\mu = 0.0$).

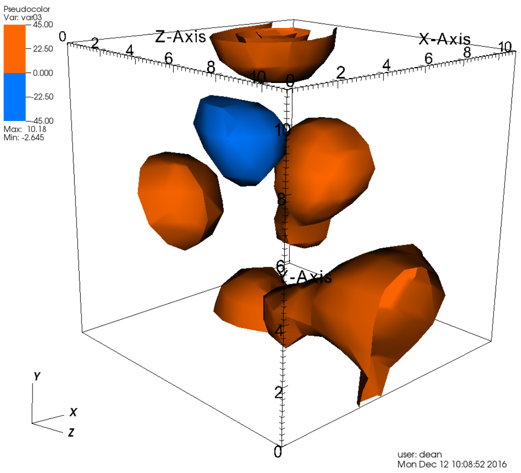


Figure 4.15: A time slice of topological charge density data, cooled for 10 iterations and visualised as nested isosurfaces. Chemical potential in this ensemble is set to ($\mu = 0.0$).

In Figure 4.14 we present a direct volume render of the topological charge density field when cooled through 10 cooling iterations, where it is possible to observe the field has began to separate into a number of regions of positive and negative charge. Under further cooling objects that persist as field minima and maxima could be considered as potential (anti-)instantons. When the same field is visualised instead using nested isosurfaces (Fig. 4.15) it is possible to view the objects with a better perception of depth and shape through the use of directed lighting and shading.

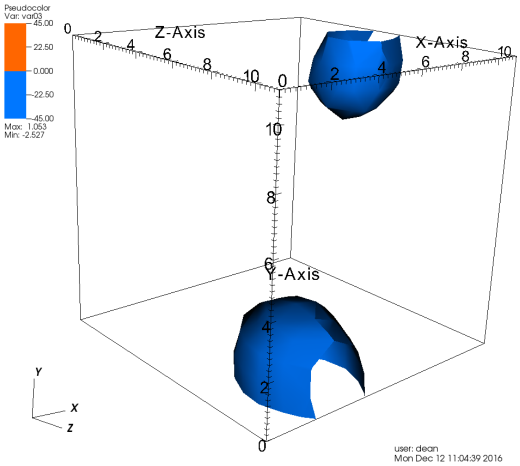


Figure 4.16: A time slice of topological charge density data, cooled for 20 iterations and visualised using isosurfaces. Chemical potential in this ensemble is set to $\mu = 0.5$.

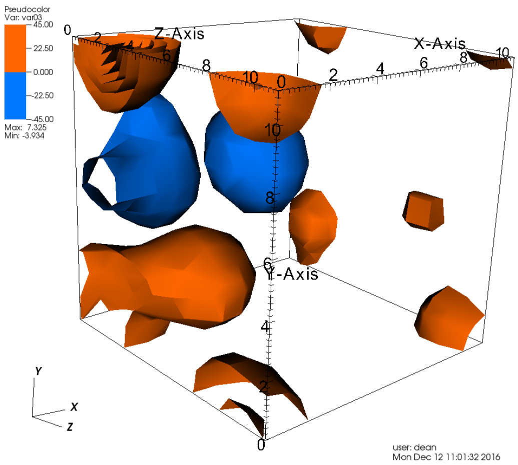


Figure 4.17: A time slice of topological charge density data, cooled for 20 iterations and visualised using isosurfaces. Chemical potential in this ensemble is set to $\mu = 1.0$.

Figures 4.16 and 4.17 use isosurfaces to visualise configurations at 20 cooling iterations with differing chemical potentials (μ). The effect of periodic boundaries is demonstrated by objects disappearing off the sides, whereas they should be visible on either face of the visualisation. Adding a layer of "ghost vertices" on the far plane of each axis, taking the scalar value of the vertices on the opposing plane helps to alleviate this problem. Whilst this provides a method of revealing objects hidden or sliced by the boundary, the technique often reconstructs surfaces incorrectly, as shown in Fig. 4.18.

4.3.3 Polyakov loop field

The Polyakov loop, otherwise known as the *Wilson line* operator, is used as a method for computing the symmetry of a lattice. Breaking of symmetry is one signal that can indicate a transition to a de-confined state, achieved by varying ensemble thermodynamic control parameters such as temperature or chemical potential. Lattice QCD allows variations in temperature T to be simulated by modifying the number of time steps, denoted as L_t (4.23). This method of locating critical temperatures is well established in $\mathrm{SU}(2)$ and $\mathrm{SU}(3)$ lattice gauge theories [150, 151, 152, 153].

In order to compute the Polyakov loop, we take the product of all time-like link variables from a given lattice site in three dimensions (4.24). On a lattice defined with a periodic time axis, such as in lattice QCD, the effect is a closed straight line, as visualised in Fig. 4.19. As

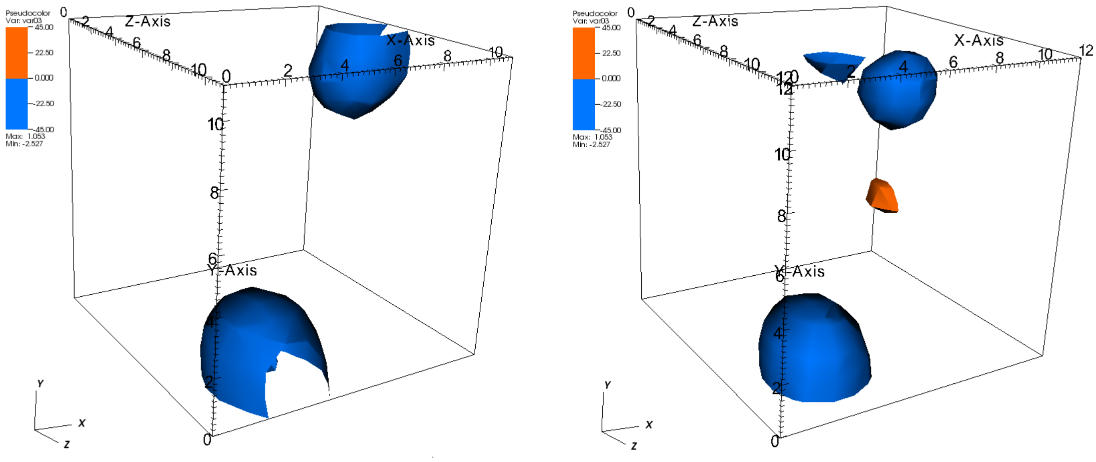


Figure 4.18: Comparison of data rendered in VisIt [149] without knowledge of periodic boundaries (left) and with periodic boundaries mimicked using ghost vertices (right).

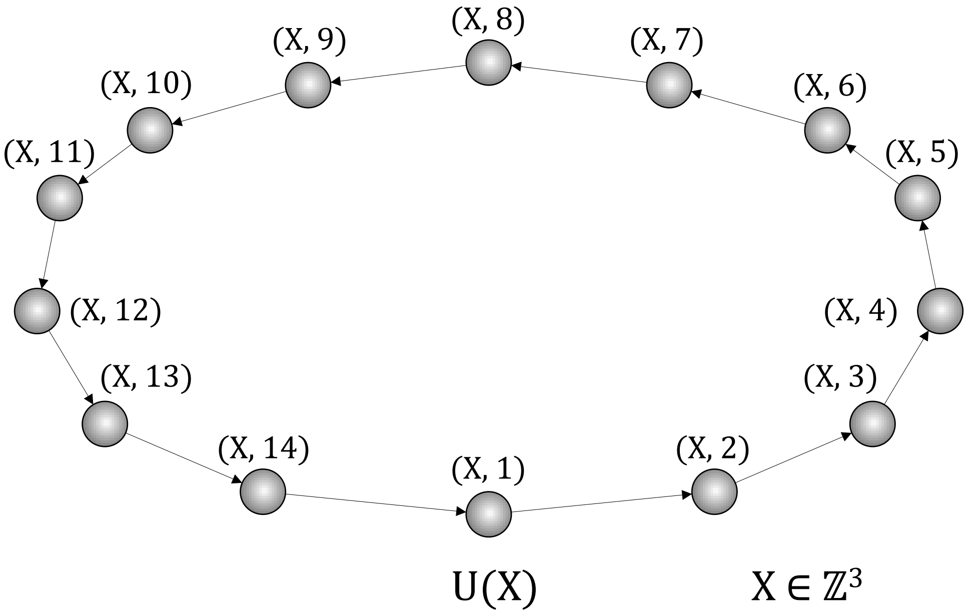


Figure 4.19: The Polyakov loop is a mapping $f : \mathbb{R}^4 \mapsto \mathbb{R}^3$ by visiting each site in a given space-time direction. Most commonly we loop over the time axis.

we effectively loop over all link variables in one direction for each spatial site on the lattice, we get the added benefit of a reduction in dimensionality from a four dimensional lattice to a three dimensional scalar output. Thus the Polyakov loop represents an attractive method for identifying de-confinement in a form that can easily be visualised and analysed using existing methods.

$$T = \frac{1}{L_t a} \tag{4.23}$$

$$\textit{where } a \textit{ is lattice spacing}$$

$$f(X) = \frac{1}{2}\Re(Tr(\prod_{n=1}^{L_t} U_\mu(X + n\hat{\mu}))) \tag{4.24}$$

$$\textit{where } \mu = 3,\ X \in \mathbb{Z}^3$$

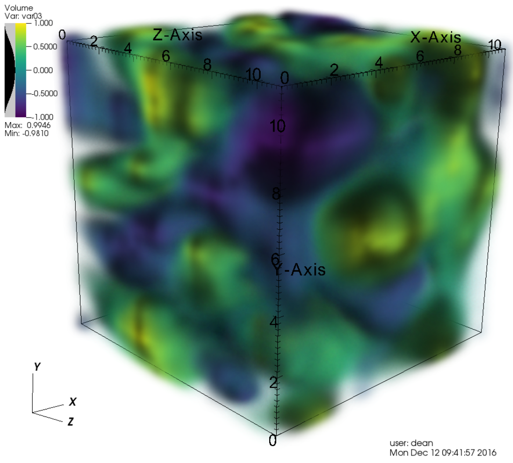


Figure 4.20: Polyakov loop data, cooled for 20 iterations and visualised using Direct Volume Rendering.

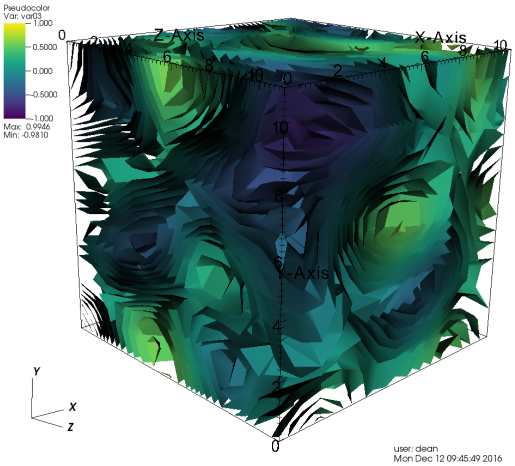


Figure 4.21: Polyakov loop data, cooled for 20 iterations and visualised as 25 nested isosurfaces.

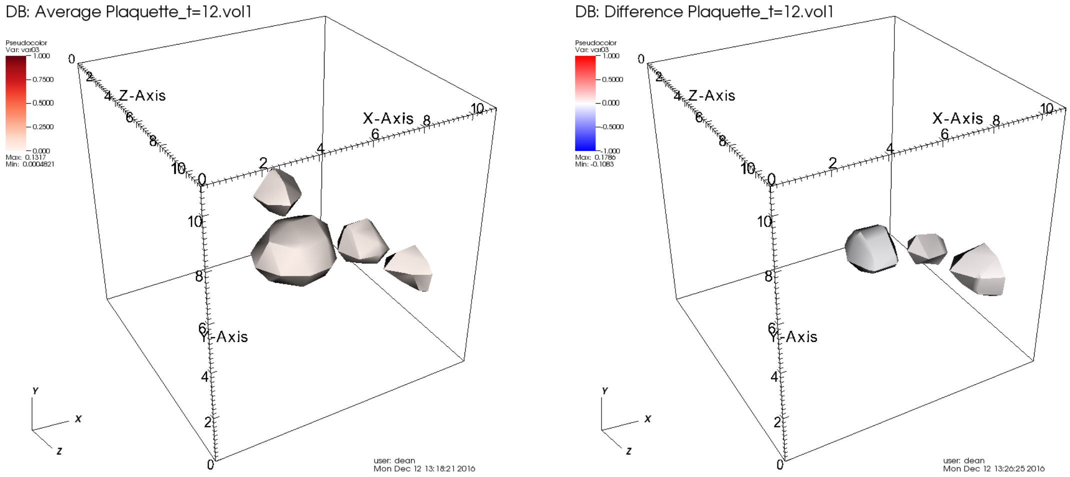
The scalar field generated by computing the Polyakov loop across the spatial volume is confined to the range $[-1.0, +1.0]$. In Figure 4.20 a sample is visualised using DVR techniques, having been cooled to 20 iterations, showing how regions of minima and maxima are separated by turbulent bands centred on zero. Using isosurfaces we are able to capture this segmentation as a series of models at varying isovalues, as shown in Figure 4.21.

4.3.4 Average plaquette field and difference plaquette field

In Section 4.3.1 we showed that two categories of plaquette fields are computable from the lattice. These form the space-like plaquettes existing in the *xy*, *xz*, and *yz* planes (4.14, 4.15) and time-like plaquettes existing in the *xt*, *yt*, and *zt* planes (4.16, 4.17). In lattice QCD it is also possible to derive secondary fields as combinations of these two categories of plaquettes.

The average plaquette is a signed scalar calculated using the average of the space-like and time-like plaquette fields. Likewise, the difference plaquette is a signed scalar calculated using the difference in space-like and time-like plaquettes for a site on the lattice. In most situations the difference plaquette field is represented using an unsigned value as the magnitude of the difference is of primary interest. As with other lattice observables a scalar field can be constructed in four dimensions by sweeping across the lattice sites.

Examining these fields in parallel gives a method of detecting self-duality, a prerequisite for the presence of a potential (anti-)instanton. A value close to zero in the difference plaquette at a site shows that the time-like plaquette and space-like plaquette are closely aligned, or self-dual. Instantons appear as maxima in the average plaquette field and minima in the difference plaquette field. (Anti-)Instantons will be seen as minima in the average plaquette field and minima in the difference plaquette. Figure 4.22 shows how objects existing in these two fields can be examined in parallel to predict the presence of an (anti-)instanton. The object present in the average plaquette field but missing in the difference plaquette field is a prime candidate for being an instanton.



Average plaquette field

Difference plaquette field

Figure 4.22: A slice of the average plaquette field and difference plaquette field rendered as nested isosurfaces.

4.4 Summary

In this chapter we have detailed the processes required to construct a model for computing a range of lattice QCD observables from pre-existing data. The complexity of the data meant that validation plays an important part in the construction of the model and subsequent field calculations. Using exiting software and visualisation techniques we were able to study various field structures, initially to help validate the constructed lattice models.

Besides verification the visualisations also provoked some questions by the physicists about the structure of the data. Indirect volume rendering techniques, such as marching cubes [75], enabled physicists to easily identify distinct objects in the data. Physicists felt that the ability to begin to isolate core observables such as (anti-)instantons was a major step forward for the analysis of lattice QCD data sets. We demonstrated in this chapter how objects can filtered from the surrounding noise by careful application of isovalue thresholds.

In conclusion the contributions of this chapter are:

- We demonstrated how data can be validated by domain experts using visual methods to determine if the data has the expected structure.
- Visualisation was incorporated into a standard lattice QCD workflow, helping to provoke the formation of new hypotheses of the data.
- It was illustrated how appropriately filtered models could potentially offer new methods of analysis to physicists.

The concept of using isosurfaces to build a mathematical model of the field structure is a technique further explored in Chapter 5.

Chapter 5

Application of topological visualisation techniques to lattice QCD

Contents

In this chapter we discuss the transition from displaying lattice QCD data using traditional volume visualisation techniques, such as isosurfaces, to a topology based visualisation model. We demonstrate how the use of topological visualisation techniques helps enhance the data exploration process and show how additional information can be extracted from the data by examining connectivity.

We begin by applying techniques previously used by domain scientists to the data, which prompted them to form new hypothesises about the data as discrete objects. We use topological visualisation techniques, which have not previously been applied to lattice QCD data sets, to explore these questions. In summary, this chapter asks “how can topological visualisation techniques enhance data exploration of lattice QCD data?” and “what features of topology driven visualisation might provide new analytical approaches for domain experts?”

The remainder of this chapter is structured as follows. In Section 5.1 we begin by replicating the state-of-the-art in lattice QCD visualisation and detail how collaboration with domain experts influenced our decisions to use topological visualisation techniques. Section 5.2 discusses how the periodic nature of lattice QCD data proved problematic in initial visualisations and analysis. We present key contributions that we believe are of use to lattice QCD scientists in Section 5.3. Much of the text in this chapter was originally presented at the 2016 Computer Graphics & Visual Computing conference [12], where it received the best full paper award.

This chapter introduces a number of data structures and requires the defining of the following terminology:

Critical point This is a sampling point in the scalar field where the function gradient is zero. This can signify the location of a local or global extrema in the data, alternatively it can represent a saddle point in the topology of the scalar field.

Contour A contour represents a connected region of an isosurface. Each isosurface may consist of multiple separate contours.

Contour tree This is a data structure that is able to capture the evolution of the topology of a scalar field by sweeping through its isovalues. The contour tree should be interpreted as follows:

- Vertices (or *supernodes*): these mark the isovalue limits of a unique topological object. Leaf vertices represent the birth or death of a contour in scalar field. Vertices with degree two or higher represent the splitting or merging of two or more unique objects. Each supernode in the contour tree relates to a critical point in the scalar field.
- Edges (or *superarcs)*: each arc represents a unique topological object existing between the two isovalues marked by the associated supernodes. A mesh can be generated for the object at any isovalue within this range using the flexible isosurface algorithm [58].

Reeb graph This is a generalisation of the contour tree that allows the computation of the topology of a scalar field without a boundary. Examples of such fields are those defined as a set of sampling points on the surface of an n-torus.

Periodic boundary This is a method for representing a scalar field without boundaries. Vertices at the extremes of each axis are connected to form a hyper-toroidal structure. The cells connecting two ends of each axis are treated as equivalent to those existing between other vertices in the scalar field.

Persistence measures These represent a basic method for summarising the importance of a topological object. This information can be used to perform topological simplification in order to remove noise. Alternatively, these measures can be used to compute basic properties, such as volume or surface area, of objects existing in the scalar field.

5.1 Application of visualisation techniques to QCD data

The following section describes the range of techniques that we used in gathering a basic understanding of the QCD structure.

5.1.1 Initial approaches to visualisation

Initial use of Visit [149] enabled us to examine our data using DVR techniques (see Fig. 5.1), in order to perform a comparison with existing visualisations from similar data sources [143]. Whilst this approach allows an overview of the data to be formed, it limits the ability to examine

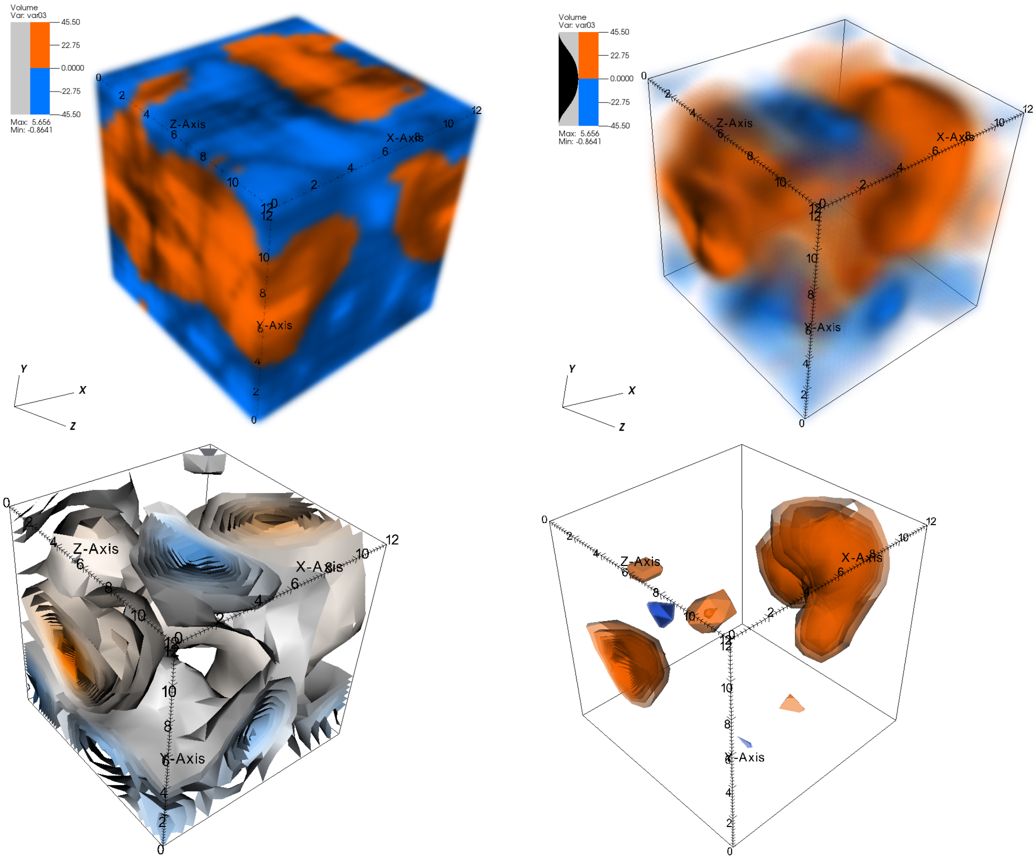


Figure 5.1: Four views of the same data; Top: a Direct Volume Render with no opacity transfer function (left), a Direct Volume Render with a opacity transfer function tuned to remove "noise" (right). Bottom: 100 isosurfaces coloured by function height (left), filtered isosurfaces (right).

the structure of discrete objects within the volume due to occlusion. Through the use of DVR we were able to observe the formation of topological object under the processing of cooling that physicists were keen to investigate further.

Moving to indirect volume techniques in VisIt, such as Marching Cubes [2], enabled visualisation of the data in a discretised form by handling the data in the form of level sets. Further refinement of the technique, through the use of multiple filters, allowed viewing of nested contours within specified bounds. This approach enabled uninteresting areas of the data to be filtered out; in lattice QCD the region centred around isovalue zero is generally considered uninteresting. By combining multiple filters we were able to focus the visualisation on instanton structures present at the positive/negative extremes of the topological charge density field.

The existing FORTRAN code used to perform cooling of the data was able to predict the locations of minima and maxima in the topological charge density field. Hence, we used this information to locate and cross examine potential (anti-)instantons using an isosurface representation. Beyond predicting the positions of minima/maxima, few other properties of the objects could be revealed using the existing cooling tool. The new visualisations allowed domain scientists to form new questions about their structure. One such question was if varying parameters of the physical simulation could alter the quantity and shape of objects in the volume. This hypothesis prompted the use of topology to perform segmentation of the data into distinct objects, allowing new data querying and exploration possibilities.

5.1.2 Topological visualisation techniques

Recent visualisation tools have successfully used the contour tree to perform topology controlled volume rendering; hence, this was the initial method we choose to use in this work. This has two benefits; first, optimized visualisation techniques can be applied to the data. Second, properties can be calculated on distinct contours as opposed to the level set as a whole, allowing the calculation of *physical* measurements of individual objects existing in the data. Instead of dealing with the data purely by isovalue, connectivity is also captured by the algorithm. The notion of connected regions is what allows the progression from level sets to discrete contours represented as triangular meshes.

The contour tree captures changes within the topology of a scalar function with respect to isovalue, in this case topological charge density. The algorithm works internally by combining split and join trees, collectively named merge trees, to compute *critical points* in the topology. Within the contour tree critical vertices are defined as those having degree 1, in the case of leaf nodes spawning completely new objects, or degree ≥ 2 where the function value splits or joins an existing topological object. Figure 5.2 presents the contour tree of a simple data set. At isovalue (1) the contour tree contains a single arc, this generates a large connected contour. As the isovalue is increased the contour changes shape and a hole appears at sample isovalue (2). This represents a change of topological genus; however, the contour tree typically doesn't present or store this information. At sample (3) the object has began to split, but remains a single connected region. Two objects are present as isovalue (4); each object is represented by a separate arc in the contour tree. By sample (5) one of these objects splits into two parts, giving a total of three active contours. Finally, at sample isovalue (6) the object has split into 4 disjoint contours.

The Reeb graph allows the handling of scalar data defined on non-simply connected domains, such as those that use periodic boundary conditions. This presents problems to the contour tree algorithm as it fails in the merge tree union-find computation. For most purposes the Reeb graph can be considered as a generalisation of the contour tree with the ability of allowing loops in the topology [64]. As with the contour tree, the Reeb graph can be applied to models of any dimension provided it is represented on a simplicial mesh [65]. For the remainder of this chapter we discuss topology using the generalised structure of the Reeb graph apart from to highlight specific differences present in the contour tree.

Potential (anti-)instantons, characterised as local extrema, are captured as leaves in the Reeb graph meaning that Reeb graphs present a technique for making the key observables easily identifiable in the data. Each arc in the Reeb graph represents a unique object captured in the topology, a property used by the flexible isosurface algorithm [76] as a method for seeding an entire contour. Each object is a simplicial mesh which can be continually deformed by varying the isovalue within the range of the critical nodes marking the extrema of the associated

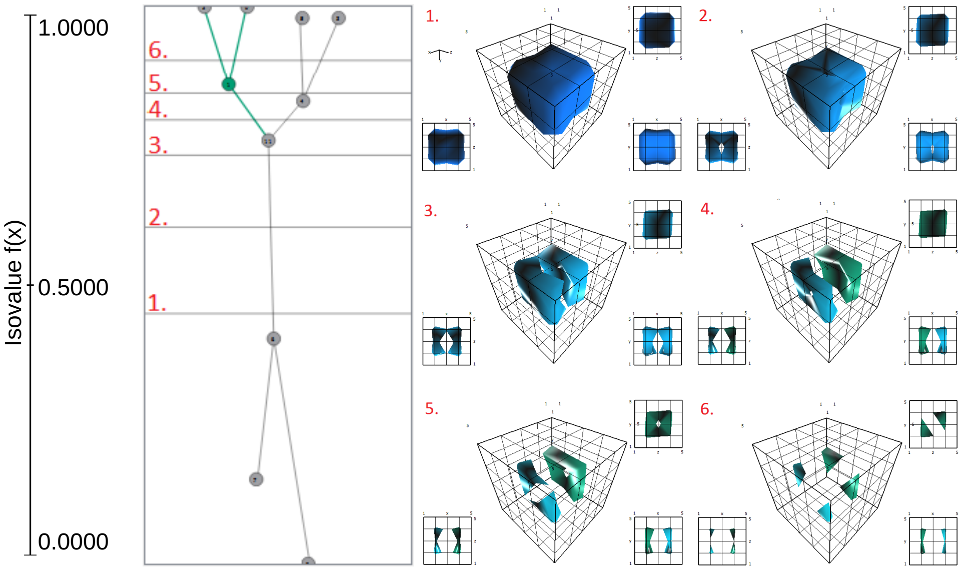


Figure 5.2: An example of a contour tree defined on a simple data set. The y-position of vertices in the graph visualisation relates to the function height. The x-position has no specific meaning and vertices are free to be moved horizontally to minimize occlusion. As the isovalue is varied the structure of contours seeded by contour tree continually change but remain path connected. The number of unique contours at an isovalue can be determined by counting the number of arcs intersected by a horizontal line draw across the contour tree.

superarc (Fig. 5.3). The object is guaranteed as being isolated from other objects in this region; however the topological genus of the object, or number of holes, is not necessarily fixed. The set of topological objects at a specific isovalue represents the distinct connected level sets making up a single isosurface.

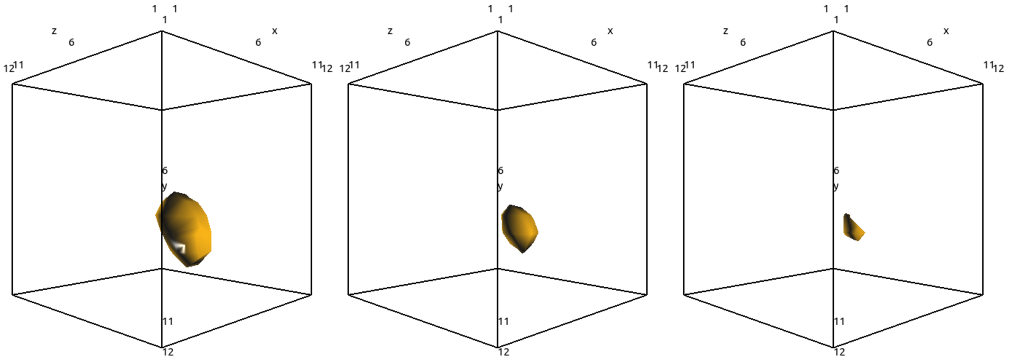


Figure 5.3: The same isolated topological object shown at three isovalues across the length of arc representing the object. As the isovalue is varied the structure of the object continually deforms but the object remains a unique entity until reaching the supernodes. Left: isovalue = 0.0068; centre: isovalue = 0.0148; right: isovalue = 0.0228.

The ability to compute distinct objects as features in the scalar topology opens up a number of possibilities for displaying and querying the topological charge density field (Fig. 5.4). For example, the concept of moments presents a generalised solution to parametrising key objects in the scalar field; these include the enclosed volume, centre-of-mass and principal component axis which is discussed in greater detail in Section 5.3.2. In lattice QCD the objects that exist as the upper and lower leaves in the Reeb graph are of greatest interest to domain scientists. These can be isolated to query physical properties of (anti-)instantons to determine how thermodynamic control parameters, particularly chemical potential μ, influence the structure of these objects. We are therefore able to use topological visualisation techniques to display and compute properties of specific lattice observables.

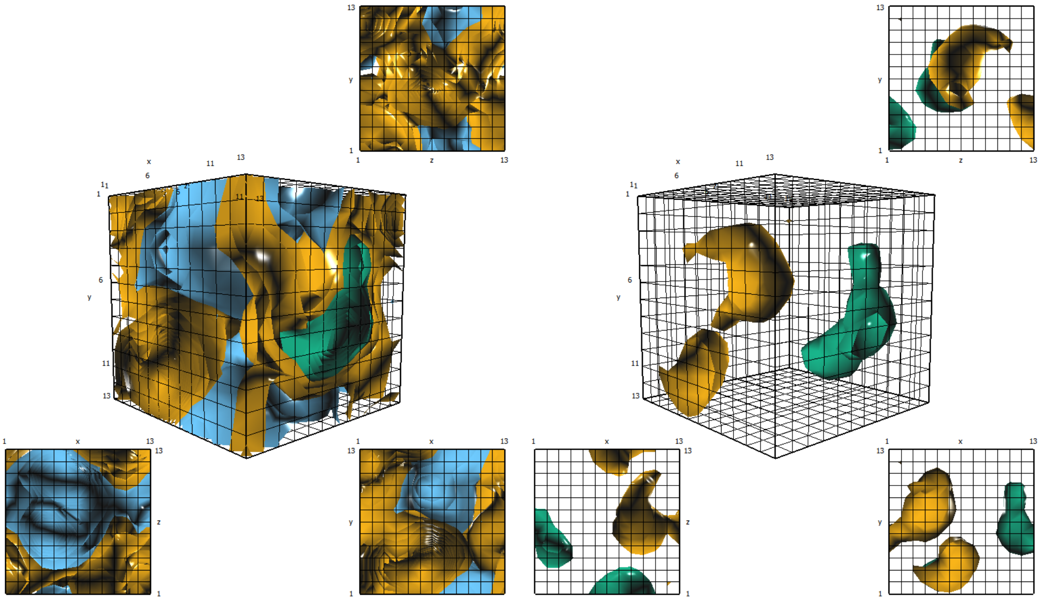


Figure 5.4: A topological charge density data set; left: shown as nested contours and right: as an individual slice at a fixed isovalue. The green region has been selected by the user using the contour tree to highlight the same region in both views.

5.2 Addressing periodic boundary conditions

In order to provide translational invariance, where it is impossible to distinguish between one point on the lattice with another, experiments are performed on a lattice with no defined boundaries. The extremes of each axis are connected to create an additional set of cells, linking the minima and maxima, in each of the four spatio-temporal dimensions (Fig. 5.5). The concept of periodic boundary conditions is not exclusive to lattice QCD but can also be found in other fields of science, such as computation chemistry [154].

The periodic boundaries conditions found in the data present challenges in the computation and display of lattice fields. Current implementations of the flexible isosurface algorithm [76] depend on the contour tree as an internal data structure. However, the contour tree is unable

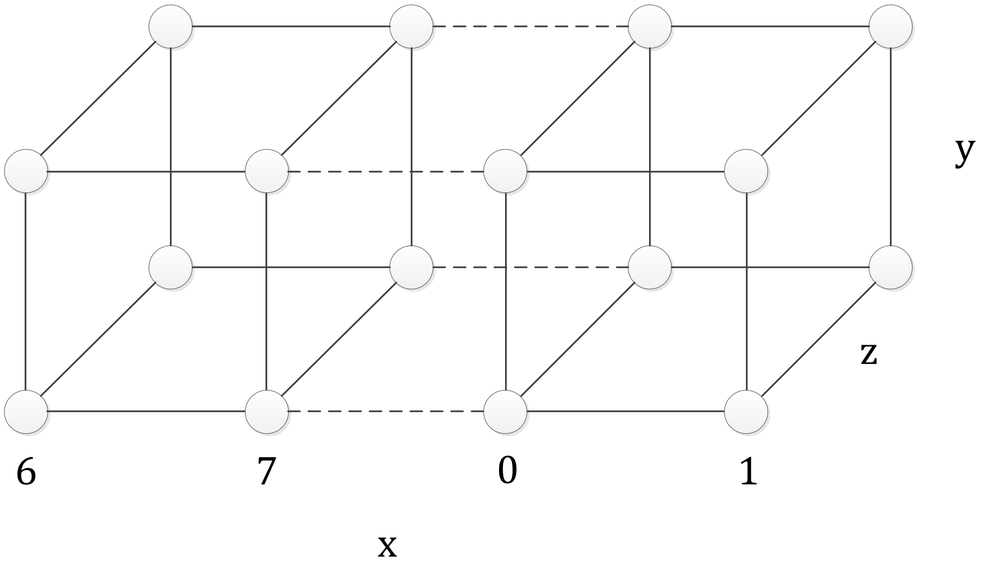


Figure 5.5: An example of a cell spanning a period boundary. The dashed cell links vertices $x = 7$ and $x = 0$ to give the illusion of an infinite axis. In lattice QCD all axes, including the time axis, use this approach to implement translational invariance.

to handle scalar fields without a defined boundary; hence, for the purposes of rendering it is necessary to cut each axis at the boundary. In order to correctly display objects spanning a the boundary (Fig. 5.6), a layer of ghost cells is created by cloning the vertices at the opposite end of each axis. Some implementations of the Reeb graph algorithm use a similar approach by making symbolic cuts in the topology [70]. This enables them to achieve better computation times as the contour tree tends to be quicker to compute that the Reeb graph.

For the purposes of rendering, the contour tree and associated flexible isosurface algorithm represent a good approach to seeding of meshes of objects spanning a periodic boundary. These objects are captured as multiple arcs in the contour tree allowing each part to be seeded and displayed separately. In comparison objects generated directly from a periodic source results in rendering artefacts such as degenerated triangles spanning an entire axis (Fig. 5.7).

Figure 5.8 provides a direct comparison of the output of the Reeb graph and contour tree algorithms. Visually the two graphs look alike; however, there are exceptions. The Reeb graph, unlike the contour tree, can contain loops in the topology (Fig. 5.9) where objects wrap around one or more boundaries. It is this property that sets the two algorithms apart. Optimised Reeb graph computations, based upon the contour tree algorithm, cut the input domain at the arcs forming these loops. As a consequence of these cuts the contour tree captures the topology using additional arcs and vertices for each part of an object spanning a boundary.

Due to the direct link between the contour tree and the flexible isosurface algorithm, we use the contour tree to seed objects for visualisation purposes. For the purposes of analysis,

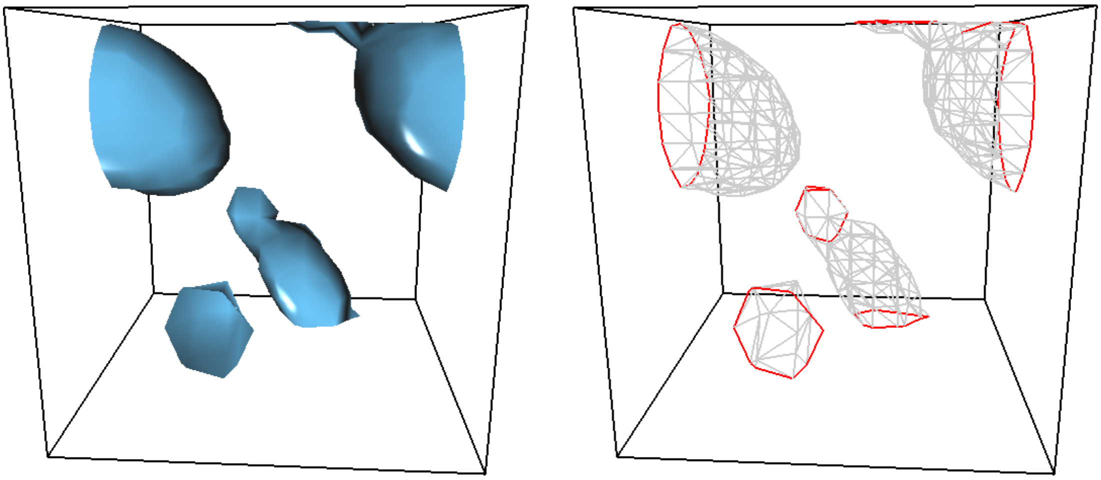


Figure 5.6: An example of a single object spanning multiple periodic boundaries. The mesh can be “glued” at each of the faces marked by the red edges in order to construct a single object. The contour tree would represent this as four separate arcs — one for each component part. However, the Reeb graph is able to resolve the periodic boundaries and captures the object as a single arc.

where a truer representation of the data is required, the Reeb graph is used instead. When examining physical properties of objects, such as the volume or surface area, objects spanning a boundary can be recovered by performing a piecewise sum of each individual mesh making up the object.

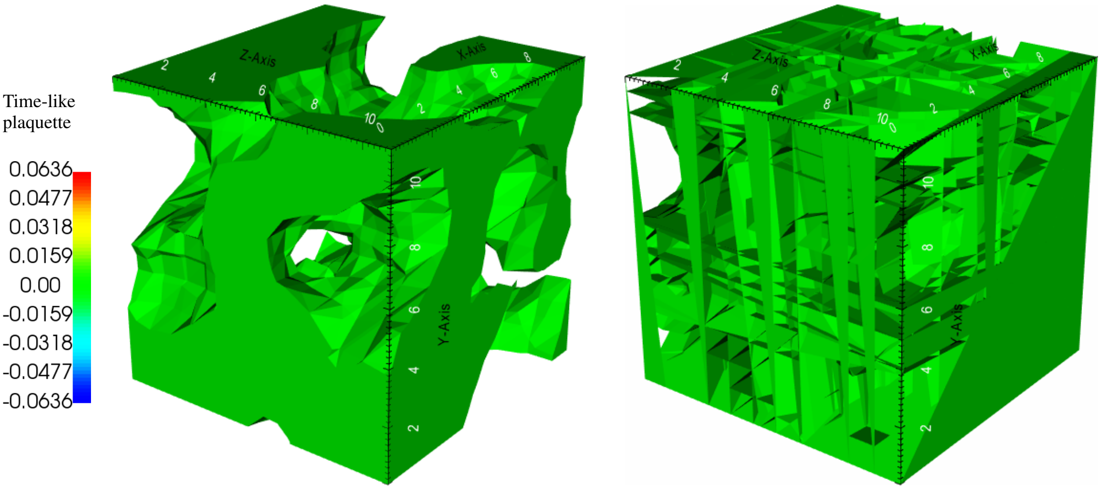


Figure 5.7: An example of the visual artefacts experienced when rendering an object spanning a periodic boundary. Multiple degenerate triangles link opposing edges of the input which occludes the structure of objects in the field. This effect is most prominent in the region around isovalue zero in each field; however, the same effect can be observed in the core observables of lattice QCD.

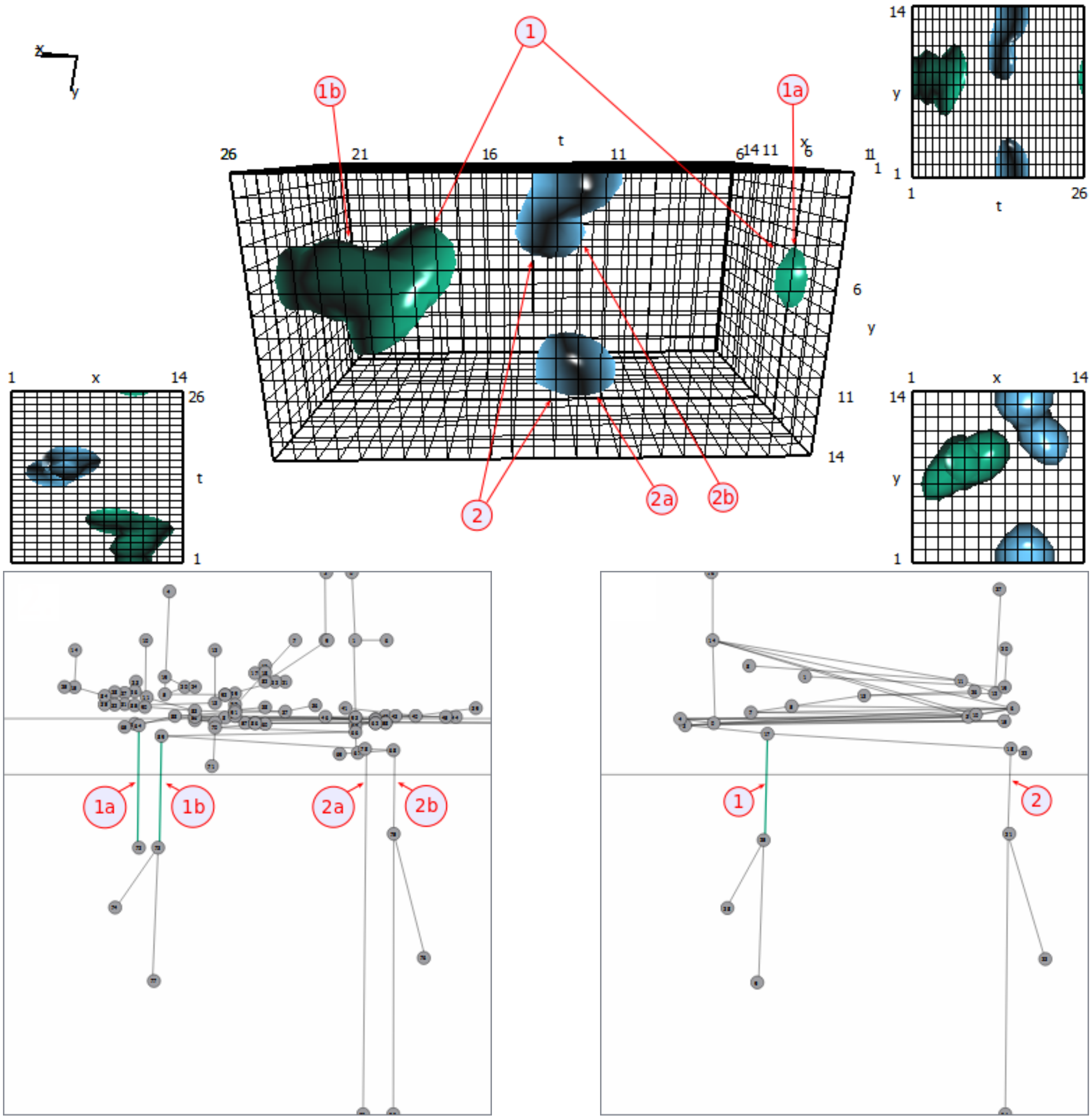


Figure 5.8: Comparison of surfaces crossing a periodic boundary in the Reeb graph and the the contour tree. Left: the contour tree captures object (1), spanning the t axis as two separate parts (1a) & (1b). Object (2) spans the y axis and is captured by the contour tree as arcs (2a) & (2b); Right: the Reeb graph is able to handle the periodic boundary enabling the two objects to be captured as single arcs (1) & (2). A union of the connected components captured as contour tree arcs (1a) & (1b) should yield the same result as that of the Reeb graph arc (1). Likewise, Reeb graph arc (2) can be recovered by merging the connected components of contour tree arcs (2a) & (2b).

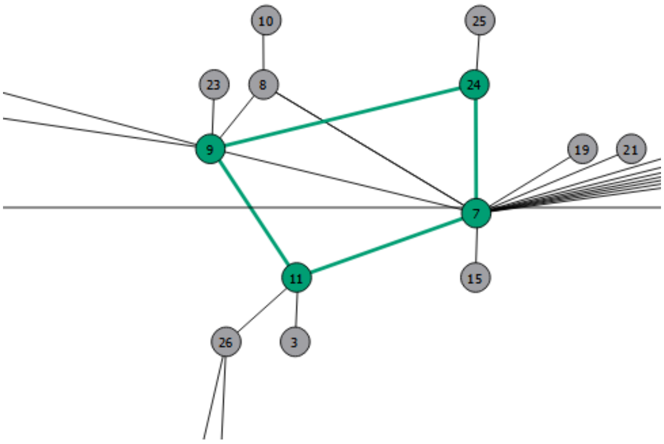


Figure 5.9: An example of a loop present in a Reeb graph. Loops are common in QCD data around isovalue zero, known as the region of percolation, representing toroidal structures in the topology.

5.3 Contributions to lattice QCD understanding

In this section we discuss how the work carried out in this chapter has the potential to further the understanding of lattice QCD by domain experts. We limit our scope to the scalar topology techniques discussed thus far, but also explore the use of temporal and multivariate algorithms in later chapters.

5.3.1 Data exploration

Initial use of the Reeb graph stimulated the QCD physicists to question if topology could be used for probing the simulations statistically by using the Reeb graph as a signature. For example, histograms can be used to calculate the distribution of objects with regard to isovalue. There is also the possibility of computing statistical measures of similarity at different time-steps by direct comparison of Reeb graphs, as discussed by Bauer et al. [8].

A number of tools were developed with the cooperation of QCD scientists to address these questions. Initially this took the form of an application that was able to generate isosurfaces at varying isovalues, allowing physicists to visually explore their data and its underlying topology (Fig. 5.4). Along with 3D rendered views of the scalar field, as seen throughout this report, we also include more traditional data visualisation techniques commonly used in the QCD domain. Histograms, in particular, are firmly established in the physics community for identifying underlying trends in data. User studies indicated that the physicists would typically use techniques that they were already familiar with to guide them towards interesting features for examination in 3D form. A full overview of the framework is given in Chapter 10.

Whilst providing interesting and novel ways of looking at QCD data, direct visual inspection of the data is infeasible, purely because of the huge quantity of configurations on offer. As a response to these views from the domain scientists we began to focus instead on computing properties of Reeb graphs, as part of an automated tool chain for comparison across an ensemble. Trends in the data appear to show that the chemical potential parameter affects the number of objects within the data. Many of these new techniques rely directly on the output of the flexible isosurface algorithm and the Reeb graph (see Section 5.3.3) and are therefore beyond the scope of what physicists were previously able to use.

5.3.2 Calculation of physical properties from simplicial meshes

The triangle meshes created for rendering also allow us to query properties of the objects. By treating each triangle as a building block of the whole object it is possible to do a piecewise sum of properties of the mesh. The simplest of these is evaluation of the surface area and volume of the mesh, calculated as the sum of the areas of each triangle. The use of Heron's formula (5.1) means this works for any triangulation of a mesh [155].

$$A = \sum_{i=1}^{n} \frac{1}{4}\sqrt{4a_i^2 b_i^2 - (a_i^2 + b_i^2 - c_i^2)^2}$$
$$where\ a_i,\ b_i,\ c_i\ are\ sides\ of\ triangle\ i\ in\ the\ mesh \qquad (5.1)$$

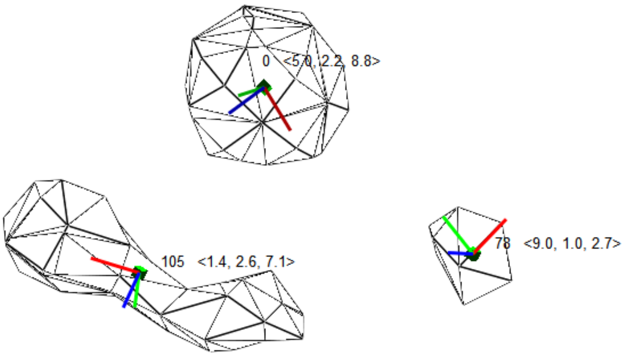


Figure 5.10: For closed objects, not spanning a boundary, it is possible to compute the centre-of-mass and moment-of-inertia quantities.

$$\chi = V - E + F \tag{5.2}$$

The volume of a mesh can be computed using two techniques; a simple measure is to count the number of scalar points that are enclosed by the mesh whilst constructing the contour tree [48]. However, the accuracy of the computed volume is limited by the resolution of the grid used in the sampling technique. A second more accurate approach, better applied to individual contours, is to compute the volume of the object directly from the triangle mesh using the mathematical concept of moments [156]. A more precise value for the volume can be calculated as the technique is insensitive to the sampling resolution of the data.

The zeroth moment represents the signed volume of a contour; the first moment, the centre of mass; and the second representing moments of inertia. A pre-requisite for correct moment calculations is for a mesh to be closed. An initial step to determine this is to calculate the Euler-Poincaré characteristic of a mesh (5.2), allowing us to determine if an object is topologically similar to a sphere, where $\chi = 2$. For toroidal objects the $\chi = 0$ and for any given n-torus a value of $\chi = -2n$. It is possible that a contour which spans two or more boundaries may be incorrectly identified as an n-torus. To counteract this, objects are also checked to see if vertices lie on the height field boundaries. A more in depth discussion of the use of moments in visualisation can be found in [157] and [158].

In order to calculate the volume of the mesh (M_{000}) we sum the individual contributions from every element. For each triangle in the mesh a tetrahedron is created using a fourth point at the origin, in order to enable calculation of a signed volume (5.3). This assumes a counter-clockwise winding of vertices in order to return the correct sign. The sum of all contributions can be negative; hence, the absolute value represents the true enclosed volume. The enclosed volume of each topological object varies considerably due to deformations of the manifolds as a result of modification of the generating isovalue. These methods are directly linked to the concept of persistence, often used for simplification of the Reeb graph or contour tree [48], and are further investigated in Chapter 6.

$$M_{000} = \frac{1}{6}(-x_3y_2z_1 + x_2y_3z_1 + x_3y_1z_2 - x_1y_3z_2 - x_2y_1z_3 + x_1y_2z_3)$$
$$where\ < x_n, y_n, z_n >\ represent\ vertices\ of\ a\ triangle \tag{5.3}$$

Centre of mass requires the calculation of three first order moments (5.4). Each tetrahedral cell is evaluated to give a weighted average in the x, y and z planes, scaled by its contributing volume. The final centre of mass for the object is calculated as a positional vector that must be scaled by the entire objects volume (5.5).

$$\begin{aligned} M_{100} &= \frac{1}{4}(x_1 + x_2 + x_3)M_{000} \\ M_{010} &= \frac{1}{4}(y_1 + y_2 + y_3)M_{000} \\ M_{001} &= \frac{1}{4}(z_1 + z_2 + z_3)M_{000} \end{aligned} \tag{5.4}$$

$$Centre\ of\ mass\ = \frac{\begin{bmatrix} M_{100} \\ M_{010} \\ M_{001} \end{bmatrix}}{M_{000}} \tag{5.5}$$

The moment of inertia requires the computation of the eigenvectors of the 2^{nd} order tensor given in (5.6), requiring further statistical measures of the mesh. First order mixed moments (5.7), which take into account averages on the plane, are calculated along with second order moments (5.8) that represent variances on each axis. Eigenvalues and eigenvectors of the matrix are computed using Schür decomposition through the use of the *Eigen* linear algebra library [159]. The three normalised vectors returned are sorted in order of magnitude of their generating eigenvalue, with the largest being the principle component axis. Comparison of the relative magnitudes of the eigenvalues, and orientation of the corresponding eigenvectors with respect to the t-axis, allows to reveal any tendency for instanton distortion, either oblate or prolate, as μ is increased.

$$Inertia\ Tensor\ = \begin{bmatrix} M_{200} & M_{110} & M_{101} \\ M_{110} & M_{020} & M_{011} \\ M_{101} & M_{011} & M_{002} \end{bmatrix} \tag{5.6}$$

$$\begin{aligned} M_{110} &= \frac{1}{20}([x_1(2y_1 + y_2 + y_3)] + [x_2(y_1 + 2y_2 + y_3)] + [x_3(y_1 + y_2 + 2y_3)])M_{000} \\ M_{011} &= \frac{1}{20}([y_1(2z_1 + z_2 + z_3)] + [y_2(z_1 + 2z_2 + z_3)] + [y_3(z_1 + z_2 + 2z_3)])M_{000} \\ M_{101} &= \frac{1}{20}([x_1(2z_1 + z_2 + z_3)] + [x_2(z_1 + 2z_2 + z_3)] + [x_3(z_1 + z_2 + 2z_3)])M_{000} \end{aligned} \tag{5.7}$$

$$\begin{aligned} M_{200} &= \frac{1}{10}(x_1^2 + x_2^2 + x_3^2 + x_1x_2 + x_2x_3 + x_1x_3)M_{000} \\ M_{020} &= \frac{1}{10}(y_1^2 + y_2^2 + y_3^2 + y_1y_2 + y_2y_3 + y_1y_3)M_{000} \\ M_{002} &= \frac{1}{10}(z_1^2 + z_2^2 + z_3^2 + z_1z_2 + z_2z_3 + z_1z_3)M_{000} \end{aligned} \tag{5.8}$$

5.3.3 Topological signatures of QCD data

We are able to use the Reeb graph to compute a “signature” of a particular lattice configuration. This allows the Reeb graphs for multiple configurations at a determined cooling iteration to be analysed statistically. Properties such as number of objects, number of loops or average arc length (a measure of topological persistence shown in Fig. 5.11) are used to quantify properties across an ensemble. Computing values between a range of ensembles is one of the measures that physicists use in evaluating lattice QCD data sets.

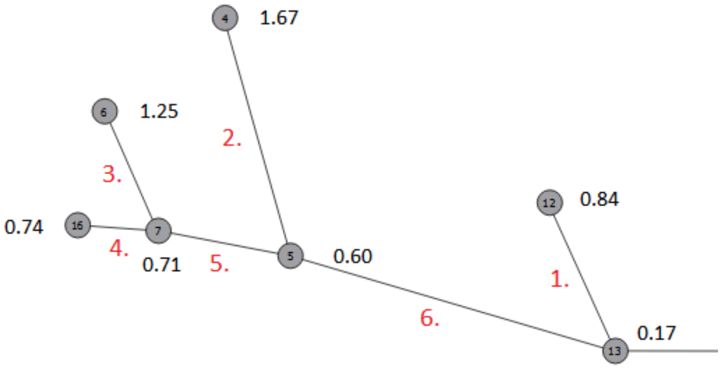


Figure 5.11: Measuring persistence on a contour tree using isovalue ranges, where the black numbers represent the isovalues at each critical vertex. Relative persistence values are: 1: $0.84 - 0.17 = 0.67$, 2: $1.67 - 0.60 = 1.07$, 3: 0.54, 4: 0.03, 5: 0.11, 6: 0.43. Note: the arc length doesn’t indicate the persistence, instead it can be understood as the difference in the y positions of critical vertices.

A more simplistic approach that is found to be interesting is to query the Reeb graph at a very basic level by summarising the number of vertices and edges. One such use is to evaluate the effect of cooling by computing Reeb graphs after each iteration, as demonstrated by Fig. 5.13. As predicted, when un-cooled the Reeb graph contains a large quantity of vertices and edges, this is due to the short lived persistence of each object. As the cooling algorithm is iteratively applied to the data this quickly drops off to become almost stable at around 15 cools. This trend appears to continue in most cases up to around the 30 cools mark (Fig. 5.12). Existing methods employed by physicists look for similar regions of flatness in graphs of peak topological charge density — the trends of the two graphs appear very similar.

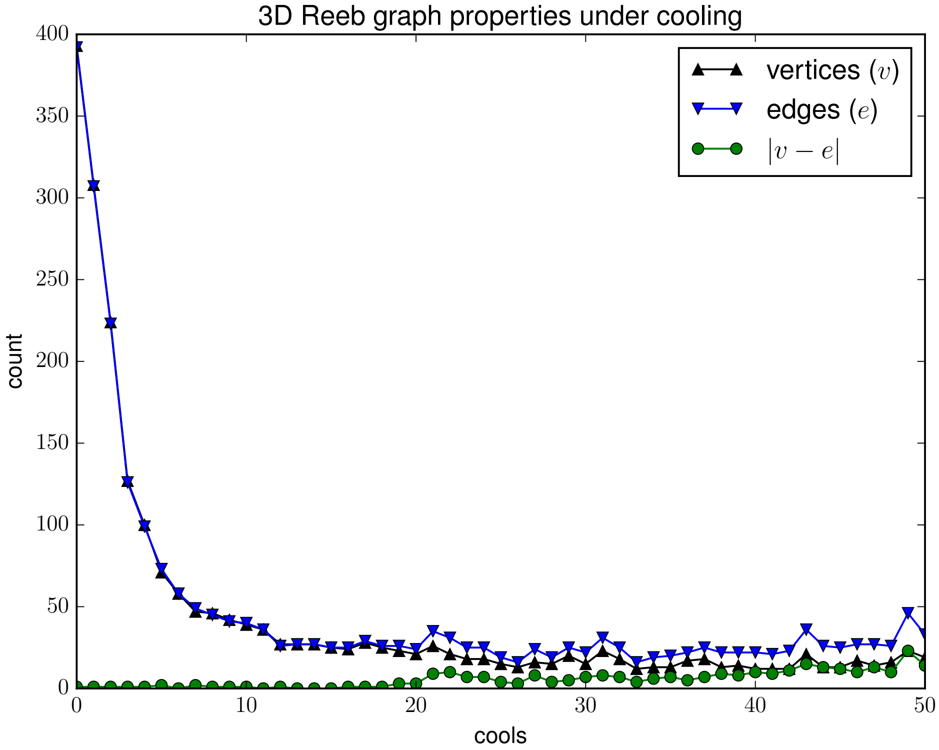


Figure 5.12: A graph in which we count the number of vertices and edges in the Reeb graph with regard to cooling. We also compute the difference; this can give a basic understanding of the number of cycles, or loops, present in the graph.

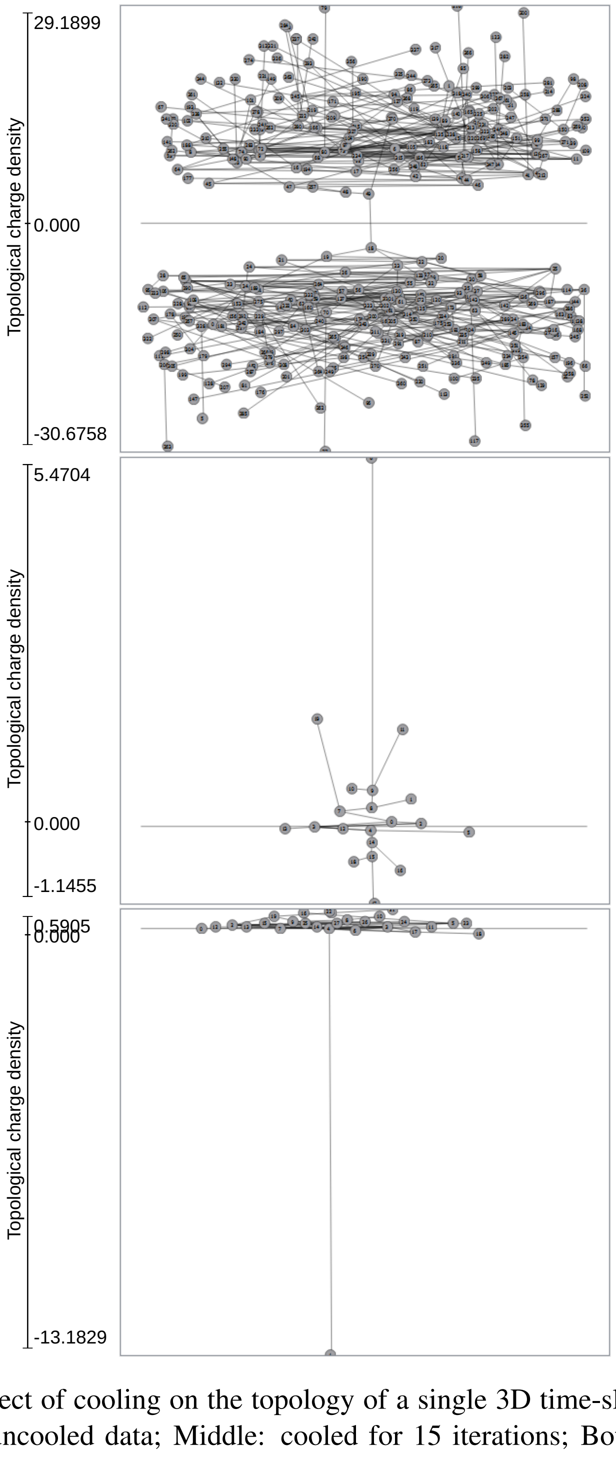


Figure 5.13: The effect of cooling on the topology of a single 3D time-slice, captured by the Reeb graph. Top: uncooled data; Middle: cooled for 15 iterations; Bottom: cooled for 30 iterations. Initially many objects in the topological charge density field are distributed around zero. As cooling takes effect the number of objects is greatly reduced and the distribution of object with regard to a neutral topological charge fluctuates as the energy of the system is reduced. By 30 cools an anti-instanton stands out in the data as the most prominent feature, indicated by the long arc below the zero line.

5.4 Summary

The use of topology to explore and analyse scalar data sets is a new approach in lattice QCD. In this chapter we have shown how close collaboration with domain experts prompted the use of topological visualisation techniques based upon the presence of interesting behaviour in the topological charge density field when visualised using isosurfaces. We were able to examine connectivity in order to split isosurfaces into multiple discrete contours, in the process forming an object based model of the data for exploration. Topology allows physicists to isolate individual lattice objects, such as potential (anti-)instantons, in order to examine their structure in greater detail. The graph based visualisations offered by the Reeb graph and contour tree enable connectivity to captured and presented in a compact format.

The mathematical models created using topological algorithms can also be used to query more advanced features of various lattice fields. The concept of moments enables properties, such as centre-of-mass and volume, to be computed directly on individual lattice observables. Properties of entire configurations can be summarised using quantities taken directly from the Reeb graph.

A summary of the contributions of this chapters is as follows:

- It was explained how topological data structures, such as the contour and Reeb graph, can be used to seed meshes for visualisation and analysis of lattice QCD observables.
- We used topology to provide domain experts new ways of probing their data that were previously unavailable to them.
- Techniques were proposed for analysing the physical properties of lattice objects.
- We demonstrated how topology can be used to compute a signature of a specific lattice configuration.

In the next chapter we continue to use topological visualisation techniques to perform an in-depth exploration of a configuration of lattice QCD data.

Chapter 6

Display and analysis of lattice QCD data

Contents

This chapter builds upon the findings of the initial experiments presented in Chapter 5 to perform an in-depth analysis of the topological charge density field. We show how topological algorithms can be used to compute a descriptor of the data, taking into account connectivity, at a range of levels of cooling. Topological visualisation techniques are used for the first time to isolate and analyse an (anti-)instanton — allowing us to examine its structure in detail under cooling. We also examine nested object meshes and associated topological graph structures at a range of isovalues to demonstrate how the techniques can enhance exploration and analysis of lattice QCD data. Finally, we apply the concepts of topological persistence to the (anti-)instanton to quantitatively analyse its structure using techniques previously unavailable to domain scientists.

In Section 6.1 we give details of the approach to topological visualisation used in this work. These techniques are used to visualise the topological charge density field at a range of cooling iterations in Section 6.2 in the form of a case study. We also show how persistence measures, computed directly from the topological graph structures, can be used to assign additional meaning to structures on the lattice in Section 6.3.

6.1 Implementation

Using knowledge and feedback gained from initial experiments with lattice QCD data, topological visualisation techniques are used to capture and generate individual object meshes.

This has two distinct advantages; data is automatically captured as unique objects, and can be displayed without the additional overhead of direct volume rendering. This approach is well suited to our application as it enables computation of numeric properties of meshes, such as range based persistence, at a level of accuracy that satisfied our users.

6.1.1 Surface construction

The basis of this work is the flexible isosurface algorithm [58] which uses the contour tree as an internal data structure. The contour tree provides seeds to generate contours for a given isovalue using marching cube style lookup tables. The algorithm has been heavily modified to suit our needs; in particular it forms the basis of an object oriented approach to working with individual lattice objects. Whilst this approach cannot deal directly with the periodic boundaries present in lattice QCD, it is suitable to allow domain scientists to visually explore their data.

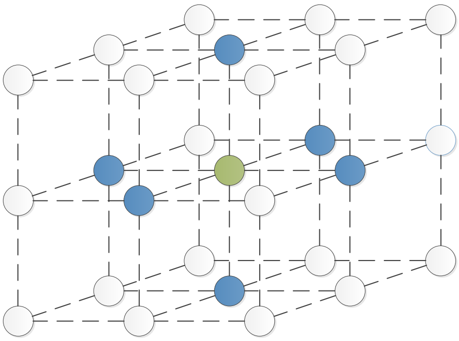


Figure 6.1: Neighbourhood used in construction of the join tree.

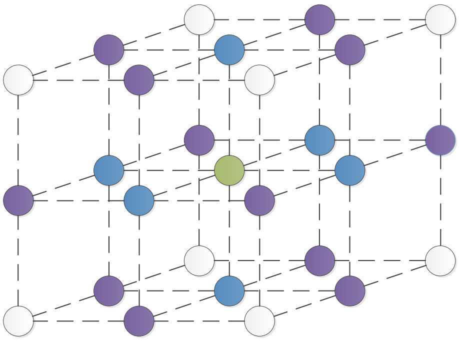


Figure 6.2: Neighbourhood used in construction of the split tree.

The construction phase of the contour tree, requires merge trees of the data to be computed by performing a union-find segmentation [4] of the scalar volume. For computation of the join tree, samples are presorted from lowest to highest to boost the computation speed of the algorithm [51]. Each sample is compared against a neighbourhood of *six* nearest neighbours (Fig. 6.1), components are merged if a neighbour has a higher function value. The split tree algorithm uses samples sorted from highest to lowest and considers a neighbourhood of *eighteen* neighbours (Fig. 6.2). Components in the split tree are merged with neighbours with a lower function height.

Reeb graphs of the data are generated using VTK 6.3 [160]. This uses the *vtkUnstructuredGridToReebGraphFilter* algorithm to extract the Reeb graph from a simplicial subdivision of the input field. Each cubic cell in the input is divided into six tetrahedra using a Freudenthal subdivison (Fig. 6.3), to provide an optimal number of tetrahedra for the computation [161]. In order to handle periodic boundaries an additional cell linking the minimum and maximum samples on each axis is defined. For four dimensional fields the input is first sliced on a user defined axis to reduce the dimensionality by one. The decision to work in three dimensions rather than four is linked to the complexity involved in performing simplicial subdivisions of hypercubic cells. This is discussed in greater detail in the future work section of this thesis (Section 11.2).

[redacted]

Figure 6.3: Freudenthal subdivison of a cubic cell into 6 tetrahedra.
Graphic is taken from Carr et al. [161].

6.2 Display and analysis of lattice QCD data

Due to the vast array of configurations available from the lattice QCD data sets we present the results in the form of a case study chosen with the help of domain specialists. This demonstrates how a physicist may investigate some of the advanced properties of lattice QCD fields accessible through the use of topology.

In order to locate regions of interesting topological structure output from the cooling code, described in greater detail in Sec. 4.3.2, is used to predicate the locations of global minima and maxima. Cooling is iteratively applied to the lattice with objects that persist at the isovalue extremes being candidate (anti-)instantons. Upon locating and viewing potential (anti-)instantons it is possible to observe the effect cooling has on the object by viewing it at multiple cooling iterations. The Reeb graph and contour tree are used to demonstrate how these objects are identified in the scalar topology. Finally, some basic persistence measures of the objects are computed using the Reeb graph and contour tree.

The goals for this task are summarised as follows:

- What happens to the topological charge density as the level of cooling is increased?

- How can topology be used to compute the effect cooling has on objects in the scalar field?

Cools	S_{MAX}	Q_{MAX}	Q_{MIN}
24	(6, 11, 12, 6)	(5, 12, 7, 16)	(12, 9, 8, 17)
25	(6, 11, 12, 6)	(5, 12, 7, 16)	(12, 9, 8, 17)
26	(6, 11, 12, 6)	**(6, 11, 12, 6)**	(12, 9, 8, 17)
27	**(5, 10, 7, 9)**	**(5, 12, 7, 16)**	(12, 9, 8, 17)
28	(5, 10, 7, 9)	(5, 12, 7, 16)	(12, 9, 8, 17)
29	(5, 10, 7, 9)	(5, 12, 7, 16)	(12, 9, 8, 17)
30	**(12, 9, 8, 17)**	**(5, 10, 7, 9)**	(12, 9, 8, 17)
31	**(5, 12, 6, 16)**	**(5, 12, 7, 16)**	(12, 9, 8, 17)
32	(5, 12, 6, 16)	**(6, 8, 10, 22)**	(12, 9, 8, 17)
33	(5, 12, 6, 16)	(6, 8, 10, 22)	**(5, 12, 6, 16)**
34	**(11, 9, 8, 17)**	(6, 8, 10, 22)	(5, 12, 6, 16)
35	(11, 9, 8, 17)	**(11, 9, 8, 17)**	**(9, 9, 5, 15)**
36	(11, 9, 8, 17)	(11, 9, 8, 17)	(9, 9, 5, 15)
37	(11, 9, 8, 17)	(11, 9, 8, 17)	(9, 9, 5, 15)

Table 6.1: The location on the lattice of global minima and maxima in the *conp0005* configuration. Changes between cooling iterations are highlighted in **bold**.

- Does the orientation of slicing in space-time affect observed properties of (anti-)instanton objects?

- What limitations are presented by lattice QCD data?

6.2.1 Selecting a candidate configuration for visual analysis

The data used in this case study is pre-cooled using the *suNtopology* cooling code [134] and has a fixed chemical potential $\mu = 0.9$. The lattice has an extent of 12 sites in space and 24 sites in time ($12^3 \times 24$) and is commonly referred to as a "cold" lattice by physicists. However, the process used here can be applied to any lattice data.

The first step undertaken by a physicist is to select an interesting configuration to be viewed from the entire ensemble. The graph of total topological charge Q for each lattice configuration can be viewed across the full range of cooling iterations, observing plateaus in the data to signify the lattice is in a stable state. Further cooling results in a net jump in the topological charge density, signifying an instanton-anti-instanton annihilation or a topological object falling through the lattice. In Fig. 6.4 we observe that most configurations become stable at around 30 cooling iterations with the exception of *conp0015*. This case study focuses on the *conp0005* configuration which is stable between 20 and 25 cool, but sees a net increase in topological charge density between 25 and 30 cools.

6.2.2 Preparation of configuration for visual analysis

Topological charge density fields are generated at multiple cools using feedback from the peak topological charge density graph (Fig. 6.4) and peak Wilson action graphs (Fig. 6.5). The Wilson action appears to fluctuate between cools 24 and 34, before dropping off and stabilising after 36 iterations. This range of the cooling process is chosen to observe the cause of this behaviour and the effect it has on the scalar topology.

The peak topological charge density Q_{PEAK} (Fig. 6.6) is a very basic prediction of the size of (anti-)instantons, and remains relatively smooth throughout the cooling process. However, also present are large deviations that can be cross referenced with the predicted locations of (anti-)instantons from the cooling code (Table 6.1). Fluctuations in peak topological charge density appear to be coupled to changes in lattice position of the global minima and maxima.

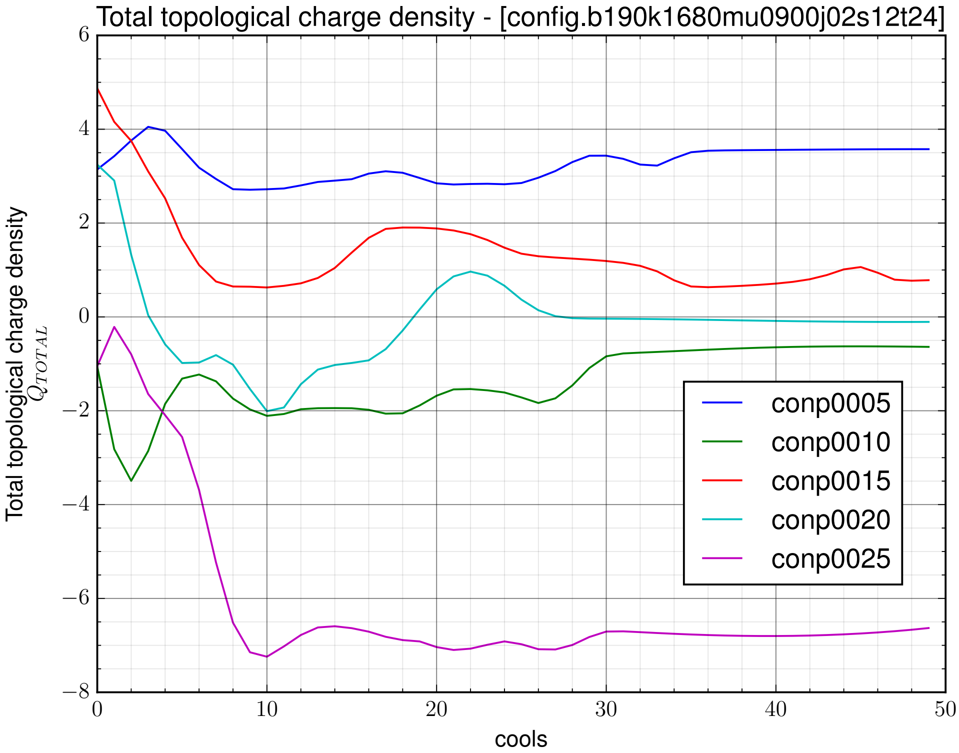


Figure 6.4: Viewing total topological charge density for multiple configurations in an ensemble.

6.2.3 Visual inspection of configuration

Using the ensemble module, further discussed in Sec. 10.2.1, the *conp0005* configuration is selected at cooling iterations 24 through to 36. However, the data must first be reduced from a four dimensional space-time hypervolume to a three dimensional volume using slicing. As lattice QCD treats space and time as equivalent it is equally as valid to slice on the space-like axes as it is on the time-like axis. This unique feature of lattice QCD is demonstrated in Fig. 6.7 where we visualise an instanton object predicted by the cooling code at 33 cools. The location of this object, taken from Table 6.1, is $(6,8,10,22)$; orthogonal views of the data help to verify that the correct object is being visualised. We can also use this approach to confirm that (anti-)instantons are identified as the lowest and highest superarcs in the Reeb graph and contour tree at positions predicted by the cooling code.

6.2.3.1 Inspecting (anti-)instanton structure

Referring to Table 6.1 we are able to observe that the location of the global minima in the topological charge density field Q_{MIN} remains constant up until cooling iteration 33. When

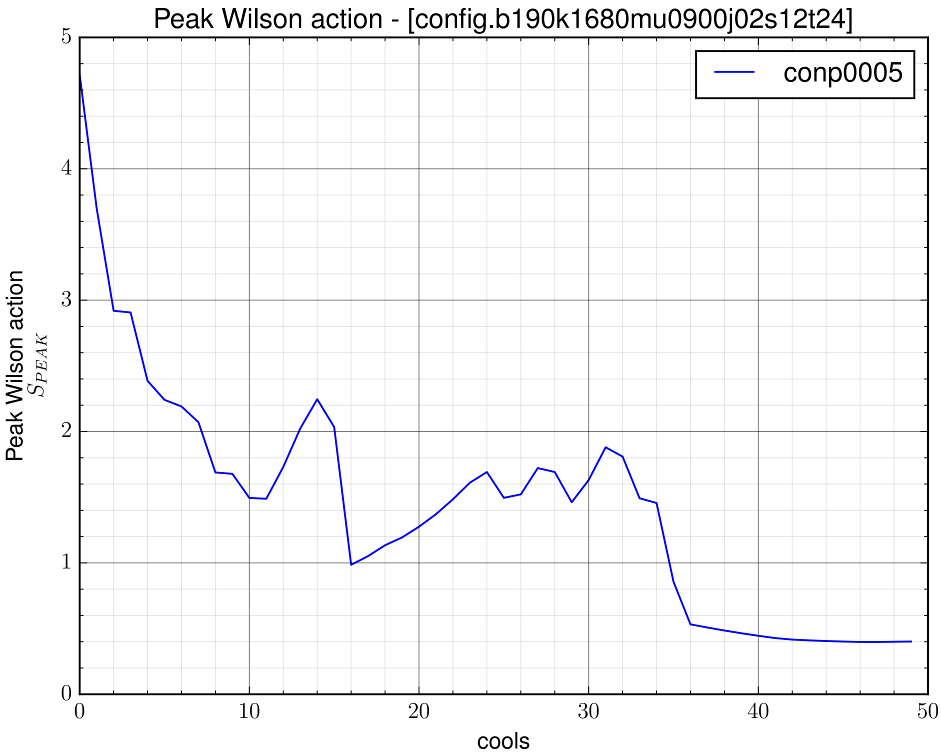


Figure 6.5: Viewing peak Wilson action for the *conp005* configuration.

visualised in isolation, using the selection features of the flexible isosurface algorithm, its structure can be studies in depth when subjected to cooling. Due to the objects proximity to a periodic boundary ($x = 12$) we slice on the x axis to obtain a view of the object without it crossing the boundary.

In each Reeb graph visualisation (Fig. 6.8) the green superarc captures the anti-instanton Q_{MIN}. Each Reeb graph is shown alongside a rendered surface view of the data it represents. Critical nodes in the Reeb graphs are mapped to the horizontal axis using the function $f(x) = (v_x + v_y + v_z) \div (dim_x + dim_y + dim_z)$ to preserve a sense of spatial relation between the nodes, a technique initially proposed by Heine et al. [162].

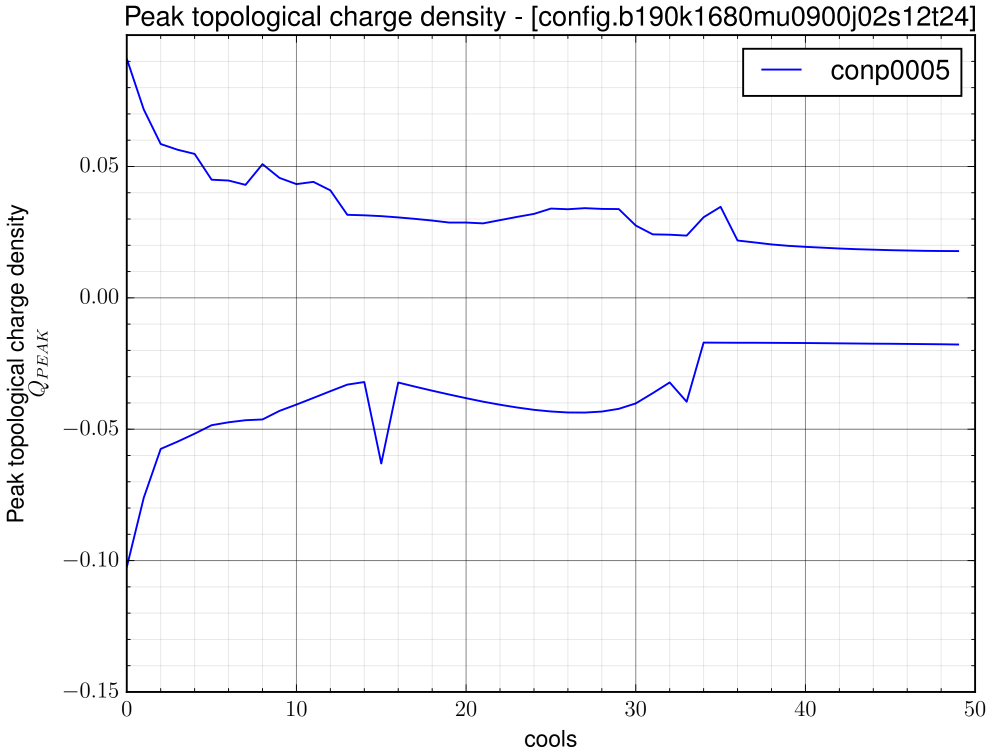
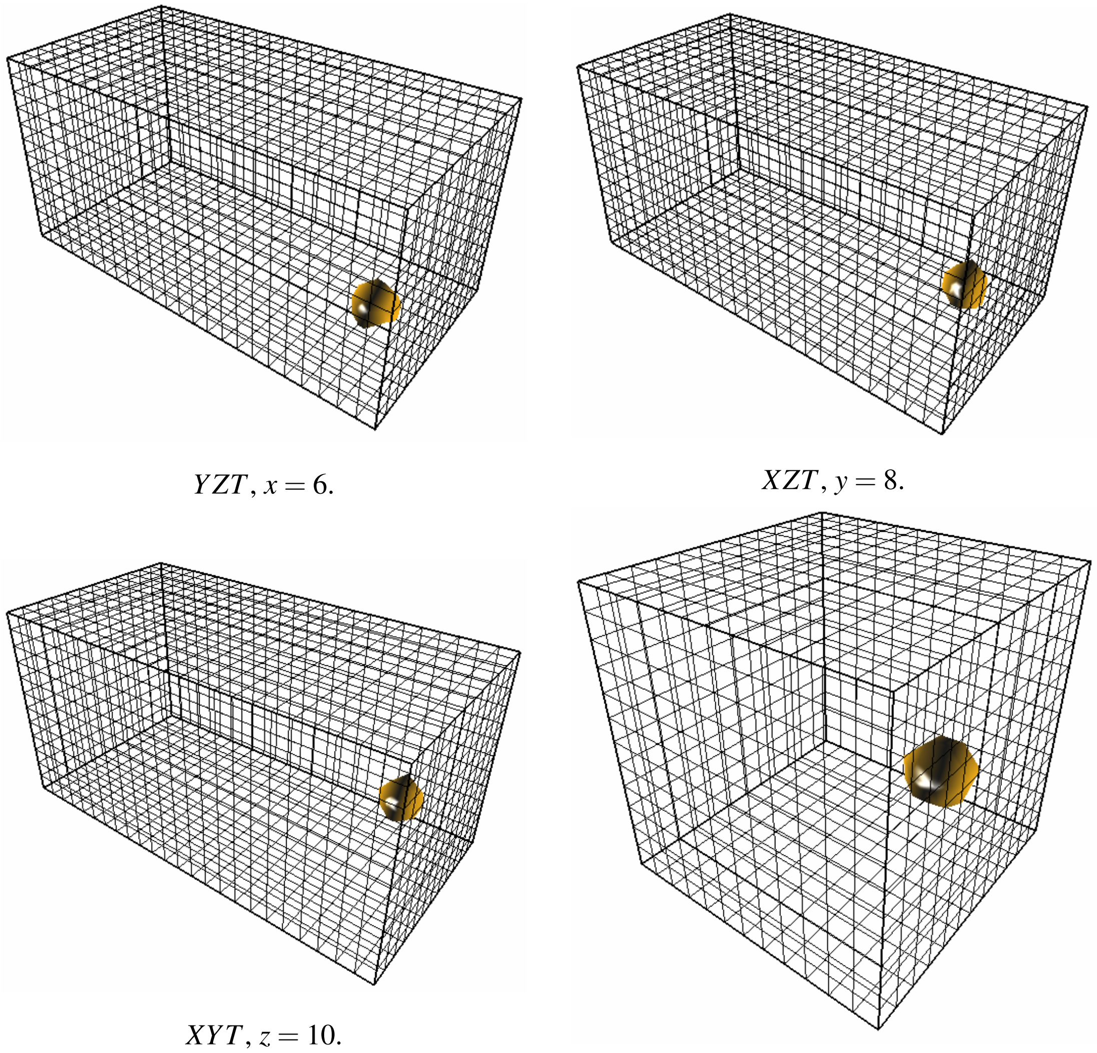


Figure 6.6: Viewing peak topological charge density for the *conp005* configuration.



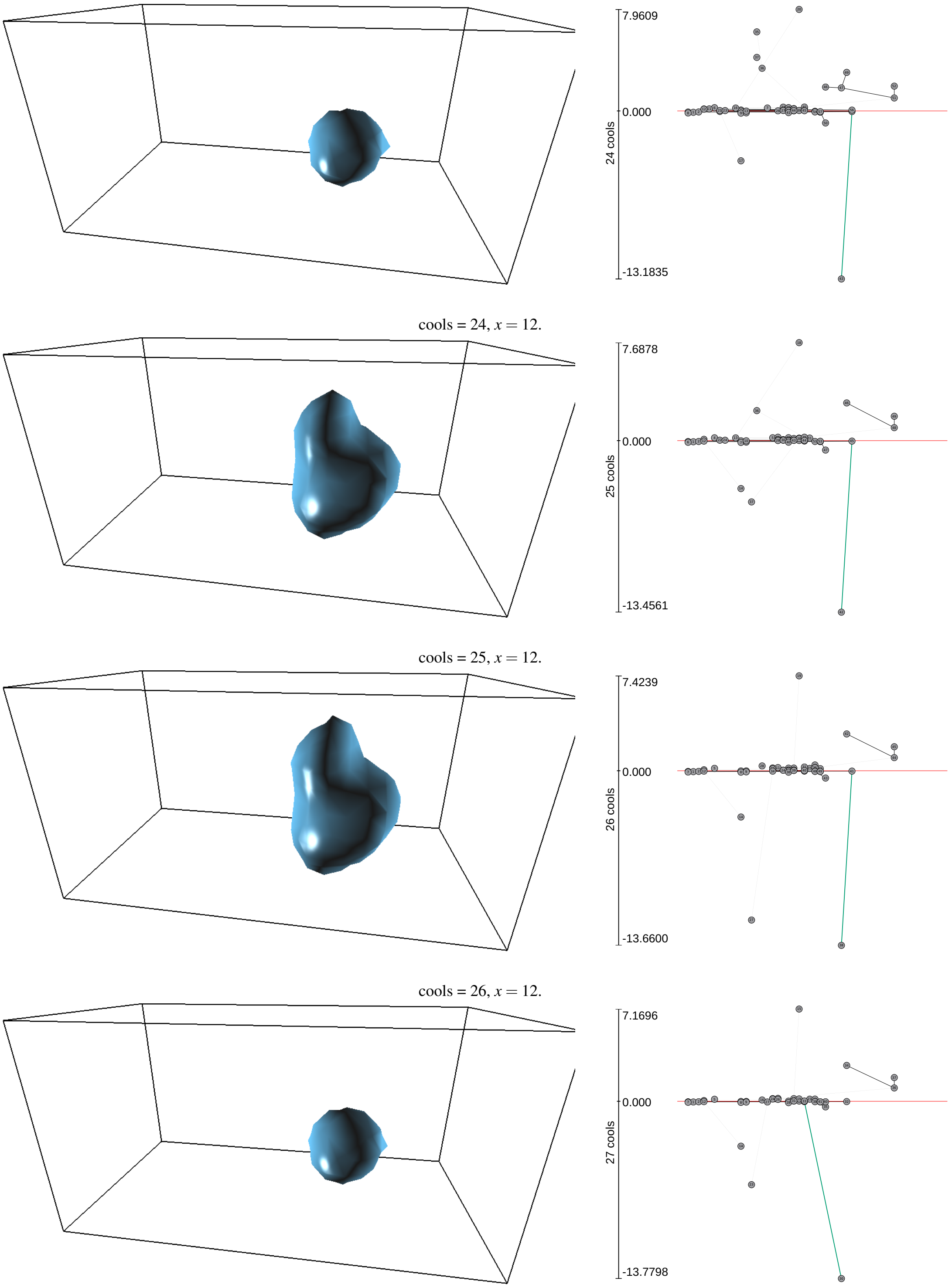
YZT, $x = 6$.

XZT, $y = 8$.

XYT, $z = 10$.

XYZ, $t = 22$.

Figure 6.7: Viewing the same instanton in four different space-time volume configurations.



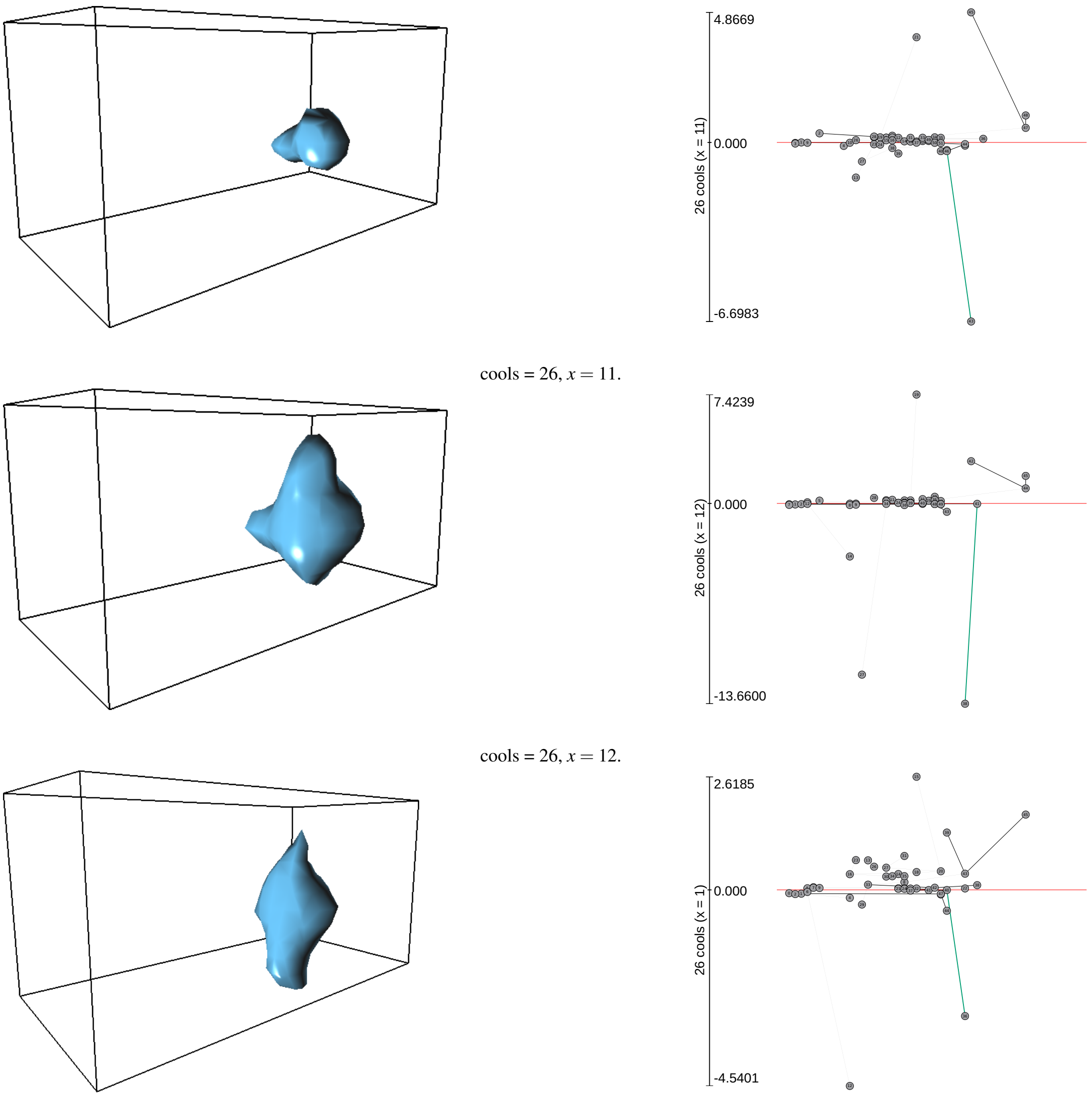
cools = 24, $x = 12$.

cools = 25, $x = 12$.

cools = 26, $x = 12$.

cools = 27, $x = 12$.

Figure 6.8: Visualising an anti-instanton as isolated contours (left) at cools 24 – 27; the green Reeb graph arcs (right) represents the anti-instanton at each stage of cooling.



cools = 26, $x = 11$.

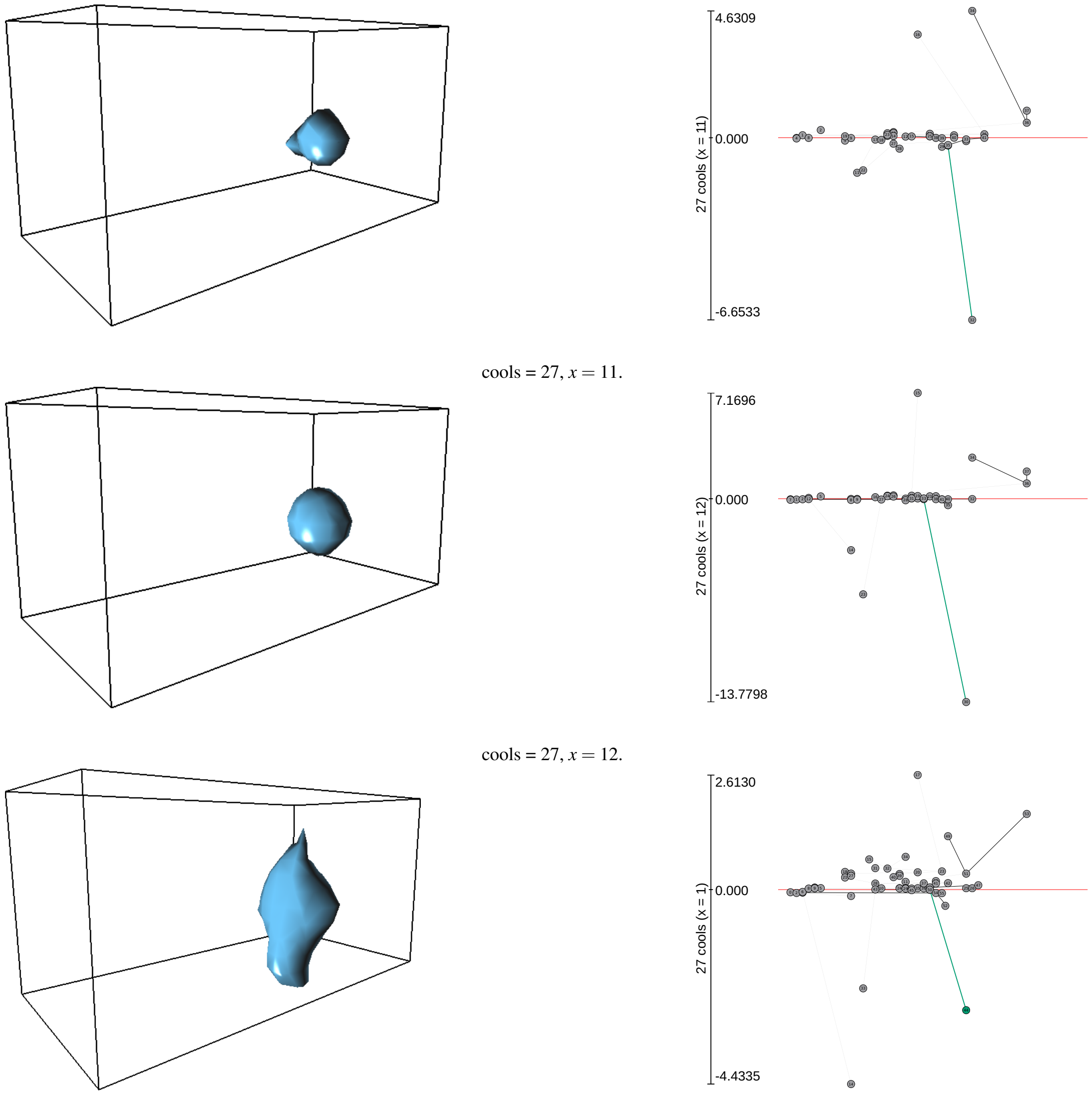
cools = 26, $x = 12$.

cools = 26, $x = 1$.

Figure 6.9: Visualising an anti-instanton in isolation at different x slices at 26 cools. The contour surfaces (left) suggests that the object structure fluctuates in space-time. The green Reeb graph arcs (right) captures the anti-instanton in the context of objects on the lattice across the isovalue range.

The Reeb graph presents the topological structure in a context that is not available using existing statistical physics techniques. Using this approach an unexpected oscillating behaviour is observed over the cooling axis, characterised by a deformation from a uniform spherical object suggesting that another nearby object is influencing the shape of the object by distorting its shape. However, it is difficult to make conclusions directly from this viewpoint as it is possible the object also distorts in the fourth x dimension. Figures 6.9 and 6.10 further investigate the oscillating behaviour by moving the x axis forward and back one step at 26 and 27 cools. This enables the structure viewed in Fig. 6.8 to be observed in the context of its neighbouring slices in space-time. Similar structures are observed in the $x = 11$ and $x = 1$ slices at both cooling iterations — we are unsure how to completely explain the variation in shape between cools 26 and 27 in the $x = 12$ view. One possible explanation is that part of the structure is lost to neighbouring isosurfaces with higher function value. This could explain the appearance of a prominent secondary arc in the Reeb graph alongside the main observable in cools 25 and 26 (Fig. 6.8). This object contracts again at 27 cools suggesting it may correspond to the anti-instanton returning to a spherical form.

The oscillation repeats at later cooling iterations, up until cooling slice 33, where there is the emergence of different behaviour. Figure 6.11 shows the same topological object between cooling iterations 33 and 36. At iteration 33 the object is spherical but as cooling proceeds it begins to split into two objects, with splitting behaviour captured in the Reeb graph at 35 cools as a fork. The large superarc representing the object in earlier configurations is reduced in size and split into two smaller length arcs part way along its length. Moving forward to 36 cooling iterations the two objects are virtually removed from the lattice by the algorithm. This behaviour encouragingly corresponds to an increase and subsequent flattening of the total topological charge (Fig. 6.4) and a plateau in the Wilson action (Fig. 6.5). Both these features indicate that what we are observing is a candidate for an object falling through the lattice as a result of over cooling. Viewing the object as a contour alongside the Reeb graph appears to agree with this assumption.

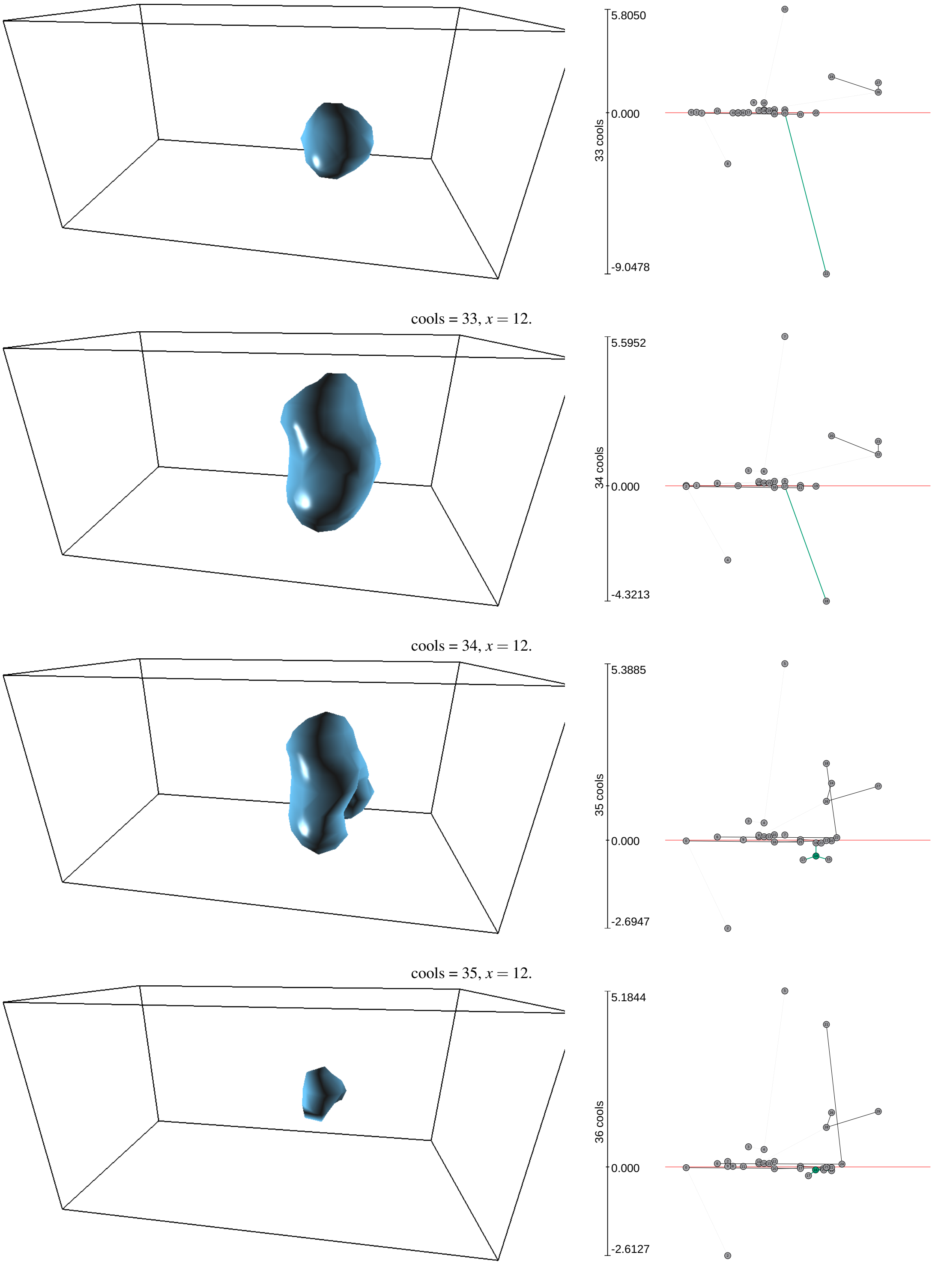


cools = 27, $x = 11$.

cools = 27, $x = 12$.

cools = 27, $x = 1$.

Figure 6.10: Visualising an anti-instanton in isolation at different x slices at 27 cools. The anti-instanton structure (left) continually changes in space-time; this effect could be caused by the existence of other objects on the lattice with similar isovalues in the Reeb graph (right).



cools = 33, $x = 12$.

cools = 34, $x = 12$.

cools = 35, $x = 12$.

cools = 36, $x = 12$.

Figure 6.11: Visualising an anti-instanton in isolation at cools $33 - 36$. The anti-instanton begins to split into two parts at 35 cools (left); this process is captured in the Reeb graph (right) by the green arcs and vertices.

6.2.3.2 Inspecting topological field structure

Figures 6.12, 6.13, 6.14 and 6.15 present an example of the cooling process using a different approach. Instead of viewing an instanton in isolation the same space-time slice, with $t = 1$, is viewed as a series of 25 nested contours distributed across the scalar range. Objects with a positive isovalue are coloured orange and those with a negative isovalue appear blue.

Cooling iterations 24 to 27 are illustrated in Fig. 6.12. Rendered contours allow a sense of the balance of positive and negative objects to be formed by the user. A few small regions of negatively charged objects are present within a large region of positive surfaces. Reeb graphs capture this by displaying more objects, represented as superarcs, above the zero line. Counting the number of critical nodes existing above v_+ and below v_- the zero line enables the balance to be captured as a numeric quantity.

Moving along the cooling axis makes it possible to witness the growth of regions of negative charge. This characterises the short scale smoothing effect that the cooling algorithm uses to remove noise. Whilst the regions continue to grow with cooling they remain unique topological objects and the number of negative critical points v_- remains relatively stable. The global maxima in the field is visible at the bottom centre as a number of nested contour shells that persist throughout the entire process.

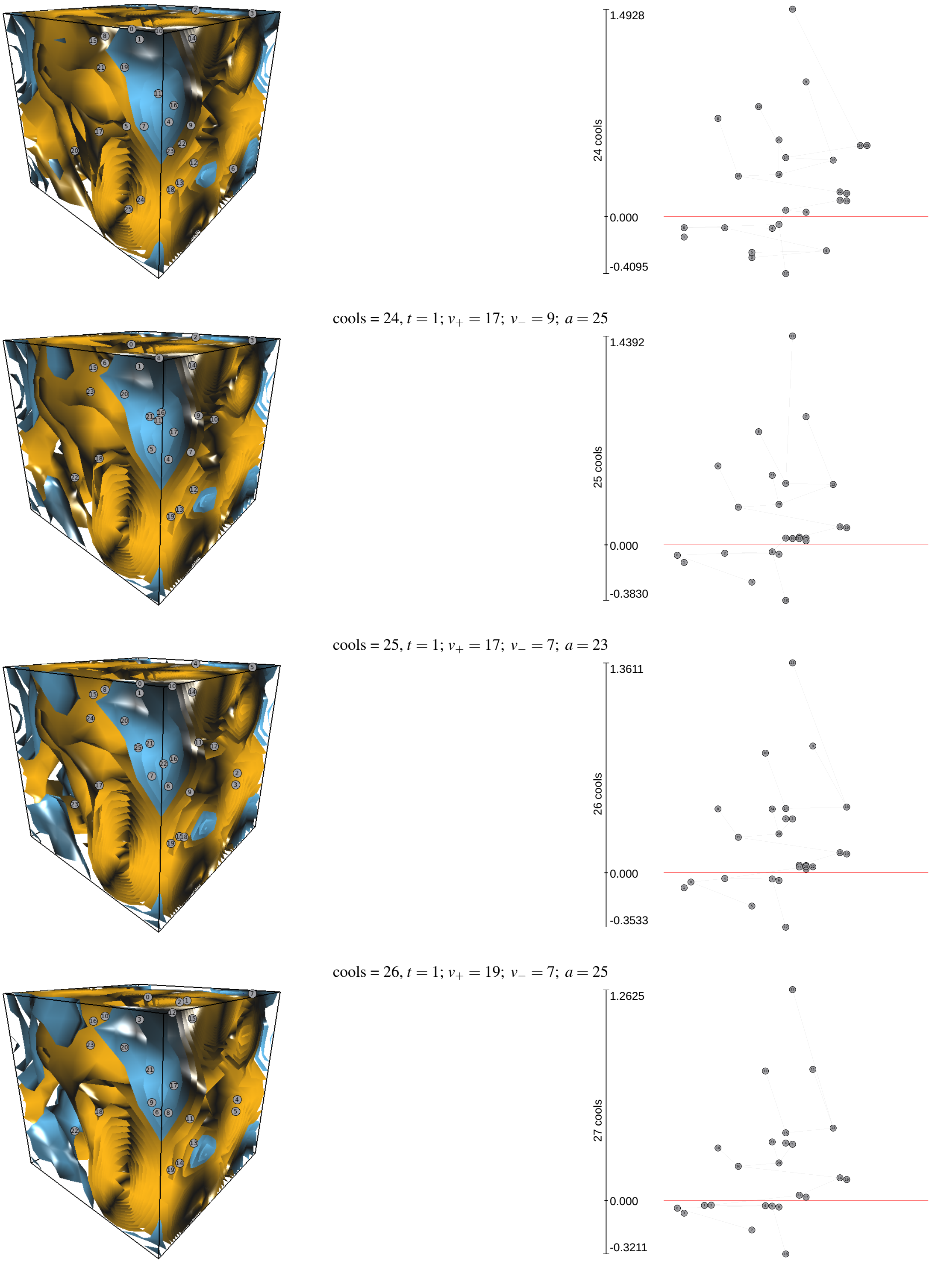
Further cooling, at iterations 27 to 30 (Fig. 6.13), shows an interesting phenomena emerge in the cooled configurations at iteration 29. A large region of negative charge suddenly become positive, illustrating how fluctuations in the data centre on zero. A clustering of superarcs and supernodes in the Reeb graph signifies many small objects have an isovalue close to zero.

Also present at cool 29 is a small surge of complexity in the topology as the number of critical nodes and objects briefly increases — examining the scalar field reveals that these objects are shown to relate to small isolated regions. One example is the small negatively charged surface in the top left of slice 29 which is surrounded by a positive contour shell that previously had negative isovalues. In cool 30 this region is further expanded but remains topologically isolated from any other regions of negative charge.

As the data is further cooled to 33 iterations (Figure 6.14) the object that appeared in cool 30 continues to increase in size until it merges with another large region of negative charge in cool 32. This merge sees a number of objects switch from a positive to negative isovalue, captured by the Reeb graphs nodes and arcs crossing the zero line. These objects appear to again switch back to a positive charge in cool 33.

Also observable in the Reeb graph is the bunching up of a group of geographically close objects either side of the zero line in cools 33 – 35. The two clusters appear to be have a large spatial separation, indicated by the horizontal location in the Reeb graph. This suggests that the regions of positive and negative isovalue are becoming increasingly large, as we expect from the smoothing effect of the cooling algorithm. The rendered view enables a link to be made with the large positive region separating a negative region spanning the periodic boundary.

In the final series of visualisations (Fig. 6.15) a large negatively charged region continues to grow in size until it is almost self connected across the periodic boundary in cool 36. Throughout the process it is possible to observe the merging of a number of previously distinct regions. This process is captured in the Reeb graph as a drop in the overall number of negative critical nodes. Visualising the entire process demonstrates how the cooling algorithm is able to allow the desired (anti-)instantons to persist throughout cooling as the minima and maxima in the graph.



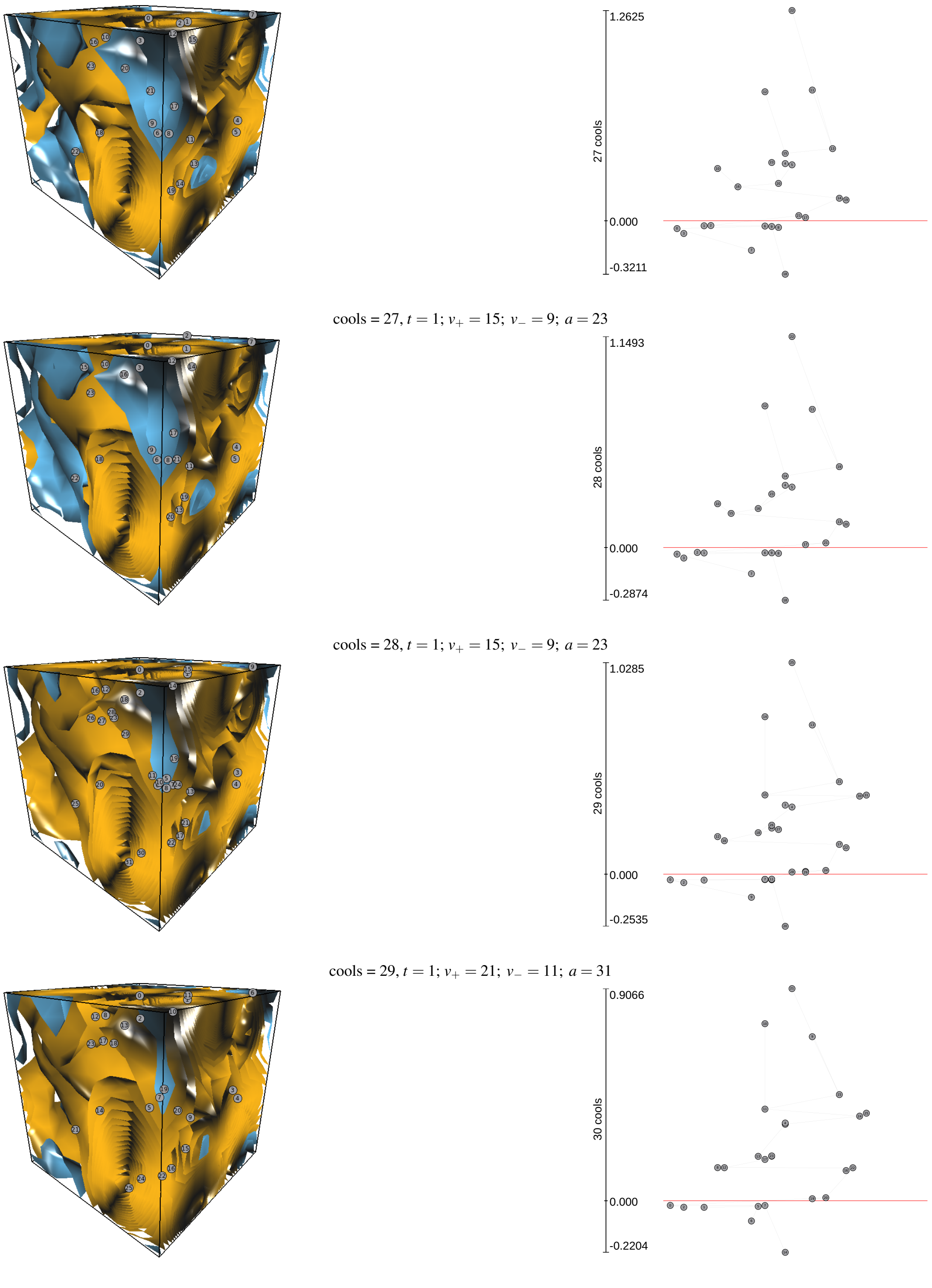
cools = 24, $t = 1$; $v_+ = 17$; $v_- = 9$; $a = 25$

cools = 25, $t = 1$; $v_+ = 17$; $v_- = 7$; $a = 23$

cools = 26, $t = 1$; $v_+ = 19$; $v_- = 7$; $a = 25$

cools = 27, $t = 1$; $v_+ = 15$; $v_- = 9$; $a = 23$

Figure 6.12: Nested contours (left) and associated Reeb graphs (right) at cooling iterations $24 - 27$. Cooling sees the number of objects (arcs a) fluctuate between iterations.



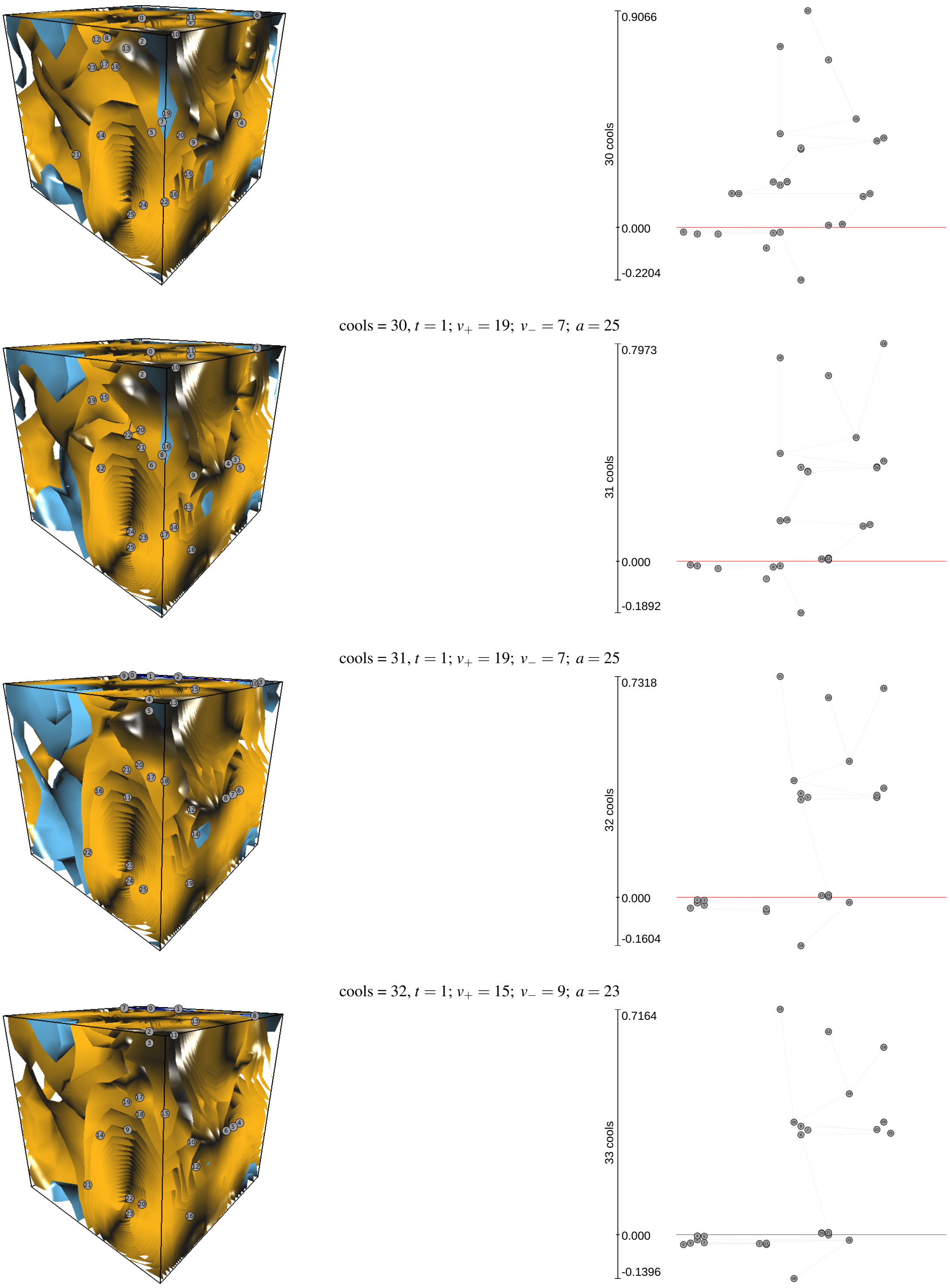
cools = 27, $t = 1$; $v_+ = 15$; $v_- = 9$; $a = 23$

cools = 28, $t = 1$; $v_+ = 15$; $v_- = 9$; $a = 23$

cools = 29, $t = 1$; $v_+ = 21$; $v_- = 11$; $a = 31$

cools = 30, $t = 1$; $v_+ = 19$; $v_- = 7$; $a = 25$

Figure 6.13: Nested contours and associated Reeb graphs at cooling iterations 27 – 30. The surge in the number of objects a at 29 cools is thought to be caused by the splitting of a large connected region into several smaller parts.



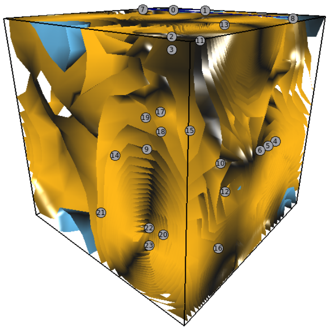
cools = 30, $t = 1$; $v_+ = 19$; $v_- = 7$; $a = 25$

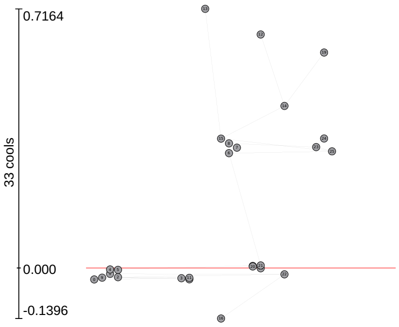
cools = 31, $t = 1$; $v_+ = 19$; $v_- = 7$; $a = 25$

cools = 32, $t = 1$; $v_+ = 15$; $v_- = 9$; $a = 23$

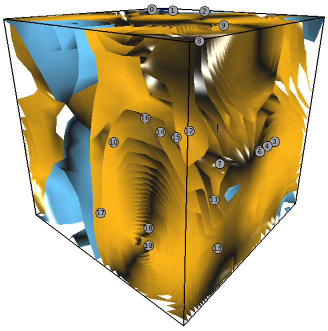
cools = 33, $t = 1$; $v_+ = 14$; $v_- = 12$; $a = 25$

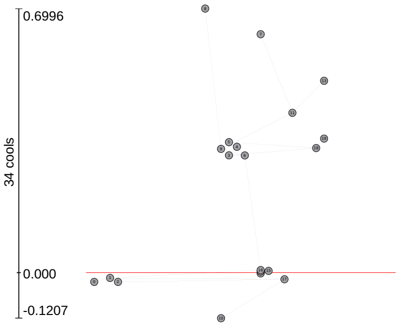
Figure 6.14: Nested contours and associated Reeb graphs at cooling iterations $30-33$. As cooling progresses isovalues and critical vertices v move towards a neutral isovalue.



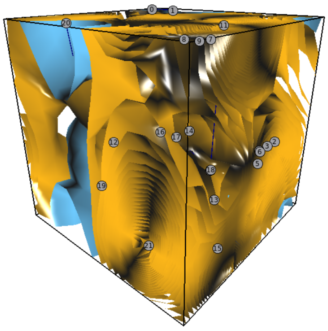


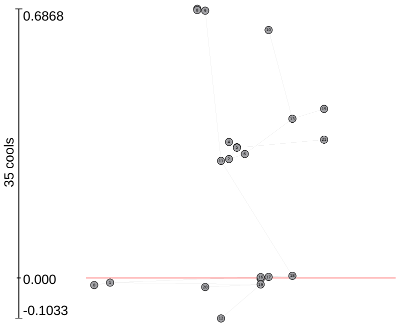
cools = 33, $t = 1$; $v_+ = 14$; $v_- = 12$; $a = 25$



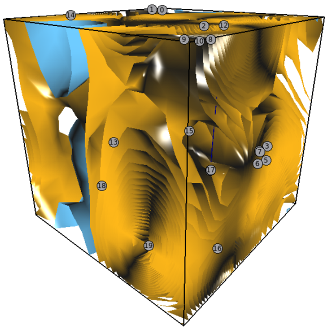


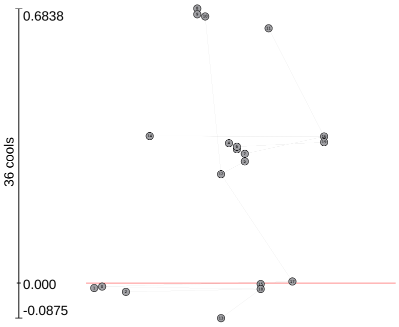
cools = 34, $t = 1$; $v_+ = 14$; $v_- = 6$; $a = 19$





cools = 35, $t = 1$; $v_+ = 16$; $v_- = 6$; $a = 21$





cools = 36, $t = 1$; $v_+ = 14$; $v_- = 6$; $a = 19$

Figure 6.15: Nested contours and associated Reeb graphs at cooling iterations $33 - 36$. The Reeb graph shows an imbalance in the number of critical points above v_+ and below v_- zero.

6.3 Quantitative analysis of the Reeb graph under cooling

When inspecting the topological charge density field (Q) it becomes clear from the Reeb graph that the topological structure is more complex than simple visual inspection via isosurfaces may suggest. The number of unique objects (represented by superarcs) and topological events (the supernodes) fluctuate throughout the process, rather than being a smooth transition to a simpler topological structure. Objects centred around zero frequently switch sign at several points in the process which is characterised by a change in colour in the rendered view and the displacement of the supernodes and arcs in the Reeb graph. We attempt to quantify this behaviour by graphing basic attributes of the Reeb graph as presented in Table 6.2.

Figure 6.16 plots the number of supernodes and superarcs in the Reeb graph as cooling progresses. Whilst initial predictions were that the topology would progressively move towards a simpler state, fluctuations are caused by the field being split into distinct positive and negative regions. As cooling takes effect these regions gradually spread out, merging with others of the same charge. In some instances the merging of two regions of positive charge cuts through an area of connected negative charge, splitting the region into two distinct parts. This is demonstrated by the switching of a topological object from negative to positive charge at 28 to 29 cools (Fig. 6.13), resulting in the creation of many smaller unconnected objects with negative charge. This corresponds to a spike in topological complexity at 29 cools, which is possible to capture from the Reeb graph by examining the number of supernodes and superarcs.

Further investigation of the fluctuating behaviour by plotting the relative distributions of positive v_+ and negative (v_-) supernodes is demonstrated in Fig. 6.17. The graph shows that the number of positive critical points v_+ always outnumbers the negative v_- in this particular configuration, appearing to agree with the observed contours. Regions of positive charge seem to dominate, particularly at high levels of cooling (Figs. 6.14 and 6.15). Calculating the difference presents a fluctuating pattern capturing the behaviour of clusters of supernodes close to zero in the Reeb graph.

6.3.1 Using topological persistence to measure (anti-)instantons

In this section various topological persistence measures are used to estimate properties of the anti-instanton object studied in this case study. The visualisation module allows the user to in-

Cools	v_+	v_-	v_{TOT}	v_{DIF}	a
24	17	9	26	8	25
25	17	7	24	10	23
26	19	7	26	12	25
27	15	9	24	6	23
28	15	9	24	6	23
29	21	11	32	10	31
30	19	7	26	12	25
31	19	7	26	12	25
32	15	9	24	6	23
33	14	12	26	2	25
34	14	6	20	8	19
35	16	6	22	10	21
36	14	6	20	8	19
37	14	6	20	8	19

Table 6.2: Quantifying Reeb graph complexity across the cooling process.

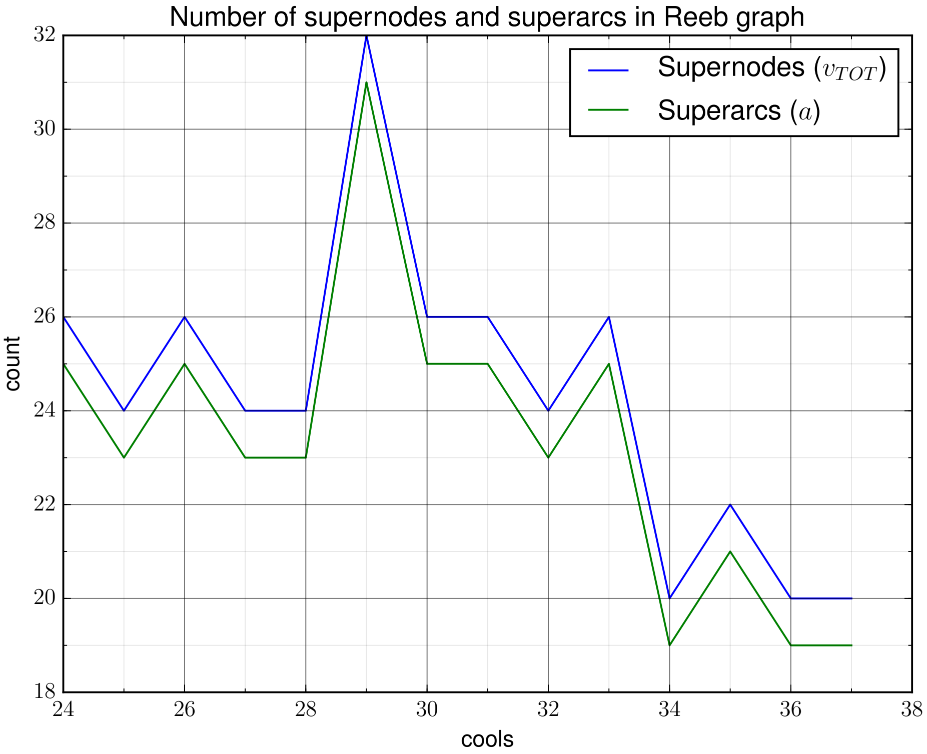


Figure 6.16: Total number of supernodes and superarcs in the Reeb graph during cooling.

teractively query persistence in the visualisation module by selecting an object with the mouse.

The persistence measures used here are captured directly from the topological graph structures to provide a simplistic method of summarising persistence for each object. An alternative but slower approach is to compute properties from each mesh using the techniques described in Chapter 5.

6.3.1.1 Function height range.

The simplest measure of persistence is to compute the difference in function height between the top and bottom supernodes. This provides a measure for the scalar range at which an object exists as a unique entity. Prominent features have a large scalar range, and noise is characterised by small arcs. Figure 6.18 shows how this measure of persistence varies with cooling; the oscillating behaviour present in the nested surface view of the object (Fig. 6.8) is not present in the persistence measure. Upon hitting a peak at 28 cools it begins to drop showing gradual erosion of the object by the cooling algorithm as it progresses. At 35 iterations a distinct change in the trend of the graph apparently coincides with observed forking behaviour in the Reeb graph as was expected due to the splitting observed in the contour view.

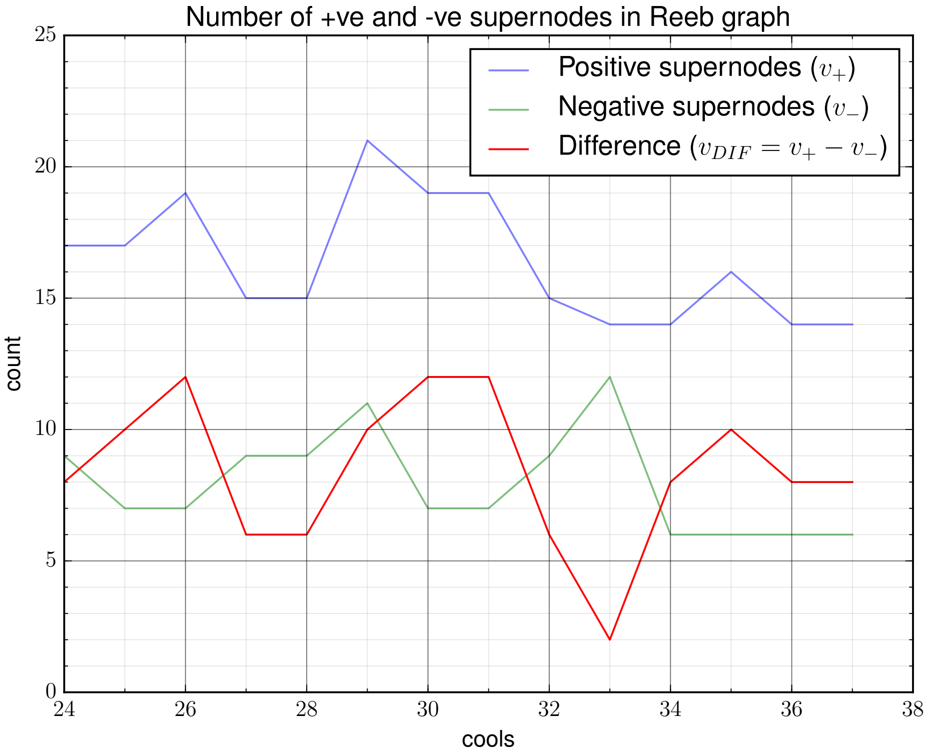


Figure 6.17: Number of positive and negative supernodes in the Reeb graph during cooling.

6.3.1.2 Node count.

The node count relates to the number of vertices in the scalar field traversed by the superarc between its two critical nodes. The superarc represents a region of connected space as it traverses vertices in the scalar field to provide a rough estimate of volume. Figure 6.19 presents the approximate volume of the anti-instanton whilst subjected to cooling, showing an encouraging correlation to behaviour predicted by the physics. This states that the volume of an (anti-)instanton should remain roughly constant under cooling regardless of the deformation of the object. The behaviour witnessed also predicts the splitting behaviour by capturing a sharp decrease in volume at 35 cools.

6.3.1.3 Sample sum.

The final measurement of persistence is the sample sum for the superarc. As with the node count, this measurement traverses the connected region of the scalar field represented by the arc. However, rather than counting the nodes a sum of the samples attached to each vertex is computed to give an approximation of the hyper-volume of the instanton. In Figure 6.20 we graph the absolute value of the sample sum which shows less of a correlation with function

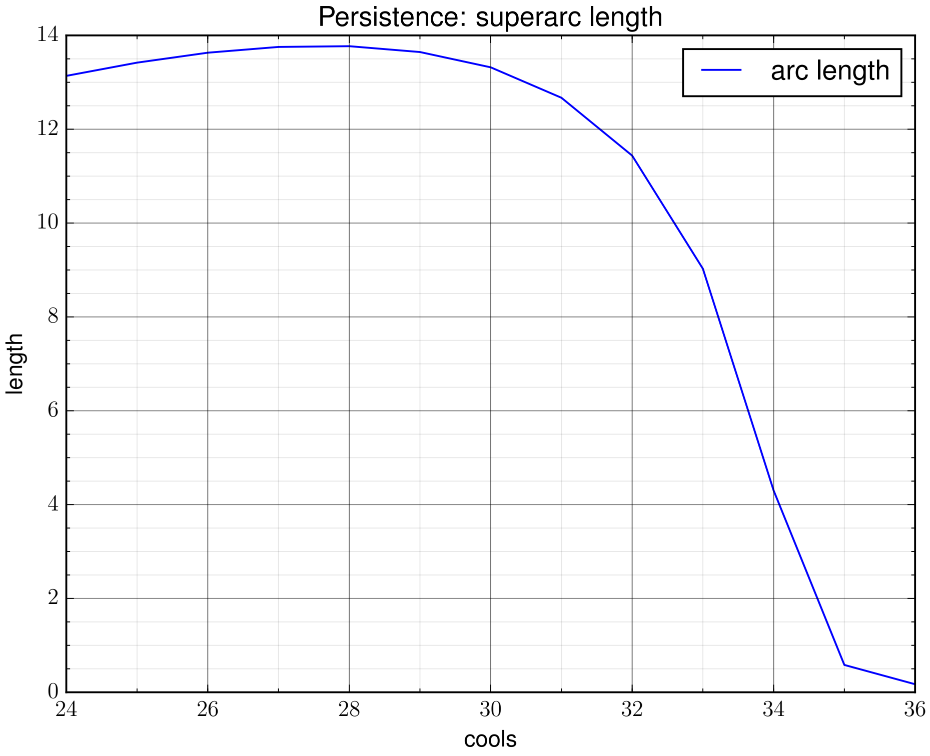


Figure 6.18: Measured length (scalar range) of the superarc representing an anti-instanton.

height range (Fig. 6.18) and node count (Fig. 6.19). It is believed this can be explained by the gradual reduction in energy caused by cooling as the overall system energy tends towards a vacuum state. Encouragingly the trend seems to project towards zero, appearing to agree with this prediction.

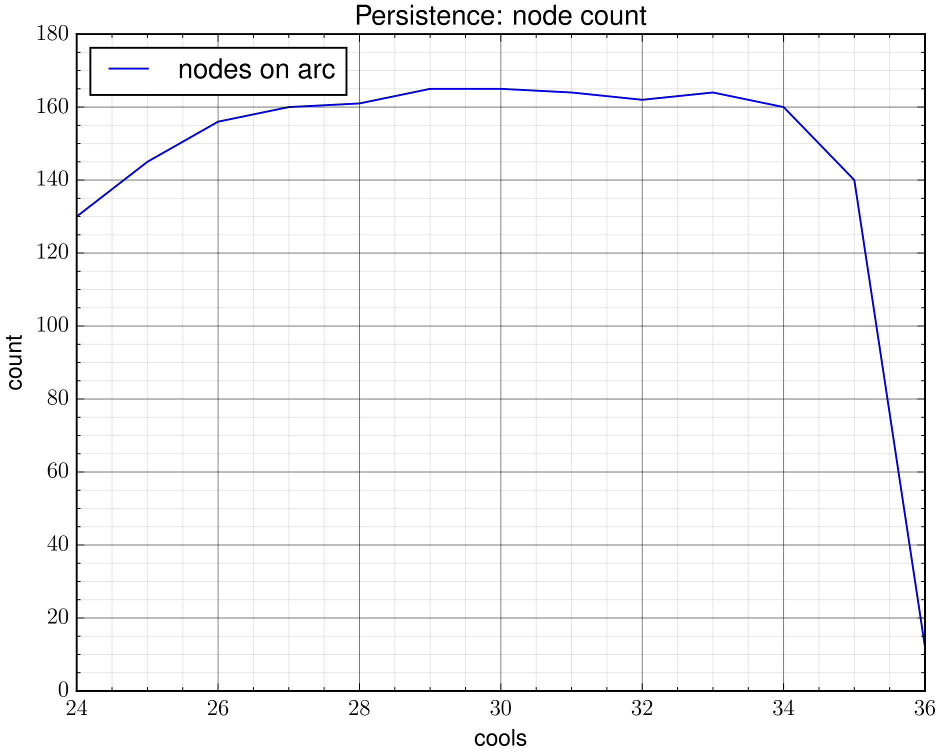


Figure 6.19: Number of heightfield nodes on the superarc representing an anti-instanton.

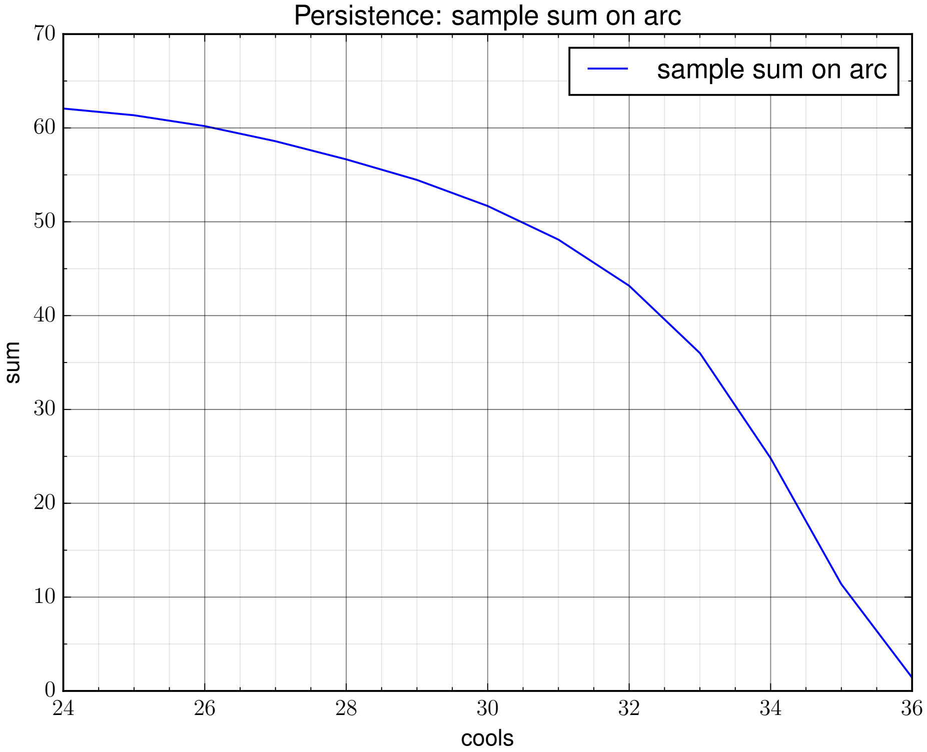


Figure 6.20: Sum of samples on the superarc representing an anti-instanton.

6.4 Summary

In this chapter we have shown how topological visualisation techniques can be applied to data from lattice QCD simulations. We used topological visualisation techniques to isolate an (anti-)instanton for the first time, enabling us to examine its structure in detail under cooling. During the analysis we examined the Reeb graph structure in conjunction with rendered contours to show how the techniques can be used to better understand lattice configurations. Besides being useful as a method for generating visualisations, we have also showed how persistence measures computed from the Reeb graph can summarise properties of individual objects. We feel persistence measurements provide a valuable new technique for physicists to analyse their data.

Contributions of this chapter can be summarised as:

- We used topology to isolate and analyse the structure of lattice observables.
- The Reeb graph was used to examine the evolving nature of a lattice field as cooling is applied.
- An optimised technique was shown for computing properties of lattice object by evaluating the topological data structures directly.

A limitation of the current approach is the need to slice the data to a 3D input. In Chapter 7 we show how a recent development in multivariate topology, the Reeb space, can help to overcome the higher-dimensional aspect of the data. This technique has the combined advantages of being able to handle multiple input fields defined on a common domain and natively working on inputs defined in higher dimensions.

Chapter 7

Using the JCN to track lattice objects in four dimensions

Contents

This chapter builds upon the topological visualisation techniques explored in Chapter 6 by generalising the concepts to multiple variables existing at common sampling points. The case study presented in this chapter applies the technique to a temporal lattice field in order to understand the relation between time-steps. As previously discussed in Chapter 3 many fields in lattice QCD are 4D in nature; whilst not seen as a direct replacement for handling the data in its native 4D, we feel these techniques offer an interesting method for capturing events at particular moments in space-time.

We begin this chapter by introducing the implementation of the JCN algorithm used in this work in Section 7.1. In Section 7.2 we carry out a visual analysis of a lattice using the JCN to identify and track an (anti-)instanton across the time axis. We use multivariate persistence measures and simplification to perform a quantitative analysis of the data in Section 7.3. Finally, in Section 7.4 we present an experimental use of the JCN — using it to compute a signature of an entire 4D lattice field.

A summary of terminology used throughout this chapter is presented below. In addition many of the concepts introduced in earlier chapters, such as the notion of critical vertices, also generalise to multivariate data.

Reeb space This represents a generalisation of the Reeb graph to handle multivariate data. In its discrete form the Reeb space can be considered as a decomposition of an input domain into a number of m-dimensional regions.

Interval A discrete technique for capturing isovalues within a range, provided as an external parameter to computations. When selecting the interval it is important to avoid aliasing of the data; this occurs when two separate features merge due to falling into the same discrete sampling range. The interval is generally referred to as the slab size when used in the context of the JCN,

Slab A slab represents a region of the m-dimensional input domain where the isovalue of n fields fall into a isovalue interval. For a single variable a slab roughly equates to the enclosed space between two isovalues.

Joint contour net (JCN) This is a graph structure used to capture properties of the Reeb space using an interval based approach. Occasionally the JCN may be presented in domain layout — meaning each vertex is roughly mapped to the barycentric coordinates of its region. However, due to the high possibility of occlusion it is often presented using a spring layout. The JCN should be interpreted as:

- Vertices: represent slabs where the isovalue of n functions fall into a defined interval. The glyphs used to represent each vertex are split into n sectors to reflect the isovalue in each field.
- Edges: represents adjacency between slabs.
- Glyph scale: this is used to reflect the volume of a slab.

Multivariate persistence measures This is an extension of the persistence measures used in topological analysis of a single field. Typically they represent some geometric property of a slab, such as its volume or surface area.

Reeb skeleton The Reeb skeleton presents a simpler view of the Reeb space by reflecting only important topological changes in the multivariate topology. The Reeb skeleton visualisation should be interpreted as follows:

- Red vertices: capture major changes in the topological structure including the birth or death of a path-connected region of the Reeb space.
- Blue vertices: capture minor changes in the topology.
- Edges: represent adjacency between regions of the Reeb space.
- Glyph scale: this is used to relay information about multi-dimensional persistence measures.

7.1 Implementation

In this work we use the JCN implementation supplied as part of the Multifield Extension of Topological Analysis (META) project [163]. This provides a number of filters for multivariate data that can be applied to the visualisation pipeline as part of a VTK [160] workflow. Filters are included for creating an initial JCN decomposition of the input fields that are able to be presented in graph form. As part of the process of creating the JCN individual joint contour slabs, in the form of polygon meshes, are created as the union of multiple smaller fragments. Joint contour slab and fragment data is available for the purposes of simplification using additional filters. A more comprehensive review of the simplification process is given in Section 7.3.2.

Input fields are placed on to a common set of sampling points in three dimensions and each cubic cell is subdivided into tetrahedra. By default each cell is split into 6 tetrahedra using a Freudenthal subdivison — as used in the Reeb graph case study (Chap. 6). The user can also choose to use a 5 tetrahedra Minimal division if required; a comparison of the two methods is shown in Fig. 7.1. In order to handle periodicity an additional cell is constructed to link the minimum and maximum samples on each axis.

[redacted]

Minimal subdivision of a cubic cell into 5 tetrahedra.

Freudenthal subdivison of a cubic cell into 6 tetrahedra.

Graphic is taken from Carr et al. [161].

Figure 7.1: Comparison of available cell subdivision techniques.

7.1.1 Interactive user interface

This case study was performed using a modified version of the interactive tool used by Geng et al. for the analysis of hurricane data [90]. Modifications of the software were largely made to facilitate collection of quantitative measurements for analysis. In addition, the transfer functions used to colour the glyphs have been modified to enhance feedback for lattice QCD fields that are often centred on zero [164, 165, 166]. Figures 7.2 and 7.3 give a visual overview of the user interface.

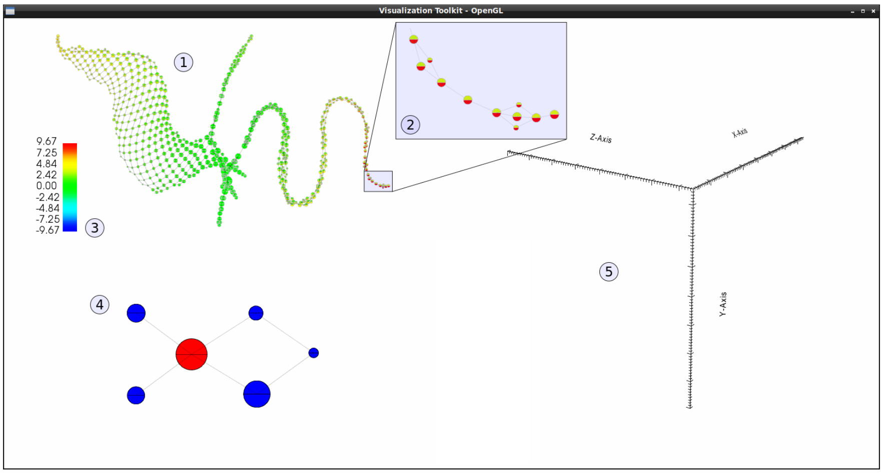


Figure 7.2: The user interface used to explore temporal lattice QCD data as captured by the JCN. The JCN captures the topology as a graph structure (1) — displayed here in a spring layout. Each JCN glyph (2) represents a slab (or quantised contour) using scale to provide feedback on the overall size of the slab. Glyphs are coloured by isovalue (3), where the top half of each glyph represents the first input field and the bottom the second input. Also visible is the Reeb skeleton (4) which captures the JCN in a simplified form; red glyphs represent major changes in topological connectivity and blue glyphs represent relatively stable regions of topological structure. Slabs are only rendered (5) to reflect any selections made by the user; in this view nothing has been selected.

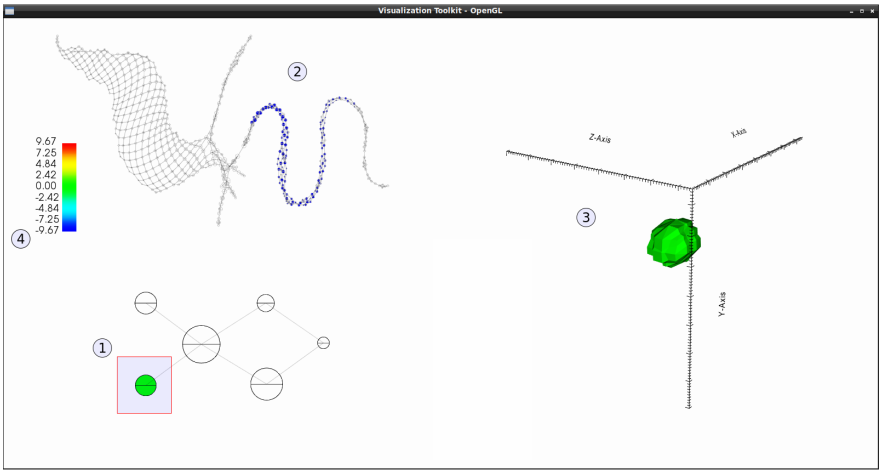


Figure 7.3: The user interface allows the user to select nodes in the Reeb skeleton (1) or JCN (2) using rubber band selection. Green glyphs in the Reeb skeleton correspond directly to the blue glyphs in the JCN view. Selected vertices are displayed as quantised contours (3) using a colour transfer function (4) determined by the bivariate input fields.

7.2 Visual analysis of lattice objects in four dimensions.

The purpose of this case study is to use the JCN to track objects in the topological charge density field across multiple time steps. (Anti-)Instantons are localised to specific locations in 4D Euclidean space-time, but using this approach it is possible to consider these observables as the temporal index is varied. To demonstrate this, we present the technique as a method for evaluating topological charge density as the time parameter is varied, with a focus on a single instanton identified by the cooling code.

Due to the Euclidean nature of lattice QCD, the techniques used in this chapter will also generalise to scanning volumes with a temporal component (xyt, xzt, yzt) along a spatial axis. Whilst not seen as a direct replacement for viewing these fields in their native 4D embeddings, this technique presents an interesting approach to pin-point critical events in the topology of a single evolving field.

To summarise, we hope to answer the following questions in this case study:

- Can we use the JCN to track an instanton between multiple neighbouring time slices?
- Can extra properties about instantons be determined through multivariate persistence measures?
- Can the Reeb skeleton be used to simplify the field and summarise properties of observables?
- How can multivariate topological visualisation techniques be applied to other lattice QCD observables?

7.2.1 Selecting a candidate configuration for visual analysis

Data used in this case study originates from a lattice with 16 spatial sites and 8 time sites, otherwise denoted as a $16^3 \times 8$ or "hot" lattice on account of the short temporal extent. This particular lattice is chosen as it features relatively few time slices in comparison to other ensembles where the temporal extent can exceed 32 steps. The JCN is demonstrated capturing a single topological object across the temporal axis including the periodic boundary between $t = 8$ and $t = 1$. We also use the short temporal axis to push the limits of the JCN by queuing multiple time slices on the multi-field to see if we are able to observe a signature for the entire 4D hyper-volume (Sec. 7.4).

The configuration used in this study *conp0050* originates from an ensemble with a chemical potential $\mu = 0.7$. For a hot lattice, such as the one used here, this represents a region of potential deconfinement. The lattice is pre-cooled for 30 iterations to a stable state, validated by inspection of the total topological charge density and peak Wilson action graphs. At this point in the cooling process the total topological charge Q_{TOT} remains flat for many cooling iterations (Fig. 7.4) and the peak Wilson action S_{PEAK} follows a smooth trajectory (Fig. 7.5).

The stability of the lattice is also reflected by monitoring the location of minima and maxima in the topological charge density Q and Wilson action S fields (Table 7.1). In this interval of the cooling process the predicted locations of field minima and maxima are extremely stable,

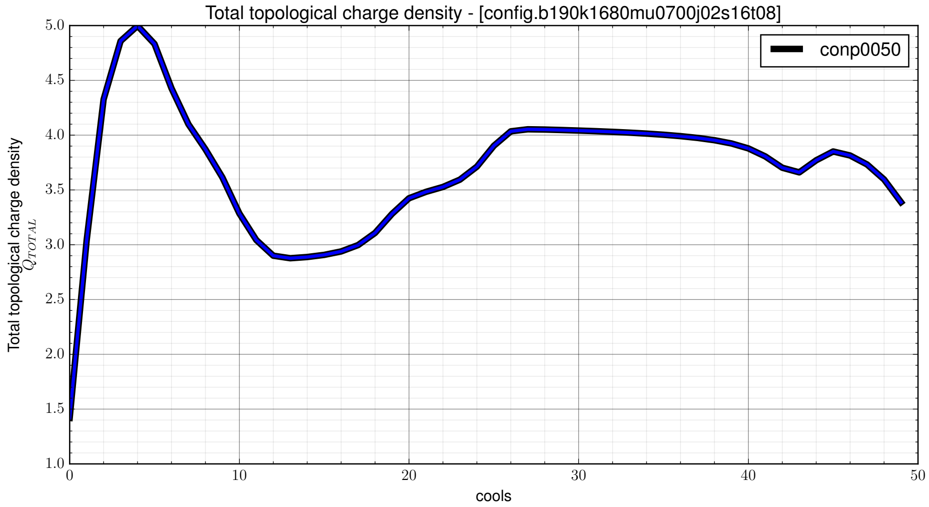


Figure 7.4: Total topological charge Q_{TOT}

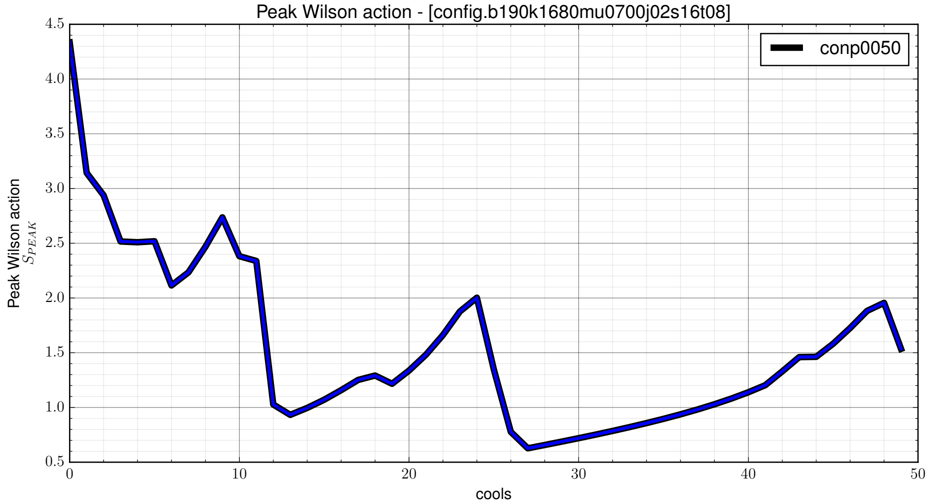


Figure 7.5: Peak Wilson action S_{PEAK}

indicating that the same object is persisting throughout. The location of maxima in the Wilson action S_{MAX} and topological charge density Q_{MAX} coincide predicting the presence of an (anti-)instanton. The location of the instanton in relation to the periodic boundaries means that the object can be simply visualised — whilst it is possible to track an object spanning one or more periodic boundaries using the JCN, displaying them visually as slab geometry becomes problematic.

Cools	S_{MAX}	Q_{MAX}	Q_{MIN}
27	(11, 4, 13, 5)	(11, 4, 13, 5)	(6, 2, 7, 2)
28	**(9, 8, 10, 5)**	**(9, 8, 10, 5)**	(6, 2, 7, 2)
29	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
30	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
31	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
32	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
33	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
34	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
35	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)
36	(9, 8, 10, 5)	(9, 8, 10, 5)	(6, 2, 7, 2)

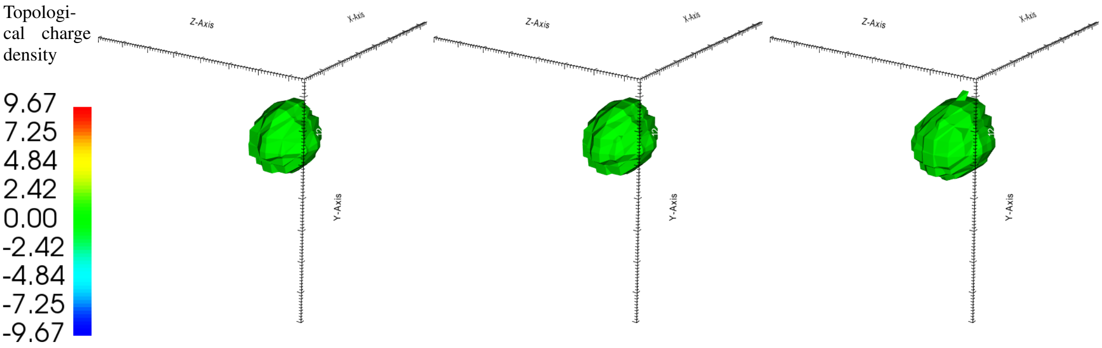
Table 7.1: The location on the lattice of global minima and maxima in the *conp0050* configuration. Changes between cooling iterations are highlighted in **bold**.

7.2.2 Input fields

In order to use the 4D topological charge density with the JCN algorithm [163] the dimensionality is reduced to 3D by slicing along the the t axis. Each JCN in this case study is constructed from two neighbouring time-slices (t_n, t_{n+1}), additionally as the time axis is also periodic in lattice QCD the JCN ($t_{max(n)}$, $t_{min(n)}$) is a valid input configuration. Hence, for the 16×8 ensemble, as used in this study, we preserve temporal periodicity by computing a JCN for (t_8, t_1).

7.2.2.1 Slab size parameter

Throughout this study we keep the JCN slab size parameter fixed dividing the topological charge density, with approximate range $[-45.0, 45.0]$, into $2^9 = 512$ intervals to give a slab size of 0.17578125. Figure 7.6 visualises the instanton as 3 consecutive temporal objects, generated as interval contours at the specified slab size. The green shades of each object, used to show neutral isovalues, show the outer layers of the instanton hide a dense core at the field maxima. The same object is present in each time step; however, its structure constantly evolves with time. Multivariate persistence measures can be used to capture this information at the specified slab size as demonstrated in Sec. 7.3.



$t = 4$ $t = 5$ (maxima) $t = 6$

Figure 7.6: View of the main instanton observable as univariate slabs.

7.2.3 Global location of (anti-)instanton observables

Figure 7.9 presents the eight separate JCNs created by evaluating each pair (t_n, t_{n+1}) of temporal fields. A fixed colour transfer function is used, based upon on the peak magnitude in four-dimensions, in order to present a true representation of potential (anti-)instantons in the data. This technique was chosen as lattice QCD observables must be considered as global extrema in 4D, rather than as localised between two time steps.

Initially it is possible to locate an anti-instanton in the JCN for $t = (1,2)$ by identifying the object as a global minima using the coloured glyphs (Fig. 7.7). The object can also be detected in $t = (2,3)$ using the same approach; however, the bottom half of the glyph turns green to indicate a neutral isovalue. The structure of the anti-instanton continues to persist in the data beyond this time slice despite the change in isovalue. Recovery of the anti-instanton slab structure in later time slices requires some exploration of JCN branches using knowledge of the objects location. The anti-instanton becomes a prominent feature in the bivariate topology at $t = (8,1)$ where the coloured glyphs indicate the presence of a global minima.

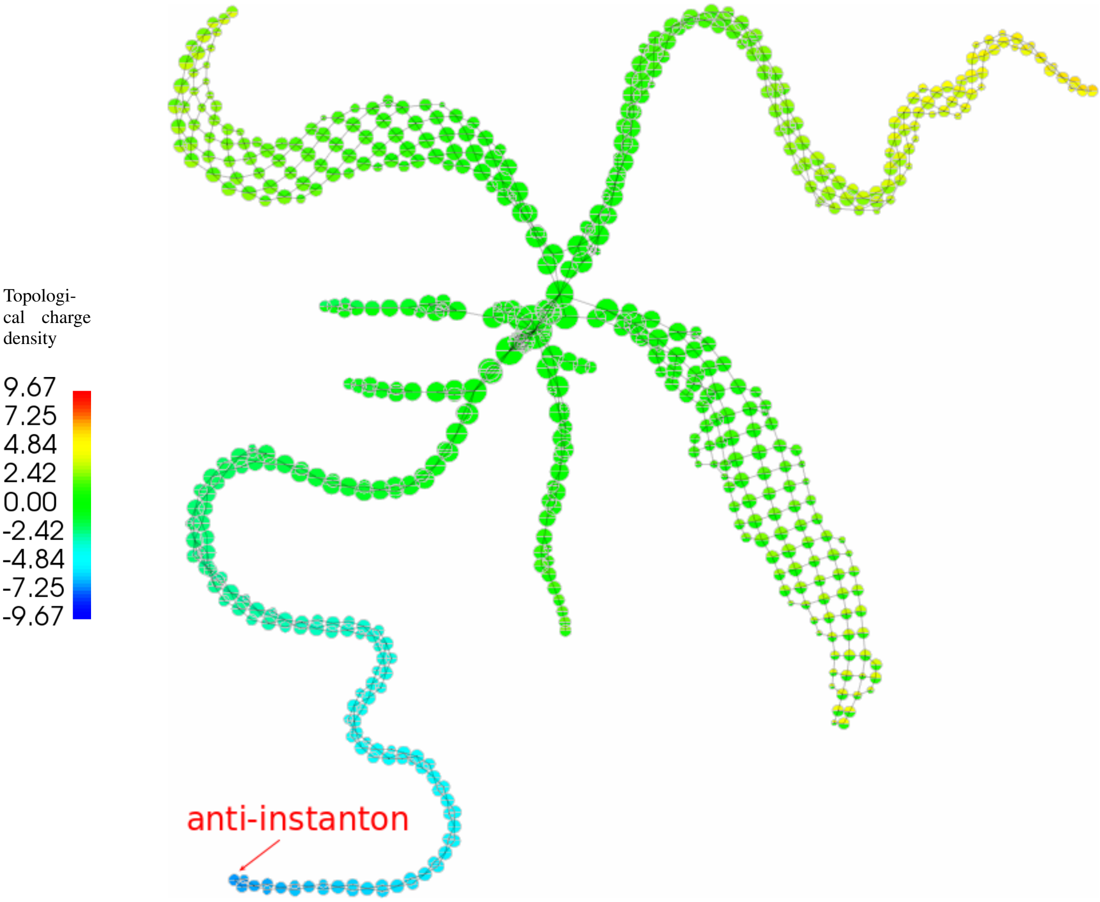


Figure 7.7: An anti-instanton can be located in the $t = (1,2)$ JCN as the branch with blue glyphs. This object continues to exist for several time steps.

Also present in the JCN overview (Fig. 7.9) is a global maxima which first becomes prominent in the multivariate glyphs at $t = (4,5)$. This coincides with the output of the FORTRAN code which predicts the presence of an instanton at $t = 5$. The instanton quickly begins to fade into the surrounding lattice structure at $t = (5,6)$ where the multivariate glyphs indicate a return to more neutral isovalues (Fig. 7.8).

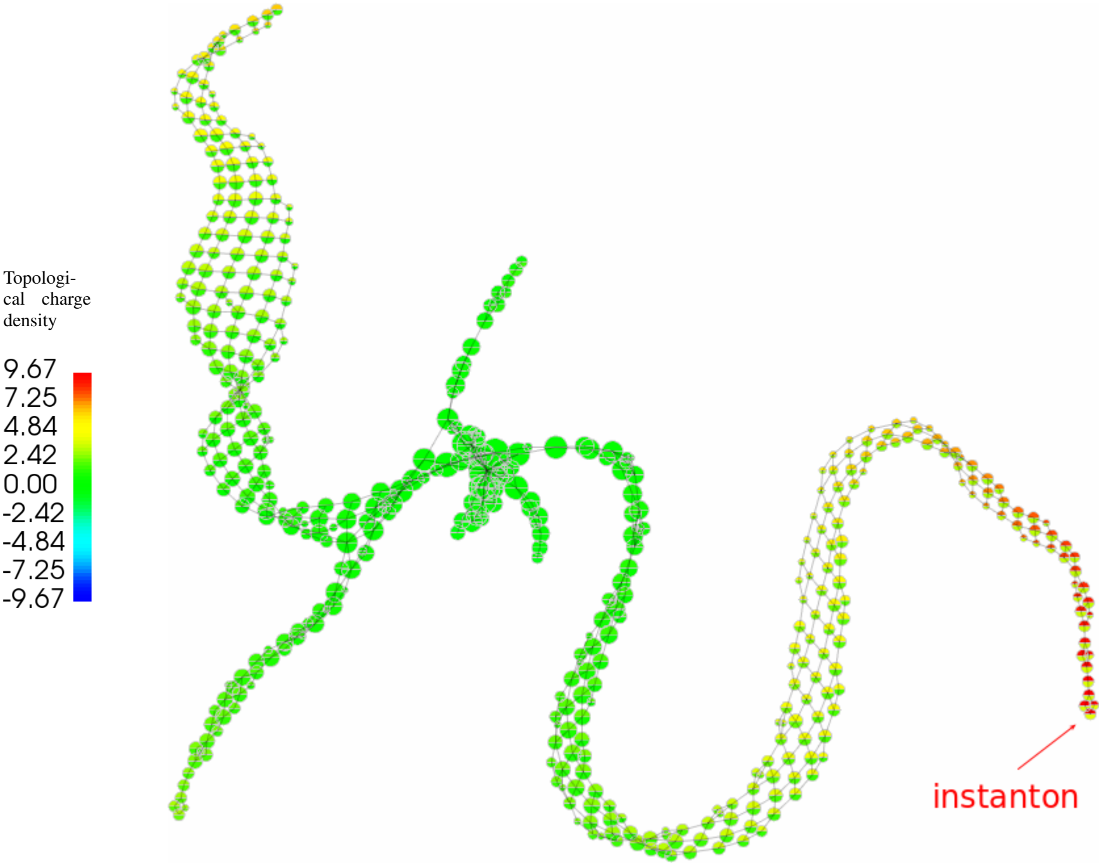
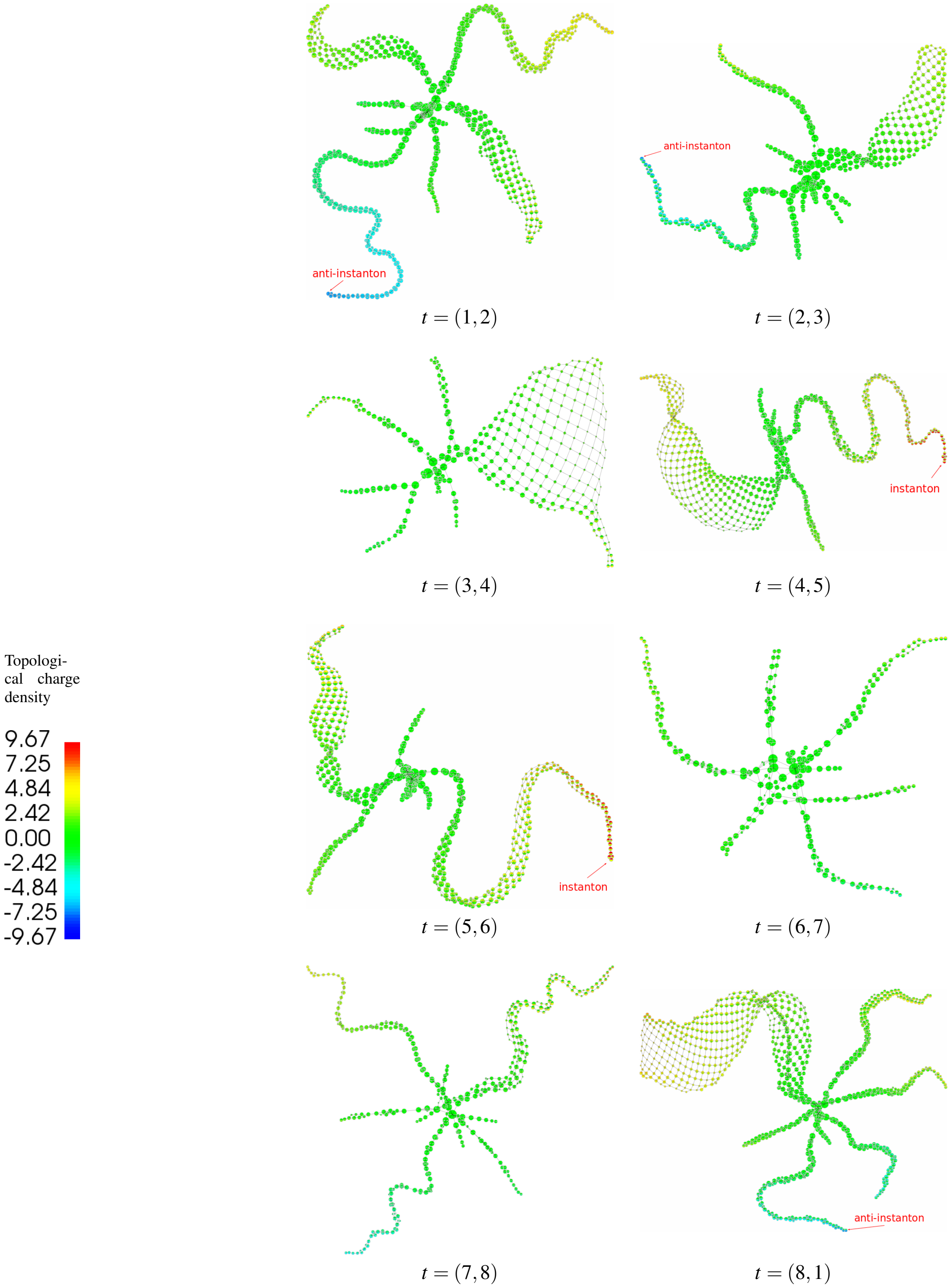


Figure 7.8: An instanton can be located in the $t = (5,6)$ JCN as the branch with red & yellow glyphs. The bottom half of the glyphs indicate that the topological charge density quickly drops off from a global peak at $t = 5$.

Several other localised features exist in the data at various points in the temporal JCNs. A second potential anti-instanton appears in the JCN at $t = (7,8)$, continuing to exist in parallel to the main anti-instanton observable at $t = (8,1)$. The JCN offers an interesting approach to examining the interactions between multiple temporally localised lattice objects. Lattice observables in neighbouring time slices could potentially have an influence over the geometric structure of other objects.



$t = (1,2)$ $t = (2,3)$

$t = (3,4)$ $t = (4,5)$

$t = (5,6)$ $t = (6,7)$

$t = (7,8)$ $t = (8,1)$

Figure 7.9: Using the JCN to locate the most prominent anti-instantons (by magnitude) across the time axis. Less prominent anti-instantons also appear in the data but are not labelled.

7.2.4 Visually tracking an instanton across the temporal axis

In the following section the bivariate topology, as captured by the JCN and unsimplified Reeb skeleton, is analysed visually. For each combination of inputs the instanton is presented as detected by user guided selection of vertices in the topological graph structures. We begin by looking at the identified global maxima in four dimensions, predicted by the cooling code as being at $(9,8,10,5)$ and continue across the periodic boundary back to the origin.

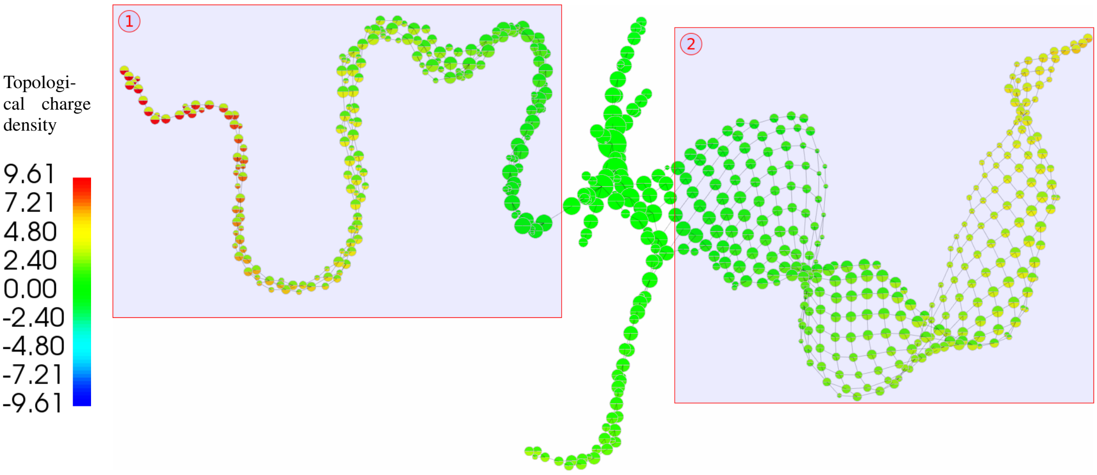
These views of the JCN use a localised colour transfer function based upon the two input temporal fields. This allows interesting features to standout amongst the surrounding topology. When comparing multiple JCNs a global colour scale is applied to avoid misleading interpretations of the data.

7.2.4.1 Time steps $t = 4$ and $t = 5$

Initially the instanton is located by examining the JCN vertices using the coloured glyphs relating to isovalue. The highlighted region (1) in Figure 7.10 shows how the instanton relates to a maxima at $t = 5$ slices and has slightly elevated levels at $t = 4$ in the topological charge data field. The JCN, when drawn in domain layout, shows the approximate location of barycentre of the slabs making up the instanton. Displaying the slab geometry makes it possible to validate that the location of the object agrees with the predicted location $(9,8,10,5)$ from the cooling code. Figure 7.11 shows how the instanton structure (1) is the most prominent region of the bivariate topology captured in the Reeb skeleton structure.

Also present in the Reeb skeleton (Fig. 7.11) are smaller branches relating to other branches in the JCN (Fig. 7.10). Of the secondary features the most interesting is the branch labelled (2) relating to a large sheet-like structure in the JCN. Examination of object (2) by isovalue shows that at time-steps $t = 4$ and $t = 5$ the object features elevated levels of topological charge density. The sheet-like structure of this object appears linked to complex layering of isovalues in the two fields coupled with a large spatial separation of the joint contour slabs. We omit a graphical presentation of the object here as the object spans a periodic boundary making it difficult to display using the slab geometry.

Selecting Reeb skeleton vertices (Fig. 7.11) highlights common vertices in the JCN (Fig. 7.10). Vertices in the Reeb skeleton share a one-to-many relationship with the JCN as the Reeb skeleton collapses regions of path connected slabs into a single vertex. Close observation of the two graphs shows that the Reeb skeleton (Fig. 7.11) captures slightly less of the instanton structure than user guided selection via the JCN (Fig. 7.10). Figure 7.12 shows how this affects the geometric representation of the instanton by displaying the joint contour slabs represented by the JCN vertices. In both fields the instanton is surrounded by slabs with low isovalue, represented by the green glyphs — relating to the region of percolation on the lattice where topological charge density centres on zero. The Reeb skeleton facilitates the removal of the outer layers of the instanton by considering it as topological noise due to low persistence. Instead the Reeb skeleton merges it into the percolation structure, represented by the highly connected vertex (3) in the Reeb skeleton (Fig. 7.11).



Glyphs; top: $t = 4$; bottom: $t = 5$.

Figure 7.10: JCN for $t = (4,5)$ coloured by isovalue. The instanton is captured as branch (1) of the JCN; a second object (2) represents an object with elevated topological charge density which is dispersed over a large area.

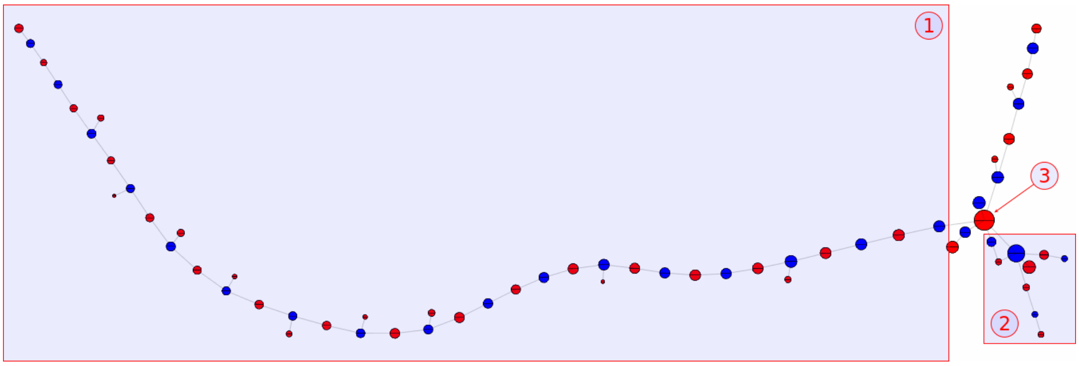
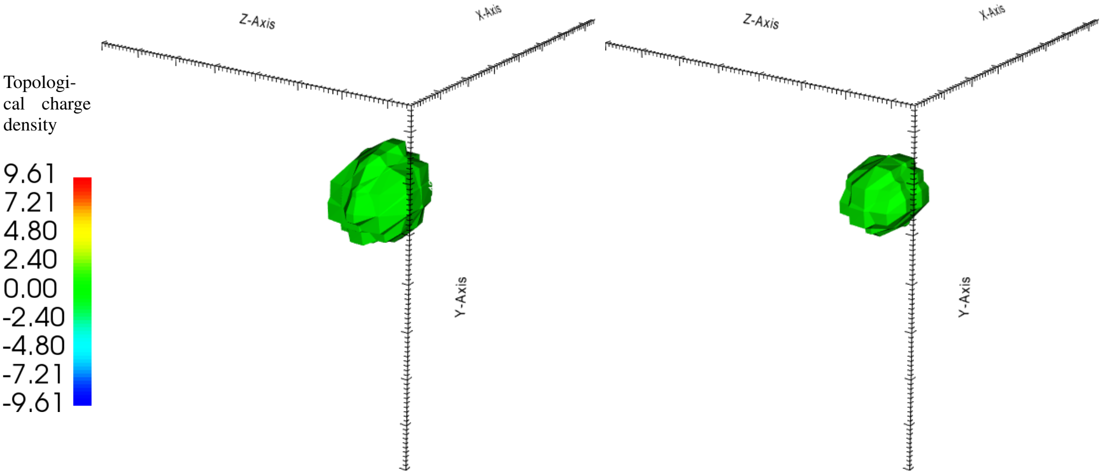


Figure 7.11: Reeb skeleton for $t = (4,5)$. The two observables (1) & (2) in the JCN also appear in the Reeb skeleton. The highly connected vertex (3) represents a large region with a neutral isovalue.



User selection via JCN.

Identified by the Reeb skeleton.

Figure 7.12: Joint contour slabs relating to the instanton object for $t = (4,5)$. The slab structure, as selected by the user (left), captures more of the instanton structure than the Reeb skeleton (right).

7.2.4.2 Time steps $t = 5$ and $t = 6$

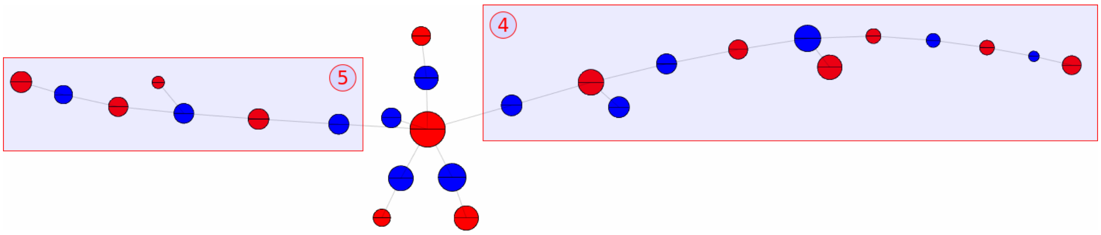


Figure 7.13: Reeb skeleton for $t = (5,6)$. The instanton (4) appears as the most prominent feature; however, a second feature (5) also stands out — this is further examined using the JCN.

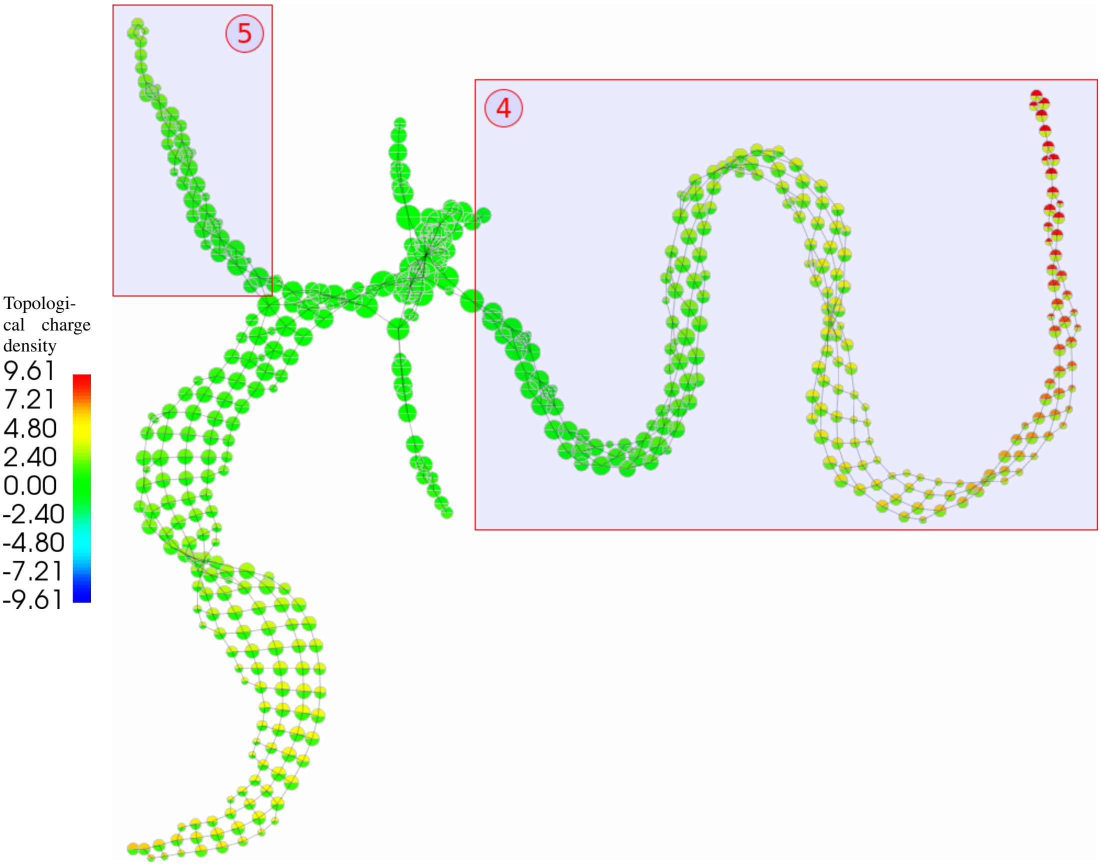
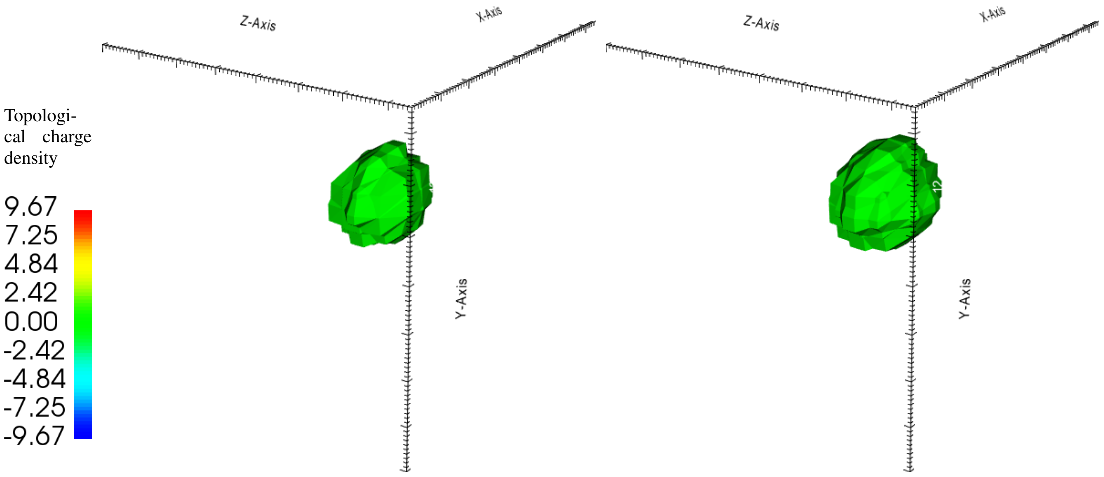


Figure 7.14: JCN for $t = (5,6)$. The instanton (4) can be identified by the red / light-green glyphs. The secondary object (5), detected by the Reeb skeleton, has a dense shell-like structure but is not a potential (anti-)instanton due to the neutral isovalues in the two time steps.

Analysis of the JCN generated comparing time-slices $t = 5$ and $t = 6$ is initially lead by examining the Reeb skeleton (Fig. 7.13). As with the previous JCN, generated for $t = (4,5)$, the Reeb skeleton highlights two significant features for $t = (5,6)$. The most prominent feature is the instanton (4), but a secondary feature (5) also stands out in the Reeb skeleton and JCN views (Fig. 7.14). Visual examination of the secondary object as slabs and using the JCN shows a relatively small feature made up of a number of densely packed layers. Isovalues of the object (5) in both temporal fields reveal that the topological charge density at the inner core is neutral, indicated by the green coloured glyphs. This allows it to be concluded that although a significant feature in the lattice topology this object is not a potential (anti-)instanton.



Instanton structure from $t = (4,5)$. Instanton structure from $t = (5,6)$.

Figure 7.15: Joint contour slabs relating to the instanton object for $t = (4,5)$ and $t = (5,6)$. The structure of the instanton appears to vary little over the time-steps.

The JCN and Reeb skeleton capture the main instanton observable (4) as a large branch in the topology. Comparison of vertices in the Reeb skeleton (Fig. 7.13) and JCN (Fig. 7.14) shows the two graph structures identify more common vertices than in the previous time slices. This suggests that the Reeb skeleton considers less of the outer shell of the instanton as topological noise existing in the region of percolation. Visually examining the structure of instanton through the slab geometry shows a great deal of similarity with the slab structure in the earlier time-steps (Fig. 7.15) — showing that the structure of the instanton changes very little across the three steps $t = (4,5,6)$. This is much as expected as a large contribution to the topology in the two bivariate JCNs is from the common time-slice $t = 5$ that identifies the instanton as the global maxima.

7.2.4.3 Time steps $t = 6$ and $t = 7$

The next JCN (Fig. 7.16) visualises the topological charge density at time-steps $t = (6,7)$. Initially the instanton can be located using the JCN vertices, coloured by isovalue, appearing as significant feature (6) of the topology. At $t = 6$ the instanton appears as a maxima in the topological charge density field; however, at $t = 7$ the isovalue of the instanton has drastically reduced to a neutral, as indicated by the green vertex glyphs.

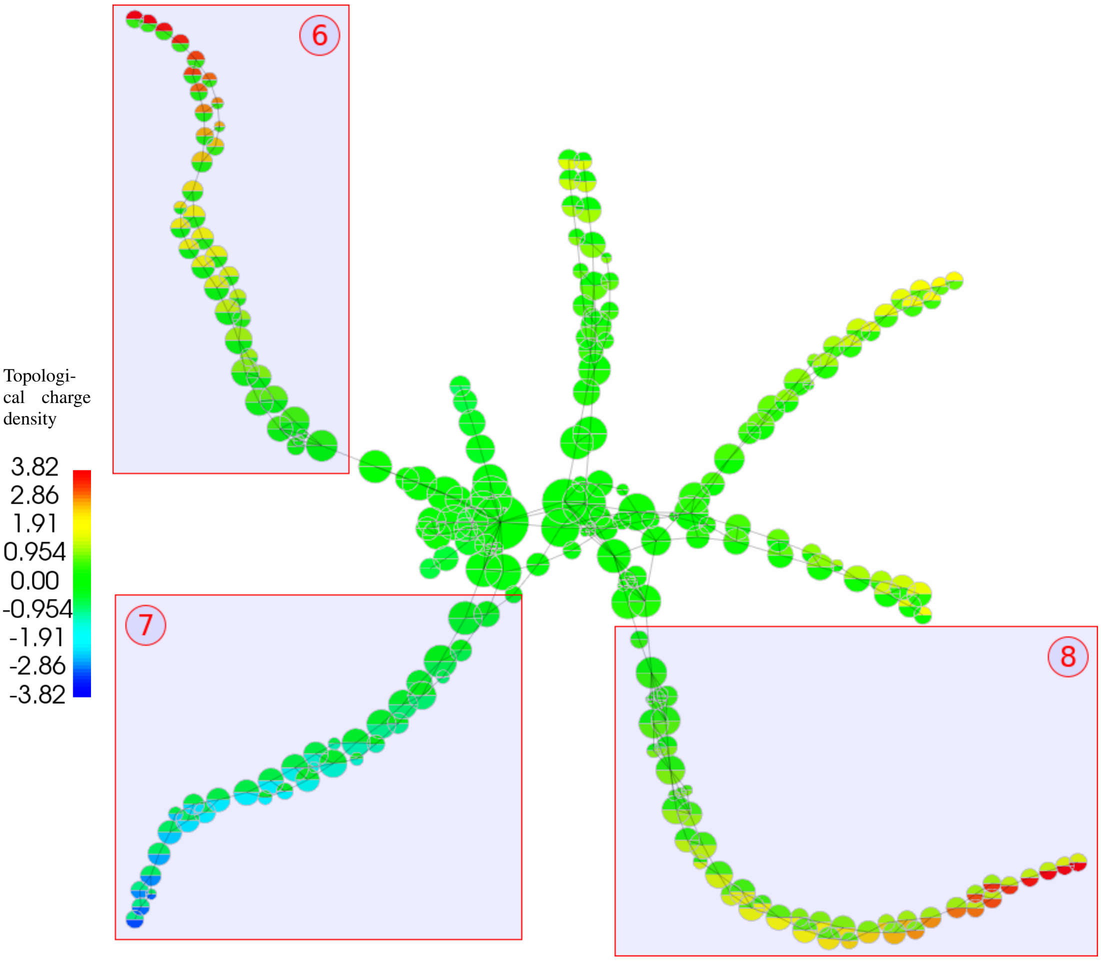


Figure 7.16: JCN structure at $t = (6,7)$. The instanton appears as object (6); also present is a potential anti-instanton (7) and another potential instanton (8).

User guided selection of the vertices on the branch relating to the instanton in the JCN allow it to be located in the Reeb skeleton. Repeating the process in reverse, with the object (6) selected from the Reeb skeleton shows that the Reeb skeleton determines significantly less JCN vertices (Fig. 7.17) to be part of the instanton structure. Basic simplification appears to determine that the outer layers of the instanton are topological noise as their function values deviate little from zero.

The Reeb skeleton potentially considers the main instanton observable (6) to be of less significance in comparison to other lattice objects based upon the number of vertices used to capture it in its unsimplified form. Further investigation of other branches of the JCN and Reeb skeleton highlights other interesting structures on the lattice. The Reeb skeleton identifies a potential anti-instanton, existing as a local minima (7) at $t = 7$ (Figs. 7.16). Also present is a secondary maxima (8) that replaces the instanton structure as the local maxima at time slice $t = 7$.

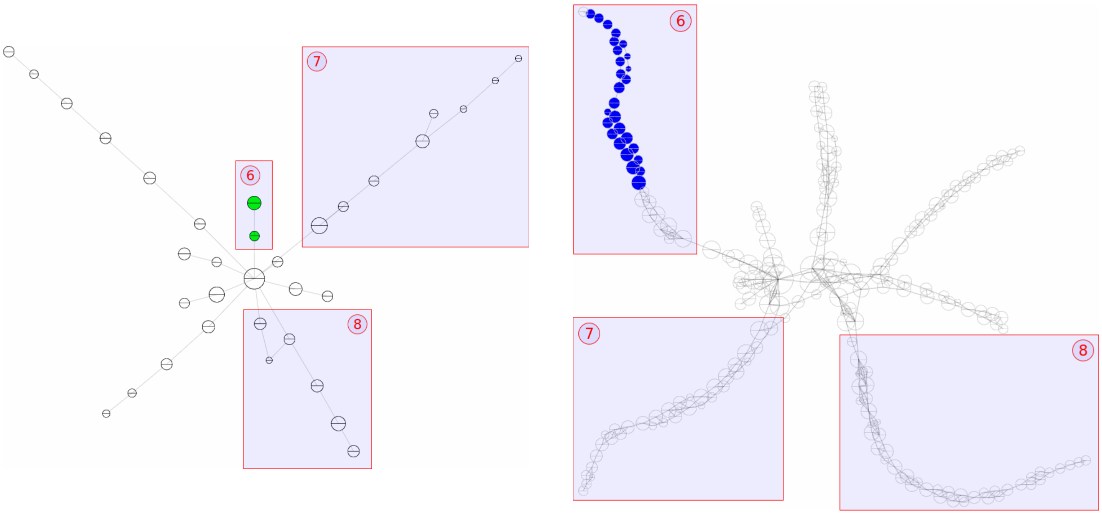


Figure 7.17: Time-steps $t = (6,7)$ captured using the Reeb skeleton (left) and JCN (right). Blue glyphs in the JCN and green glyphs in the Reeb skeleton are common to both structures and identify the instanton (6).

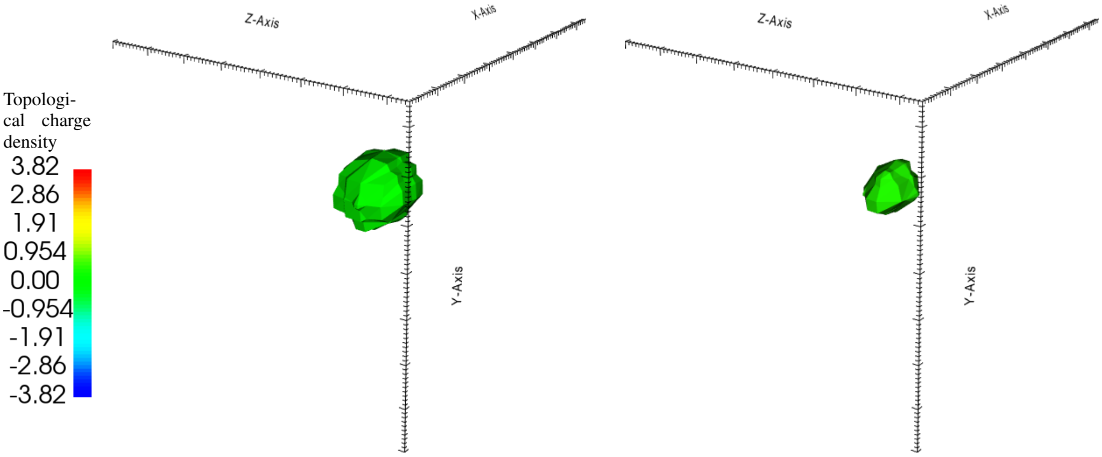


Figure 7.18: Left: instanton structure recovered by user selection in the JCN. Right: instanton structure recovered via the Reeb skeleton.

In Figure 7.18 we demonstrate how user guided selection of JCN vertices often captures more of the slab structure than that offered by the Reeb skeleton. Outer layers of the instanton, where the isovalues are closest to zero, are discarded as topological noise. More aggressive simplification using the Reeb skeleton gradually erodes the outer layers of the objects, with the slabs merging into the surrounding topological structure. The same effect is also frequently observed at the core of lattice observables where slabs are discarded due to their reduced isovalue range.

7.2.4.4 Time steps $t = 7$ and $t = 8$

The instanton structure continues to remain visible in the $t = (7,8)$ JCN. The object is selected by the user using prior knowledge of its location due to it becoming a less significant feature of the JCN. Examining the isovalues of the instanton at time-steps $t = (7,8)$ (Fig. 7.19) shows green glyphs in the JCN revealing that the isovalue has reduced to near zero. However, enough topological structure remains to separate the instanton from the surrounding region of percolation. Due to reduction of isovalue range representing the object the Reeb skeleton discards the instanton, instead determining it to be topological noise.

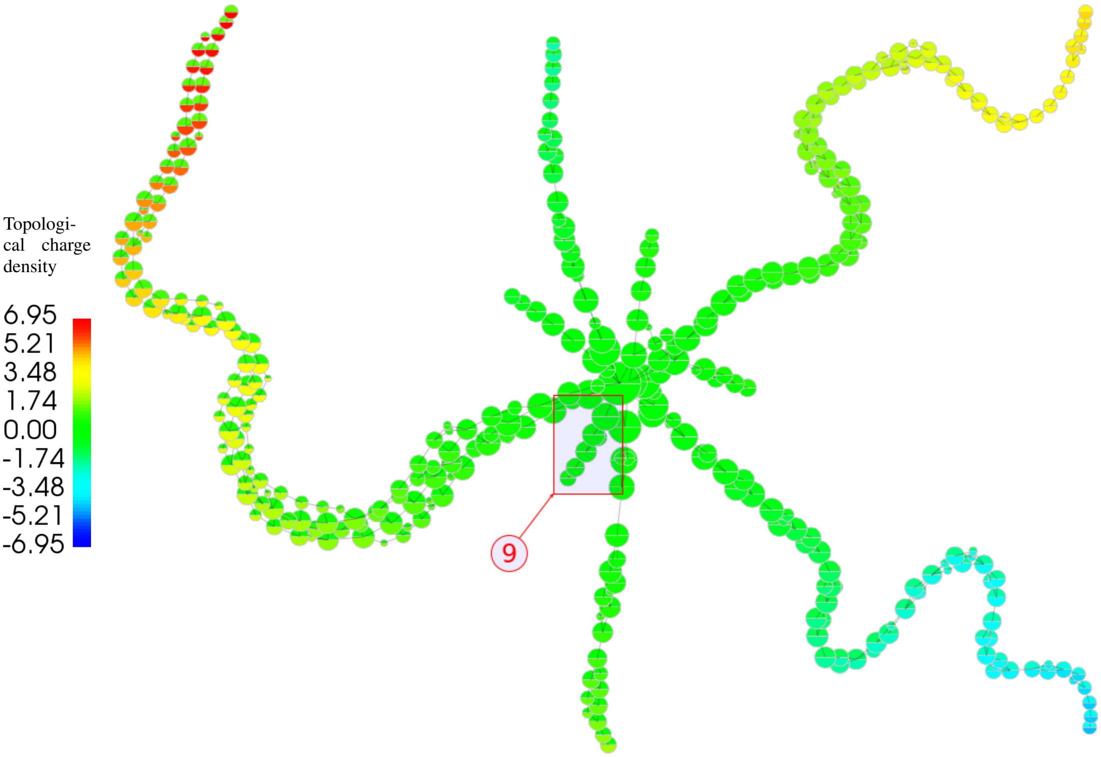


Figure 7.19: JCN structure at $t = (7,8)$. Isovalues of the instanton structure (9) are neutral but can still be found in the bivariate topology.

The user is able to recover the structure of the instanton as joint contour slabs by carefully selecting vertices in the JCN. In Figure 7.23 the slab geometry captured from the $t = (7,8)$ JCN is compared against those at $t = (4,5)$. The instanton is instantly recognisable in the two visualisations despite the relative changes in isovalues between the inputs, reflected in the colour change of the object. Visually, the two objects are very similar with only minor deviations in the outer structure. However, closer examination shows that the outer layers of the instanton continue to disperse as the time-steps move away from the maxima at $t = 5$.

It is possible to confirm that the structure of the instanton is present in the Reeb graph of time slice $t = 7$ by cross referencing the quantised Reeb graph (Fig. 7.20). Viewing the quantised contours (Fig. 7.22) selected from the Reeb graph (Fig. 7.20) confirms that the instanton is still present on the lattice at $t = 7$. The joint contour slabs and quantised contours also show that the instanton structure is relatively unchanged in comparison to time-step $t = 5$, with the relative isovalues being the most dramatic change.

The Reeb skeleton (Fig. 7.21) can also be applied to the quantised Reeb graph, yielded by the univariate JCN, to simplify the topology of the lattice. Using this approach it is possible to witness how topological changes, such as the birth, death, and merging of topological objects, map to the red singular vertices of the Reeb skeleton. Regions where the topology remains unchanged map to the blue vertex glyphs; these also reflect the scale of the simplified region by mapping the enclosed volume to scale.

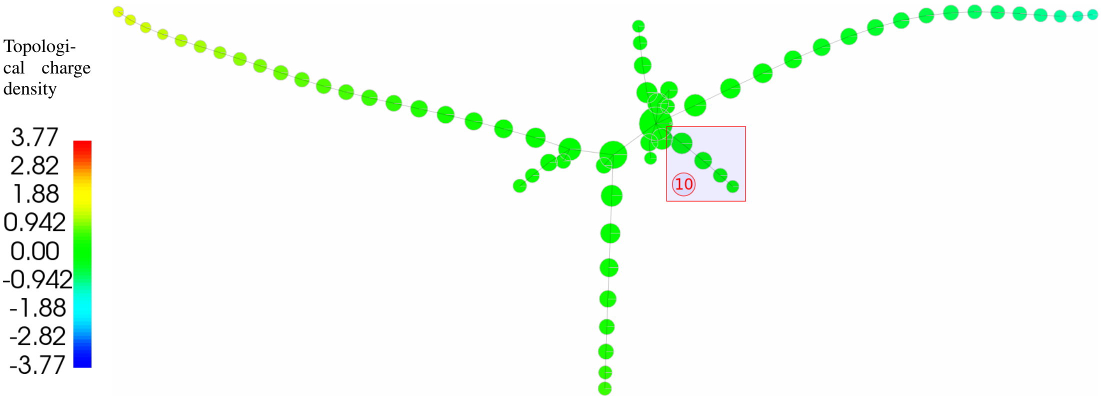


Figure 7.20: JCN for $t = 7$. The instanton appears as branch (10) in a quantised representation of the Reeb graph.

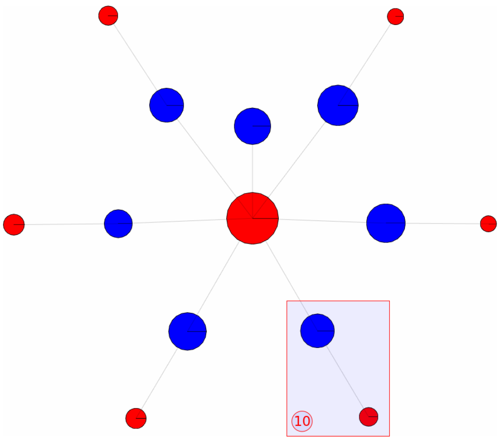


Figure 7.21: Reeb skeleton for $t = 7$ showing the instanton (10).

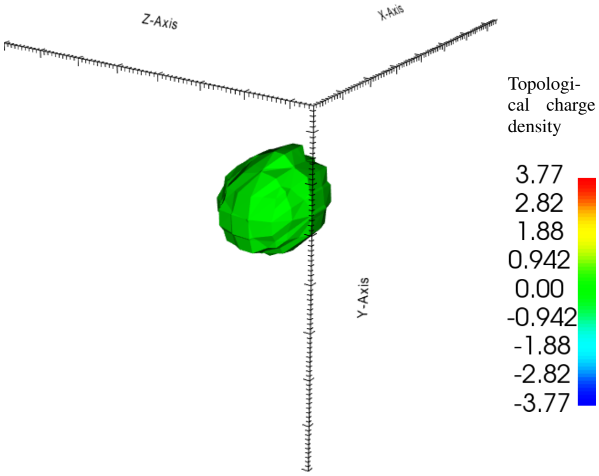
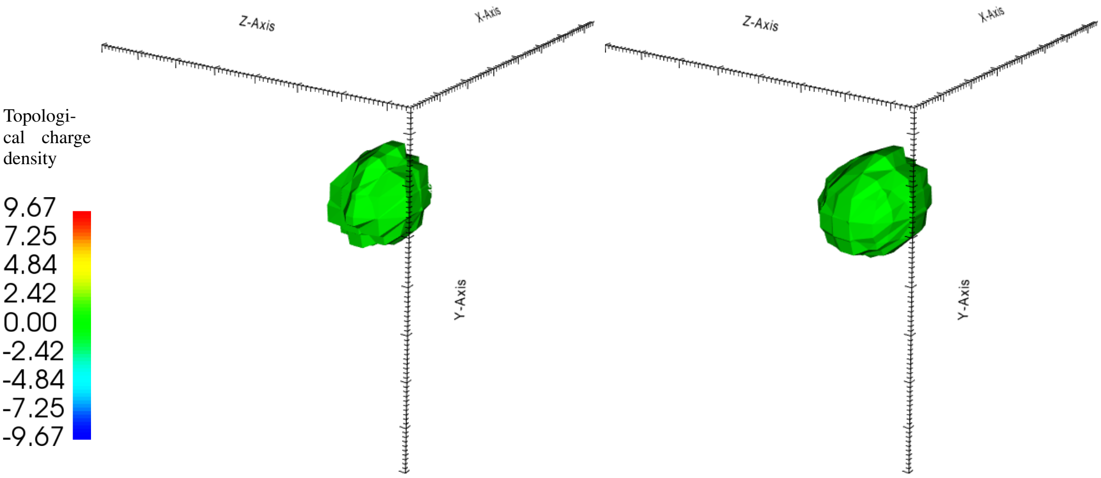


Figure 7.22: Instanton object for $t = 7$.



Instanton structure from $t = (4,5)$.

Instanton structure from $t = (7,8)$.

Figure 7.23: Joint contour slabs relating to the instanton object for $t = (4,5)$ and $t = (7,8)$. The structure of the instanton appears to be fairly stable over several time-slices.

7.2.4.5 Time steps $t = 8$ and $t = 1$

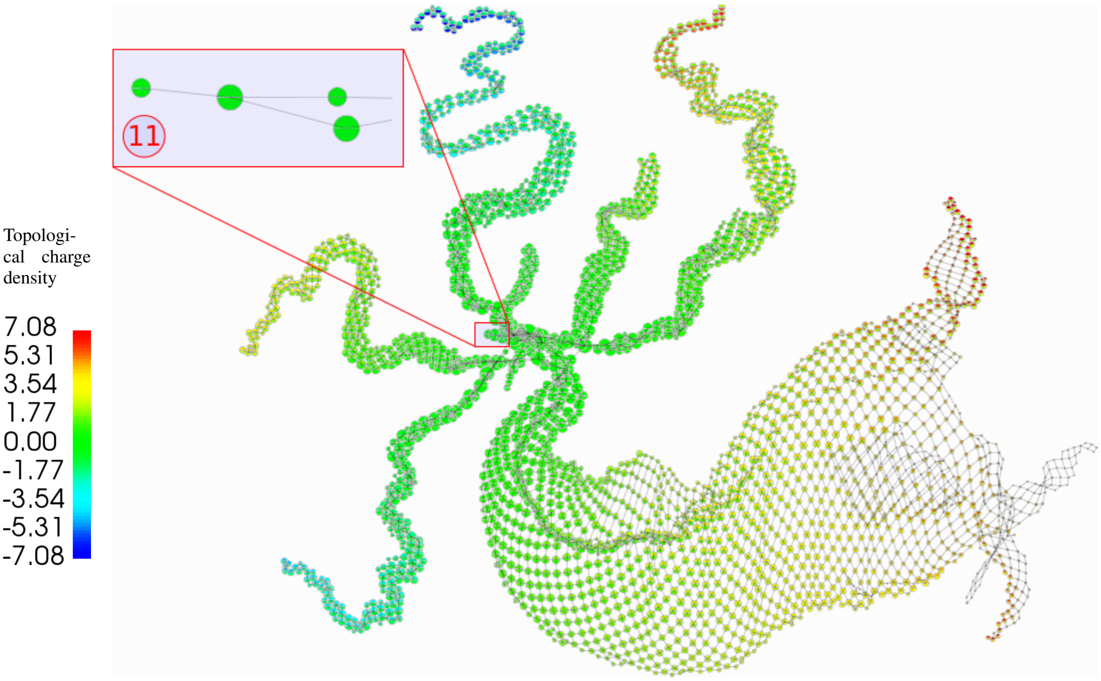


Figure 7.24: JCN for $t = (8,1)$. The instanton structure (11) is present as a minor feature in the topology with relatively neutral isovalues.

Initial user selection of the vertices in the JCN with inputs set to $t = 8$ and $t = 1$ failed to reveal any structure relating to the instanton observable. When the slab size is decreased the instanton can just about be isolated from the surrounding topology charge density (Fig. 7.24). A halving of the slab size to 0.087890625, giving $2^{10} = 1024$ intervals, is sufficient to allow the object to be located. Examination of the geometric structure of the instanton, through the joint contour slabs, reveals that the shape found in earlier time slices begin to merge into the surrounding lattice field. However, it is possible to confirm that the selected object (11) relates to the instanton by observing the JCN in domain layout (Figure 7.25).

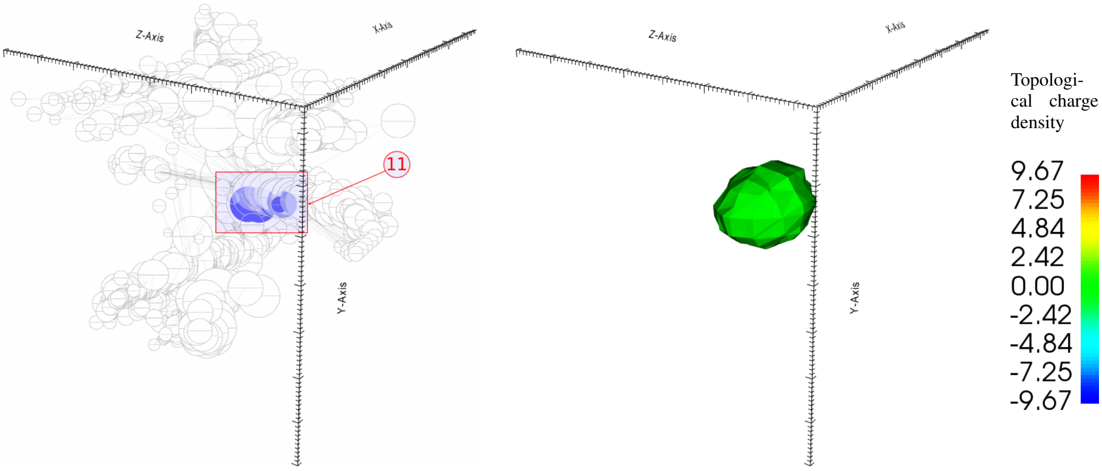


Figure 7.25: The JCN in domain layout for $t = (8, 1)$ displayed alongside the joint contour slabs. The selected glyphs (11) represent the instanton and correlate with the slab structure.

7.2.4.6 Time steps $t = 1$ and $t = 2$

The JCN for time-steps $t = (1, 2)$ reveals that the instanton can still be differentiated from the surrounding topological charge density by again halving the slab size to 0.087890625. The instanton is present within the JCN as two vertices, with the corresponding joint contour slabs retaining much of the geometric structure that is observed in the $t = (8, 1)$ joint contour slabs. Returning to examining the JCN and Reeb skeleton at the standard resolution shows evidence of the emergence of a potential anti-instanton in the data. This coincides with the global minima Q_{MIN} estimated by the cooling code to be present on the lattice at $(6, 2, 7, 2)$.

7.2.4.7 Time steps $t = 2$ and $t = 3$

The main instanton observable reappears in the bivariate topology at standard resolution with inputs $t = (2, 3)$. The JCN (Fig. 7.26) and Reeb skeleton (Fig. 7.27) identify the instanton (12) as a minor feature of the topology alongside several more significant features. Two other features, interesting in the context of lattice QCD, are present; object (13) identifies the global minima in 4D – a potential anti-instanton. Object (14) spreads across a large region of the quantised Reeb space, leading to a sheet-like JCN structure and features a sparse inner core.

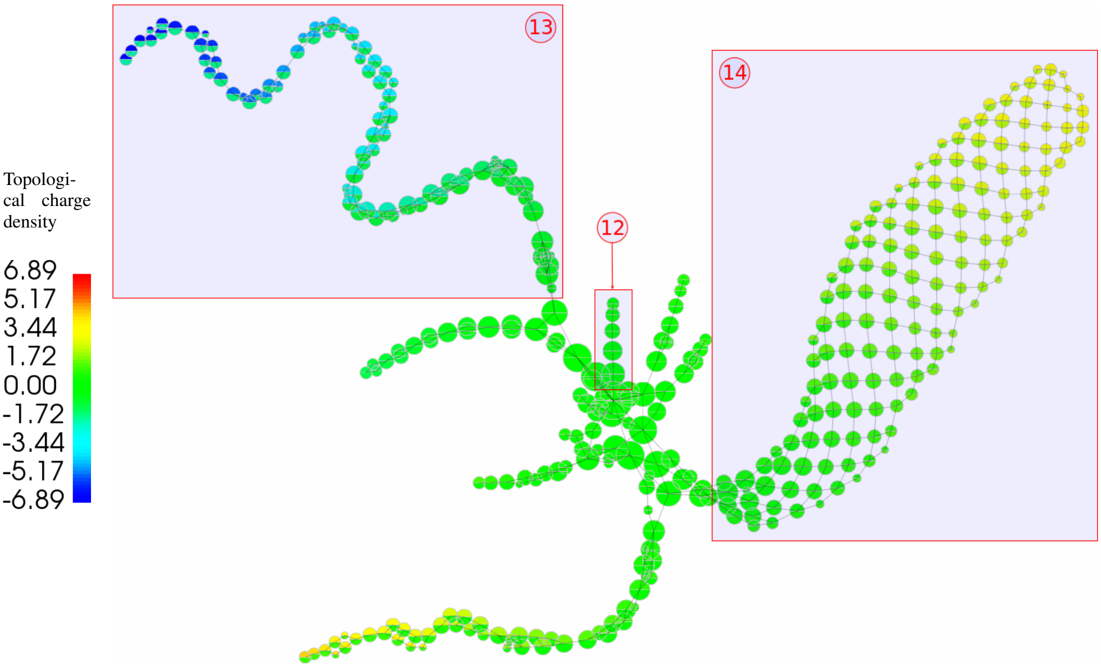


Figure 7.26: JCN for $t = (2,3)$. Branch (12) represents the instanton observables; branch (13) relates to an anti-instanton; branch (14) represents a large object dispersed across the Reeb space.

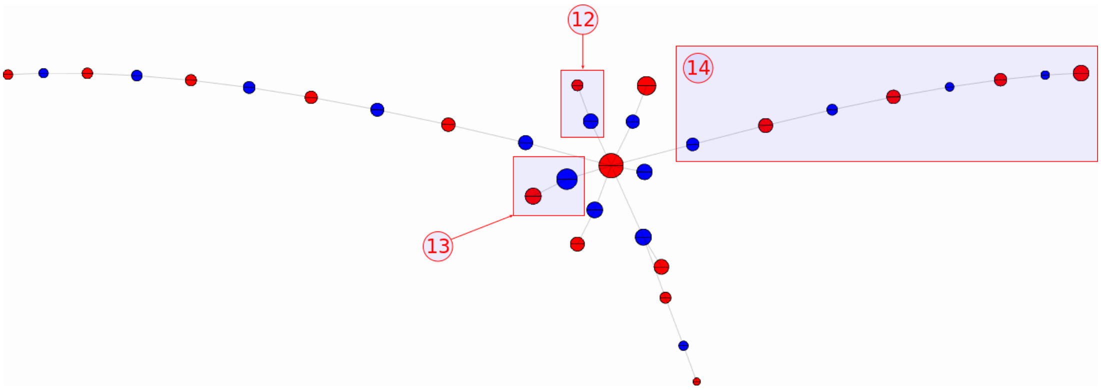
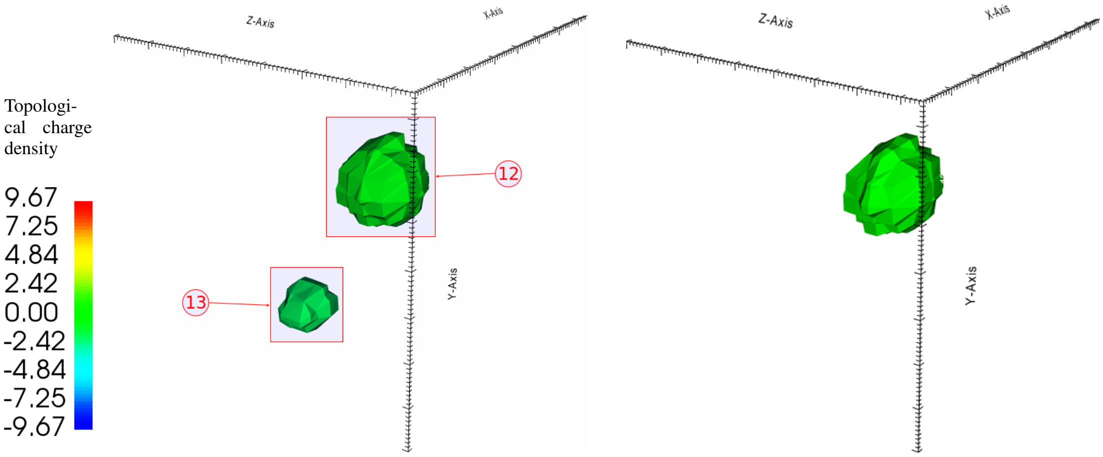


Figure 7.27: Reeb skeleton for $t = (2,3)$. The labelled objects correspond to those in the JCN; (12) is the main instanton observable; (13) is an anti-instanton; (14) is the large dispersed object.

The instanton (12) and anti-instanton (13) can be examined together as separate objects as shown in Figure 7.28. Despite occupying additional JCN vertices, the anti-instanton occupies a smaller volume when viewed as joint contour slabs. The presence of the anti-instanton appears not to distort the structure of the instanton. When compared to the $t = (4,5)$ JCN the joint contour slabs making up the instanton show very little structural difference at $t = (2,3)$.



$t = (2,3)$.

$t = (4,5)$.

Figure 7.28: On the left the instanton structure (12) and anti-instanton (13) from $t = (2,3)$ are shown together. It would appear the relative proximity of the anti-instanton (13) to the instanton (12) has little effect on its structure. When compared to the instanton at $t = (4,5)$ (right) the structure changes little.

7.2.4.8 Time steps $t = 3$ and $t = 4$

The final JCN returns to the origin, examining time-steps $t = 3$ and $t = 4$. The Reeb skeleton, shown in Figure 7.29, detects the instanton as the most prominent feature (14). Visually the slab structure captured by the JCN is well defined and resembles that of other time slices. The isovalues associated with the instanton object at $t = (3,4)$ (Fig. 7.30) shows a large jump in isovalue between the two time-slices, as captured by the colour change from red to green in the JCN.

The second most significant feature in the Reeb skeleton (15) relates to the local minima. The bivariate topological structure associated with this object, that is also identifiable in the $t = (2,3)$ JCN, exhibits a splitting behaviour. The global minima, first identified in the previous JCN (Fig. 7.26), transforms into two local minima (15) and (16) caused by the object splitting into two at an isovalue close to zero. The JCN captures this event as a fork at vertex (17) and leading to creation of two geographically distinct objects. The branching behaviour is absent from the simplified topological view presented by the Reeb skeleton (Fig. 7.29), where the associated values are merged into the region centred on zero, due to low multidimensional persistence.

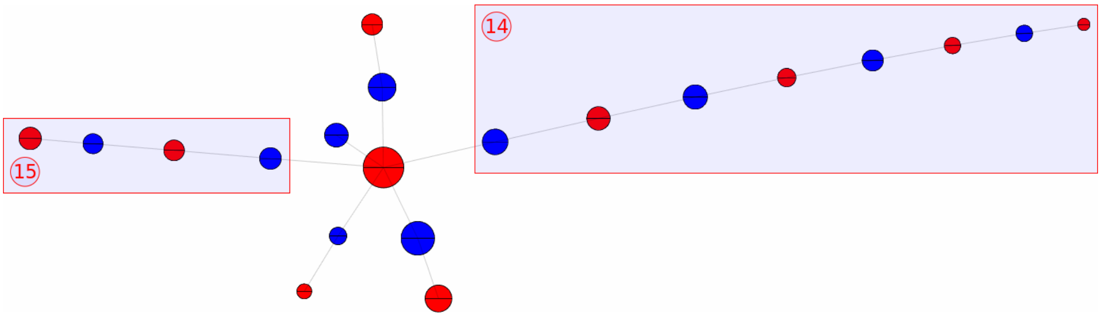


Figure 7.29: Reeb skeleton for $t = (3,4)$. The tracked instanton (14) appears as a major feature; also present is the anti-instanton (15) identified in the previous JCN.

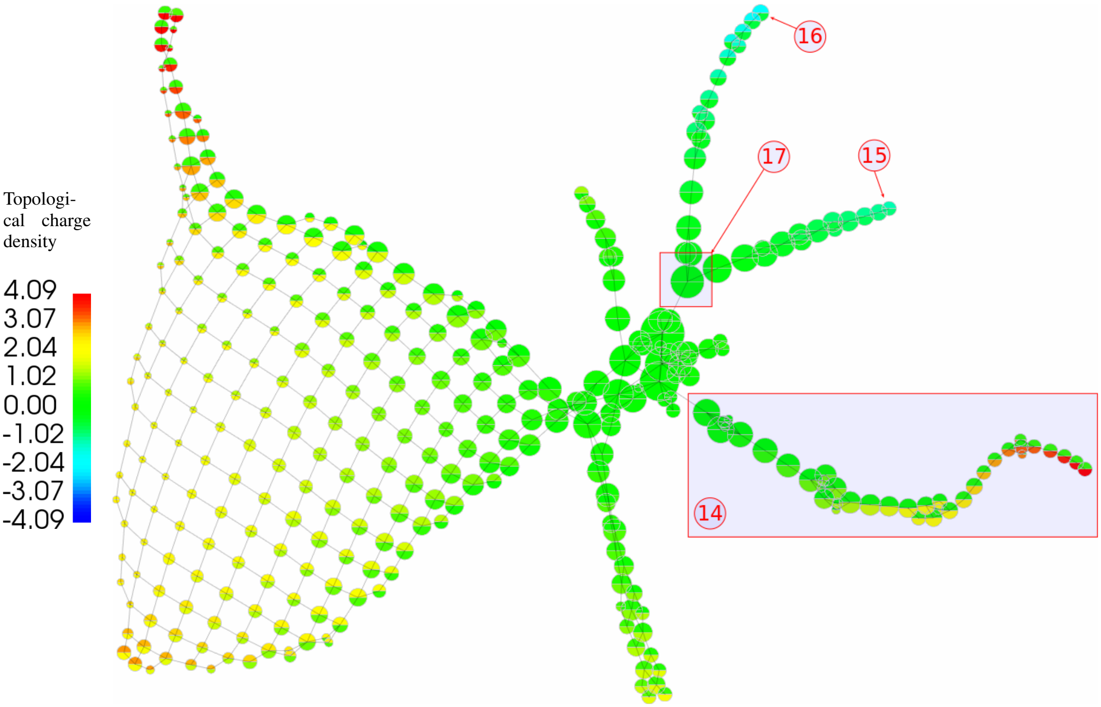


Figure 7.30: JCN for $t = (3,4)$. Instanton (14) shows a heightened isovalue at $t = 4$; the anti-instanton features a fork-like behaviour, splitting at vertex (17) into its inner (15) and outer components (16). The inner part of the anti-instanton (15) corresponds to the branch present in the Reeb skeleton.

7.3 A quantitative approach to instanton tracking

We have shown how the instanton can be tracked by the JCN visually; however, for domain scientists a more quantitative approach is required. In the following section we look at what statistical measurements are available using bivariate topological structures. It should be noted that each of these process require manual locating of the target object.

7.3.1 Slab count

A basic measure of persistence can be computed from the JCN by counting the number of slabs making up a topological region. Each vertex in the JCN represents a joint contour slab, or region of the quantised Reeb space, associated with a pair of isovalues. Throughout the case study it was found that the number of slabs linked to an object tended to vary with two properties of the object; the volume of the slabs, and the isovalue range of the object. Sparse objects with a wide range of isovalues frequently appear as sheet-like structures in the JCN, and densely packed objects appear with a branch-like structure.

We found that it was possible to get a rough estimate of properties of lattice objects in the quantised Reeb space by calculating the percentage of vertices in the JCN contributing to the object, as detected through manual selection.

Input fields	Instanton sub-graph	Entire JCN structure	Percentage
$t = (1,2)$	0	572	0
$t = (2,3)$	5	371	1.35
$t = (3,4)$	39	339	11.50
$t = (4,5)$	199	584	34.08
$t = (5,6)$	212	472	44.92
$t = (6,7)$	44	237	18.57
$t = (7,8)$	6	360	1.67
$t = (8,1)$	0	806	0

Table 7.2: Number of JCN vertices contributing to instanton structure
All values are taken using a slab size of 0.17578125.

Table 7.2 presents the number of vertices present in each JCN across the temporal axis at the standard slab size. When plotted as a histogram, as in Figure 7.31, a peak in the number of vertices in the manually identified instanton sub-graph coincides with Q_{MAX}.

The JCNs show an increase in the number of total vertices on the lead up to Q_{MAX} and Q_{MIN}, followed by a drop in graph complexity after each event. The peaks were expected to coincide with the emergence of the anti-instanton and instanton; however, both peaks seem to proceed the time-steps containing the (anti-)instanton structures rather than directly matching them.

Viewing the number of vertices in the instanton sub-graph as a percentage of the entire JCN (Fig. 7.32) suggests that the structure dominates the quantised Reeb space at its peak value, where $t = 5$, agreeing with the predicted global maxima Q_{MAX}. In the $t = (1,2)$ and $t = (8,1)$ JCNs the percentage of vertices is zero — this is where the instanton structure was not detected at all at the standard resolution.

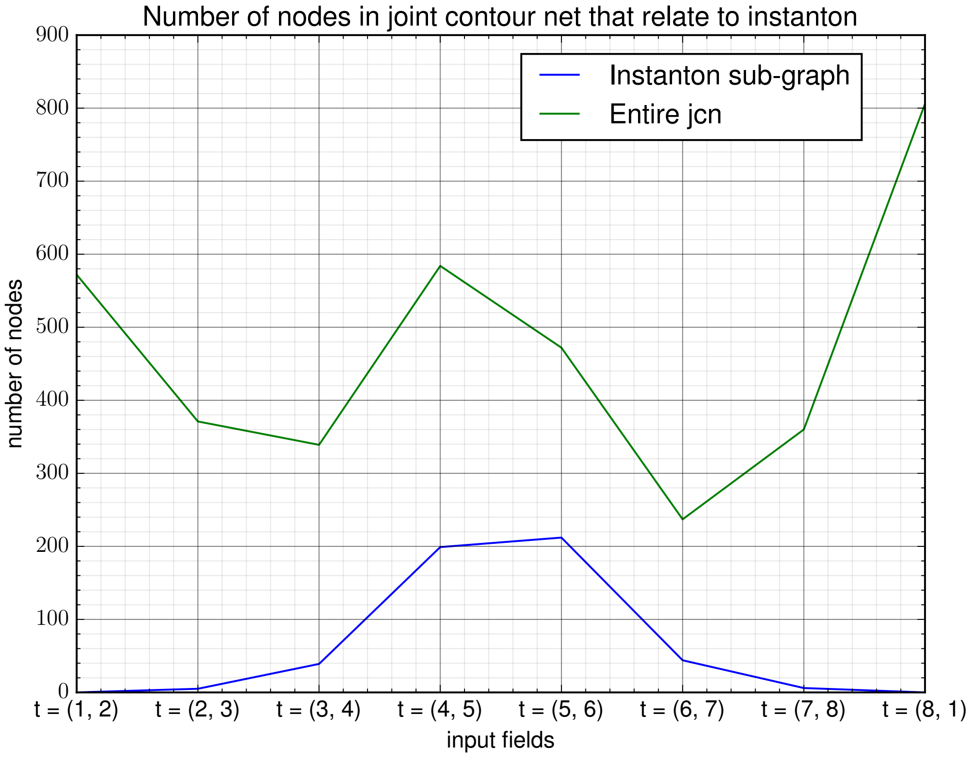


Figure 7.31: Number of slabs in the JCN representing the instanton structure.

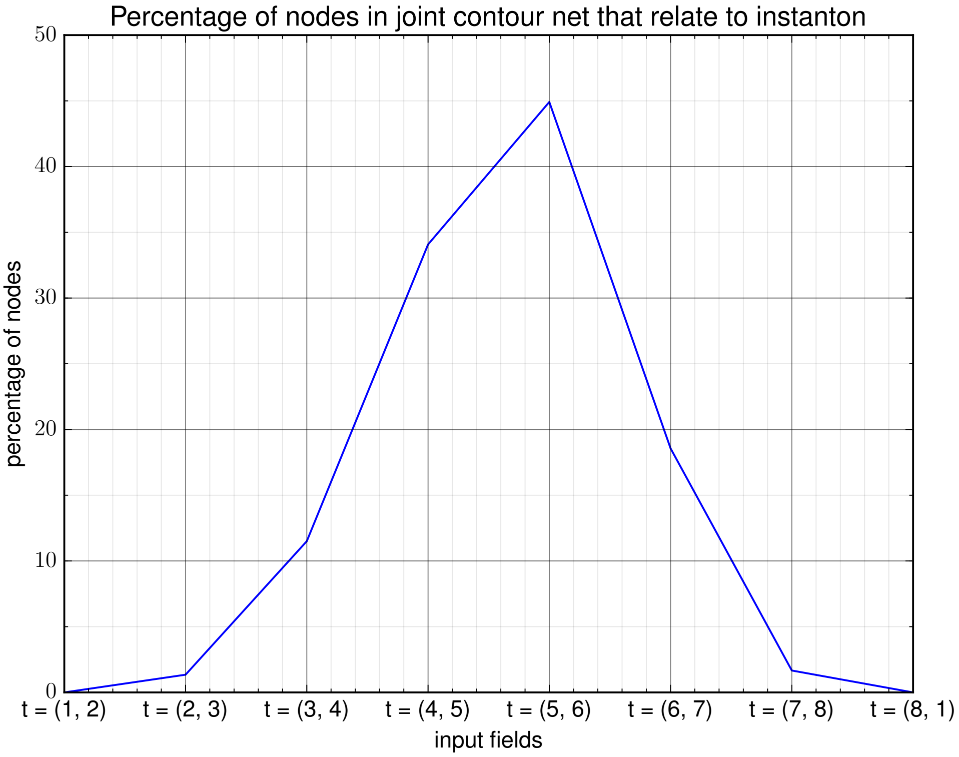


Figure 7.32: Percentage of slabs in the JCN representing the instanton structure.

7.3.2 Simplification and persistence data using the Reeb skeleton

The Reeb skeleton summarises the multivariate topology in a more compact graph structure. Adjacent slabs are merged, provided no critical events coincide with the slab. This allows entire path connected regions to be summarise as a single unit, represented by a vertex in the Reeb skeleton. Besides simplification of the quantised Reeb space, the Reeb skeleton enables the generalisation of persistence measures used in univariate topology to multivariate topology. Each vertex in the Reeb skeleton has persistence measures attached representing the connected components (the joint contour slabs), allowing us to further analyse the instanton structure.

This section examines how different persistence measures, used during simplification, affect the ability of the Reeb skeleton to detect the instanton structure. The simplification measures are defined as follows:

Reeb skeleton Collapses adjacent slabs into a single vertex, meaning entire sheet or branch-like structures in the JCN can be summarised by a single branch of the Reeb skeleton. Topological events, such as splits and merges, are captured as vertices with degree 3 or higher. No simplification is performed, frequently leading to the creation of multiple leaf vertices along branches of the Reeb skeleton.

Simplified Reeb skeleton This is the full Reeb skeleton except a basic pruning of non-critical vertices is performed. First, all degree one singular vertices (leaf nodes) are removed from the Reeb skeleton and replaced with regular vertices. Next any degree two singular vertices are replaced with regular nodes. Finally, any regular leaf vertices are merged with their neighbours until a singular vertex is encountered. This has the effect of collapsing large branches of regular vertices to a single vertex representing the entire set of connected components.

Persistence simplified Reeb skeleton The Reeb skeleton is first pruned of non-critical vertices, as described above. Following this each of the remaining components in the graph are assigned a level of persistence based upon the quantity of JCN nodes that make up the sub-graph represented by the vertex. Next regular vertices of the JCN that fall below a specified threshold are removed from the graph. Finally, any remaining non-critical vertices are removed by repeating the basic pruning technique.

Volume simplified Reeb skeleton This method of simplification is similar to "persistence simplification" except instead of counting the number of JCN nodes in a sub-graph of connected components, the volumes are approximated by counting the number of fragments passed through. Each fragment represents a tetrahedra cell in the quantised Reeb space, counting the number of these in each slab gives a rough estimate of volume.

Table 7.3 shows the effect that various forms of simplification have on the Reeb skeleton, also visualised in Figure 7.35. This confirms that under simplification the Reeb skeleton often reduces regions of the bivariate topology representing the instanton to a single vertex.

Simplification using basic leaf pruning and volume measures at first appear to give similar results. However, when viewing the effect by cross referencing the JCN (Table 7.4, Fig. 7.36)

	$t = (1,2)$	$t = (2,3)$	$t = (3,4)$	$t = (4,5)$	t = (5,6)	$t = (6,7)$	$t = (7,8)$	$t = (8,1)$
Unsimplified Reeb skeleton	0	2	8	43	12	2	0	0
Simplified Reeb skeleton	0	1	1	1	4	1	0	0
Persistence simplification	0	0	1	1	1	0	0	0
Volume simplification	0	1	1	1	4	1	0	0

Table 7.3: Number of vertices in Reeb skeleton after simplification

	$t = (1,2)$	$t = (2,3)$	$t = (3,4)$	$t = (4,5)$	$t = (5,6)$	$t = (6,7)$	$t = (7,8)$	$t = (8,1)$
Manual selection	0	5	39	199	212	44	6	0
Unsimplified Reeb skeleton	0	5	34	186	212	28	0	0
Simplified Reeb skeleton	0	2	33	167	122	1	0	0
Persistence simplification	0	0	26	137	120	0	0	0
Volume simplification	0	2	33	167	122	1	0	0

Table 7.4: Number of node in JCN relating to instanton after simplification

variations appear, as smaller slabs are filtered out under simplification. The simplification preserves the observed forking behaviour in the instanton structure at $t = (5,6)$, resulting in four nodes in the Reeb skeleton (Fig. 7.33). Investigating the object by viewing the slab geometry (Fig. 7.34) shows the instanton splits into an outer (20) and inner (21) shell structure.

Figure 7.36 shows how under simplification the number of vertices in the JCN judged to be topological noise varies by the persistence measure used. The unfiltered Reeb skeleton captures most of the vertices that are chosen by manual selection. Minor variations are found at the point where the instanton branch of the Reeb skeleton attaches to the region of percolation around zero. This appears as a very highly connected vertex at the centre of the JCN and Reeb skeleton. Under simplification most techniques merge the JCN vertices closest to isovalue zero into the central vertex due to their low relative persistence.

We also observed that the vertices at the ends of JCN branches, where the isovalue is furthest from zero, representing the core of an (anti-)instanton are often removed from the intended observable. There were other situations where simplification resulted in a single disconnected joint contour slab halfway along the instanton branch (Fig. 7.37). This is often linked to the creation of many small and low persistence slabs — most likely this is an indication that the slab size parameter is too high.

The most aggressive form of simplification witnessed in this case study was based upon the number of connected components. This removed the intended instanton observable from the simplified topology several time-steps earlier than other simplification techniques. We believe this was likely caused by an overly strict threshold level and highlights the need for careful selection of simplification parameters.

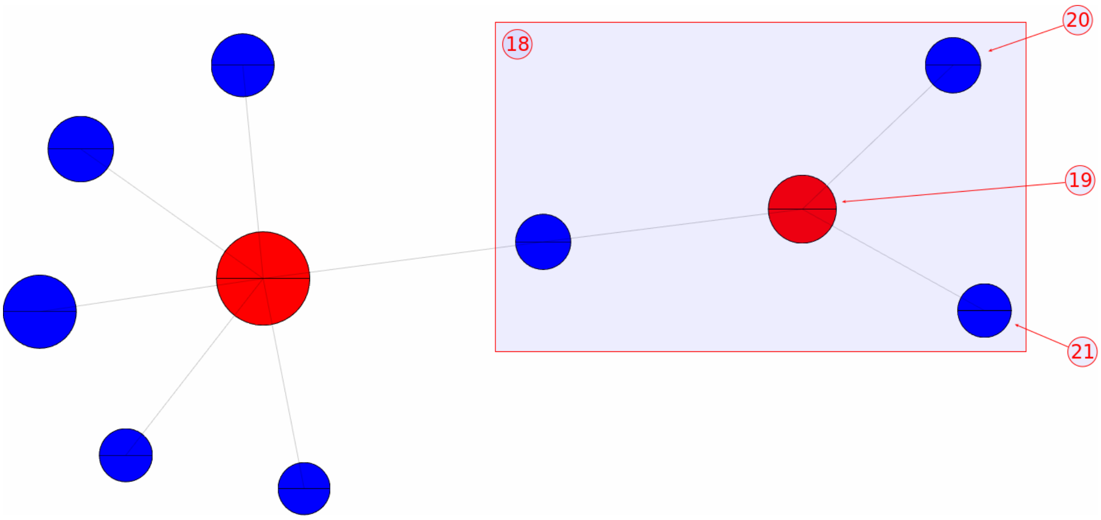


Figure 7.33: Reeb skeleton for $t = (5,6)$, simplified using volume persistence measures. The instanton (18) is captured as a branch in the simplified Reeb skeleton. At vertex (19) the instanton splits into two; the outer part (20) and inner core (21).

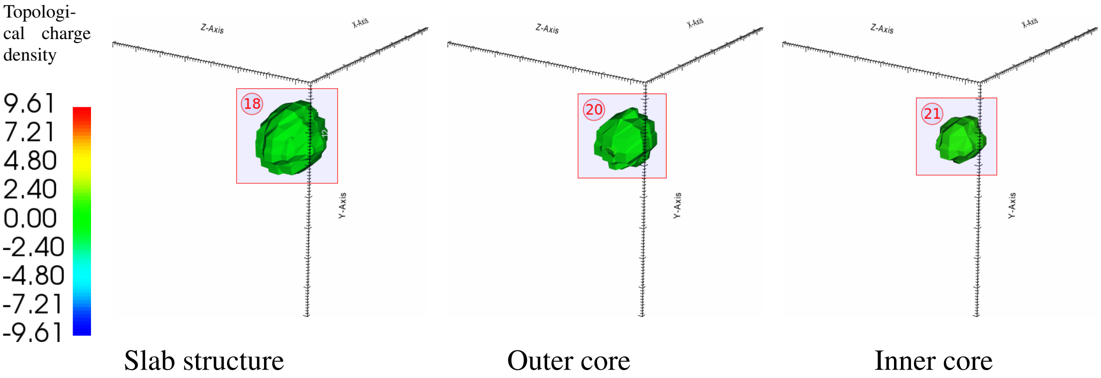


Figure 7.34: The forking behaviour visualised using the slab geometry. The full instanton structure (18) splits into outer (20) and inner (21) parts.

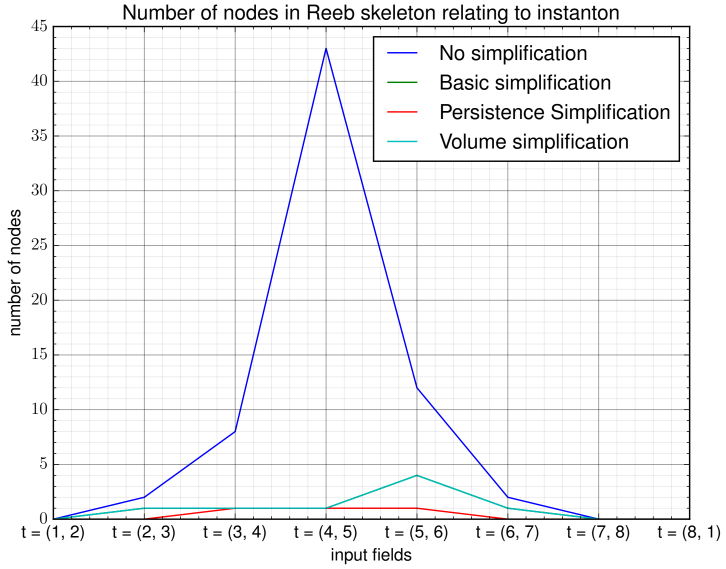


Figure 7.35: Number of nodes in the *Reeb skeleton* representing the instanton under simplification.

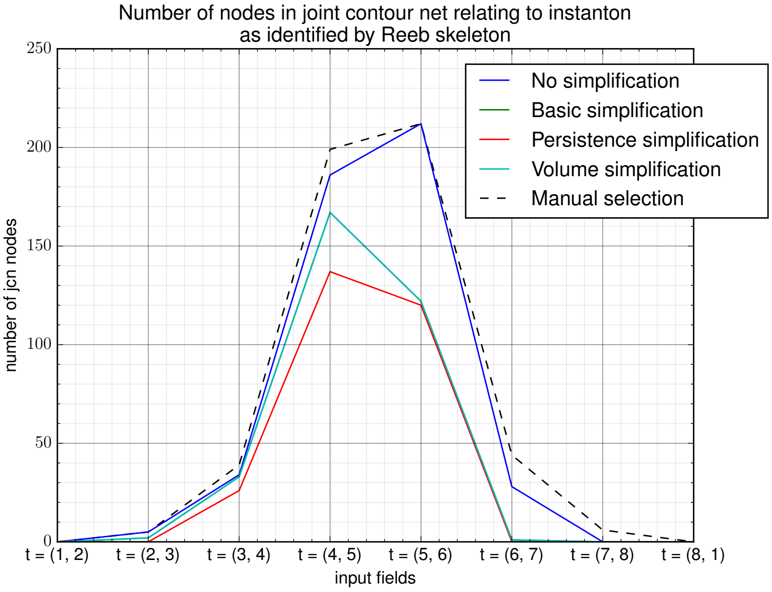
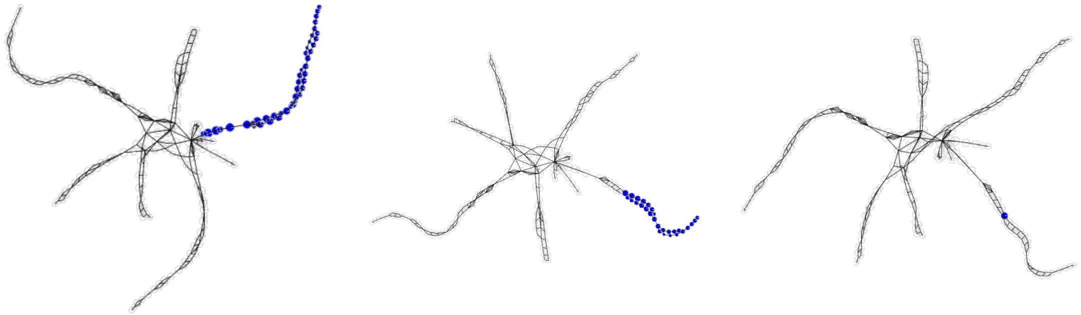


Figure 7.36: Number of nodes in the *JCN* representing the instanton under simplification.



Manual selection.

Unsimplified Reeb skeleton selection.

Volume simplified Reeb skeleton selection.

Figure 7.37: Comparison of simplification techniques on the $t = (6,7)$ joint contour net.

7.4 Feature detection using the entire lattice data

The final use of the JCN is an experimental attempt to detect the instanton using all eight time-slices as input. Whilst the Reeb space is only well defined for situations where $m > n$, the JCN algorithm can be used on data where this condition is not met. In this situation we will be dealing with data defined on an $m = 3$ dimensional mesh with $n = 8$ function values. The output of the algorithm is a subdivision of the input field with regard to all 8 functions; embedded amongst this we expect to find structure relating to the instanton.

The JCN for $t = (1,2,3,4,5,6,7,8)$ is shown in Figure 7.38, featuring 36450 vertices and 104373 edges. The structure clearly splits the input space into distinct topological objects, meaning it is possible to select a single branch of the JCN to isolate the instanton structure (1). Viewing the instanton as joint contour slabs (Fig. 7.39) shows an object very similar to that output by the bivariate JCNs.

Stepping through the data, colouring the graph by isovalue for each time-step, allows the variation in isovalues of each branch to be observed. When reaching peak topological charge density Q_{MAX} at time-step $t = 5$ all other vertices in the graph have very low relative isovalues, indicated by blue glyphs. This is largely a side-effect of the colour transfer function being relative to the data set in view — for Q_{MAX} this offsets the values further from zero than other time-steps. The same effect is less obvious for Q_{MIN} at $t = 2$, but can still be observed.

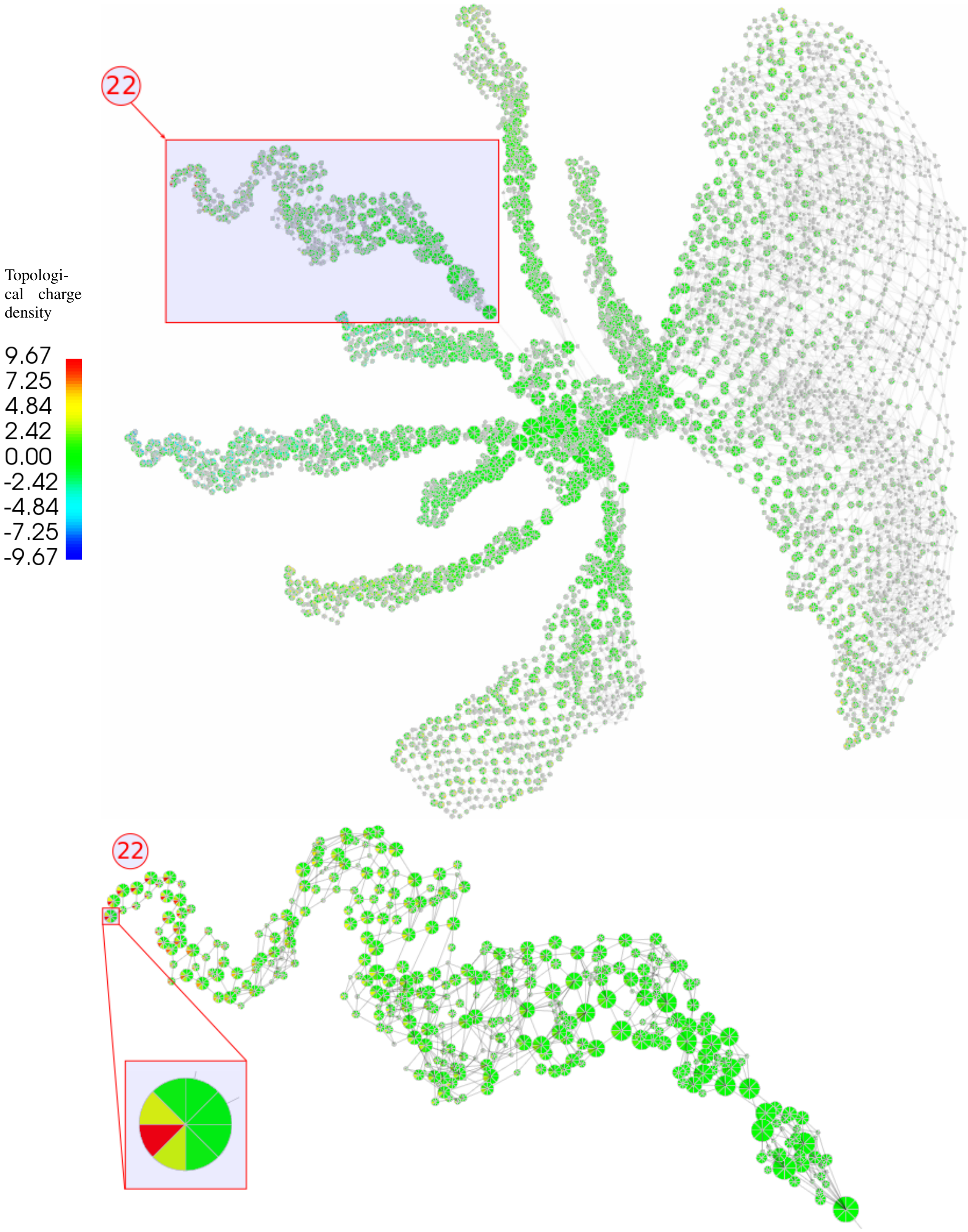


Figure 7.38: Top: The JCN for $t = (1,2,3,4,5,6,7,8)$. captures the instanton as branch (22). Bottom: Examining the branch in greater detail shows that the peak in topological charge density is capture in only a few vertices at the end of the branch. The peak in the topological charge density is captured by the multivariate glyph at $t = 5$. It is also possible to observe elevated levels in the topological charge density in neighbouring time slices.

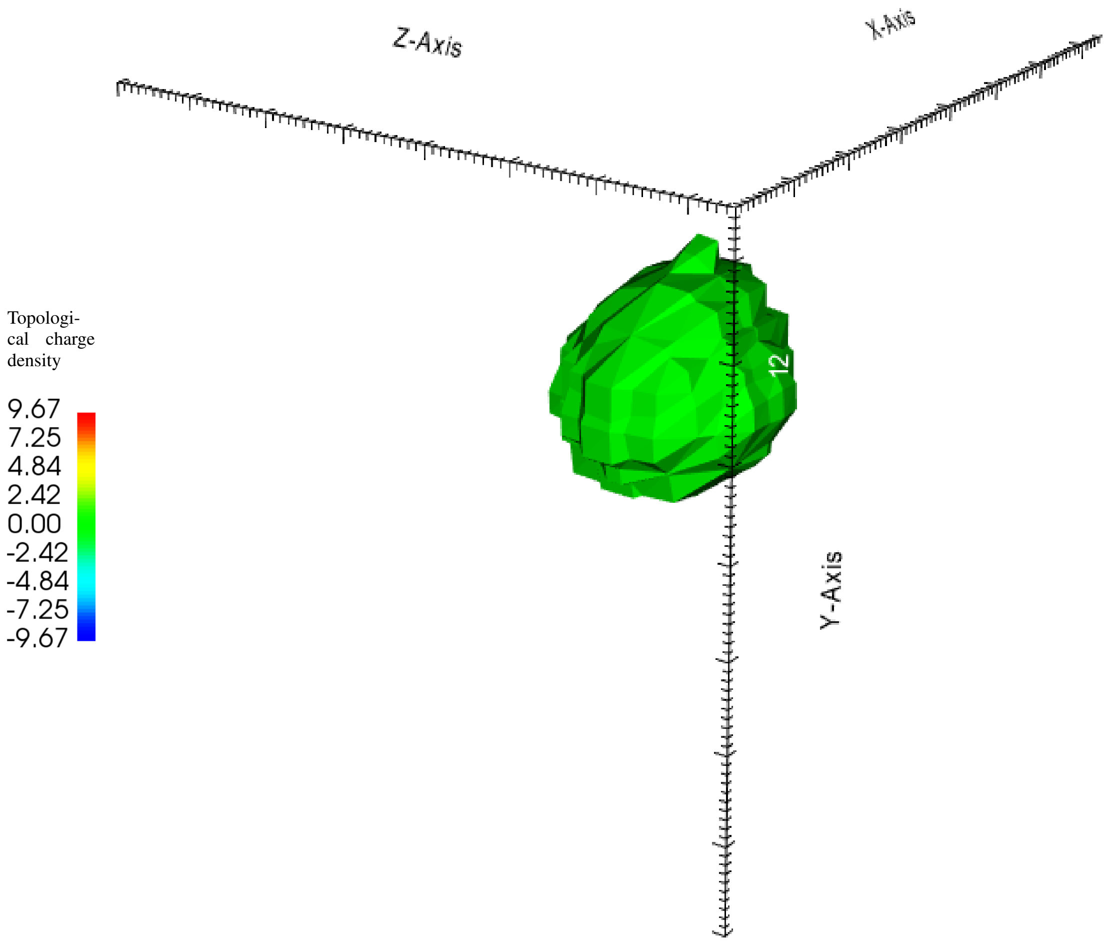


Figure 7.39: The slab structure of the instanton recovered from the JCN of $t = (1,2,3,4,5,6,7,8)$.

7.5 Summary

We have shown in this chapter how recent advances in multivariate topological analysis can be applied to lattice QCD data sets. We have demonstrated how multivariate topology can be used for comparing data with a temporal element in order to observe critical topological events as the time step is varied. This case study also considered some basic multivariate persistence measures available in the data, such as the number of slabs. These measures are suited to evaluation as lattice simulation parameters are varied as their values can be collected autonomously.

In summary this chapter makes the following contributions:

- The JCN was used to track and observe the structure of an instanton in 4D space-time.
- Quantitative measures were taken directly from the JCN and Reeb skeleton to evaluate the importance of lattice observables in the context of the topology of the lattice.
- We demonstrated how multivariate simplification metrics could potentially be utilised to locate important observables in lattice QCD.

The JCN provides an interesting opportunity to evaluate the effect of the cooling process, where we may consider the cooling iteration as an imaginary time-step, this is studied in greater detail in Chapter 9. In Chapter 8 we use the JCN in an alternative way to study correlations between two fields defined on the lattice at the same point in space-time.

Chapter 8

Using the JCN to compare two variables on the same lattice

Contents

The following chapter presents a second case study where we use the JCN to analyse potential correlations between fields on the same lattice. We study various lattice field combinations to examine how the JCN can offer new methods of predicting the presence of (anti-)instantons and their properties.

In Section 8.2 we carry out a benchmarking study to analyse how quantisation effects affect the JCNs ability to capture the topology of lattice fields. We use information gathered from the benchmark study to select optimal inputs to the JCN algorithm in Section 8.3 where we carry out a visual analysis of multiple bivariate field combinations to examine the presence of an (anti-)instanton. The fields used in this study all contribute to, or show evidence of, (anti-)instantons — these include the space-like and time-like plaquettes, the average plaquette, and the difference plaquettes.

In this Chapter we continue to use the JCN and Reeb skeleton introduced in Chapter 7.

The *Joint contour net (JCN)* presents the quantised Reeb space as a graph structure where:

- Vertices: represent slabs where the isovalue of n functions fall into a defined interval. The glyphs used to represent each vertex are split into n sectors to reflect the isovalue in each field.
- Edges: represents adjacency between slabs.
- Glyph scale: this is used to reflect the volume of a slab.

The *Reeb skeleton* presents the quantised Reeb space in a simplified form where adjacent slabs are merged into a single vertex:

- Red vertices: capture major changes in the topological structure including the birth or death of a path-connected region of the Reeb space.
- Blue vertices: capture minor changes in the topology.
- Edges: represent adjacency between regions of the Reeb space.
- Glyph scale: this is used to relay information about multi-dimensional persistence measures.

In addition we introduce a variant of the JCN where:

- Red vertices: represent slabs in the quantised Reeb space where the function gradients become parallel or equal to zero.
- White vertices: represent all other slabs where there is no alignment of function values.
- Edges: represents adjacency between slabs.
- Glyph scale: this is used to reflect the volume of a slab.

8.1 Implementation

The software used to perform this case study is the same tool used in Chapter 7. However, in this case study the inputs are from various combinations of two fields defined on the same lattice. We continue to use the JCN implemented in VTK as part of the Multifield Extension of Topological Analysis (META) project [163]. As with the previous study, we use a Freudenthal subdivison of cubic cells in into 6 tetrahedra.

8.1.1 Interactive user interface

Many of the software features used in this case study are similar to those used in the previous case study. However, a few additional features are introduced to help make a comparison between the bivariate inputs. Each field is assigned its own colour transfer function in order to allow the key features to be detected easily by examining the isovalues in the context of each fields expected data range (Fig. 8.1). Figure 8.2 shows how the ability to display Jacobi nodes in the JCN is used to examine alignment of gradients within the fields. The ability to filter parts of the JCN using domain layout of the JCN is also used extensively in this study to assist in locating specific lattice objects (Fig. 8.3).

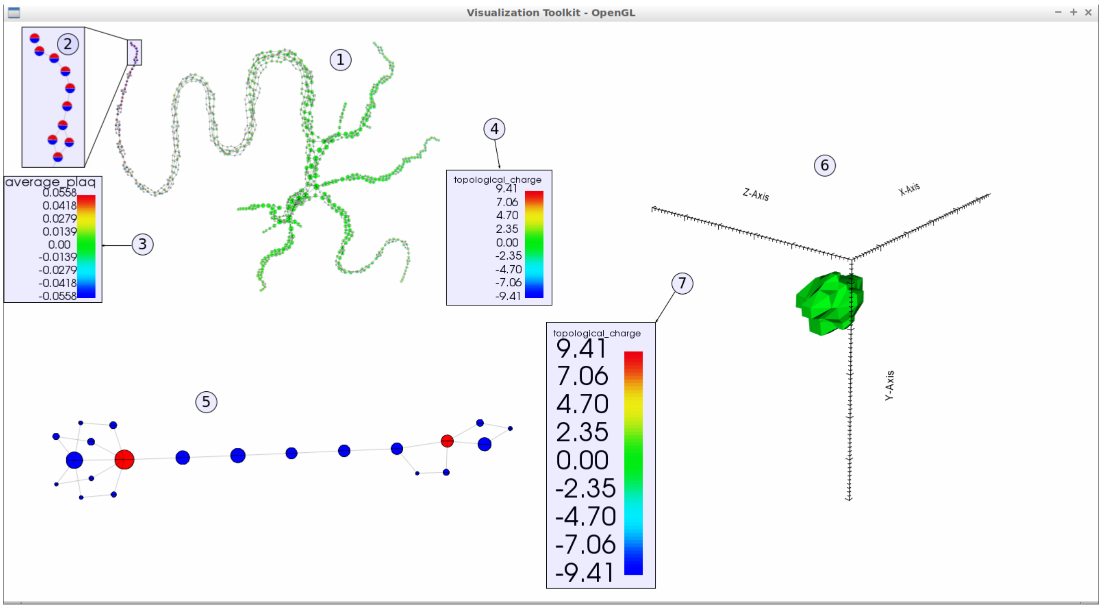


Figure 8.1: The main user interface for exploration of the JCN data. The JCN is generally displayed using a force directed layout (1) in order to prevent occlusion of vertices. Each vertex in the JCN represents a single joint contour slab, these are presented as glyphs (2) scaled by the volume of the region they represent. Glyphs are split into halves to represent the isovalues in each field for that slab; the top half represents the first field (3) and the second field (4) is represented by the bottom half. Separate transfer functions can be assigned to each field; this is useful in lattice QCD as most interesting features occur at the extrema in each field. The Reeb skeleton (5) presents a simplified view of the JCN where adjacent slabs are merged whilst preserving connectivity. Red glyphs in the Reeb skeleton represent changes in topology such as the merging of regions of the Reeb space. User selections in the JCN can be rendered as joint contour slabs (6); this is rendered using the colour transfer function of either input field (7) and can be changed at run time.

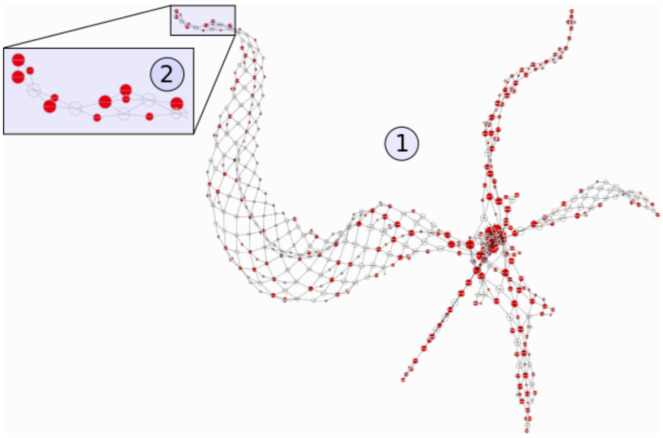


Figure 8.2: The JCN (1) can relay information about Jacobi nodes. Red glyphs (2) represent Jacobi nodes, where the gradient of the two fields align or becomes zero; regular nodes are shown in white.

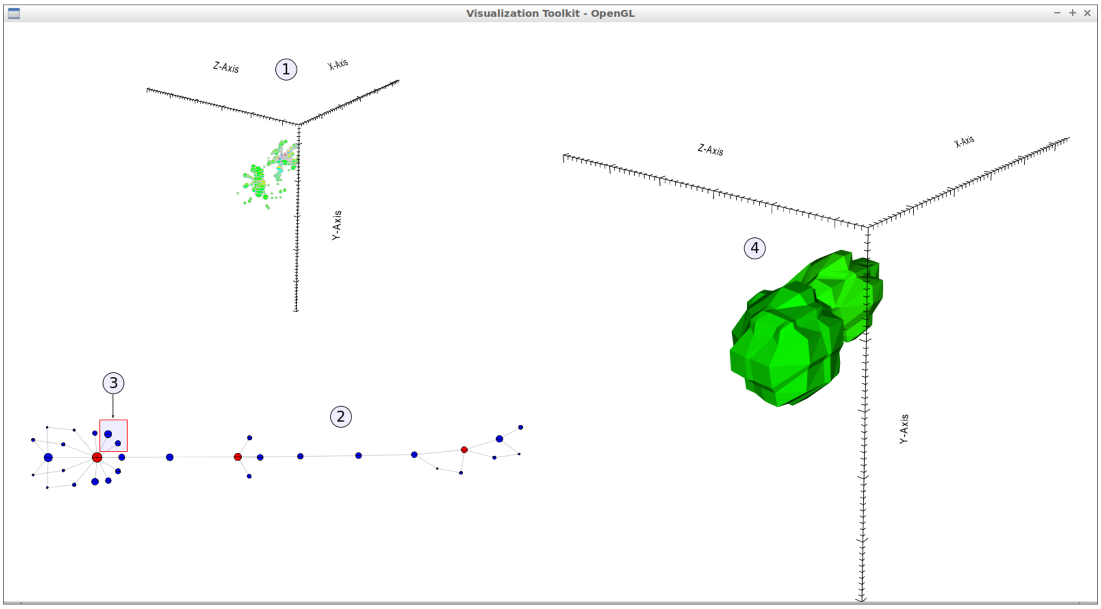


Figure 8.3: The JCN can also be placed in *domain layout* (1). This places the glyphs representing each slab at the barycentric coordinates of the slab they represent. The Reeb skeleton (2) can be used to select regions of interest in the simplified topology. Selected Reeb skeleton vertices (3) can be used to filter the JCN to show only specific parts of Reeb space. The selected regions in the JCN can be rendered as slabs (4) to recover geometric features of the objects captured by the bivariate topology.

8.2 Quantisation effect on input fields

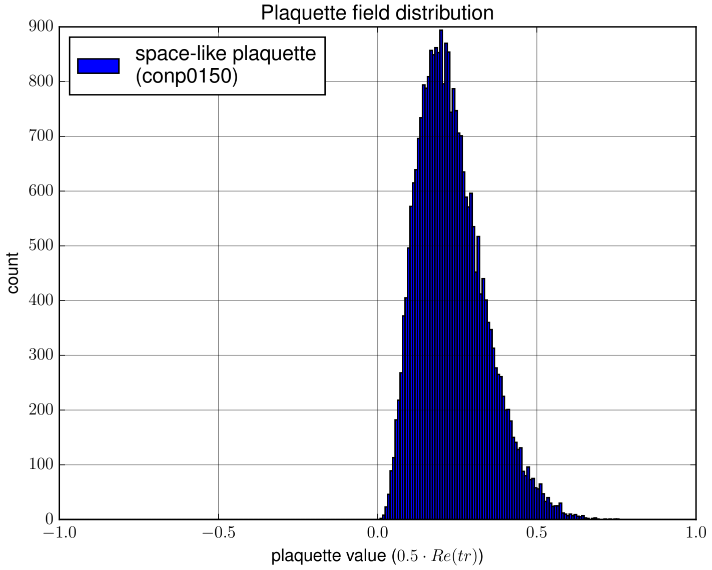
The JCN uses a discrete method of defining regions in scalar field(s) using user controlled parameters. Each vertex in the JCN represents a region of the Reeb space falling within an upper and lower boundary of isovalues in multiple fields. The interval range is determined by the “slab size” parameter, which is specified independently for each field in the input data and directly effects computational complexity. In this small case study we investigate the effect slab size has on accuracy by studying the effect it has on plaquette fields — exploiting that for a single scalar input the JCN approximates the Reeb graph.

8.2.1 Input field

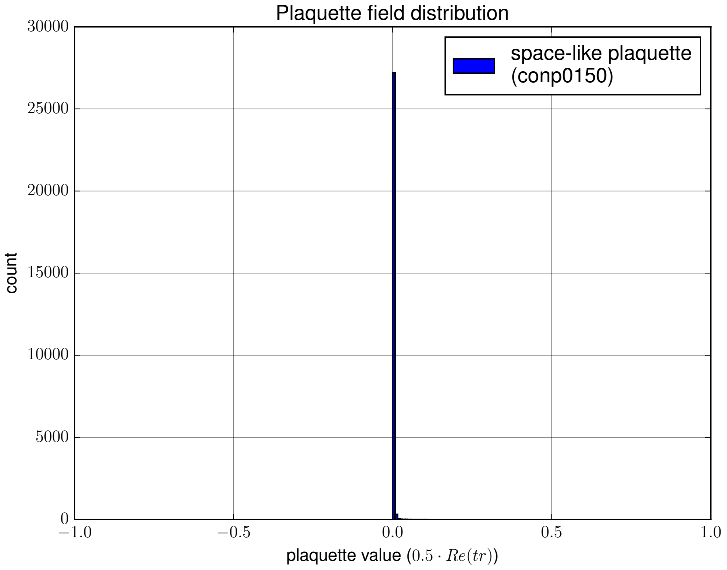
A configuration with chemical potential $\mu = 0.3$ is chosen at random from the $12^3 \times 16$ “hot” ensembles — meaning the lattice is defined with 12 sites in space and 16 sites in time. Configuration *conp0150* is analysed at 15 cooling iterations and sliced on the t axis at $t = 6$. However, the process detailed here is applicable to any field on the lattice and serves as an example work flow.

Plaquettes are calculated as matrix products and must first be reduced to scalar values for analysis (see Sec. 4.3.1). As plaquettes are represented by $\mathsf{SU}(2)$ matrices they have a theoretical range of $[-1.0, 1.0]$ (8.1). For this reason we assume that potential values in the output always fall into this range, even though in practice the actual range is significantly reduced as the data is cooled (Fig. 8.4). Because plaquette values tend toward a small region centred on zero when subjected to cooling, it is necessary to chose a slab size that is able to accurately capture the topology in this region. At 15 cools the majority of noise is removed by the cooling algorithm but prominent topological features still remain.

$$0.5 \times \Re\left(tr\begin{bmatrix}-1+0\mathrm{i} & 0+0\mathrm{i}\\ 0+0\mathrm{i} & -1+0\mathrm{i}\end{bmatrix}\right) = -1.0 \qquad 0.5 \times \Re\left(tr\begin{bmatrix}1+0\mathrm{i} & 0+0\mathrm{i}\\ 0+0\mathrm{i} & 1+0\mathrm{i}\end{bmatrix}\right) = 1.0 \quad (8.1)$$



Uncooled data.



15 cools.

Figure 8.4: Space-like plaquette field distribution in the *conp0150* configuration.

8.2.2 Studying effects of slab size on topology.

To determine optimum levels of quantisation the slab size parameter is adjusted as an iterative process by halving the value each time. A Reeb graph of the input field is computed to be used as a ground truth, allowing us to measure how closely the JCN approximates to the true Reeb graph as the slab size is reduced.

8.2.2.1 Ground truth

Figure 8.5 shows the computed Reeb graph beside a view of the field as a set of nested contours. The Reeb graph demonstrates that all scalar values exist above zero, there is one large global maxima, and several smaller local maxima that show up in the scalar field as pink regions. An appropriate slab size is required to maintain the individual local peaks by insuring they do not merge together.

The following statistics can be derived directly from the ground truth Reeb graph:

- Number of supernodes: $v_{TOT} = 44$.
- Number of superarcs: $a = 43$.

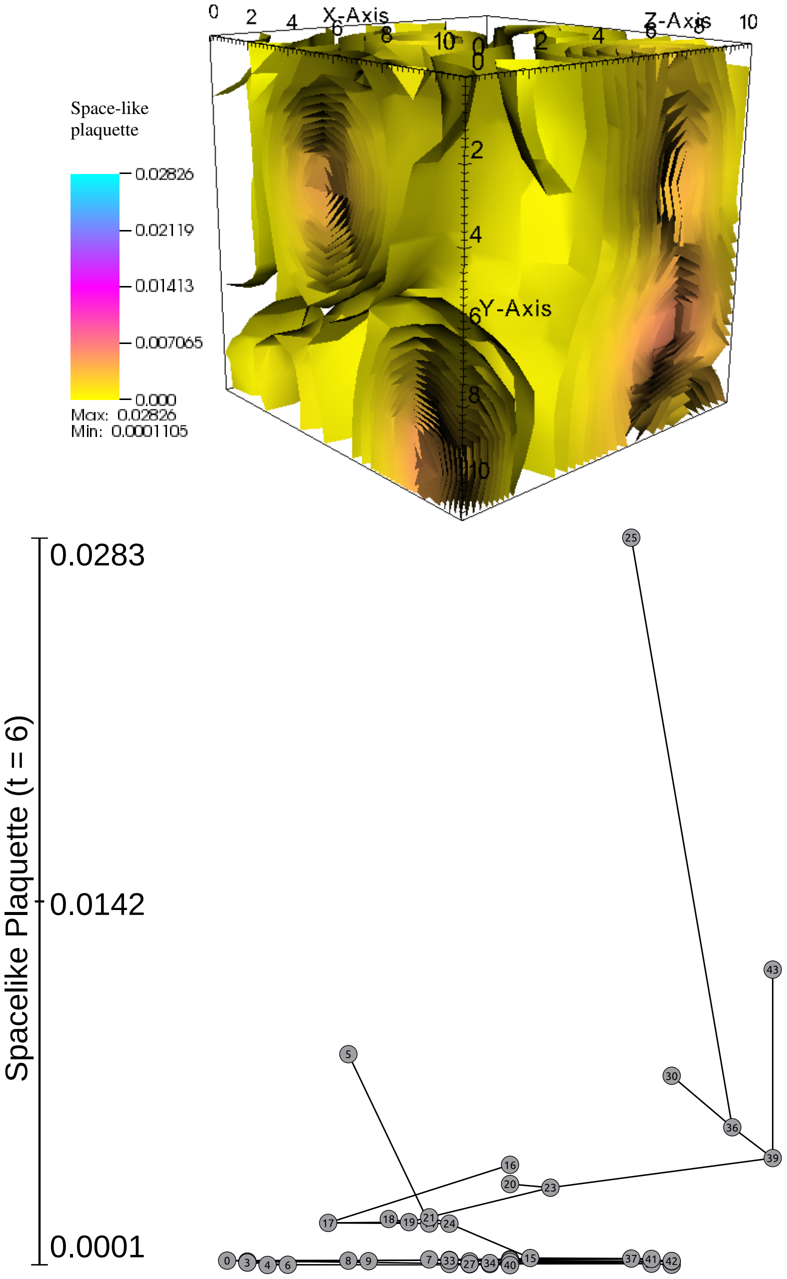


Figure 8.5: Nested contours and associated Reeb graph for space-like plaquette field at $t = 6$. The orange / pink regions relate to the upper leaves of the Reeb graph.

8.2.2.2 Approximating the Reeb graph using the JCN

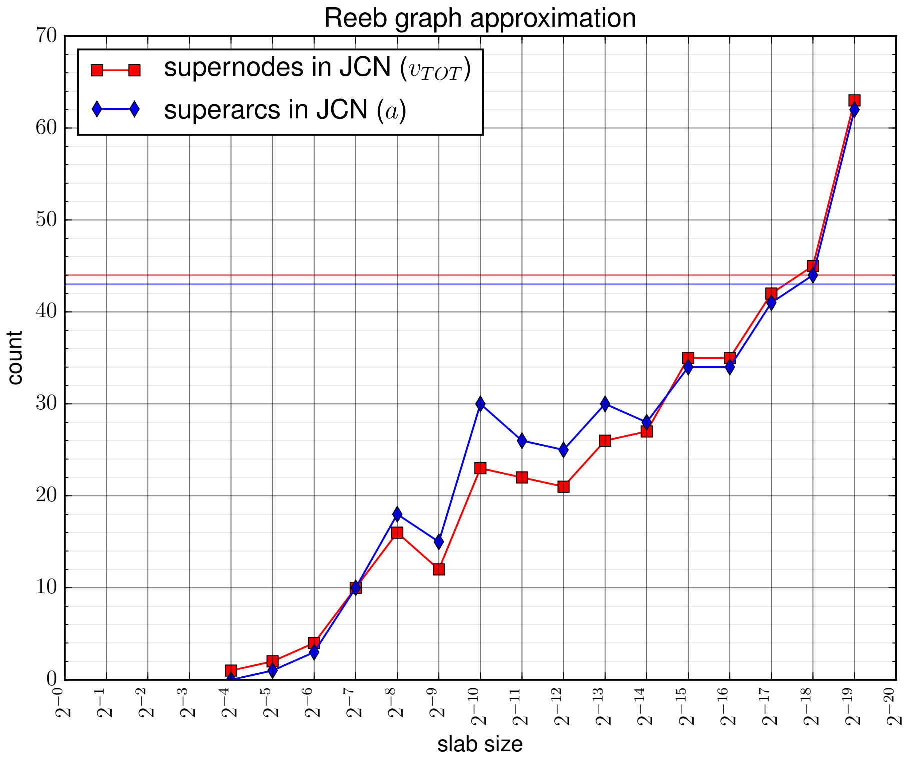
Table 8.1 presents results of varying the slab size parameter as each iteration doubles the data resolution. Figure 8.6 shows the convergence of the single field JCN and the Reeb graph as the slab size is decreased. At a slab size of 2^{-18} output from the Reeb graph algorithm and the

Slab size	v_{TOT}	a
2^{0}	n/a	n/a
2^{-1}	n/a	n/a
2^{-2}	n/a	n/a
2^{-3}	n/a	n/a
2^{-4}	1	0
2^{-5}	2	1
2^{-6}	4	3
2^{-7}	10	10
2^{-8}	16	18
2^{-9}	12	15
2^{-10}	23	30
2^{-11}	22	26
2^{-12}	21	25
2^{-13}	26	30
2^{-14}	27	28
2^{-15}	35	44
2^{-16}	35	34
2^{-17}	42	41
2^{-18}	**45**	**44**
2^{-19}	63	62
2^{-20}	n/a	n/a

Table 8.1: JCN approximation of the Reeb graph as the slab size is varied.

JCN become approximately equivalent.

It also appears possible to view the topology at a resolution beyond that of the Reeb graph algorithm when the slab size is set to 2^{-19}. Beyond this the algorithm fails, as the JCN sees an explosion in the number of topological objects ($v_{TOT} = 88918; a = 256347$). This is thought to be due to floating point limitations in the representation of isovalues required at this scale.

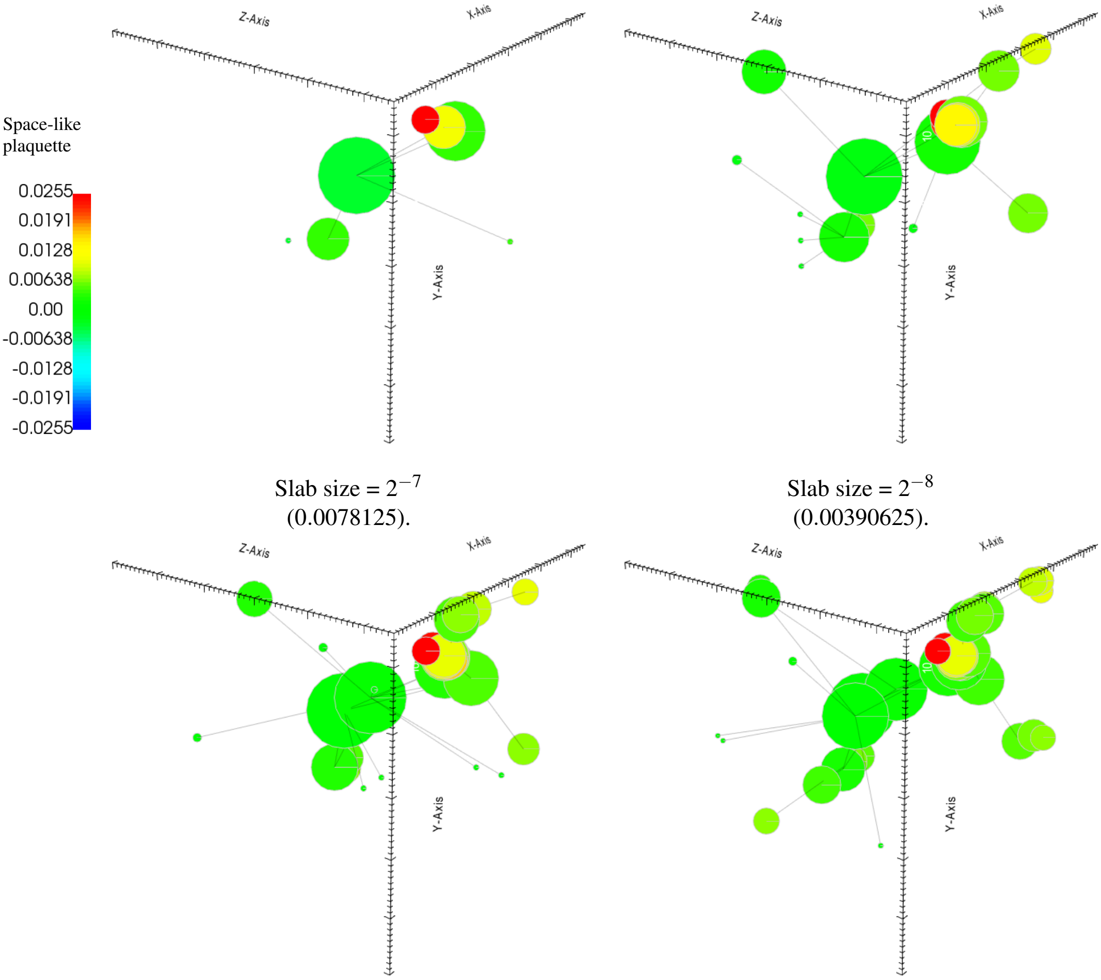


Flat lines represent “ground truth” Reeb graph.

Figure 8.6: Measuring properties of the JCN approximation of the Reeb graph as the slab size is varied.

An unexpected feature of the data is the appearance of loops in the topology between slab sizes 2^{-7} and 2^{-16}, characterised in Fig. 8.6 by an excess of superarcs (a) in comparison to supernodes (v_{TOT}). This appears to be an artefact of the interval based technique used by the JCN algorithm as loops are not present in the Reeb graph ground truth. Arranging the JCN in domain layout, as shown in Fig. 8.7, shows that loops are due to the adjacency of joint contour slabs. The domain layout view of the JCN also demonstrates the emergence of the principle observables, the objects associated with the field maxima, as the slab size is decreased. Each of these maxima are potential contributors to self-dual objects in topological charge density field, otherwise referred to as (anti-)instantons.

A slab size of 2^{-7} captures much of the region centred on zero as a single connected component in the topology. A single maxima is also visible, represented by the arc with a red leaf node, and a smaller connected region with a low isovalue is captured as a cyan leaf node. At 2^{-8} samples the maxima merges into the central cluster and the existing low peak is joined by three other local minima represented by blue and cyan nodes. At a slab size of 2^{-9} the four most prominent features in the topology, the four maxima, become visible as separate objects. The increased resolution allows the computed barycentre of each slab to become increasingly accurate, meaning that the JCN vertices move towards their true spatial position.



Slab size = 2^{-7} (0.0078125).

Slab size = 2^{-8} (0.00390625).

Slab size = 2^{-9} (0.001953125).

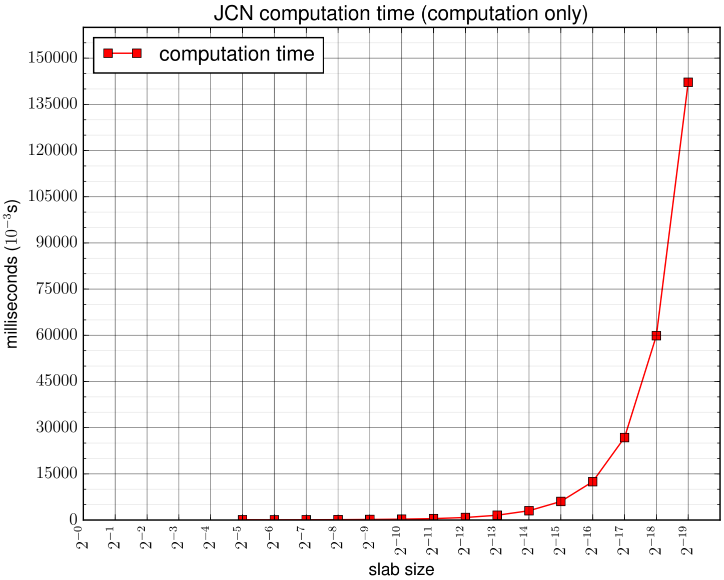
Slab size = 2^{-10} (0.0009765625).

Figure 8.7: The space-like plaquette field displayed using the JCN in *domain layout*. Glyphs are placed at the barycentre of the slabs they represent. As the slab size is decreased the JCN is able to capture the topological structure at increasing levels of detail.

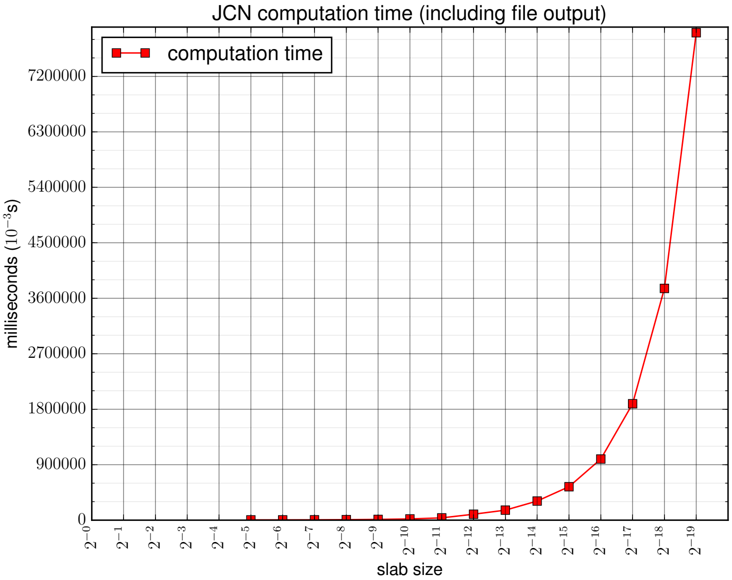
8.2.3 Complexity

As part of the investigation into the effect of slab size on the accuracy of the quantised Reeb graph statistics were collected regarding complexity. Figure 8.8 plots computation time as a function of slab size — showing exponential growth as the resolution is increased. However, for the relatively small input grids in lattice QCD this rarely becomes problematic.

A better overall understanding of the bottleneck is revealed by examining the number of triangles generated by the algorithm. Figure 8.9 shows how the algorithm generates exponentially more triangles as a consequence of the Reeb space being split into increasingly small



Execution time for JCN algorithm only.



Execution time including output to disk.

Figure 8.8: Measuring time complexity of JCN algorithm as slab size is decreased.

fragments. This is more problematic in terms of overall memory use of the program; this remains under a gigabyte until the slab size is decreased to 2^{-17} . From this point the memory usage continues to double, reaching just over 2.5GB as it converges to the Reeb graph ground truth. Besides creating problems in memory usage, the number of fragments substantially slows the algorithm during the merge phase where neighbouring fragments are joined to form output slabs.

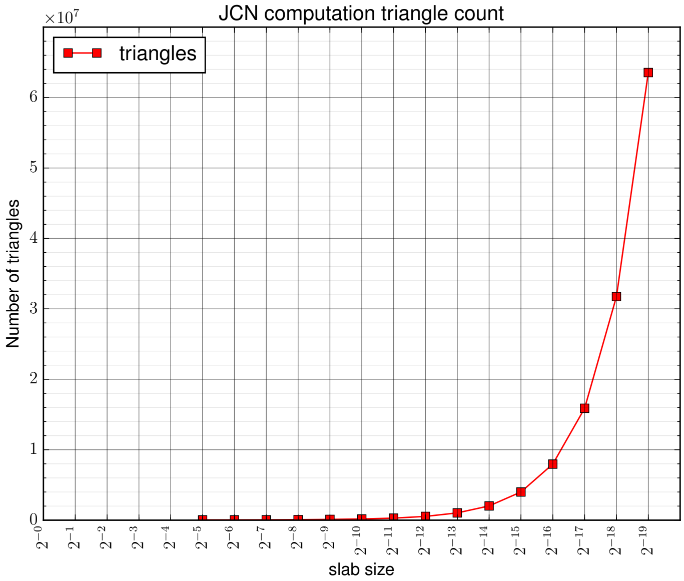


Figure 8.9: Number of triangles generated by JCN algorithm as slab size is decreased.

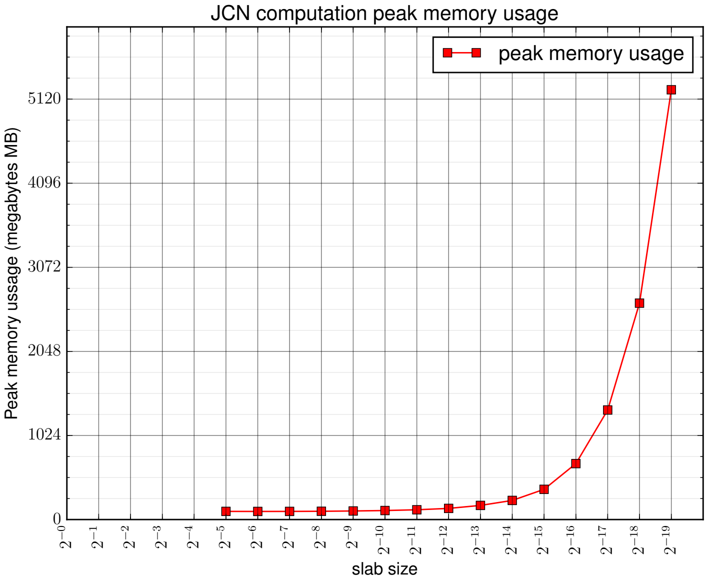


Figure 8.10: Peak memory usage (entire process) as slab size is decreased.

The main performance bottleneck experienced was linked to disk output. The vast quantity of information computable from the topology, including the persistence measures of individual slabs, saw a huge jump in overall execution time from minutes to hours.

8.3 Using the JCN to compare two variables on the same lattice.

In this case study we use the JCN as a method of inspecting potential (anti-)instantons for self-duality. Whilst it can be said that an (anti-)instanton will always appear as a global minima or maxima in the topological charge density field, not all minima and maxima will be an (anti-)instanton. In Section 4.3.4 we showed how the classical definition of an (anti-)instanton is a self-dual object, meaning that an (anti-)instanton must be a peak in both the space-like and time-like plaquette fields. A peak in either, but not both plaquette fields, may still be correlated to a local peak in topological charge density; however, this does not predict the presence of an (anti-)instanton.

We therefore formalise a set of research questions as follows:

- Can we use the JCN to identify and isolate (anti-)instantons in lattice QCD data?
- What does an (anti-)instanton look like in various bivariate fields?
- Does the JCN allow validation that an object is self-dual?
- What limitations are presented by lattice QCD data?

8.3.1 Selecting a candidate configuration for visual analysis

The process begins using data that is pre-cooled using the FORTRAN code [134] for 15 iterations. During the early stages of our research we visualised various lattice QCD configurations as isosurfaces of different fields. The data visualised in this study originates from this work and is previously viewed as nested isosurfaces using VisIt [149] in Section 4.3.4. There we demonstrated, but were not able to capture, the topological differences between the average plaquette field and difference plaquette field (Fig. 4.22). Potential (anti-)instantons appear in the difference plaquette field as values approaching zero, indicating the two fields are closely aligned. However, this must also coincide with an associated peak in the average plaquette field as general alignment of plaquette values is not a valid indicator of an (anti-)instanton. This case study demonstrates how the JCN can be used to capture the relative differences and similarities in the two fields in a compact graph structure.

Data is sourced from the “cold” ($12^3 \times 24$) lattice with a chemical potential of $\mu = 0.0$. Configuration *conp0068* is sliced along the t axis, focusing upon the 12^{th} time-step which represents a global minima of topological charge density in four-dimensions. This object is well suited to visual analysis as it is sited far enough away from a periodic boundary to make visualisation via indirect volume rendering a simple task.

Field	Possible range	Slab size
space-like plaquette	[-1.0, 1.0]	0.001953125
time-like plaquette	[-1.0, 1.0]	0.001953125
average plaquette	[-1.0, 1.0]	0.001953125
difference plaquette	[-1.0, 1.0]	0.001953125
topological charge density	[-45.0, 45.0]	0.087890625

Table 8.2: Slab sizes used for the first multi-variate case study.

8.3.2 Input fields

This study visually inspects a number of lattice fields all of which contribute to topological charge density and are linked to the presence of (anti-)instantons. Whilst it is possible to view each of these fields in isolation using the Reeb graph, it is the ability to interpret these fields in the context of others that make bivariate topology well suited to this task.

8.3.2.1 Slab size parameter

Slab sizes are based upon the expected scalar range of an input field and the desired number of intervals. Dividing plaquette inputs into $2^{10} = 1024$ intervals captures major topological structures whilst minimising overall noise. The same number of intervals are used in the topological charge density field which has a potential range of approximately $[-45.0, +45.0]$. The determined slab sizes remain constant throughout this study and are summarised in Table 8.2.

8.3.3 Approximated Reeb graphs of input fields

Each field is first evaluated using the approximated Reeb graph of the JCN at the pre-determined slab sizes — enabling the intended observable to be located in each. An anti-instanton is known to exist in the data and can be verified by cross referencing the JCN in domain layout and viewing the slab geometry of each field. In each of the following visualisations the colour map, running blue-cyan-green-yellow-red, represents the isovalue mapped to the range of the inputs.

Topological charge density The topological charge density JCN identifies the anti-instanton as branch (1) in the quantised Reeb graph (Fig. 8.11). The colour map shows the anti-instanton as a field minima indicated by blue glyphs in the spring layout. The location of the instanton can be observed by arranging the JCN in domain layout (Fig. 8.12), grouping the joint contour slabs that make up the anti-instanton branch in their barycentric positions. It is also possible to view the quantised contours captured by the JCN by rendering the joint contour slab meshes. Joint contour slabs are coloured by the isovalue of a user specified input field, with only the outer neutral isovalue layer visible in the nested structure.

Plaquette fields Structures present in the space-like (Fig. 8.13) and time-like plaquette fields (Fig. 8.14) contribute towards the (anti-)instanton structure. Interesting objects are identified

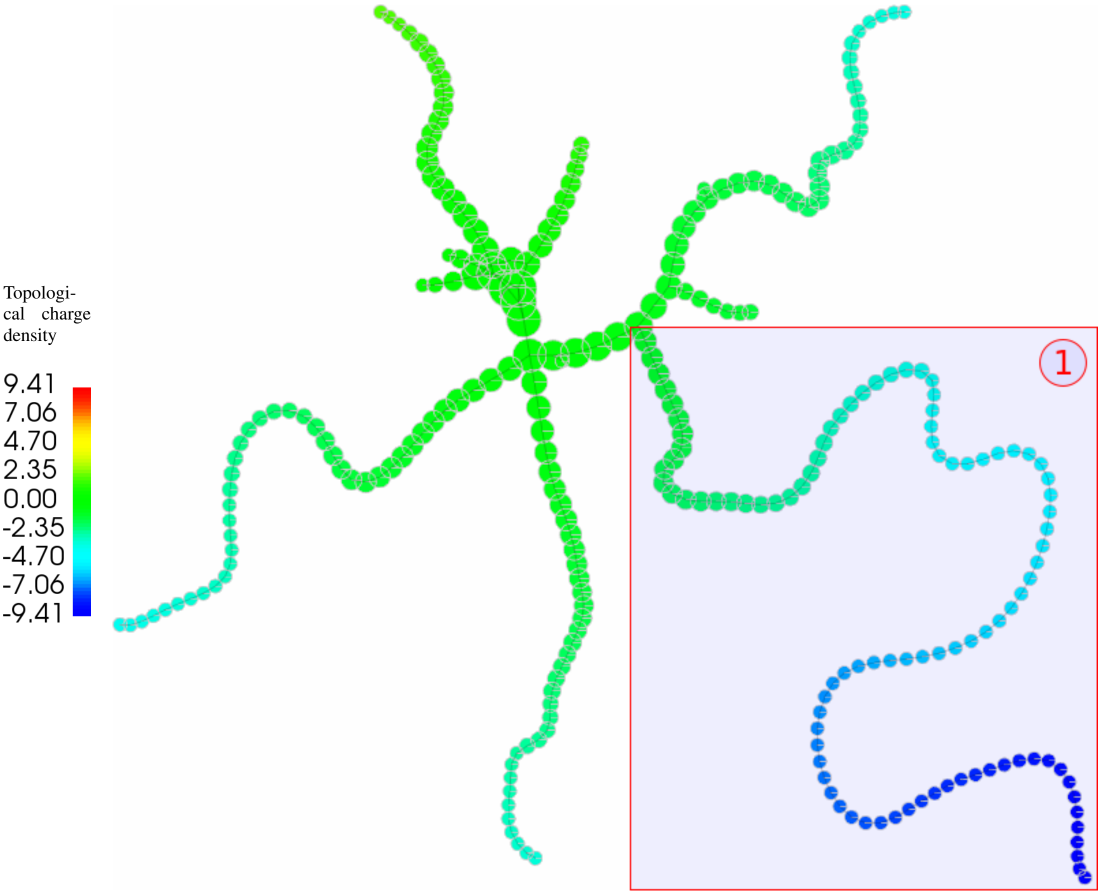


Figure 8.11: When viewed in a spring layout the anti-instanton is captured as branch (1).

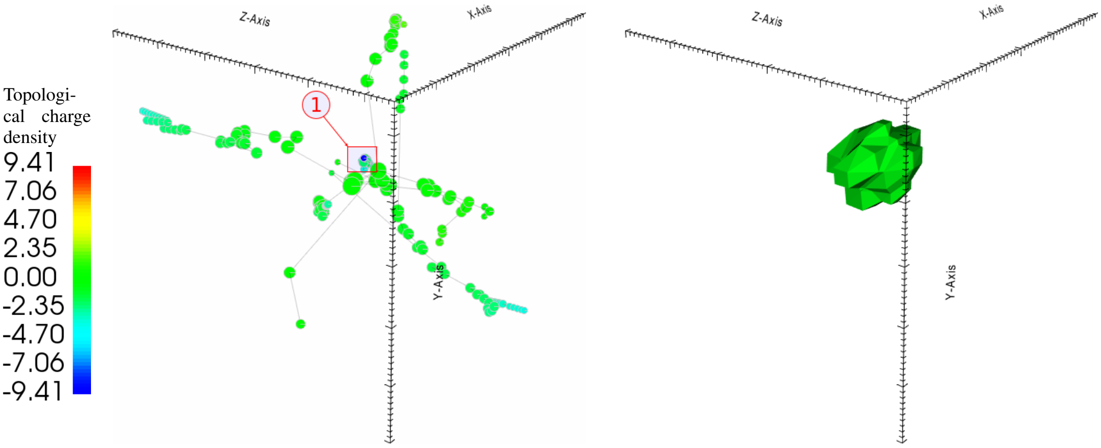
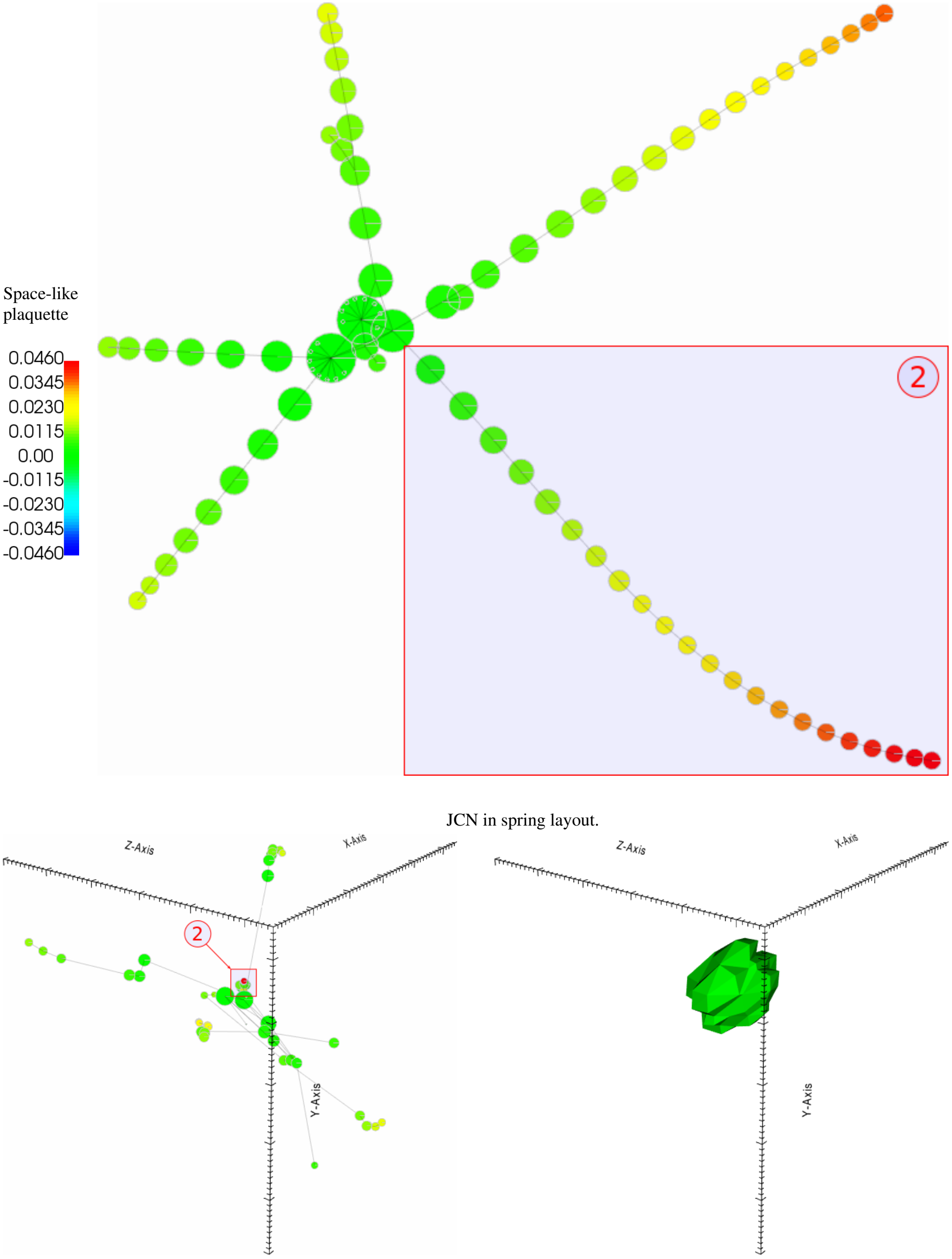


Figure 8.12: Placing the JCN in domain layout (left) enables the anti-instanton vertices (1) to be viewed in their geometric positions. The joint contour slabs represented by the anti-instanton (1) are shown on the right.

as maxima in the space-like plaquette (2) and time-like plaquette (3) fields and show a spatial correspondence in the domain layout.

Each vertex in the JCN graph represents a joint contour slab with m isovalues attached to it — in the case of univariate data $m = 1$. In domain layout, where vertex glyphs are projected to the barycentre of their joint contour slabs, glyph scale is used to reflect the volume of the slab region. Edges in the JCN represent adjacency of slabs; hence, a spring layout can be used to observe structures obscured in domain layout. Here it allows us to observe how peaks in the plaquette fields consist of multiple nested regions. The terminating vertex on a branch represents the inner core, in this case representing the peak isovalue in each plaquette field.

Plaquette derived fields Two fields derived from the space-like and time-like plaquette fields are presented in Figures 8.15 and 8.16. The average plaquette field (4) captures the anti-instanton as the field maxima due to high energy in both the space-like and time-like plaquette fields. The difference plaquette (5) shows relatively small variation in the two fields in the outer slab shells, indicated by blue glyph. However, the inner core presents higher variation remains relatively high resulting in an orange glyph. Visual inspection using domain layouts reveals that these slabs form a very dense and closely packed core. Under additional cooling it would be expected that the variation at the centre of the anti-instanton would reduce.

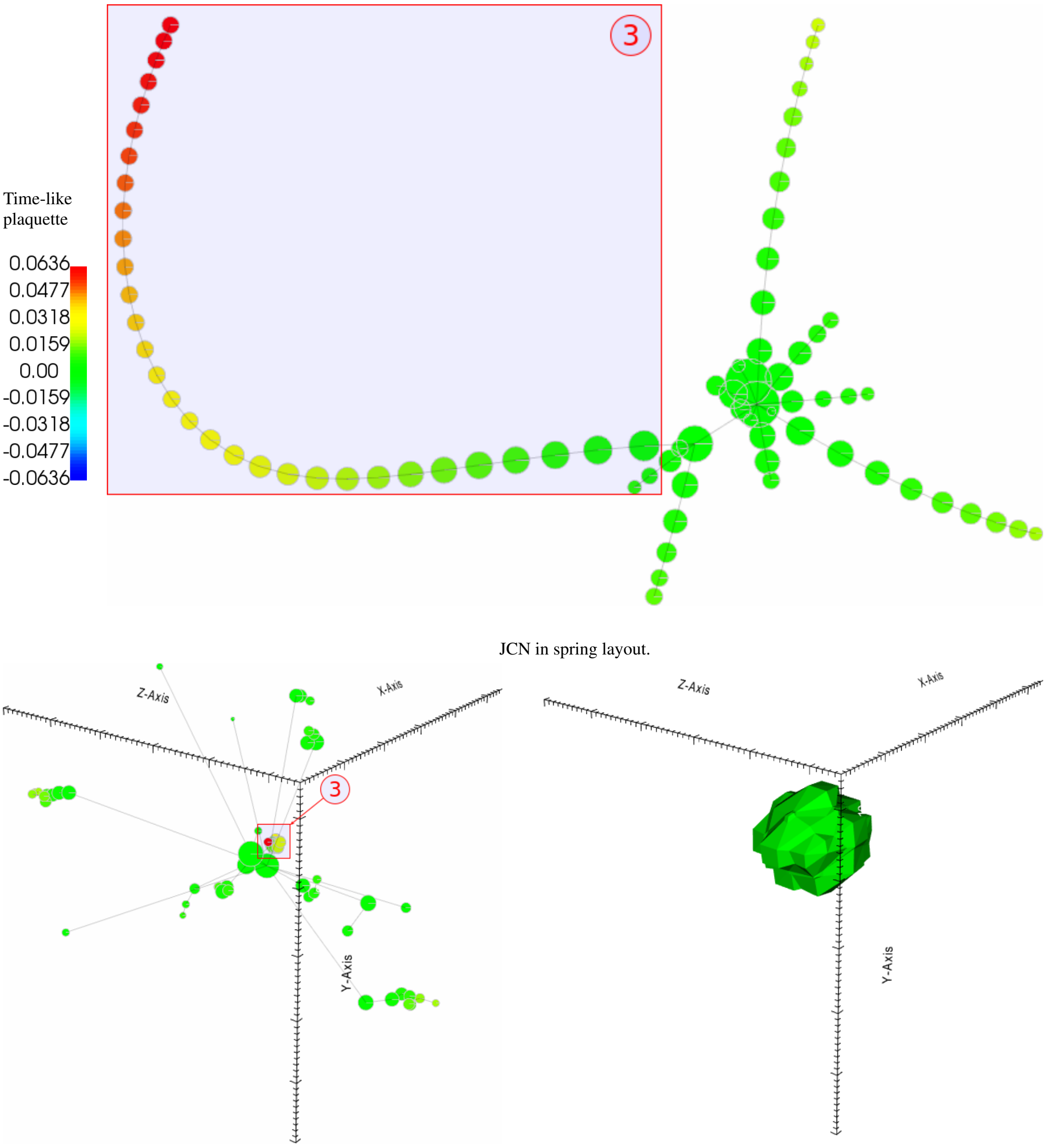


JCN in spring layout.

JCN in domain layout.

Joint contour slabs of object (2).

Figure 8.13: The anti-instanton (2) identified within the JCN of the space-like plaquette field. The anti-instanton can be identified as one of the peaks in the space-like plaquette field using the spring view (left). This can then be viewed in the context of the geometry (centre) and a joint contour slabs (right).

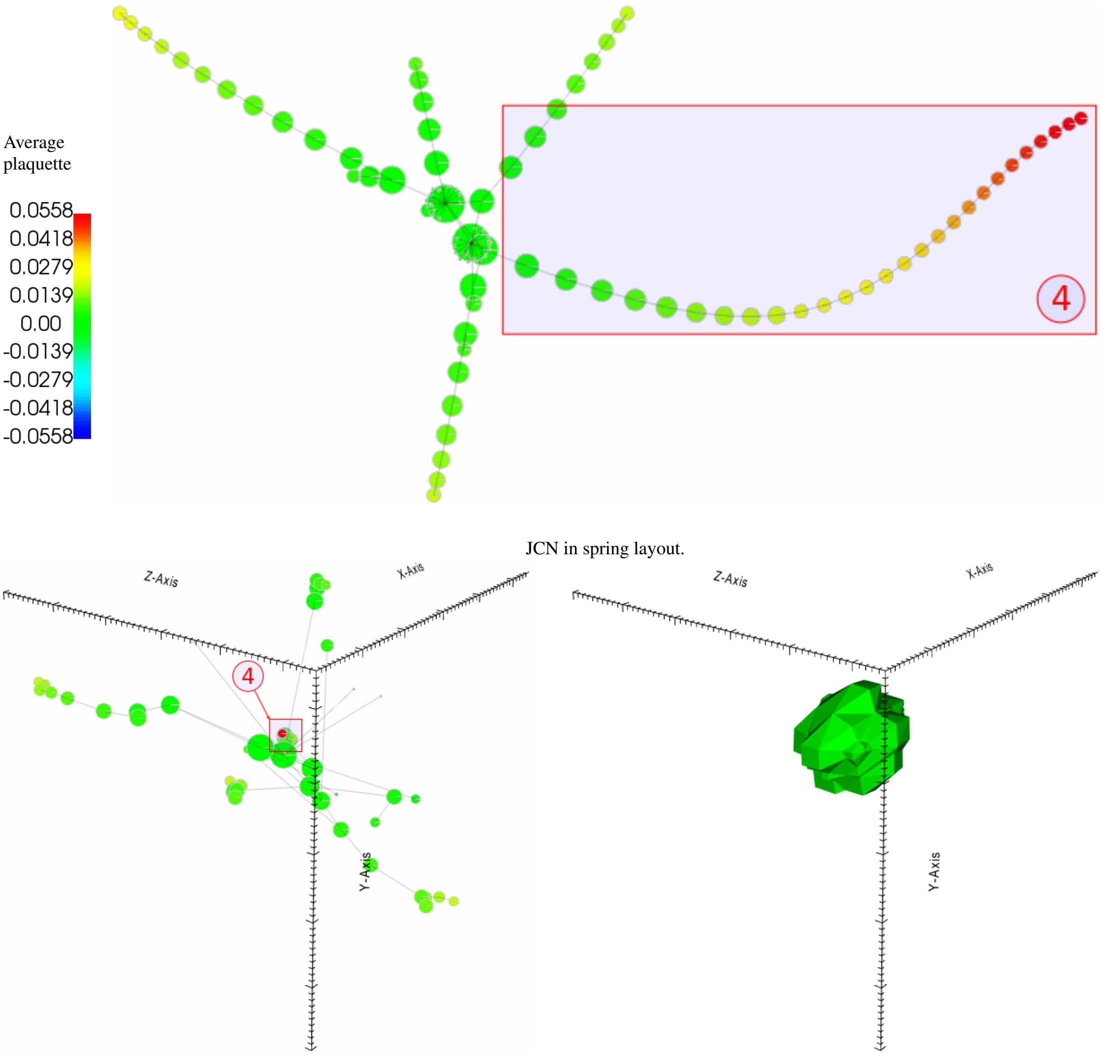


JCN in spring layout.

JCN in domain layout.

Joint contour slabs of object (3).

Figure 8.14: The anti-instanton (3) identified within the JCN of the time-like plaquette field. The anti-instanton can be identified as the major peak in the time-like plaquette field using the spring view (left). This can then be viewed in the context of the geometry (centre) and a joint contour slabs (right).

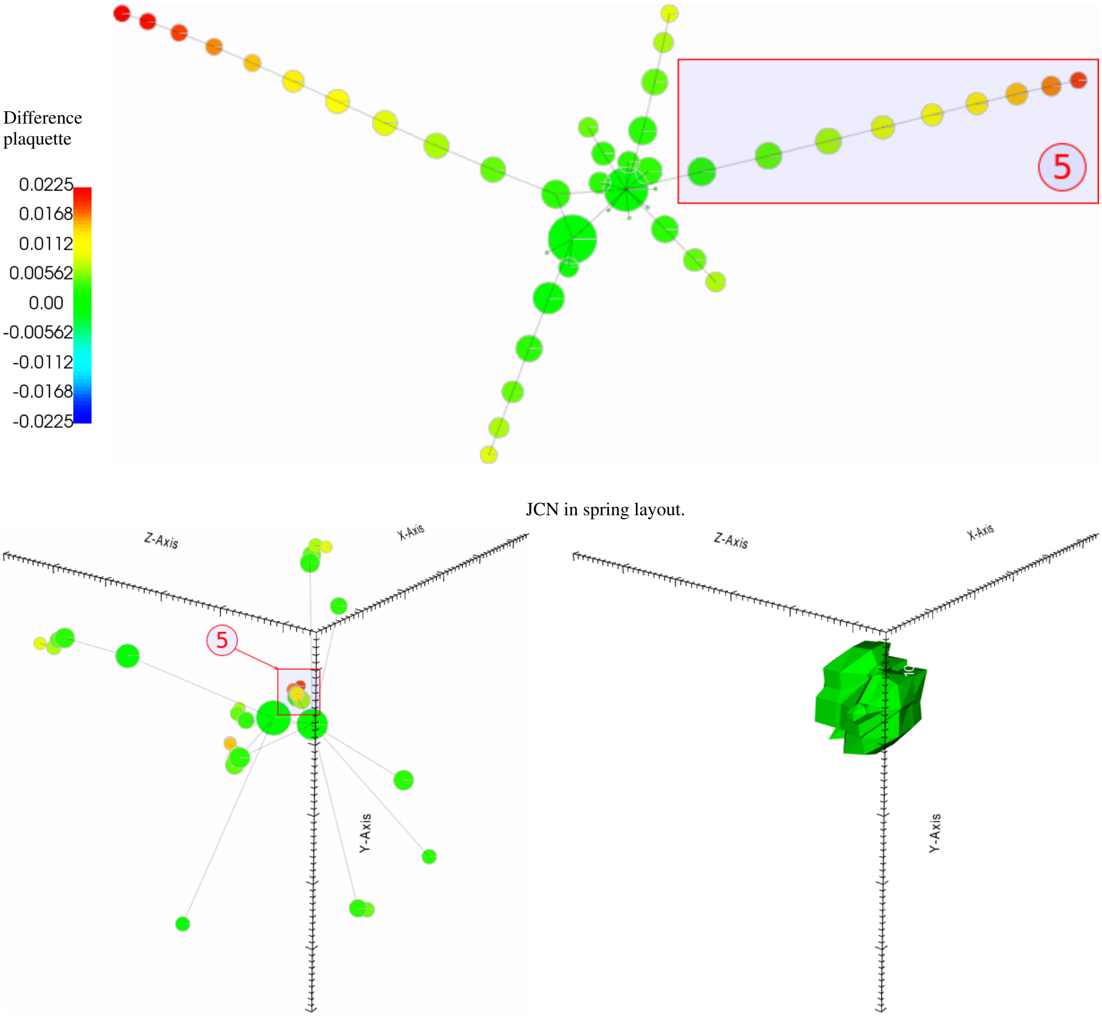


JCN in spring layout.

JCN in domain layout.

Joint contour slabs of object (4).

Figure 8.15: The anti-instanton (4) identified within the JCN of the average plaquette field. The anti-instanton appears as the major peak in the average-like plaquette field using the spring view (left). This can then be viewed in the context of the geometry (centre) and a joint contour slabs (right).



JCN in spring layout.

JCN in domain layout.

Joint contour slabs of object (5).

Figure 8.16: The anti-instanton (5) identified within the JCN of the difference plaquette field. The anti-instanton appears as one of two peaks in the difference plaquette field when observed using the spring view (left). This can then be viewed in the context of the geometry (centre) and a joint contour slabs (right).

8.3.4 Bivariate topology

The quantised Reeb graph offered by the JCN enabled us to locate and view an anti-instanton in the single lattice fields — we now explore how it can be used to view the anti-instanton in various bivariate field combinations. It is expected that some fields give more useful feedback than others; for example, we expect the *space-like and time-like plaquette* JCN and *average and difference plaquette* JCN to give encouraging results. However, we consider all combinations of the lattice fields linked to topological charge density in this study.

Many of these studies also use the Reeb skeleton, explained in greater detail in Section 7.3.2, as a means to summarise bivariate persistence in the quantised Reeb space. Throughout this study the persistence measures reflect the number of JCN vertices, or joint contour slabs, enclosed by regions captured in the Reeb skeleton. It is used in this example as a means of quantifying multivariate persistence, rather than to illustrate the effect of simplification.

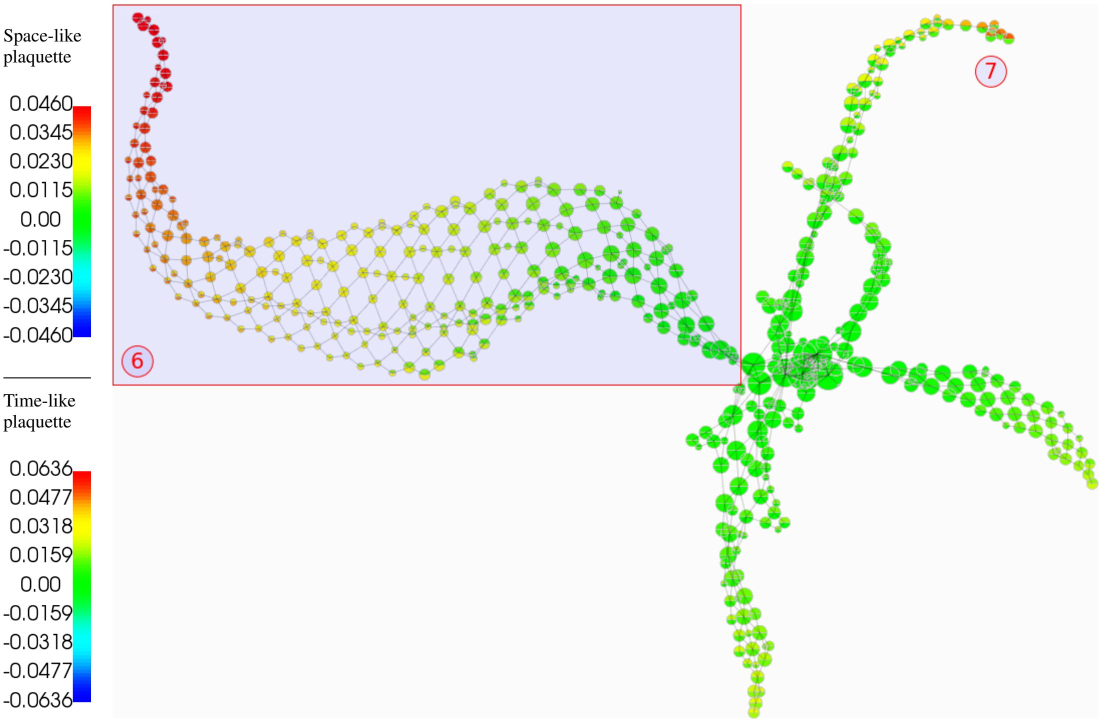


Figure 8.17: Inspecting the time-like and space-like plaquette fields using the JCN. The anti-instanton (6) appears as a maxima in both fields. A second object (7) appears in the topology as a local maxima in the space-like plaquette field but has a neutral time-like plaquette isovalue. As the object is not self-dual it cannot be classified as a potential (anti-)instanton.

Space-like and time-like plaquette fields The underlying physical theory predicts that correlations in these two fields indicate self-duality, a key signature of (anti-)instantons on the lattice. The Reeb skeleton (Fig. 8.18) isolates the self-dual object relating to the anti-instanton as the highlighted branch (6) in the discrete Reeb space. By referencing the coloured glyphs in the JCN it is possible to verify that the branch relates to maxima in both the space-like and time-like plaquette fields (Fig. 8.17). The terminating vertex, represented by a red glyph in each field, indicates a maxima surrounded by layers of decreasing isovalues in both fields. Other branches in the JCN feature a similar net-like structure; however, comparing the coloured glyphs shows that they do not fit the strict self-dual criteria required to be an (anti-)instanton. Viewing the object as joint contour slabs allows the user to validate that the selected vertices correspond to the intended observable.

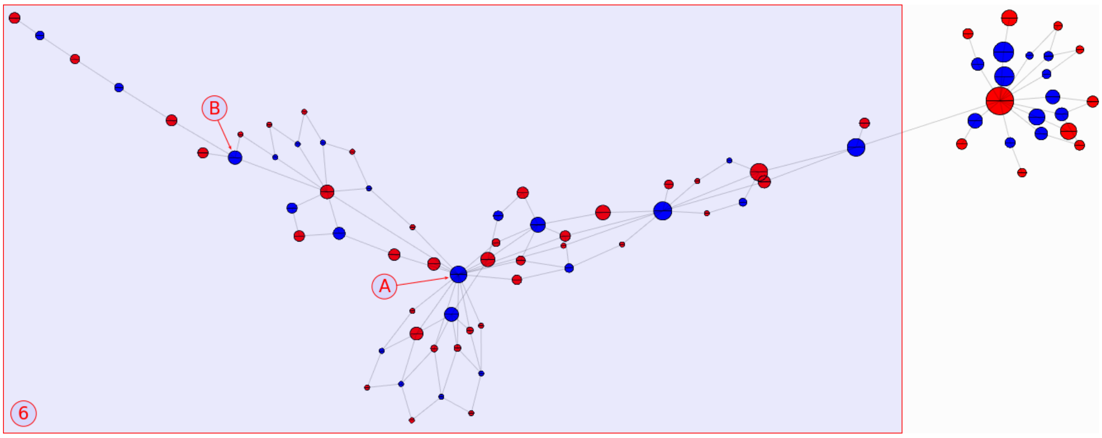


Figure 8.18: The Reeb skeleton provides an alternative view of the bivariate topology that allows selection of connected regions of the JCN. The anti-instanton is capture by sub-graph (6); this splits into smaller parts at vertices (A) and (B).

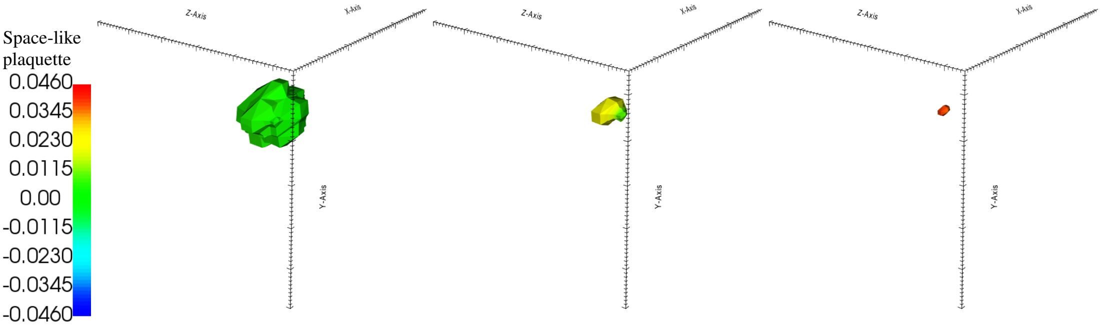


Figure 8.19: The Reeb skeleton can be used to strip away layers of the anti-instanton structure; left: viewing the whole structure (6); centre: the sub-tree from vertex (A); right: the sub-tree from vertex (B).

The Reeb skeleton provides a technique to further investigate slab structure by allowing the selection of path connected regions of the JCN. Figure 8.19 demonstrates how it is used to focus on the highlighted sub-graph where isovalues of the two fields align; although both fields approach a maximum, subtle differences remain in the isovalues of each field. The result is a sheet-like structure in the JCN as a number of small adjacent fragments are tightly packed into a dense region. The vertices of the Reeb skeleton reveal branch-like behaviour rather than a sheet, indicating joint contour slabs with stable isovalues in both fields that leads to a layered structure. Viewing this region of the Reeb skeleton as slab geometry shows that this corresponds to the core of the anti-instanton.

Average plaquette and plaquette fields This experiment considers the *average plaquette and space-like plaquette* and *average plaquette and time-like plaquette* as two separate bivariate fields. Prior to analysis it was expected that the two JCNs would appear similar in form due to the close relation of the fields. Expectations were that the anti-instanton should appear as a maxima in both the space-like and time-like plaquette fields due to self-duality. The anti-instanton can also be observed in the average plaquette field as the maxima shown in Figure 8.15.

When viewed as in the JCN (Figs. 8.20 and 8.22) the anti-instanton appears as a field maxima, marked using red glyphs, in the average plaquette field. The anti-instanton also correlates to the peak values in the space-like (Fig. 8.20) and time-like (Fig. 8.22) plaquette fields. The JCN structure appears to validate our initial prediction of similarity in the two Reeb spaces, with the anti-instanton appearing as a sheet-like structure in both graphs, correlating to well formed regions of joint contour slabs resembling the expected anti-instanton object.

Further investigation reveals interesting differing behaviour is present in the Reeb skeleton of the related two bivariate fields. It is found that the simplified view of the topology captured by the Reeb skeleton varies significantly. Figure 8.23 shows the Reeb skeleton captures the object (10) as a distinct branch in the *average plaquette and time-like plaquette* topology. However, the Reeb skeleton of the *average plaquette and space-like plaquette* disperses connected slabs of the anti-instanton structure around the central vertex (Fig. 8.21).

Examining the connectivity of the anti-instanton subtree in the context of the JCN allows predictions to be made about why this happens. In the *average plaquette and time-like plaquette* skeleton the branch attaches by a single arc to the central group of nodes. However, in the case of the *average plaquette and space-like plaquette* the branch is attached by a more complex arrangement of arcs and vertices showing a fragmented region around zero. Additionally, it is observed that the two most prominent features captured by the Reeb skeleton in the *average plaquette and space-like plaquette* relate to two other non self-dual objects on the lattice.

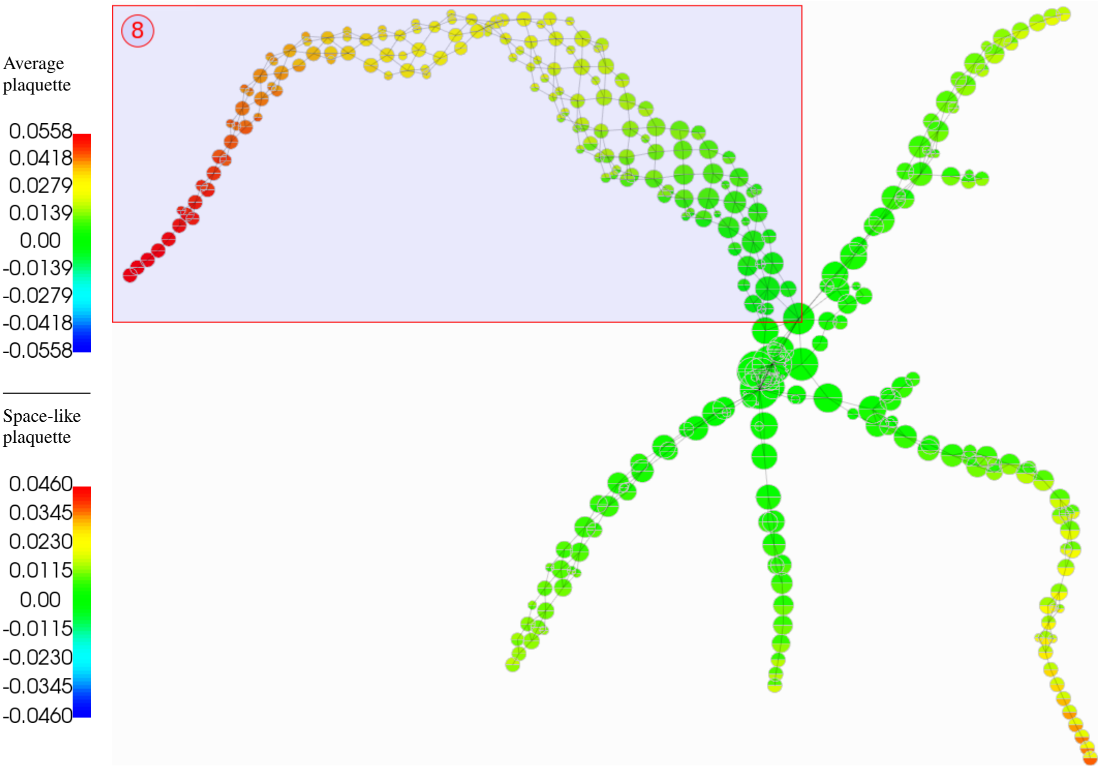


Figure 8.20: The anti-instanton (8) appears in the average plaquette / space-like plaquette as maxima in both fields.

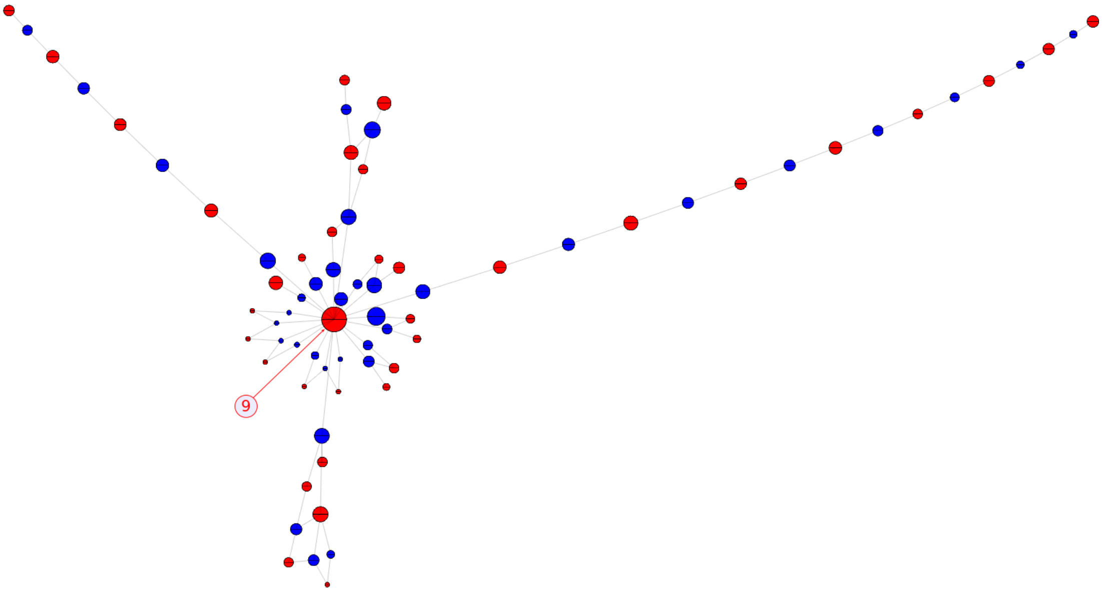


Figure 8.21: The anti-instanton appears in the average plaquette / space-like plaquette Reeb skeleton as a collection of vertices around the central highly connected vertex (9).

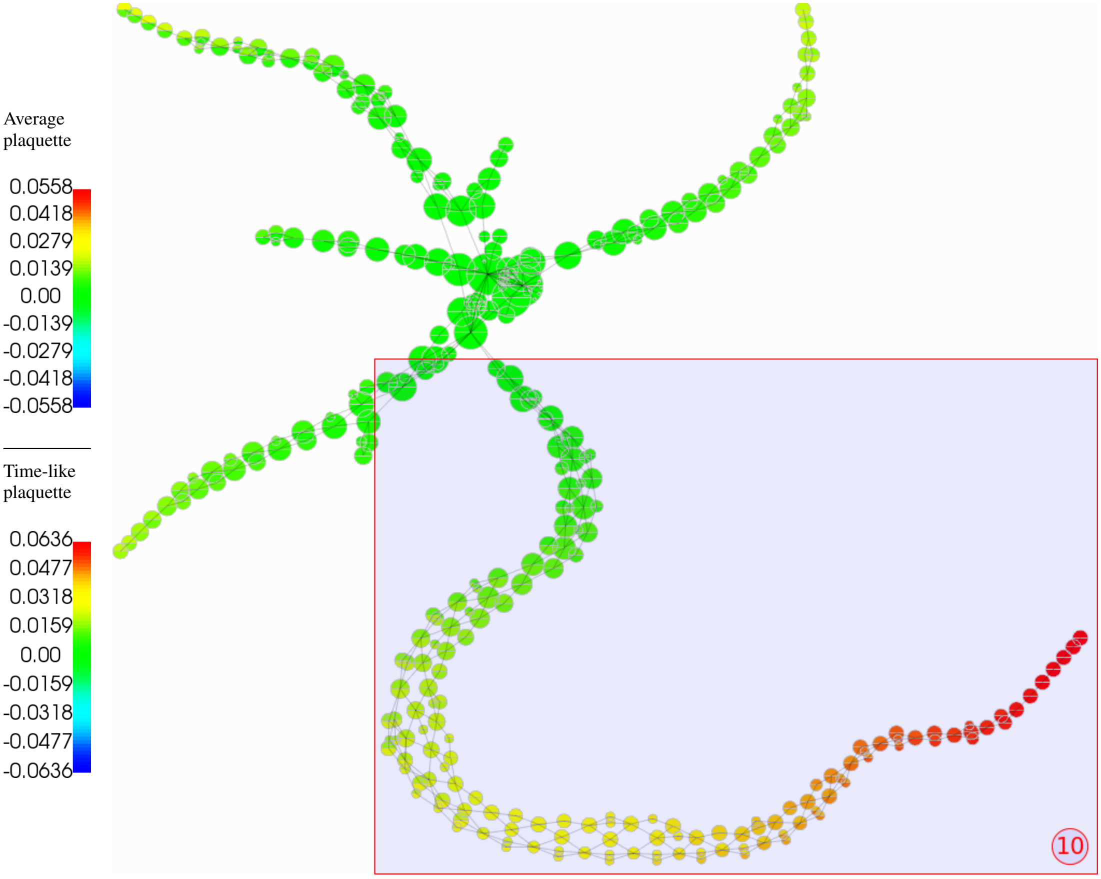


Figure 8.22: The anti-instanton (10) appears as a maxima in both the average plaquette and time-like plaquette fields.

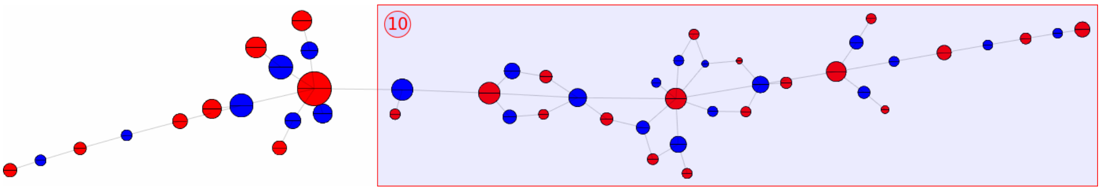


Figure 8.23: The Reeb skeleton of the average plaquette and time-like plaquette captures the anti-instanton as a distinct branch in the topology (10).

Topological charge density and average plaquette fields It is expected that this bivariate field presents a good method of capturing (anti-)instanton structure. The well defined self-dual regions predicted by the average plaquette and prominent global minima and maxima in the topological charge density lead us to believe there will be a good cross over of features in the two fields. The JCN captures the anti-instanton as a long sheet in the multi-field topology (Fig. 8.24) representing it as a multi-layered well-correlated region of space. The JCN in spring layout shows the anti-instanton with blue glyphs in the topological charge density field and red glyphs in the average plaquette field, indicating minima and maxima in the respective fields, as expected.

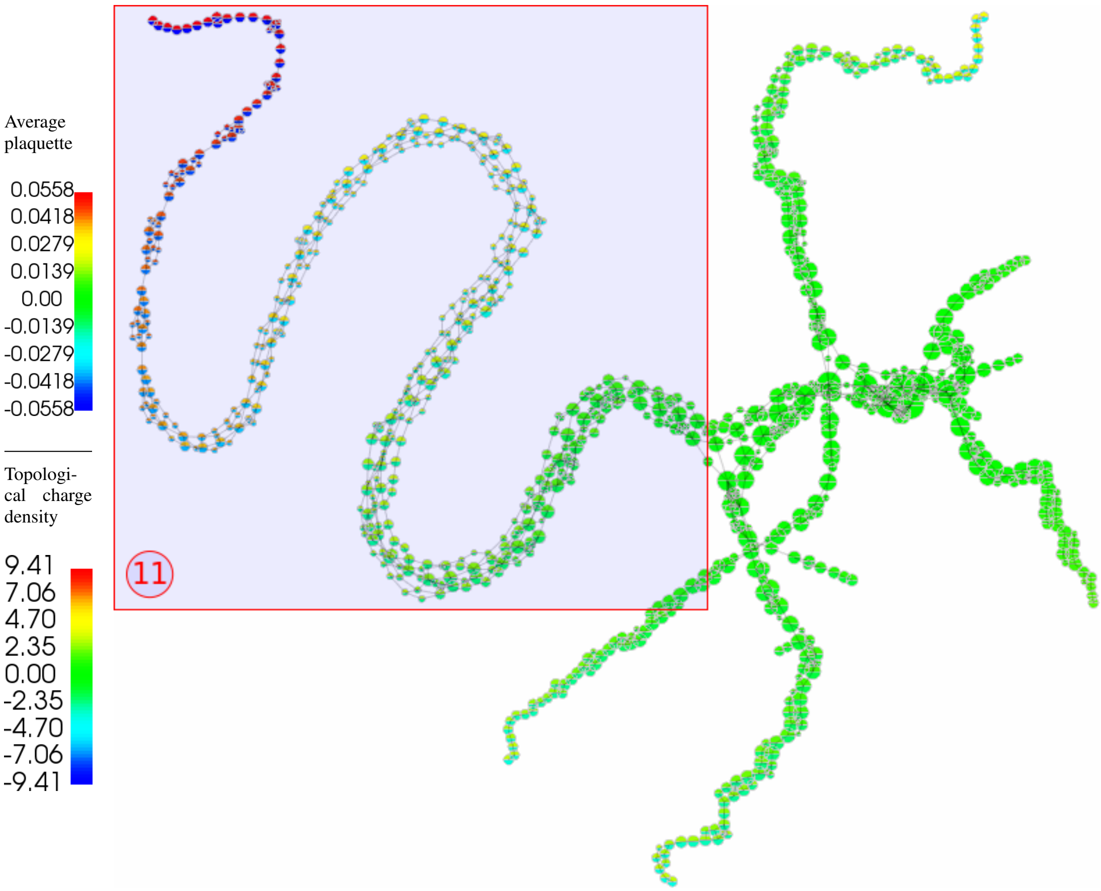


Figure 8.24: Capturing the anti-instanton (11) in the topological charge density and average plaquette fields using the JCN. The anti-instanton appears as a maxima in the average plaquette field and a minima in the topological charge density.

The Reeb skeleton (Fig. 8.25) shows some similarity to the quantised Reeb graph of the topological charge density (Fig. 8.11); a similar fork is observed where the anti-instanton separates from other local minima at vertex (12). The anti-instanton sub-graph within the Reeb skeleton contains multiple vertices, each capturing nested regions of the anti-instanton. Many smaller branches also split from the main structure at the regular (blue) vertices (13); under closer examination, alongside the JCN, these structures are seen to relate to small regions of the JCN where the bivariate topology momentarily splits into small fragments before re-merging. Cross referencing the Reeb skeleton with the JCN (Fig. 8.26) shows that the vertices relate to star-burst like features similar to those described by Schunck et al. [88]. Visual inspection of the slab geometry shows the star-burst vertices relate to well formed features that are aliased into multiple smaller slabs. Therefore, we believe this is an artefact of the discrete nature of the algorithm, rather than a feature of lattice QCD data, and would probably disappear with smaller slab size parameters.

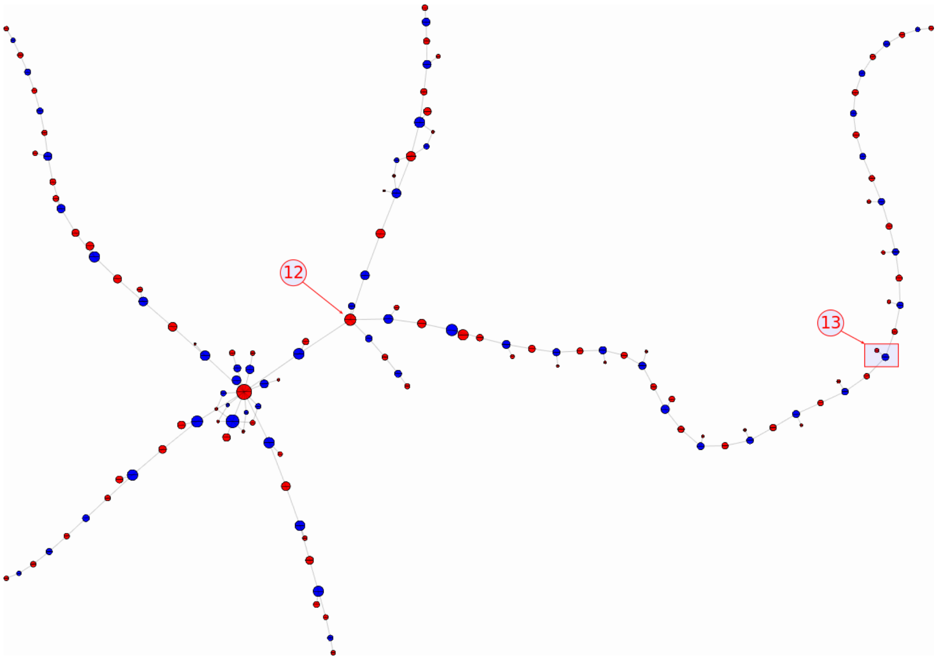


Figure 8.25: The Reeb skeleton captures the anti-instanton as a long branch in the topology, separating from the surrounding topology at vertex (12). The anti-instanton branch shows a number of micro-features along its length (13). These can be selected and examined further using the JCN.

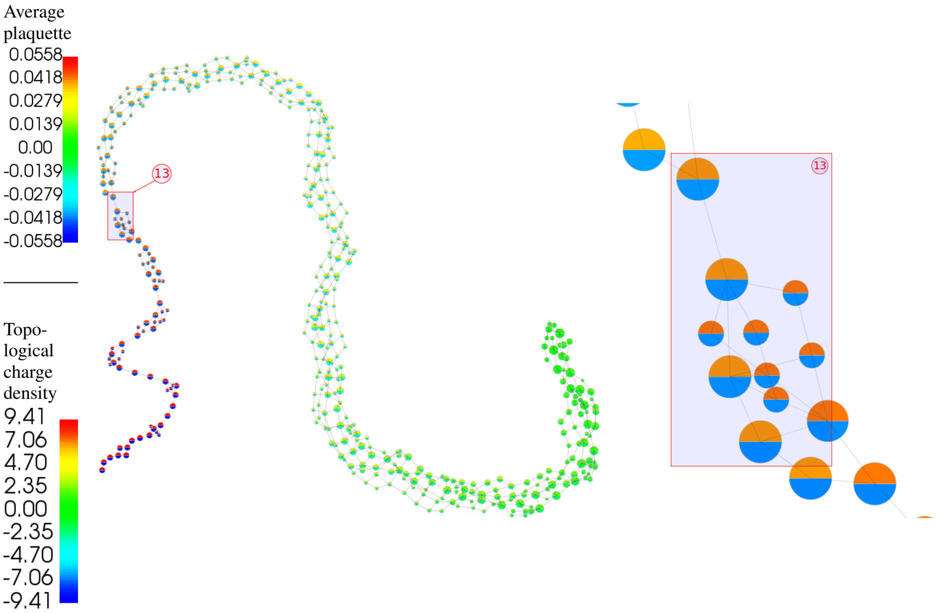


Figure 8.26: The small micro-features (13) present in the Reeb skeleton can also be examined in the JCN by focusing upon the anti-instanton sub-graph. From this view the micro-features can be seen to resemble the starburst effects also witnessed by Schunck et al. [88].

Topological charge density and plaquette fields In this experiment we compare the JCNs of the *topological charge density and space-like plaquette* and *topological charge density and time-like plaquette* bivariate fields. Previously observed variations between the two plaquette fields in an earlier study lead us to question if the same effect would be present here.

The JCN structure of the *topological charge density and space-like plaquette* bivariate topology reveals very little, besides glyph colour, to distinguish the anti-instanton from the surrounding field (Fig. 8.27). However, in the *topological charge density and time-like plaquette* field the anti-instanton immediately stands out from the rest of the data by appearing as a long sheet-like structure (Fig. 8.29). Additionally, the JCN appears to split into more distinctive branches when paired with the time-like plaquette in comparison to the *topological charge density and space-like plaquette*.

In order to gain further understanding of the JCN it is helpful to visualise it in the context of the data domain. The time-like plaquette variant captures the JCN vertices placed at the barycentre of each slab, in a cross shaped configuration within the field (Fig. 8.30), whilst the space-like plaquette presents a flatter arrangement of the vertices (Fig. 8.28). Viewing the JCN as slab structures reveals the alignment effect present in the JCN vertices is not reflected in the arrangement of the slabs. Instead, the two multi-fields capture the instanton with very little variation in geometric structure. The alignment pattern of the slab barycentres is an effect we are unable to explain and is predicted to be further evidence of an aliasing problem in the two overlapping fields.

Other differing behaviour is visible when viewing Jacobi nodes of the JCN which specify regions in the topology where the two function gradients become parallel or equal to zero. Initial predictions were that behaviour in the two bivariate fields would be approximately equivalent; however, less Jacobi nodes are present on the *topological charge density and space-like plaquette* branch (Fig.8.31 (14)) than in the *topological charge density and time-like plaquette* multi-field (Fig. 8.32 (15)). This is also noticeable in non-instanton like branches of the JCN where we have very different isovalues in the *topological charge density and time-like plaquette* field but an apparent alignment of function gradients. Whilst visually appearing similar, there appears to be a divergence in behaviour in the plaquette fields that could provide further insight into how the two plaquette fields contribute to topological charge density when observed across a range of thermodynamic control parameters, such as chemical potential μ.

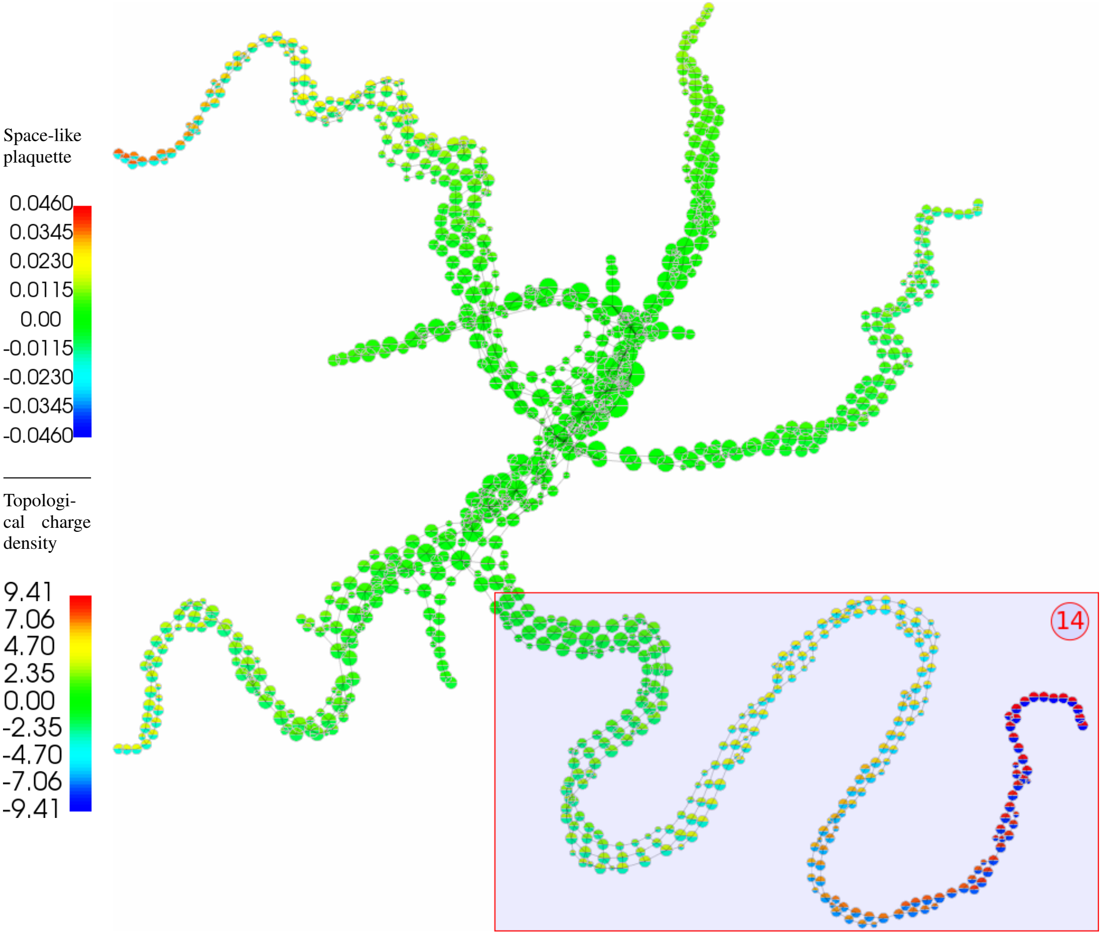


Figure 8.27: Anti-instanton structure (14) captured within the space-like plaquette and topological charge density JCN. The anti-instanton appears as a maxima in the space-like plaquette field and a minima in the topological charge density field.

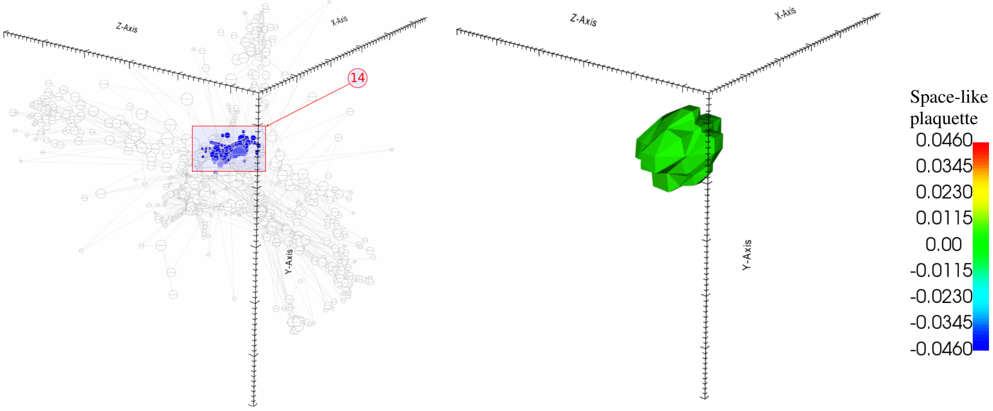


Figure 8.28: Anti-instanton structure (14) captured in domain layout (left) and as joint contour slabs (right). The blue highlighted vertices represent the geometric location of the JCN vertices making up the anti-instanton structure.

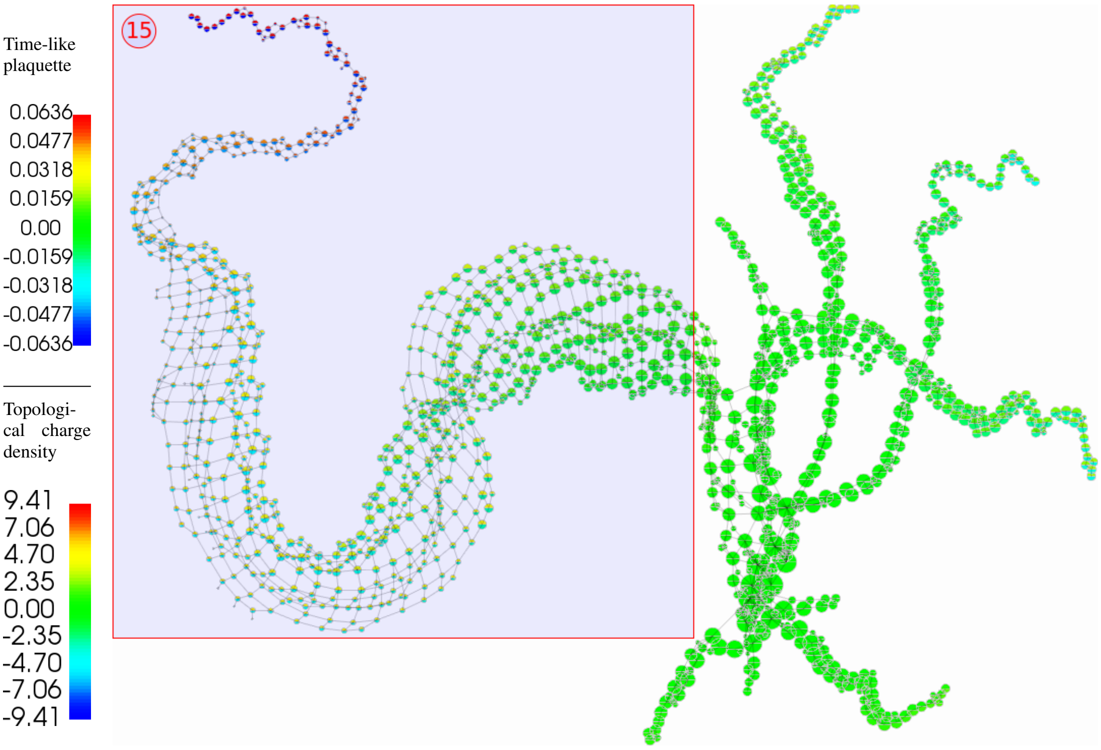


Figure 8.29: Anti-instanton structure (15) captured within the time-like plaquette and topological charge density JCN. The anti-instanton appears as a maxima in the time-like plaquette field and a minima in the topological charge density field.

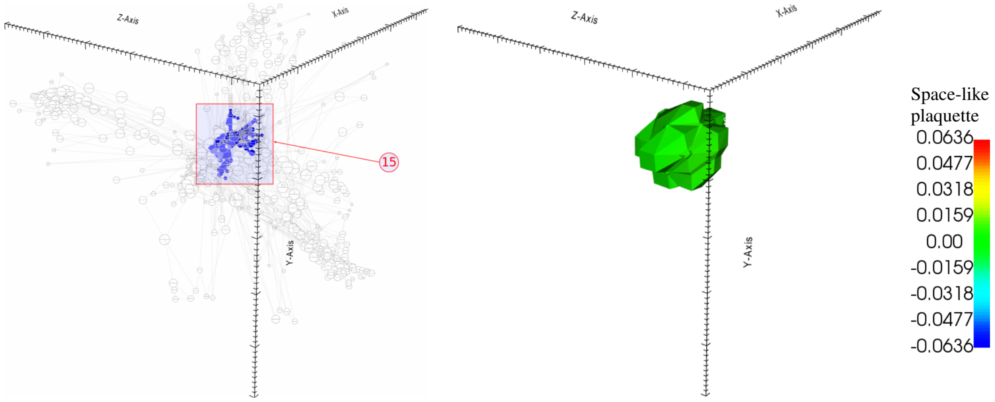


Figure 8.30: Anti-instanton structure (15) captured in domain layout (left) and as joint contour slabs (right). The blue highlighted vertices represent the geometric location of the JCN vertices making up the anti-instanton structure.

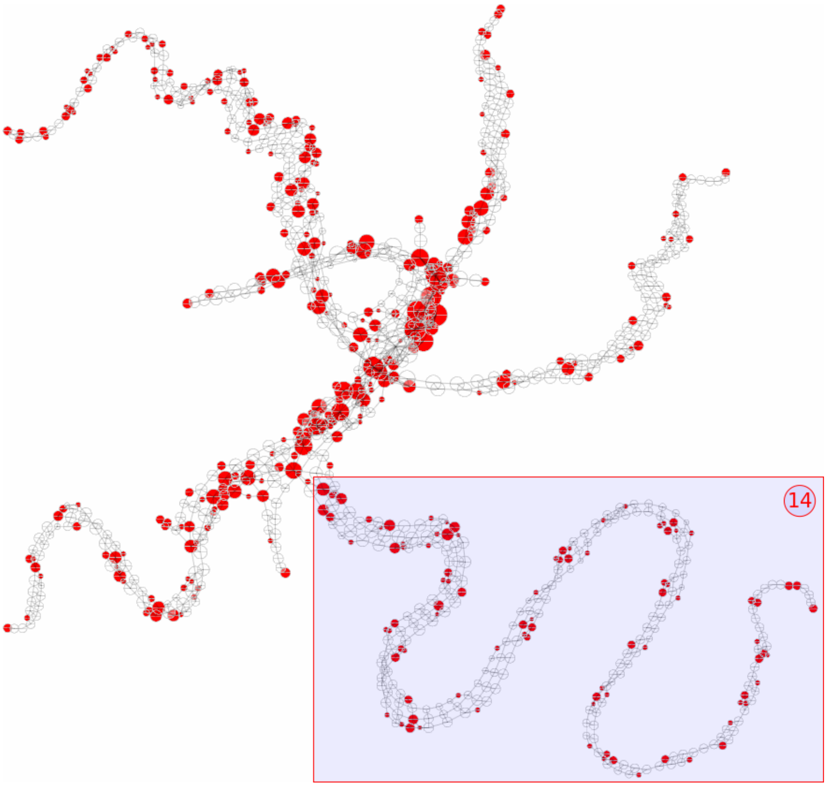


Figure 8.31: Jacobi nodes within the *topological charge density and space-like plaquette* JCN including the anti-instanton (14). The red glyphs represent Jacobi nodes.

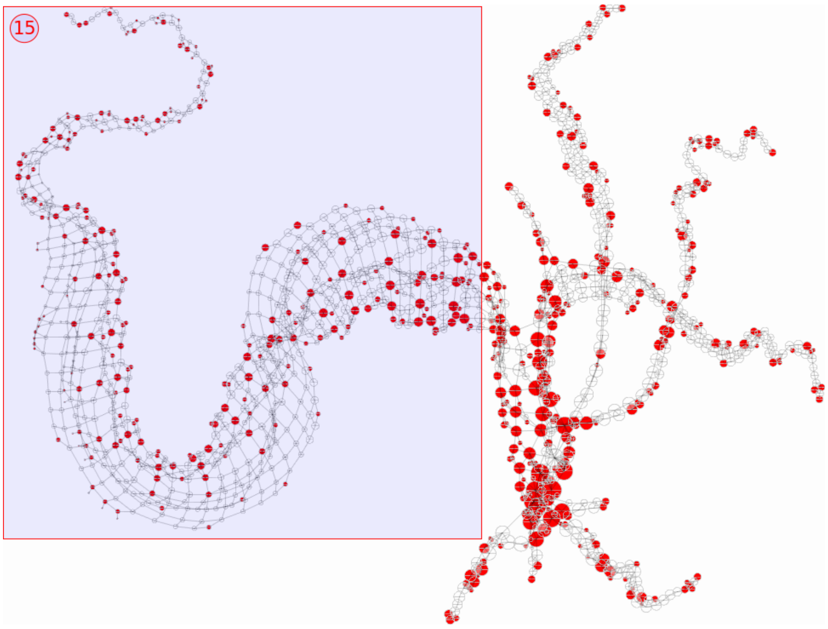


Figure 8.32: Jacobi nodes within the *topological charge density and time-like plaquette* JCN including the anti-instanton (15). The red glyphs represent Jacobi nodes.

Average plaquette and difference plaquette By their physical definitions these two fields should give a well defined method of detecting self-dual objects in bivariate topology. A self-dual object will show up as a peak in the average plaquette field and a minima in the difference plaquette field. The anti-instanton appears in the JCN as the most correlated sheet like region of the graph shown in Figure 8.33.

By viewing the associated isovalues in greater detail (Fig. 8.34) we observe that the anti-instanton does not actually appear as the global minima in the difference plaquette field but instead it seems to increase towards the inner core (17). The same observable shows as a maxima in the average plaquette field — we believe this unexpected behaviour could be explained by instabilities in the lattice that may resolve under further cooling. The same reasoning may be used to explain a region of the anti-instanton structure (18) where the average plaquette is close to neutral, but has a high difference plaquette isovalue.

In Figures 8.35 and 8.36 the Reeb skeleton is used to summarise the bivariate topology by selecting the branch of the Reeb skeleton that captures the anti-instanton. In comparison to manual selection this discards a large quantity of the JCN of the anti-instanton structure demonstrating how the outer layers of the anti-instanton are wrongly characterised as noise by the simplification process used to create the Reeb skeleton.

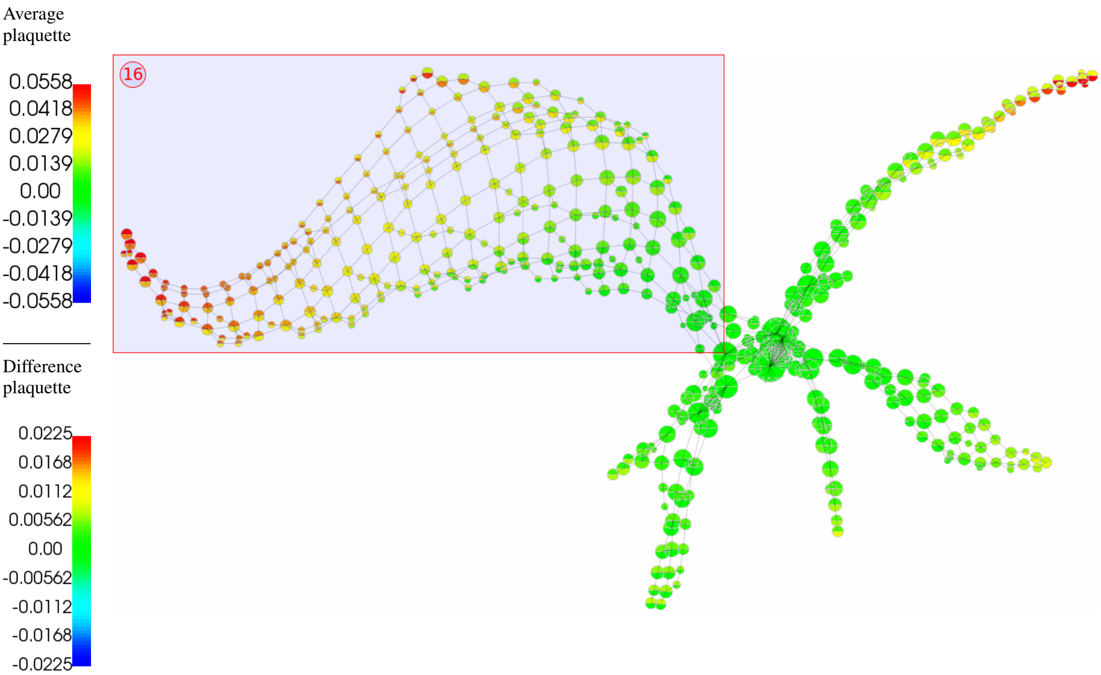


Figure 8.33: Capturing the anti-instanton structure (16) within the average plaquette and difference plaquette fields using the JCN.

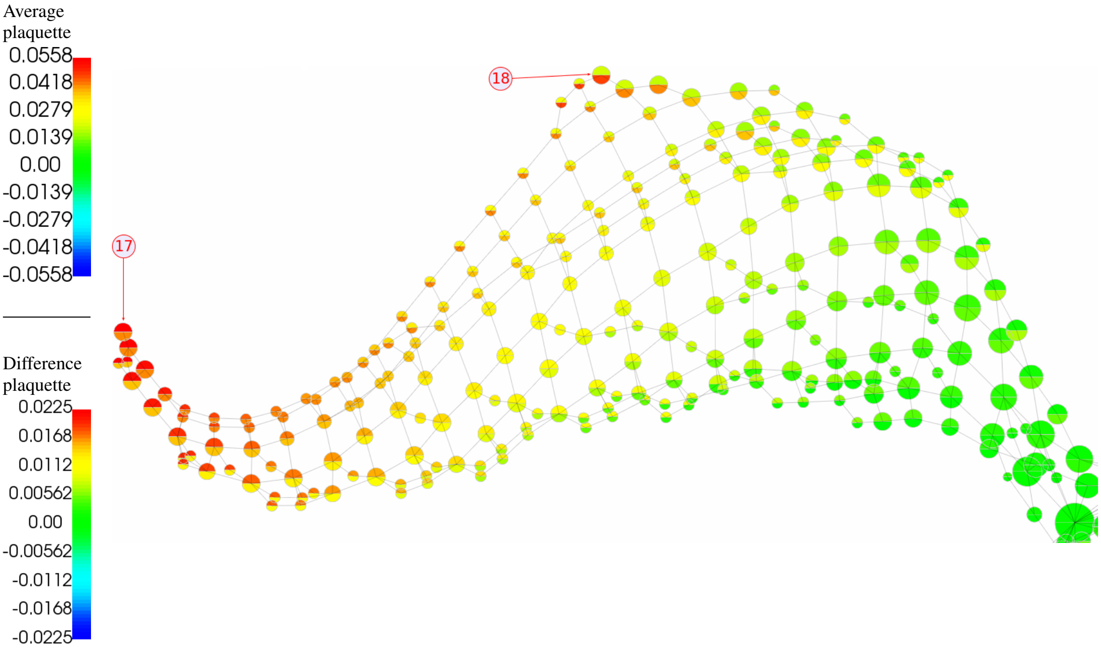


Figure 8.34: A zoomed in viewed of the anti-instanton structure. The core of the instanton (17) shows an maxima in the average plaquette field and a relatively high isovalue in the difference plaquette field. Also present is an interesting region of the anti-instanton structure (18) where the average plaquette is fairly neutral, but the difference plaquette reaches a local maxima.

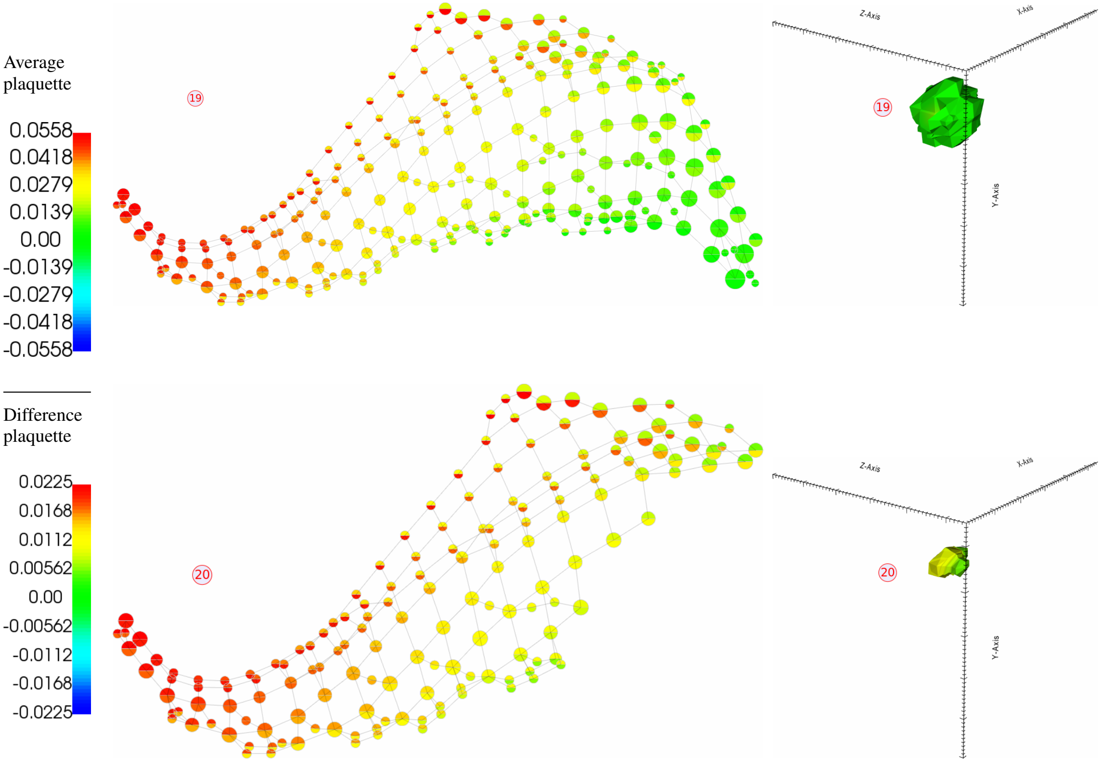


Figure 8.35: Top: anti-instanton structure (19) as determined by user selection of JCN vertices. Bottom: anti-instanton structure (20) determined by selection using the Reeb skeleton. The Reeb skeleton determines that many of the outer layers of the anti-instanton are topological noise.

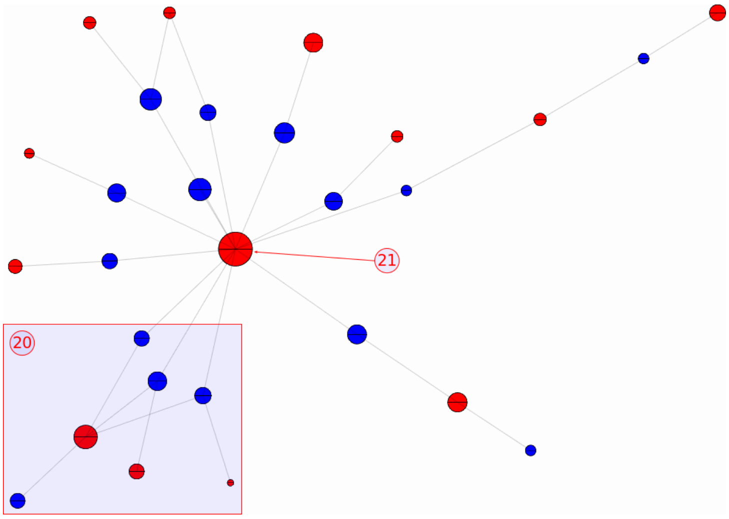


Figure 8.36: The Reeb skeleton captures the inner layers of the anti-instanton as branch (20). The outer layers are merged into the region represented by vertex (21).

Topological charge density and difference plaquette fields The JCN captures the anti-instanton as a complex sheet like structure showing a strong correlation between the two fields. Simplification using the Reeb skeleton (Fig. 8.37) captures a secondary feature in greater detail than the intended anti-instanton observable. The Reeb skeleton can be used to strip away layers of the a non-instanton-like object (Fig. 8.38), revealing an inner core (24) similar to that of the anti-instanton. Studying other JCNs reveal that this object appears throughout the case study in various fields, but is typically not present in the Reeb skeleton view of the data. Therefore, different features are sometimes determined as important by the topological algorithms in addition to the self-dual objects usually investigated by lattice QCD scientists.

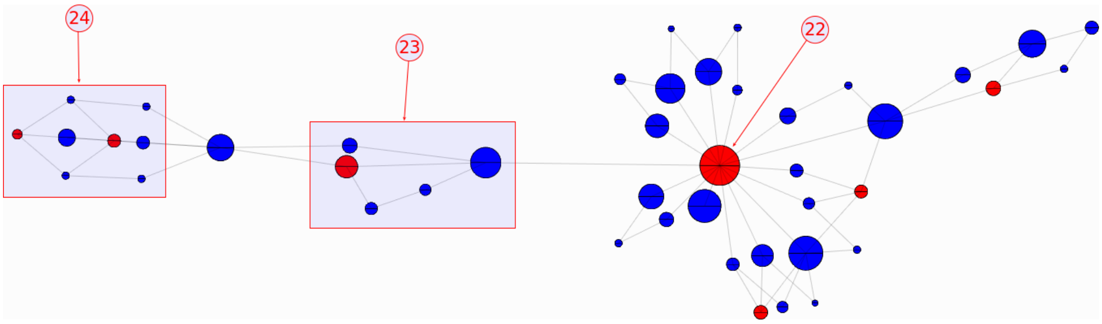


Figure 8.37: Reeb skeleton of the topological charge density and difference plaquette fields. The anti-instanton structure is distributed in the vertices around vertex (22). A secondary object is captured in detail by the Reeb skeleton; an outer region (23) surrounds an inner core (24).

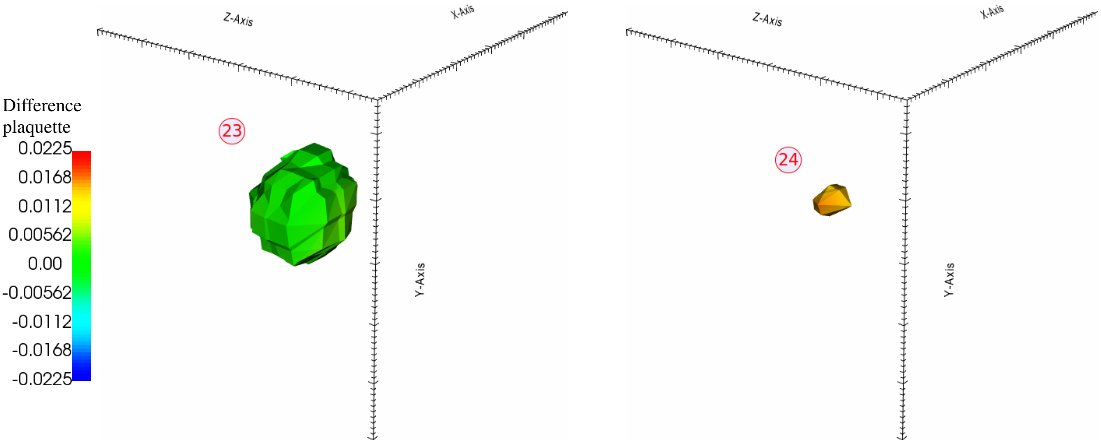


Figure 8.38: The object captured in the Reeb skeleton can be examined in greater detail by selecting groups of vertices. Vertex group (23) captures the outer structure and the inner core is captured by vertex group (24). The relatively high isovalues in the difference plaquette field mean that this object is not self-dual — hence, it cannot be a potential (anti-)instanton.

Analysis of the object, observing the relative isovalues in the two fields as shown in Figure 8.39, reveals that the object has a neutral topological charge density. When viewed in the difference plaquette field a maxima is present at the terminus of the branch, indicated by a red glyph, leading to the assertion that the object is not self-dual. Further inspection by cross referring the two plaquette fields contributing to the difference plaquette field confirms that this is because the object is a global minima in the time-like plaquette field and a local maxima in the space-like plaquette field.

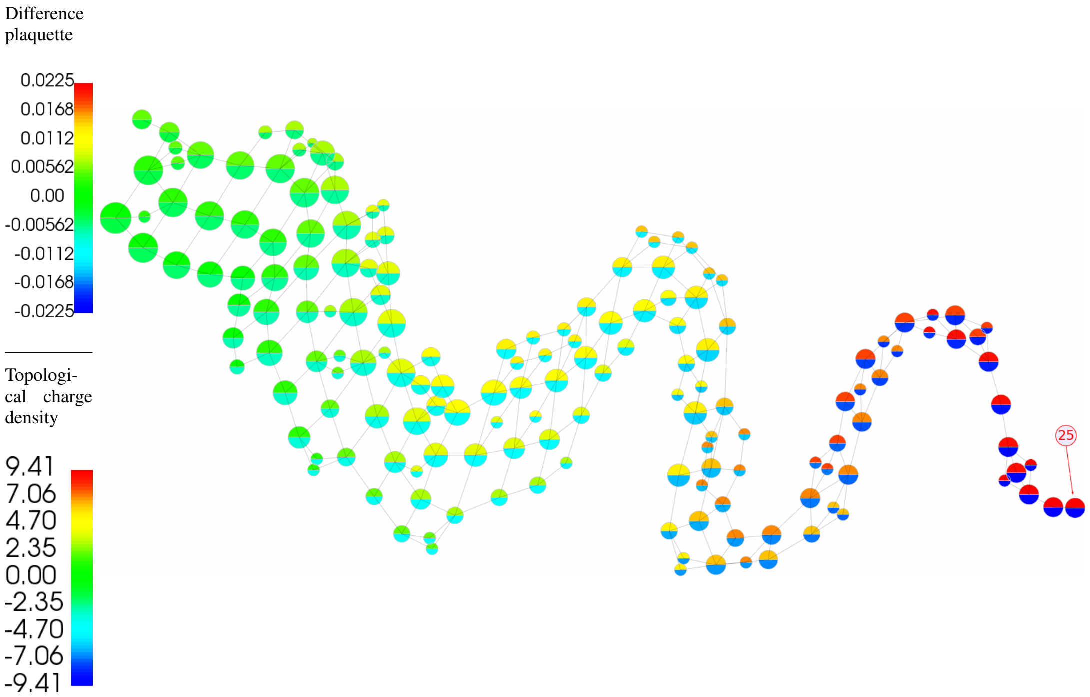


Figure 8.39: The branch relating to the prominent topological object detected in the Reeb skeleton, where vertex (25) represents the inner core of the object. By examining the isovalues in the two fields it is possible to determine that the object is non-self-dual; whilst the topological charge density at the core is a global minima, this coincides with the difference plaquette reaching a peak. For the object to be identified as an anti-instanton a neutral value would be expected in the difference plaquette.

Difference plaquette and plaquette fields Visual analysis of these lattice fields gave very similar results to other combinations. The Reeb skeleton and JCN both captured the anti-instanton as well defined sheet-like branch of the topology. However, we conclude that this combination of fields reveals little additional information that cannot be obtained using other fields. However, it may be of more interest as thermodynamic control parameters are varied.

8.4 Summary

In this chapter we have shown how the JCN can be used as a method for comparing two related variables defined on the same set of sample points. By their physical definitions many fields in lattice QCD are closely linked; using the JCN allows the relationship to be studied at a level of detail previously beyond the reach of domain scientists. When used to analyse two input fields defined on the same 3D domain the JCN evaluates to a quantised representation of the Reeb space.

A summary of the contributions of this chapter is as follows:

- We investigated how the slab size parameter affects the ability to capture features of topology and impacts the complexity of computations.
- A potential divergence in behaviour was witnessed when using the JCN to examine the space-like and time-like plaquette fields in the context of other lattice fields.
- We showed how multivariate glyphs present a simplistic method for determining if a feature is self-dual.
- Observations suggested that some fields combinations were more effective at determining the presence of potential (anti-)instantons.

In Chapter 9 we build upon the knowledge gained in this case study and that of Chapter 7 to use bivariate topology in an ensemble setting. In order to use the JCN as part of an ensemble study the analysis must be automated due to the quantity of data involved. We are able to use information gathered from visual analysis of the JCN in order to design suitable approaches to the automation process.

Chapter 9

Analysis of lattice QCD ensemble data using the JCN.

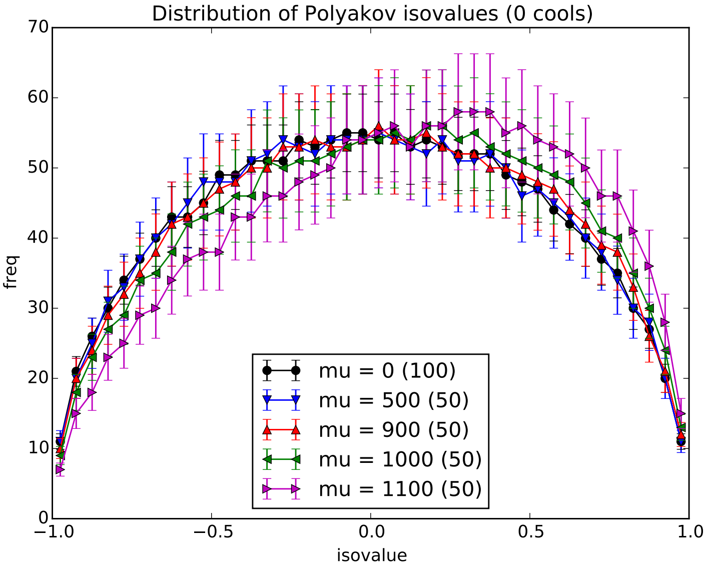
Contents

In this chapter we use multivariate topology to analyse large quantities of lattice QCD configurations in order to observe emergent behaviour as simulation parameters are varied. In order to carry out the process, the data is first examined by forming links between the graph-like structure of the JCN with a physical segmentation of the multi-field. We then examine a number of quantitative properties extracted from the topology to be compared using multiple data sets with different simulation parameters. We build upon knowledge gathered as part of two single configuration case studies, detailed in Chapters 7 and 8, to help design this experiment.

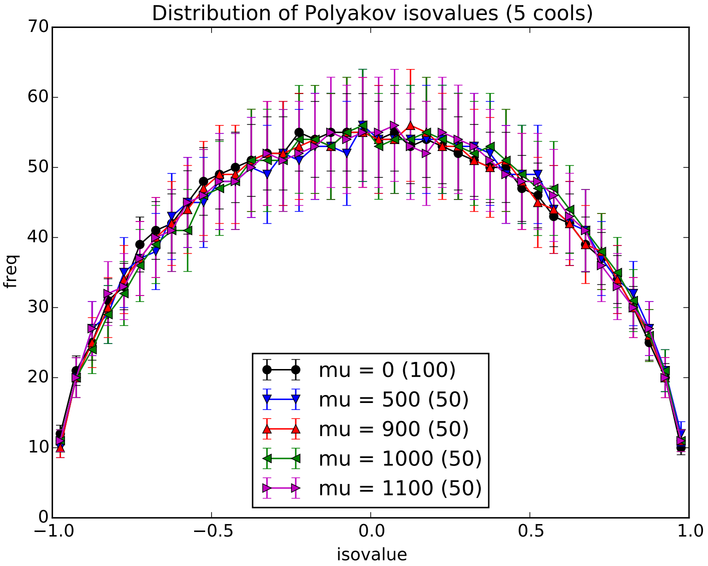
This chapter is based upon research presented at Topology In Visualisation (TopoInVis) 2017, which is currently under review for publication as a book chapter. The structure of this chapter is as follows. We begin by introducing the intended research outcomes of this case study in Section 9.1. We then perform a brief visual analysis of the data studied as part of an ensemble wide study in Section 9.2 enabling us to orientate ourselves with features of the data. The main results are presented in Section 9.3, where we perform a quantitative analysis of hundreds of configurations across the chemical potential range on two separate lattices.

9.1 Research questions

Each of these experiments are conducted as part of multi-ensemble studies [167]; in our particular work each ensemble is generated with a differing level of chemical potential (μ). This takes the form of computing around fifty separate JCNs for each ensemble; measures from



Uncooled data.



Data cooled for 5 iterations of the algorithm.

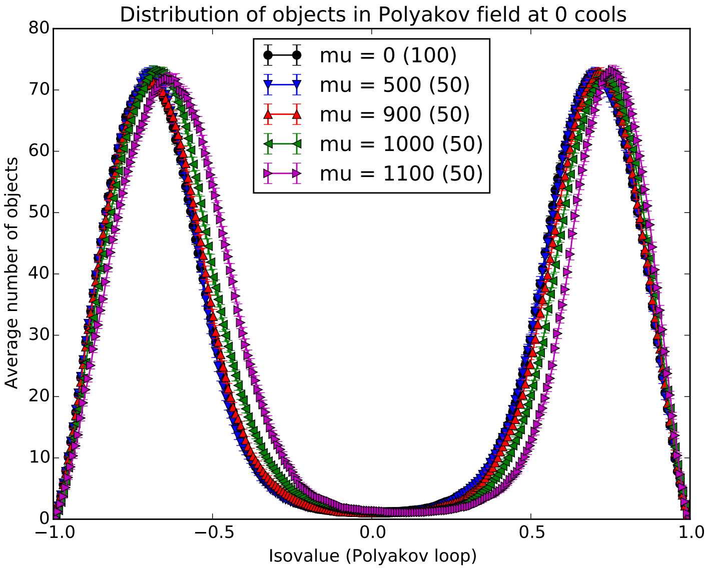
Figure 9.1: The distribution of *raw isovalues* in the Polyakov loop field plotted for multiple ensembles. In the uncooled data a lack of symmetry can be observed for high chemical potentials (μ). By 5 cools the graph appears to be roughly symmetrical.

the JCN are then computed and averaged to produce an ensemble average. Each configuration within an ensemble is a lattice with preset dimensions; typically these are expressed in the form $dim_s^3 \times dim_t$ where dim_s and dim_t represents the lattice extents in the spatio-temporal dimensions. Domain scientists generally present variations due to chemical potential as a histogram or line graph to look for signals of phase changes in the simulated quark-gluon matter.

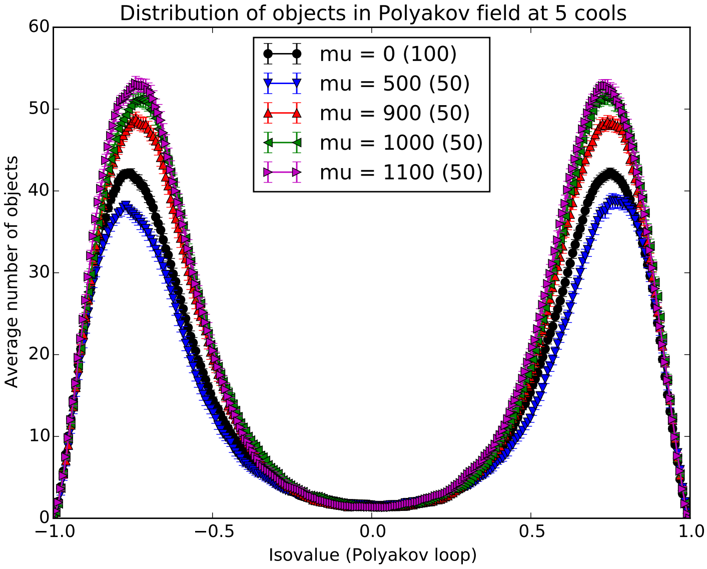
Simple analysis of the Polyakov loop observable by plotting the distribution of isovalues across multiple ensembles (Fig. 9.1), combined with a study of the distribution of objects identified using the Reeb graph (Fig. 9.2) suggest that the Polyakov loop observable may lose its ability to identify de-confinement under cooling. We are interested to see if this phenomena can be identified using multivariate topology by comparing the Polyakov loop lattice observable at neighbouring cooling iterations. From this it is hoped that the JCN can be used to provide additional information about lattice QCD simulations that are currently beyond the reach of domain scientists.

We ask the following questions, focusing on the Polyakov loop under cooling:

- Does the JCN allow us to question the relationship between objects with positive and negative isovalues in the Polyakov loop field?
- Can we use multivariate persistence measures to predict properties of the objects in the Polyakov loop field?
- By using the JCN can we learn anything regarding a lattice based noise reduction technique known as cooling?



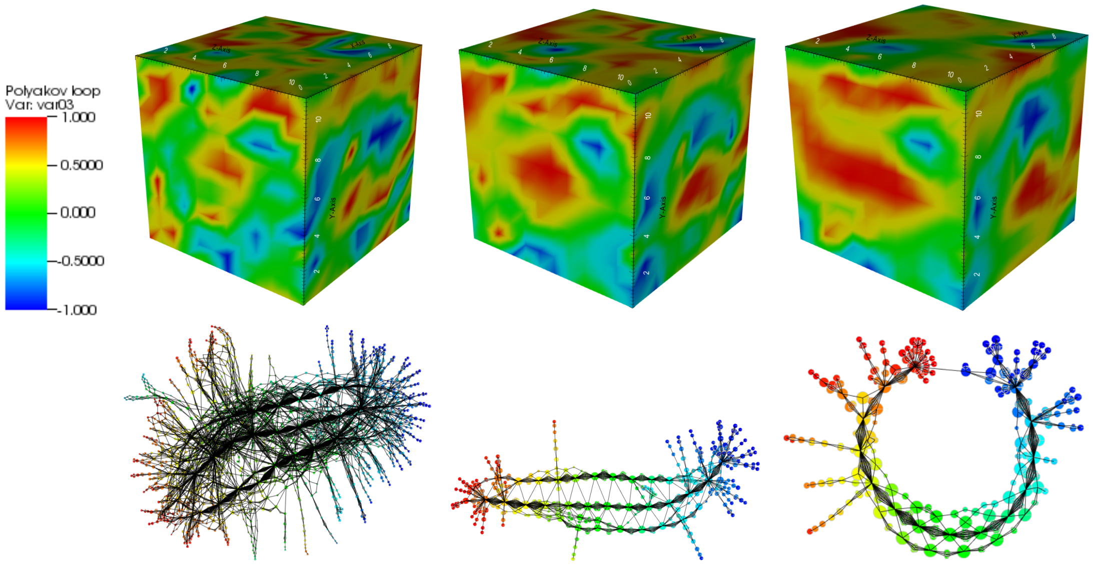
Uncooled data.



Data cooled for 5 iterations of the algorithm.

Figure 9.2: The distribution of *objects* in the Polyakov loop field plotted for multiple ensembles. For uncooled data the graph is unsymmetrical, hinting at possible deconfinement at high μ. By 5 cools the graph is much closer to being symmetrical.

9.2 Visual analysis of lattice QCD data



4 and 5 cools. 9 and 10 cools. 14 and 15 cools.

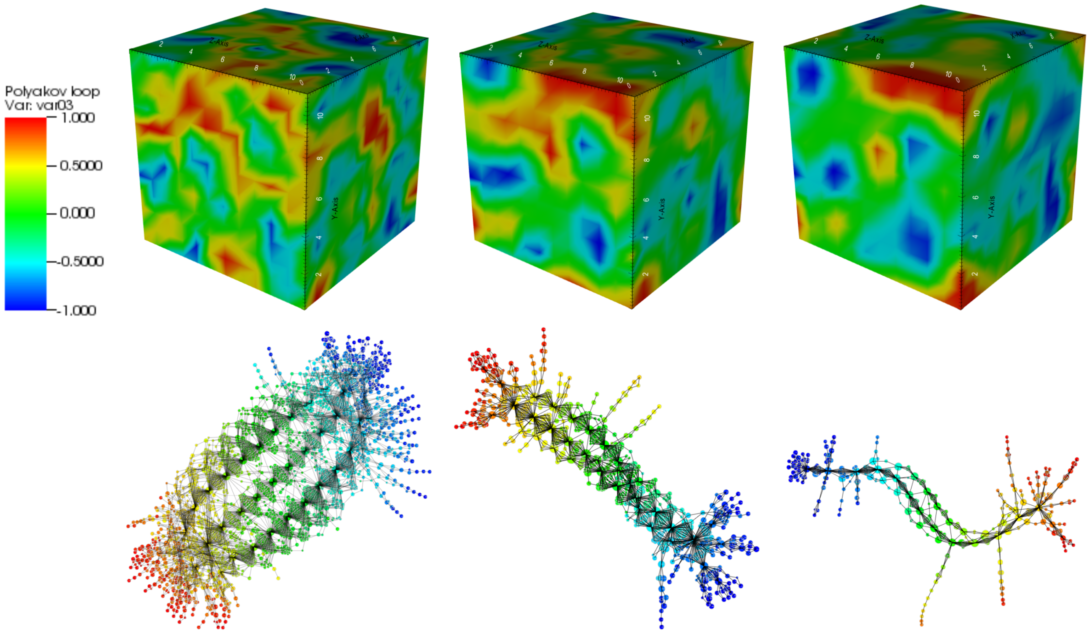
Figure 9.3: Visualising the slab geometry and JCN of Polyakov data from an $12^3 \times 16$ ensemble with chemical potential $\mu = 0.0$.

The Polyakov loop data (Figs. 9.3, 9.4, 9.5) appears as blue regions of negative isovalues and red regions of positive isovalues separated by large regions of near-zero value in green. Under cooling the slab geometry of the JCN reveals that these regions gradually merge into increasingly large topological objects.

The JCN captures this behaviour in a ladder-like structure. Each vertex represents a joint contour slab associated with pair of isovalues, with adjacency shown by edges. Individual objects in the field branch off the core structure at highly connected vertices with positive or negative isovalues. Under cooling the ladder-like structure becomes increasingly simplified and objects tend to bunch up and branch off fewer vertices, each identifying a highly connected region of the topology. For the purposes of automation and analysis it is desirable to summarise this information in a compact set of topological invariants.

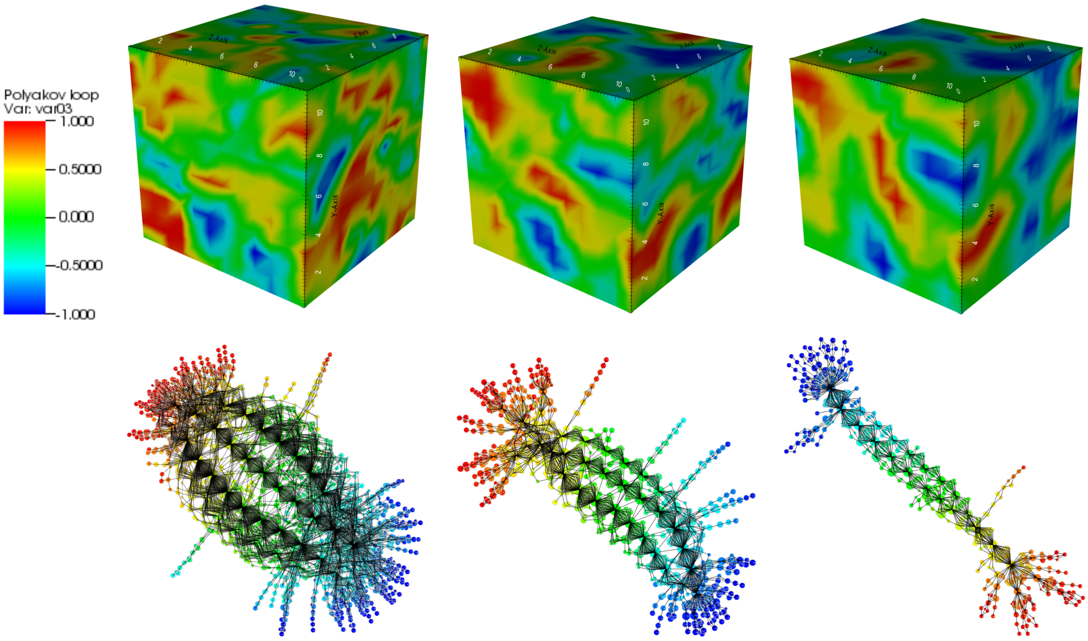
An unexpected feature captured by the JCN (Fig. 9.3) is the relationship between regions of the Polyakov loop field with positive or negative value. At higher levels of cooling positive and negative regions become direct neighbours with no near-zero region in between. The JCN indicates this by the edge between the blue and red vertices at 14 and 15 cools.

Visual analysis of the slab geometry at higher levels of chemical potential (Fig. 9.4 and Fig. 9.5) suggests an apparent change in the distribution of objects in the Polyakov field. The dominance of regions with positive isovalues at $\mu = 0.0$, seems to have been replaced by a bias towards negative regions when $\mu = 0.5$. However, at $\mu = 0.9$ the distribution looks to be approximately equivalent. Examining the JCN for each configuration shows less conclusive evidence for any bias, instead suggesting that the structures look largely similar regardless of chemical potential. It should be noted that these visual examples are taken randomly from a set of hundreds of configurations; hence, may not give an accurate summary of an entire ensemble.



4 and 5 cools. 9 and 10 cools. 14 and 15 cools.

Figure 9.4: Visualising the slab geometry and JCN of Polyakov data from an $12^3 \times 16$ ensemble with chemical potential $\mu = 0.5$.



4 and 5 cools. 9 and 10 cools. 14 and 15 cools.

Figure 9.5: Visualising the slab geometry and JCN of Polyakov data from an $12^3 \times 16$ ensemble with chemical potential $\mu = 0.9$.

9.3 Quantitative analysis of lattice QCD data using the JCN

Below we present promising results gathered by analysing different aspects of the JCN in an ensemble setting. The experiments were carried out on a Dell cluster made up of four homogenous compute nodes each with access to 16GB RAM and a separate dedicated front-end node. Each compute node contains four AMD Opteron 6376 CPUs, each with eight physical cores (16 logical cores) running 64-bit Debian 8.6. The software is built using VTK 6.1 [160] with extensions from the Multi-field Extension of Topological Analysis (META) project [163] in C++11. Quantisation parameters of the JCN were chosen to best suit the available resources in terms of memory and number of parallel processes.

9.3.1 JCN vertex count

Each JCN vertex represents a single joint contour slab in the Reeb space; hence, it can be used to count the number of unique joint contour slabs in the bivariate field as chemical potential is varied. An increasing trend in slab count is observable (Fig. 9.6) as chemical potential is increased on the hot and cold lattices. The exact reason for this is unknown; we may predict that the number of slabs may peak at region of chemical potential where deconfinement is expected to take place. This does not look to happen in the ensembles we have studied. Instead we find that lattices with higher chemical potentials tend to have more a fractured multivariate topology. The effect persists in the data throughout the cooling process, as can be observed at

19 - 20 cools (Fig. 9.7), suggesting this is a feature of the topology rather than a quirk in the data.

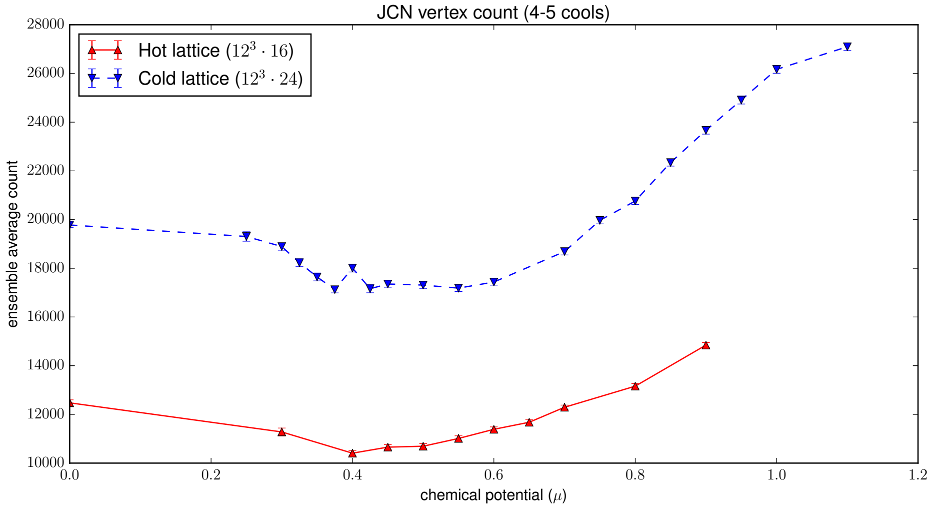


Figure 9.6: Number of vertices in the JCN, computed using the Polyakov loop at 4-5 cools. The cold lattice shows an increased number of slabs and both data sets show an upward trend as chemical potential μ increases.

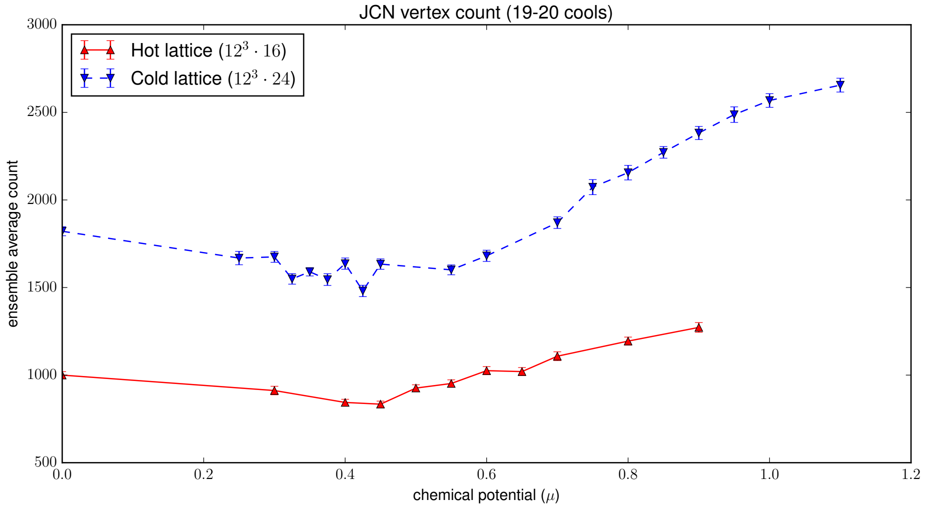


Figure 9.7: Number of vertices in the JCN, computed using the Polyakov loop at 19-20 cools. The upward trend on both lattices is preserved as cooling progresses.

9.3.2 JCN vertex connectivity

Connectivity within the JCN captures two interesting features. Starburst effects, similar to those seen in [88], are captured by vertices with medium connectivity. This represents sizeable regions of the Reeb space with two stable isovalues surrounded by multiple smaller regions where the isovalue is less stable. Duke et al. [85] suggest that these smaller regions are aliasing artefacts caused by the discrete nature of the algorithm.

Also present in the data, particularly as the lattice is cooled, are a pair of vertices with very high connectivity at the extremes of the ladder like JCN structure. These correlate to stable regions with relatively large volumes with a number of smaller features branching from them. Uncooled data show these branches throughout the isovalue range, but under cooling they gravitate to the largely connected vertices.

When considering the distributions for multiple ensembles we found interesting behaviour as the level of cooling was increased. For uncooled data there is very little difference between lattice sizes and levels of chemical potential (μ), meaning all ensembles are roughly similar using this measurement. However, as cooling takes effect we notice a distinct banding in the distributions (Fig. 9.9). This appears to not only make a distinction between differing chemical potentials, but also the hot and cold lattice.

9.3.3 Multivariate persistence measures

Vertex connectivity gives a rough estimation of the number of stable regions present under cooling. We next focus upon using persistence measures to calculate properties of these observables. It is expected that highly connected vertices in the JCN correlate to regions of the lattice with non-trivial volumes indicating the presence of stable features present at both stages of the cooling algorithm. Figure 9.10 demonstrates two persistence measures visually; working directly on the slab meshes, each of which is represented by a vertex in the JCN. The triangle count measure is able to pick-out boundaries between regions in the multi-field, whereas the surface area highlights gives a gradient effect around prominent topological features.

In the case where a single variable is evaluated using the JCN the output is equivalent to a discrete segmentation of the volume by isovalue. In Figure 9.11 we visualise this effect at two neighbouring levels of cooling alongside the bivariate topology of the same two inputs. We use this approach to assist in interpreting features persisting at the two levels of cooling.

To understand the effect of varying thermodynamic control parameters under cooling we computed persistence measures as the chemical potential (μ) and number of cooling iterations is varied. Persistence measures are averaged in each case by dividing by the total number of JCN vertices, relating to the number of slabs in the quantised Reeb space. In Fig. 9.12 we present this data as *cooling landscape* generated by counting the average number of triangles per slab on the hot and cold lattices. An upward trend is shown in the averages as the number of cooling iterations is increased and number of distinct topological objects decreases, representing the simplification effect the algorithm has on the input field and associated Reeb space. A similar trend is visible when comparing with other persistence measures including the average surface area (Fig. 9.13).

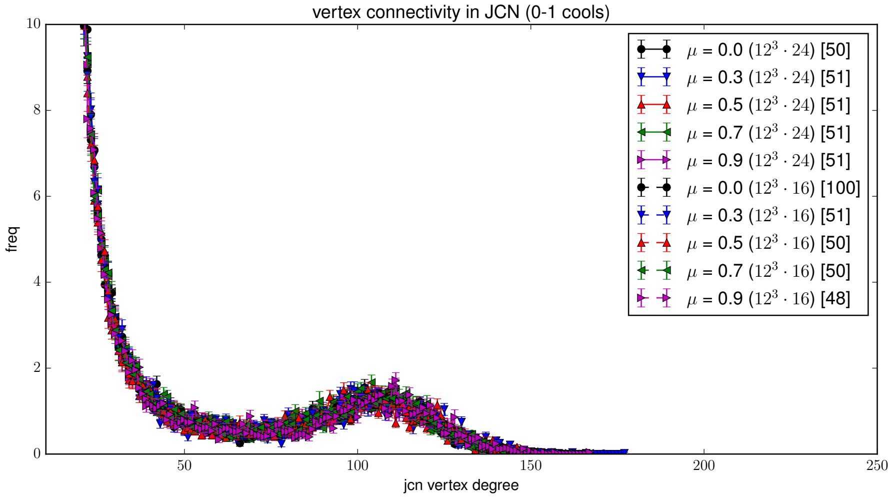


Figure 9.8: Degree of connectivity of vertices in the JCN as an average distribution for uncooled data. All ensembles, when uncooled, seem to exhibit the same connectivity distribution.

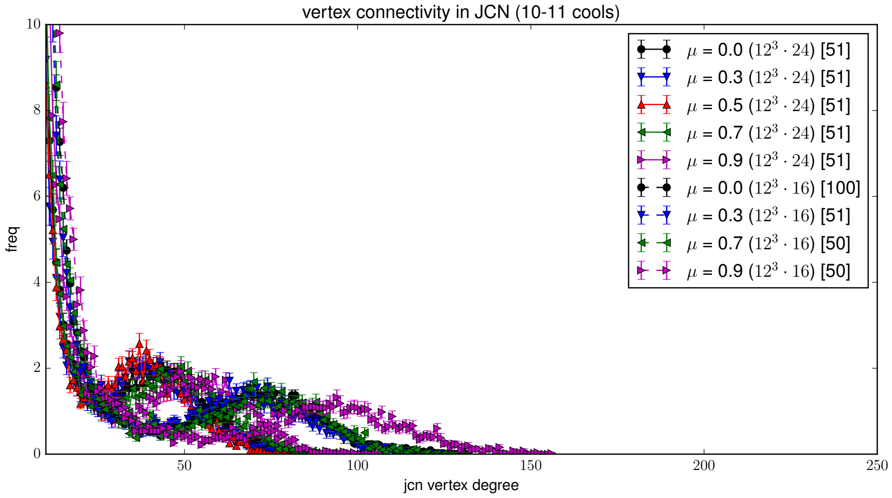


Figure 9.9: Degree of connectivity of vertices in the JCN as an average distribution for data cooled for 10 iterations. A distinct banding effect can be observed, separating the hot and cold lattices and differing chemical potentials.

In Fig. 9.14 we show a direct comparison of the persistence as a measure of the number of triangles on the two lattices at the maximum number of cools. Both lattices feature an initial largely flat region at low values of μ, followed by a global peak. A second peak follows higher in the chemical potential (μ) range following a region with a downward trend in persistence before both distributions begin to plateau at high values of μ. Initial examination of the two

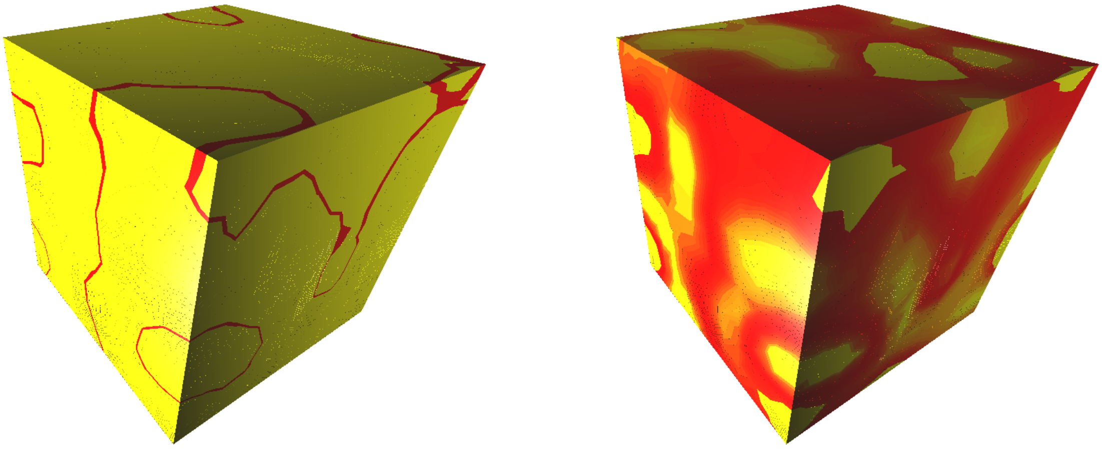


Figure 9.10: A direct comparison of the multivariate persistence measures. Left: measuring persistence as the number of triangles per slab. Right: measuring persistence as the surface area of each slab.

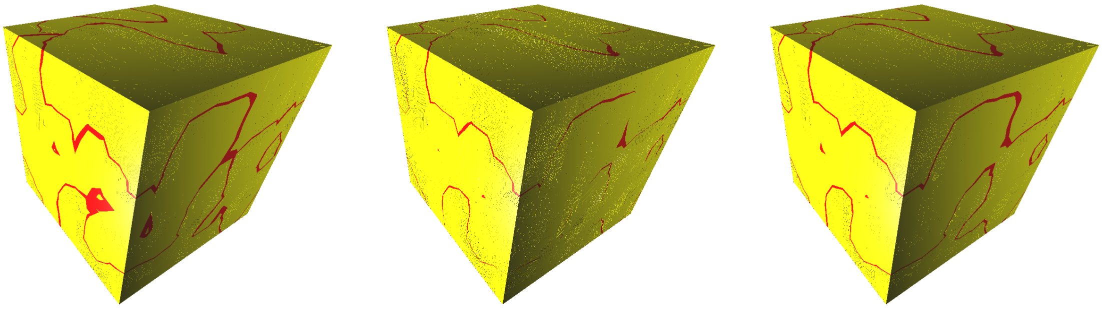


Figure 9.11: Left: Quantised contours generated from Polyakov loop at 19 cooling iterations. Centre: Joint contour slabs are a union of the quantised contours at 19 and 20 cooling iterations. Right: Quantised contours generated from Polyakov loop at 20 cooling iterations. Red regions represent most persistent slabs, measured by counting triangles.

distributions suggest that on the hotter lattice ($n_t = 16$) the trend is shifted towards the lower end of chemical potentials. This result seems to agree with recent evidence produced from statistical physics [168] that see a shift in de-confinement on hotter lattices. While this is an encouraging result, it should be noted that due to a limited number of configurations available for the analysis data exhibits large error bars. Consultation with physicists suggested that availability of a larger number of configurations will reduce the level of uncertainty. The surface area of each slab encouragingly converges to a similar distribution. There are some subtle differences in the two persistence measures; most evident in the hotter ($n_t = 16$) lattice. In particular there is a difference in the size of the peaks at $\mu = 0.45$ and $\mu = 0.65$.

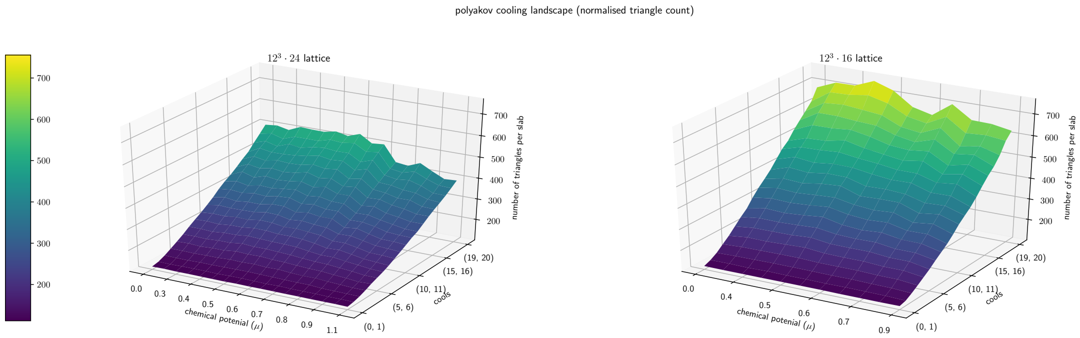


Figure 9.12: The average number of triangles per slab shown as the number of cooling iterations and chemical potential is varied.

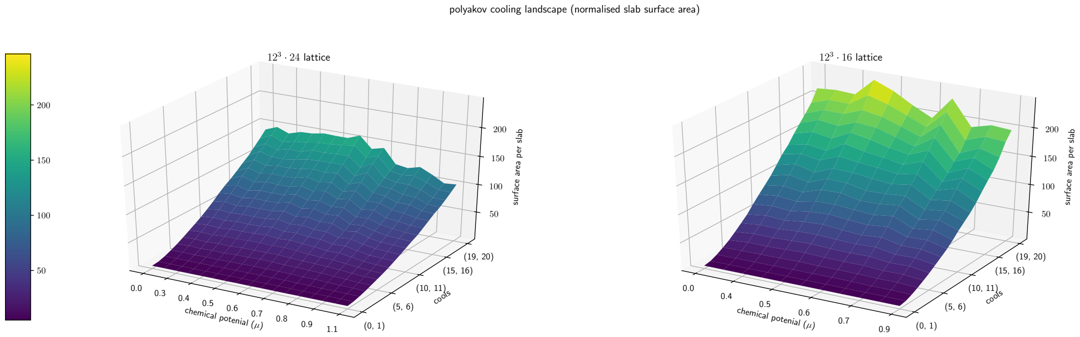


Figure 9.13: The average surface area per slab shown as the number of cooling iterations and chemical potential is varied. Peaks at specific levels of chemical potential μ appear to emerge as cooling progresses.

9.3.4 Ratio of Jacobi nodes in JCN

A Jacobi node is a generalisation of the concept of critical points for use in multivariate topology. A Jacobi node represents a point in the input domain where the gradient of multiple functions align. This can occur at local extrema and saddle points in the topology; however, Jacobi nodes can also indicate that the functions have become parallel at the sampling point.

Computing the ratio of Jacobi Nodes to JCN vertices gave an interesting overview of the effect of the cooling algorithm, with the ratio tending towards 1.0 as the level of cooling reached it's peak. This confirmed expectations as it would indicate an overall convergence to a stable configuration with minor changes between cooling iterations. A potential use for such data could be to help automate the process of determining when an optimal level of cooling has been reached — currently this is often manually decided using trends in the action and topological charge graphs.

The effect features some minor variations across the chemical potential range however, with ensembles at higher chemical potentials seemingly taking longer to converge on a optimal level of cooling. We are unable to conclusively say if this is an effect linked to deconfinement of the lattice. At low levels of cooling (Fig. 9.16) it possible to observe that the ratios form a

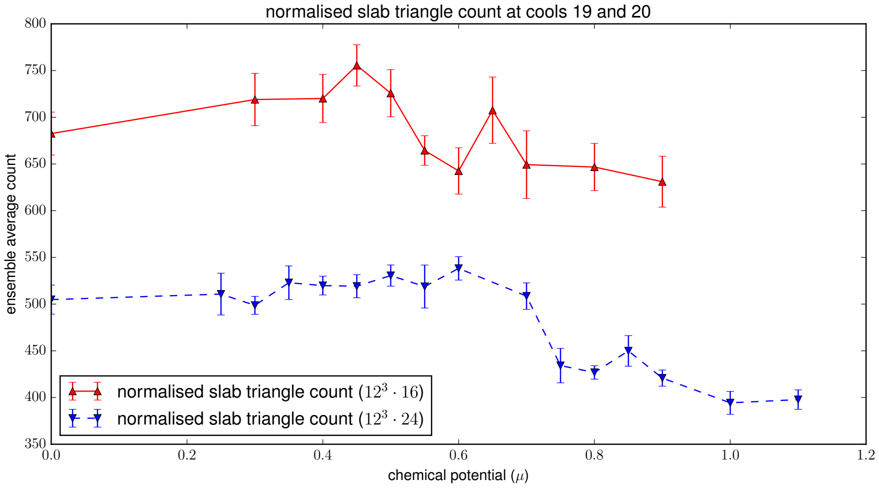


Figure 9.14: Average number of triangles per slab on the two lattices at cools 19 and 20.

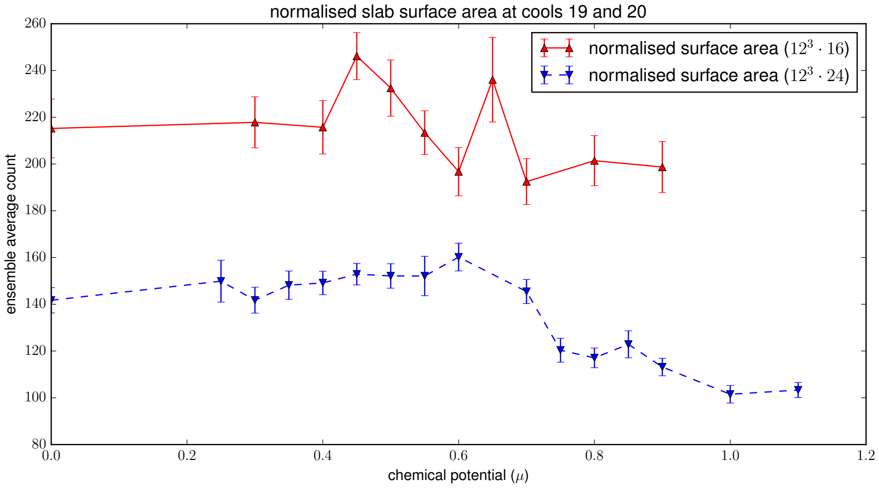


Figure 9.15: Comparing the average surface area per slab on the two lattices at cools 19 and 20.

smooth curve across the chemical potentials on both lattices. As cooling is increased (Fig. 9.17) deviations from the smooth curve become increasingly large.

When the two lattices are cooled to 14 - 15 iterations (Fig. 9.18) the graphs begin to converge to equal ratios. This suggests that the hot lattice has almost reached an almost optimal level of cooling in terms of level of change between iterations. This appears to be validated in Fig. 9.19 where both lattices have a ratio of Jacobi to JCN vertices of approximately 90%. We believe that if the lattice was cooled far beyond normal levels, a ratio of almost 100% would be observed as the lattice tended towards a vacuum state. At present we are unable to explain

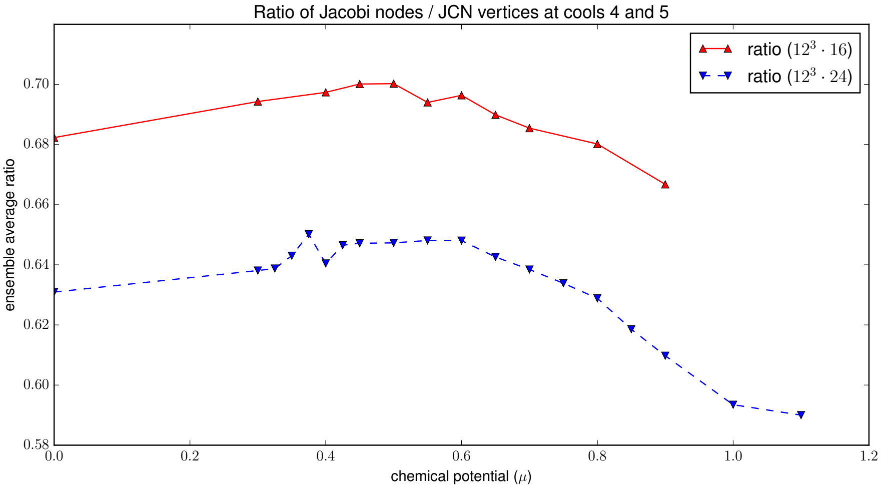


Figure 9.16: Ratio of Jacobi nodes to JCN vertices at 4–5 cools in the Polyakov field.

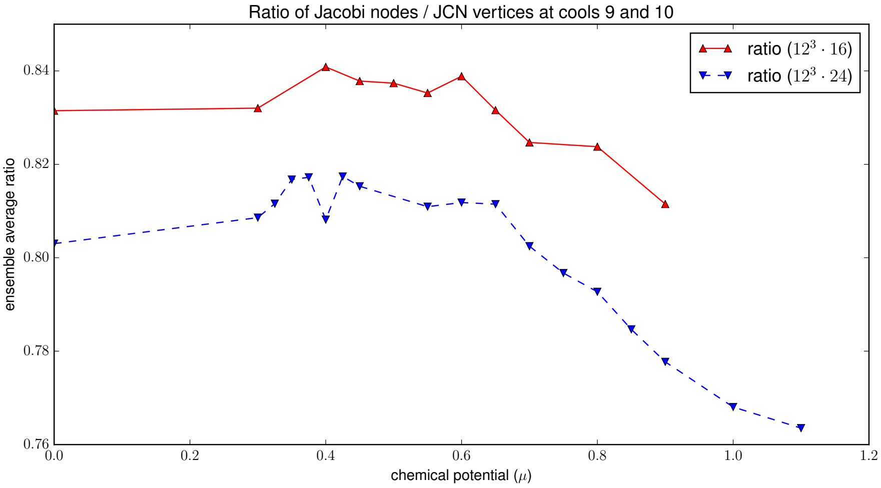


Figure 9.17: Ratio of Jacobi nodes to JCN vertices at 9–10 cools in the Polyakov field.

the erratic behaviour of the hot lattice at the highest level of cooling in this study.

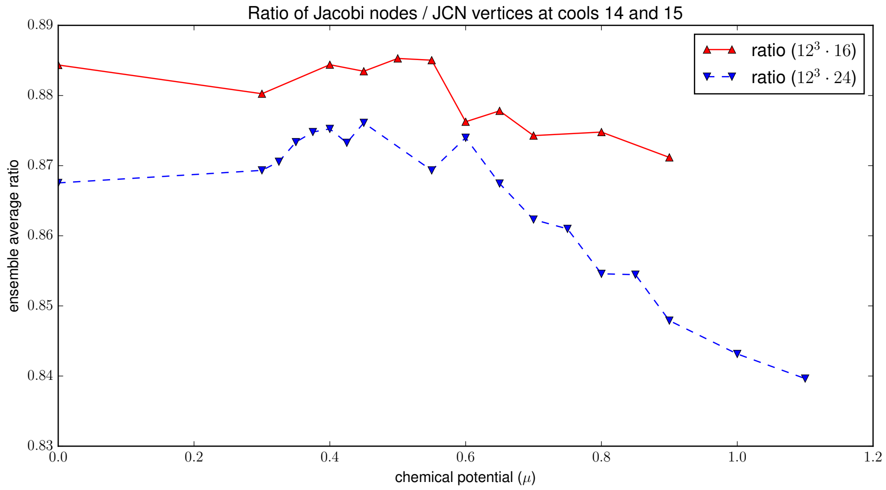


Figure 9.18: Ratio of Jacobi nodes to JCN vertices at 14–15 cools in the Polyakov field.

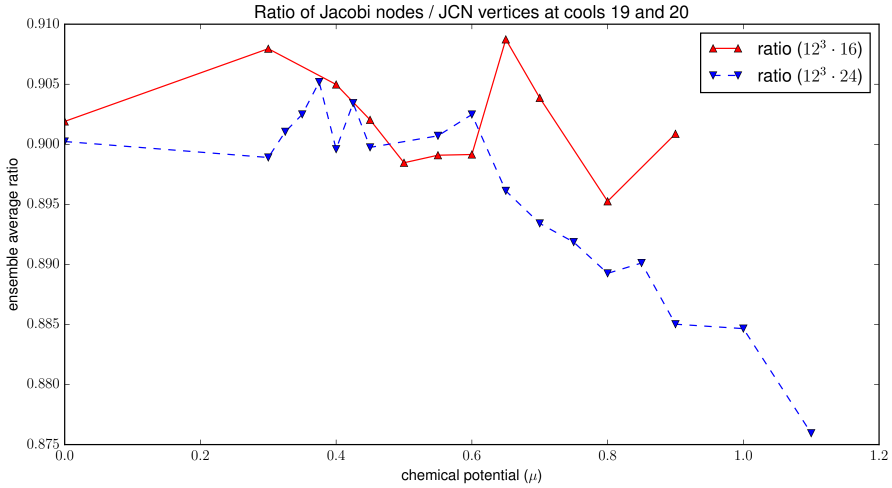


Figure 9.19: Ratio of Jacobi nodes to JCN vertices at 19–20 cools in the Polyakov field.

9.4 Summary

In this chapter we have performed an ensemble based study of lattice QCD data under cooling. This study uses elements of the knowledge we have gathered throughout this thesis to analyse data in a setting more familiar to a domain specialist. By first visually analysing several configurations we are able to make a set of predictions that can be tested at the ensemble level as thermodynamic control parameters are varied.

This case study acts as a potential prototype for further studies that may be carried out by a domain specialist. We have witnessed how simple measures of the multivariate topology can easily be computed directly from the graph structures offered by topological visualisation techniques. We have also shown how more advanced features can be taken from the lattice through the use of persistence measures.

To summarise, this chapter makes the following contributions:

- We investigated the effect of cooling on the lattice by comparing neighbouring cooling slices of Polyakov loop data in a multi-ensemble study.
- It was demonstrated how the complexity of the JCN may be influenced by the varying of thermodynamic control parameters.
- Multivariate persistence measures were used as a method for evaluating the importance of objects that exist on the lattice between cooling iterations.
- It was suggested that the ratio of Jacobi nodes in the data may help to predict an optimal level of cooling.

In Chapter 10 we describe the entire framework used throughout this thesis. This has guided us from initially viewing data as singular configurations, to computing quantitative properties of each configuration, and onto forming the basis of large automated ensemble studies.

Chapter 10

A framework for the visualisation of Quantum Chromodynamics (QCD) Data

Contents

In this chapter we detail a framework produced in order to facilitate the exploratory studies detailed within this thesis. QCDVis is inspired by previous work in visualisation of lattice QCD and offers a natively cross-platform system that builds upon the existing use of isosurfaces in lattice QCD to offer topological driven visualisation through the use of the contour tree, Reeb graph, and flexible isosurfaces. The end result is a system that is tailored specifically to the needs of QCD scientists, allowing them to interactively explore the underlying topology of the data. A video is available at https://vimeo.com/205054908 that shows the system in use.

The work this chapter is based upon was originally presented at the Spring Conference on Computer Graphics and was later published in Computers & Graphics [13]. We begin this chapter by discussing the motivations for creating such a tool including the intended contributions made to lattice QCD community in Section 10.1. We perform a detailed analysis of the framework in Section 10.2, where we describe the solution has been implemented as a modular system. In Section 10.3 we review feedback gathered in a user study performed with domain experts.

10.1 Research motivation and contributions

As we have shown in previous chapters lattice QCD data sets are relatively small in spatio-temporal dimensions. However, due to the quantum nature of the theory they are generally computed in vast quantities. Due to this complexity it is crucial to extract interesting features and present them using meaningful information. This work is therefore driven by a desire to support scientists in the validation, inspection, and understanding of the properties of lattice QCD data.

To achieve this goal we contribute a system to support physicists in the process of:

- validation and analysis of lattice QCD data sets;
- generation of visual designs that provide novel insight into the structure and evolution of phenomena represented by lattice QCD data;
- simultaneous comparison of both quantitative and qualitative aspects of QCD phenomena.

In the development of our framework we leverage the strength of topology-driven analysis, successfully introduced in different research domains and range of applications all sharing a common goal — to achieve an efficient and compact description of the basic shapes co-existing in an arbitrary set of data.

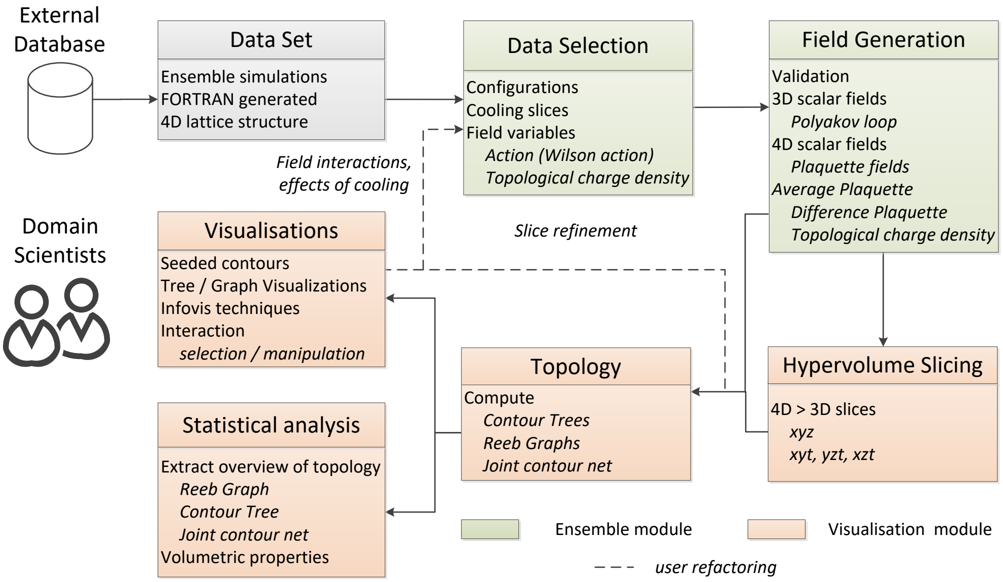


Figure 10.1: QCDVis Visual Analytics pipeline.

Existing attempts at visualisation of QCD data have so far concentrated on single aspects of the analysis [137], or have relied upon static visualisations [140]. At present no framework exists capable of supporting the scientist through the entire analytical pipeline from beginning to end and back through the use of interactive visualisations. Our framework has been developed to fill this gap. QCDVis (Figs. 10.1, 10.2, 10.7) has been designed with the user in mind. The framework addresses the fundamental aspects of data validation, visualisation and estimation of QCD objects and observables, manipulation and traversal of data. Core to the analysis is support for identification of discrete objects and their characteristics using topological structures data such as the Reeb graph. This enables research questions to be tackled in a quantitative fashion. The following sections provide detailed descriptions of each aspect of our framework and its implementation.

10.2 Implementation

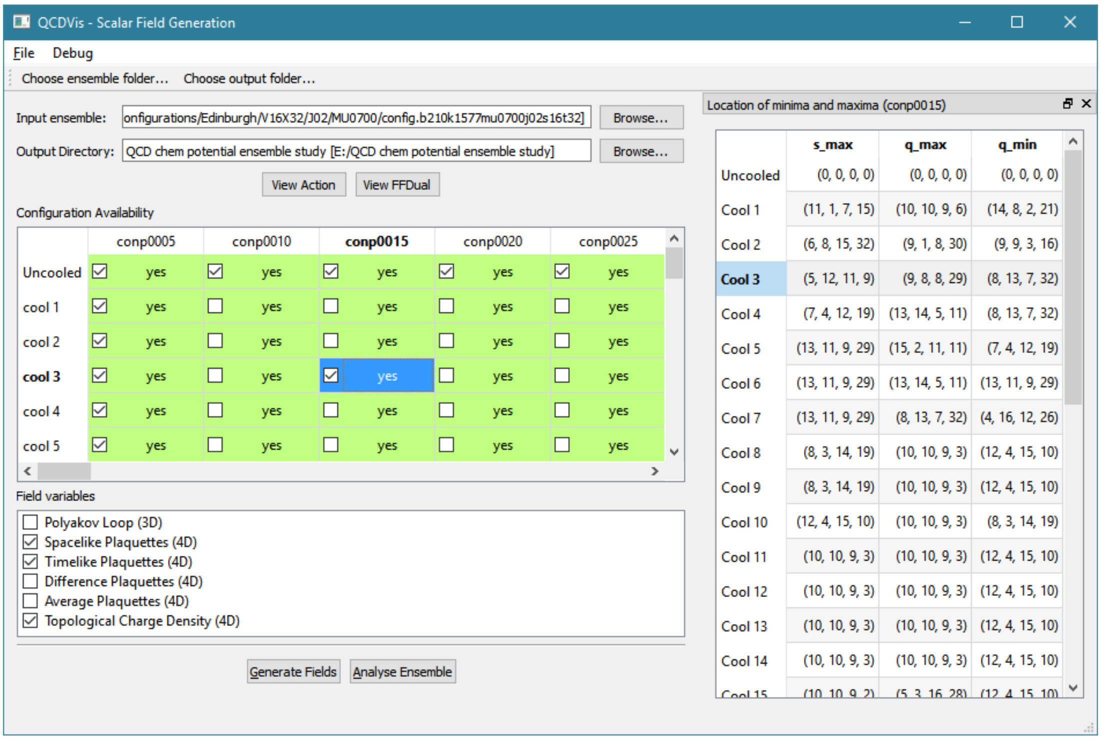
QCDVis is developed using the C++11 standard, with the Graphical User Interface (GUI) developed using Qt version 5.8 to enable cross-platform development. In order to allow a rapid development and testing cycle, the program has been written using object-oriented techniques. This allows for the quick introduction of new features by querying specific topological structures in the scalar fields as distinct objects. Furthermore, the code has a modular design making use of functional aspects of modern C++, such as lambda expressions and parallel execution, to allow new algorithms and features to be easily added to the software. Throughout development experimental aspects of the system made use of Python scripting to quickly prototype and receive feedback from users before being integrated into the core application.

QCDVis has been designed as an interactive environment to enable physicists to visually explore their data and its underlying topology. Figure 10.1 summarizes the key stages of the visual analytic pipeline upon which our system is constructed. Along with 3D rendered views of user selected scalar fields, we also include data representations that are commonly found in the QCD domain. Histograms, in particular, are firmly established in physics community for identifying underlying trends in data. Therefore, they are included to allow physicists to gain an overview of their data and promote a focus on the parts of the data they find most interesting.

The core QCDVis framework consists of two main modules; the first processes ensemble data and is used to transform raw lattice configurations into discrete scalar fields representing a number of lattice field operators. The second, working at the configuration level, is used to visualise and probe these fields using topological methods. Each module can work independently or can be used together to semi-autonomously probe the ensemble with respect to its topological features. In this scenario, the ensemble tool is used to invoke the visualisation and analysis application with given sets of configurations. In addition to the core application a number of command line tools have been created to assist in analysis of the data.

10.2.1 Ensemble module

Figure 10.2 shows the interface of the ensemble computation module made up of the main components discussed below. Lattice QCD simulations are stored as 4D array-like structures referred to as lattice configurations. These are not the direct observables and require processing to generate data for visualisation purposes. The resultant data is 4D, sliced into 3D volumes along a user defined axis.



Input ensemble: onfigurations/Edinburgh/V16X32/J02/MU0700/config.b210k1577mu0700j02s16t32]

Output Directory: QCD chem potential ensemble study [E:/QCD chem potential ensemble study]

Configuration Availability

	conp0005		conp0010		**conp0015**		conp0020		conp0025	
Uncooled	☑	yes	☑	yes	☑	yes	☑	yes	☑	yes
cool 1	☑	yes	□	yes	□	yes	□	yes	□	yes
cool 2	☑	yes	□	yes	□	yes	□	yes	□	yes
cool 3	☑	yes	□	yes	☑	yes	□	yes	□	yes
cool 4	☑	yes	□	yes	□	yes	□	yes	□	yes
cool 5	☑	yes	□	yes	□	yes	□	yes	□	yes

Field variables

- □ Polyakov Loop (3D)
- ☑ Spacelike Plaquettes (4D)
- ☑ Timelike Plaquettes (4D)
- □ Difference Plaquettes (4D)
- □ Average Plaquettes (4D)
- ☑ Topological Charge Density (4D)

Location of minima and maxima (conp0015)

	s_max	q_max	q_min
Uncooled	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
Cool 1	(11, 1, 7, 15)	(10, 10, 9, 6)	(14, 8, 2, 21)
Cool 2	(6, 8, 15, 32)	(9, 1, 8, 30)	(9, 9, 3, 16)
Cool 3	(5, 12, 11, 9)	(9, 8, 8, 29)	(8, 13, 7, 32)
Cool 4	(7, 4, 12, 19)	(13, 14, 5, 11)	(8, 13, 7, 32)
Cool 5	(13, 11, 9, 29)	(15, 2, 11, 11)	(7, 4, 12, 19)
Cool 6	(13, 11, 9, 29)	(13, 14, 5, 11)	(13, 11, 9, 29)
Cool 7	(13, 11, 9, 29)	(8, 13, 7, 32)	(4, 16, 12, 26)
Cool 8	(8, 3, 14, 19)	(10, 10, 9, 3)	(12, 4, 15, 10)
Cool 9	(8, 3, 14, 19)	(10, 10, 9, 3)	(12, 4, 15, 10)
Cool 10	(12, 4, 15, 10)	(10, 10, 9, 3)	(8, 3, 14, 19)
Cool 11	(10, 10, 9, 3)	(10, 10, 9, 3)	(12, 4, 15, 10)
Cool 12	(10, 10, 9, 3)	(10, 10, 9, 3)	(12, 4, 15, 10)
Cool 13	(10, 10, 9, 3)	(10, 10, 9, 3)	(12, 4, 15, 10)
Cool 14	(10, 10, 9, 3)	(10, 10, 9, 3)	(12, 4, 15, 10)
Cool 15	(10, 10, 9, 2)	(5, 3, 16, 28)	(12, 4, 15, 10)

Figure 10.2: Ensemble interface. Physicists can select multiple cooling slices, configurations and field variables of interest.

Ensemble cooling histogram generation (Fig. 10.3) Upon selecting an ensemble for inspection, physicists can view underlying patterns under cooling by computing graphs of the action and topological charge density statistics (e.g., Wilson action and topological charge density respectively). This can steer them towards interesting features in the derived scalar fields.

Location of field maxima and minima (Fig. 10.4) The cooling program is able to detect the position in space-time of minima and maxima of the action and topological charge density fields as the lattice is cooled. This data is available for each cooling stage for each configuration and can be called up by the user from the main interface. Physicists can use this data to determine the probable existence of (anti-)instantons in the topological charge density field, signified by corresponding minima and maxima in both fields. In QCDVis we include this data to provide an additional method for determining interesting stages of cooling in each configuration.

Configuration selection (Fig. 10.5) The vast quantity of configurations created for an ensemble under cooling mean that it is not viable for the physicists to view all possible configuration/cooling combinations. Additionally, it is unusual for a domain physicist to retain configurations early in the cooling cycle, as the configurations are too noisy for use in calculations. Therefore, physicists can select exactly which configurations they intend to further

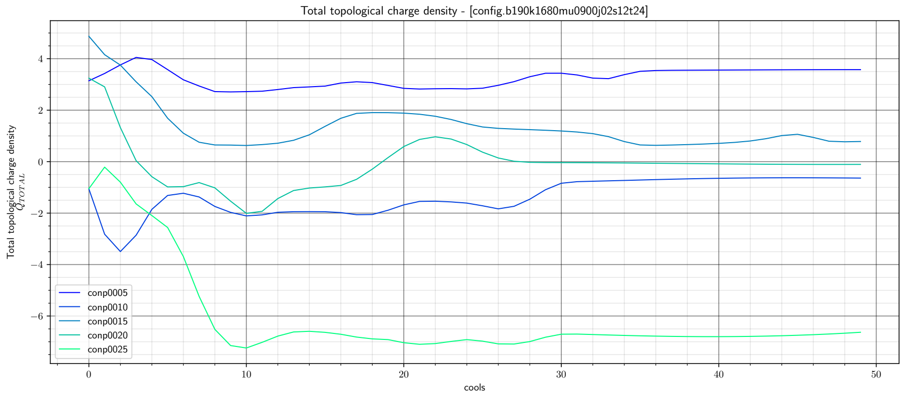
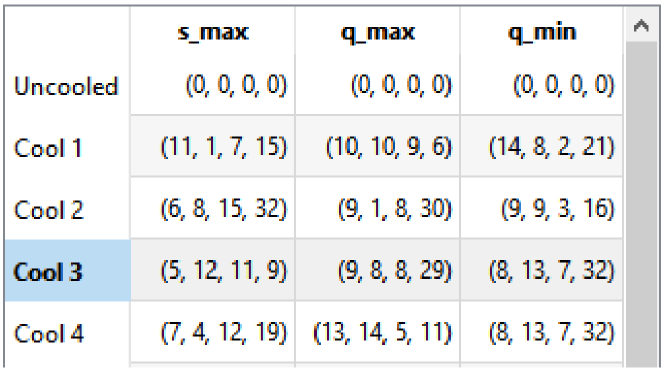


Figure 10.3: Interactive graphs are offered to the user to assist in the selection of lattice configurations at optimal levels of cooling.



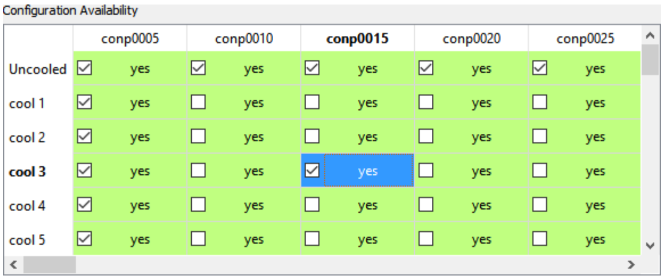
	s_max	q_max	q_min
Uncooled	(0, 0, 0, 0)	(0, 0, 0, 0)	(0, 0, 0, 0)
Cool 1	(11, 1, 7, 15)	(10, 10, 9, 6)	(14, 8, 2, 21)
Cool 2	(6, 8, 15, 32)	(9, 1, 8, 30)	(9, 9, 3, 16)
Cool 3	(5, 12, 11, 9)	(9, 8, 8, 29)	(8, 13, 7, 32)
Cool 4	(7, 4, 12, 19)	(13, 14, 5, 11)	(8, 13, 7, 32)

Figure 10.4: The location of the most prominent features in each configuration are given to guide the user towards potential (anti-)instantons in the data. The core observables appear as minima / maxima in the topological charge density field Q; examination of the peak values in the action field S can assist in interpreting cooling data.

investigate using information gathered from generated histograms and localization of maxima and minima (see above). Singular configurations can be selected by checking the relevant cell in the grid but we also offer the convenience of selecting all configurations across a level of cooling or within a configuration.

Field variable selection (Fig. 10.6) At present the system is able to calculate a range of field variables including topological charge density, polyakov loops, plaquettes, and fields derived from the averages of time-like and space-like plaquettes. Most lattice fields require the calculation of the space-like and time-like plaquettes in order to create the final scalar field. In order to optimize the computation of these values, they are computed and cached on demand.

Differing field variables may be required depending on the nature of the study being con-

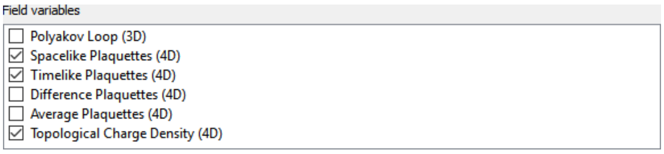


Configuration Availability

	conp0005		conp0010		conp0015		conp0020		conp0025	
Uncooled	☑	yes	☑	yes	☑	yes	☑	yes	☑	yes
cool 1	☑	yes	□	yes	□	yes	□	yes	□	yes
cool 2	☑	yes	□	yes	□	yes	□	yes	□	yes
cool 3	☑	yes	□	yes	☑	yes	□	yes	□	yes
cool 4	☑	yes	□	yes	□	yes	□	yes	□	yes
cool 5	☑	yes	□	yes	□	yes	□	yes	□	yes

Figure 10.5: Configuration selection. The user can select any number of configurations, at various levels of cooling, to view cooling data as histograms or to generate scalar fields.

ducted; hence, we provide a selection of variables that are calculated at runtime. For example, use of the Polyakov loop field is a basic method of detecting “breaking of symmetry” on the lattice. The topological charge density, average plaquette, and difference plaquette fields can all be used to predict the presence of instanton observables.



Field variables

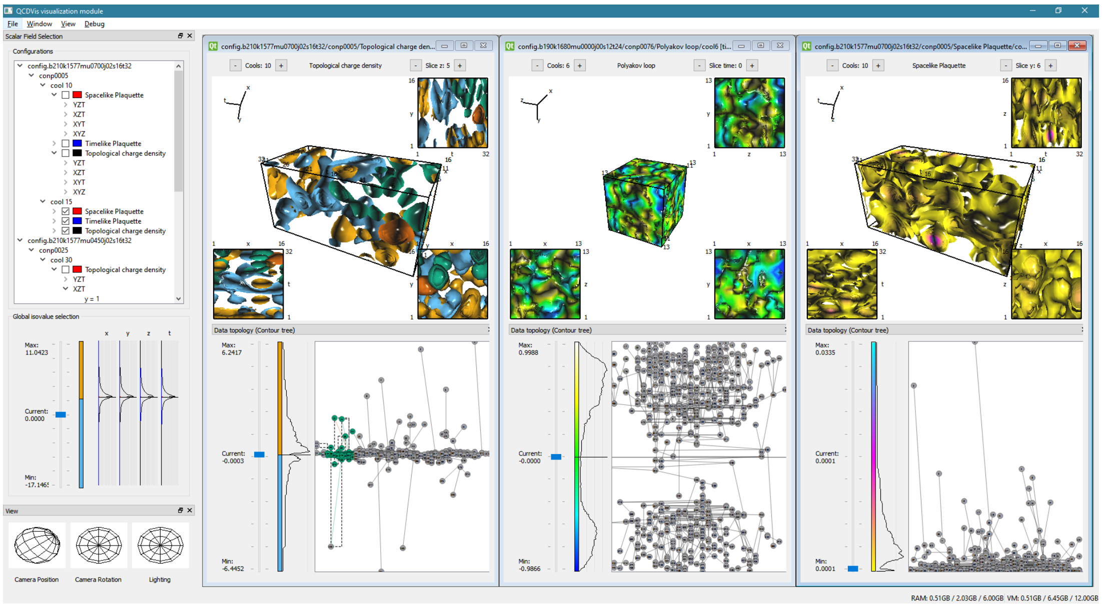
☐ Polyakov Loop (3D)
☑ Spacelike Plaquettes (4D)
☑ Timelike Plaquettes (4D)
☐ Difference Plaquettes (4D)
☐ Average Plaquettes (4D)
☑ Topological Charge Density (4D)

Figure 10.6: The user can generate several different lattice fields from the data.

Field variables calculations from configuration data. Data is stored on a four dimensional Euclidean lattice with the $\mathrm{SU}(2)$ matrices (denoted μ, ν, ρ, σ) assigned to edges in the x, y, z, and t dimensions. Field strength variables are defined by moving around the lattice from a given origin in closed loops.

To calculate different lattice fields we complete closed loops from a given lattice site in the μ, ν, ρ, σ directions. Movements across the lattice in a positive direction require multiplication by the relevant $\mathrm{SU}(2)$ link variable. Equivalently, movement in a negative direction requires a multiplication by the matrix in its adjoint, or conjugate transpose, form. Hence, to complete a unit loop in the xy plane a path-ordered multiplication of two link variables (μ, ν) and the two link variables in their adjoint form $(\mu^\dagger, \nu^\dagger)$ is performed.

The main field variables of interest are topological charge density, Polyakov loops, space-like and time-like plaquettes, and their average/difference (as shown in Fig. 10.2.4). Topological charge density for example is a loop around all four dimensions from an origin point. Thus values are a multiplicative combination of three spatial plaquettes and three time-like plaquettes. The adjoint variations of each plaquette, representing a loop in reverse, are subtracted from these to form the *difference plaquettes*.



config.b210k1577mu0700j02s16t32/conp0005/Topological charge den...

Cools: 10 — Topological charge density — Slice z: 5

config.b190k1680mu0000j00s12t24/conp0076/Polyakov loop/cool6 [ti...

Cools: 6 — Polyakov loop — Slice time: 0

config.b210k1577mu0700j02s16t32/conp0005/Spacelike Plaquette/co...

Cools: 10 — Spacelike Plaquette — Slice y: 6

Global isovalue selection: Max: 11.0423 Current: 0.0000 Min: -17.146!

Topological charge density: Max: 6.2417 Current: -0.0003 Min: -6.4452

Polyakov loop: Max: 0.9988 Current: -0.0000 Min: -0.9866

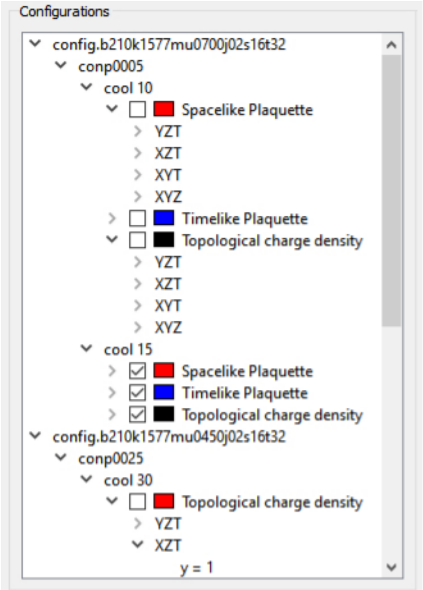
Spacelike Plaquette: Max: 0.0335 Current: 0.0001 Min: 0.0001

RAM: 0.51GB / 2.03GB / 6.00GB VM: 0.51GB / 6.45GB / 12.00GB

Figure 10.7: QCDVis Visualisation Interface. QCDVis enables physicists to slice and view multiple configurations and variables from different ensembles in parallel. Leftmost and rightmost visualisation showing topological charge density and space-like plaquettes of different samples from same configuration, central visualisation showing Polyakov loops from a sample extracted from a different ensemble.

10.2.2 Visualisation module

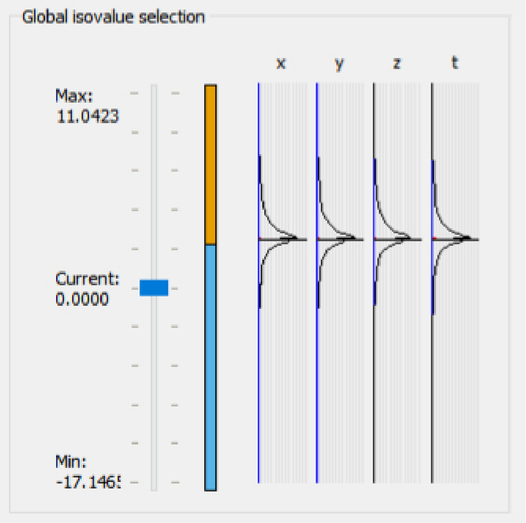
Figure 10.7 shows an overview of the visualisation module which is composed of the features discussed below.



Configurations

- config.b210k1577mu0700j02s16t32
  - conp0005
    - cool 10
      - □ 🟥 Spacelike Plaquette
        - YZT
        - XZT
        - XYT
        - XYZ
      - □ 🟦 Timelike Plaquette
      - □ ⬛ Topological charge density
        - YZT
        - XZT
        - XYT
        - XYZ
    - cool 15
      - ☑ 🟥 Spacelike Plaquette
      - ☑ 🟦 Timelike Plaquette
      - ☑ ⬛ Topological charge density
- config.b210k1577mu0450j02s16t32
  - conp0025
    - cool 30
      - □ 🟥 Topological charge density
        - YZT
        - XZT
          - y = 1

Figure 10.8: Multiple scalar fields are arranged in a hierarchical structure.



Global isovalue selection

Max: 11.0423

Current: 0.0000

Min: -17.146!

Figure 10.9: Isovalues can be controlled globally for all open data sets.

Scalar Field Selection (Fig. 10.8) QCDVis allows multiple configurations and field variables to be viewed in parallel, many of which are 4D. Each opened configuration is accessible through a hierarchy, the root of which is the source ensemble. The next level of the hierarchical view is the configuration followed by the relevant cooling slice.

For fields that are a 4D in nature, all 3D sub-volumes are computed and cached when first opening the file. An additional layer of the hierarchy sorts each of the 3D scalar fields according to the axis of slicing. The user can view multiple scalar fields by double clicking the relevant item in the tree. We therefore make no assumptions of how the user intends to interact with the data but, instead enable the user to choose combinations of fields based upon their own requirements. This makes QCDVis adaptable to viewing data concurrently from a number of points-of-view:

- Multiple ensembles can be viewed to gain an overview of how simulation parameters, such as chemical potential, impact upon the structure of objects.
- Configurations in the same ensemble can be viewed at different steps along the generating Markov chain.
- The effect of cooling on the lattice can be observed by viewing the same configuration at differing levels of cooling.
- Comparisons can be made between differing lattice variables by visual comparison of the field data.
- Objects can be viewed in both space-like (e.g. *xyz*) and time-like (e.g. *xzt*) orientations, this can give extra information about the structure of lattice observables in space time.

Global isovalue interactions (Fig. 10.9) The user is able to interact with the data globally or on an individual configuration basis. At the global scope, we provide an overview of selected configurations using a histogram of the number of objects at each isovalue across the entire 4D data set. Histograms are available for each of the space-time axes as it is expected that the space-like and time-like signatures will vary. The user can also set a global isovalue using a slider control, enabling them to understand correspondences in differing configurations or lattice operators.

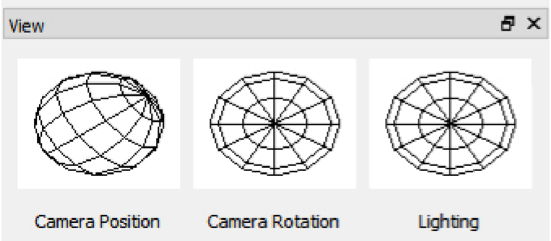


Figure 10.10: Objects can be interacted with using arc-ball rotations.

Global view interactions (Fig. 10.10) In order to visualise data from the same point-of-view we enable the user to interact globally with all open scalar fields. For rotation we provide the user with controls using the concept of arc-ball [169] rotations, this gives the freedom of movement for trackball input devices. A floating camera orbits the outside of the scalar field with the radius of the sphere it sits upon modifiable using a scroll wheel.

A number of fixed cameras can also be placed by the user; these exist at each corner of the bounding box. Additionally, a second arc-ball widget enables fine adjustment of the camera, initially fixed upon the centre of the data. The user can also move the relative position of the lighting using a third arc-ball control.

Data topology overview and isovalue selection (Fig. 10.12) Scalar fields can be explored using a localised isovalue range. The user can steer towards regions of interest in the domain via either a contour tree or Reeb graph (Fig. 10.11) representation of the domain. Selections in the contour tree or Reeb graph are reflected in the flexible-isosurface view, as shown in the leftmost visualisation in Fig. 10.7. Selected nodes, arcs, and corresponding contours are coloured green. Nodes, arcs, and corresponding contours under the mouse cursor are highlighted in orange. Following a similar approach to Bajaj et al. [5] the histogram presented alongside the isovalue selection slider shows the number of distinct objects in the scalar field across the isovalue range.

In many lattice QCD field data sets the regions of interest are the positive and negative extremes; however, the contour spectrum can also show interesting features in the region of percolation (the region around zero). Signatures of (anti-)instantons show up in the histogram as highly persistent objects in the positive or negative direction. Other field variables, such as the Polyakov loop, usually show a measure of symmetry around zero.

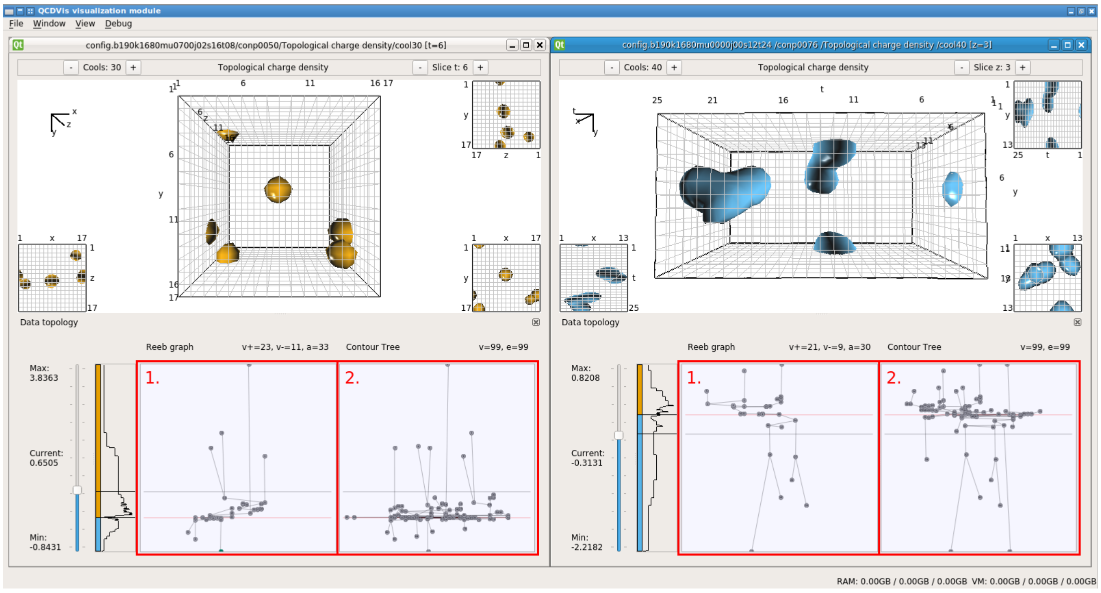

QCDVis visualization module
File Window View Debug
config.b190k1680mu0700j02s16t08/conp0050/Topological charge density/cool30 [t=6]
- Cools: 30 +
Topological charge density
- Slice t: 6 +
Data topology
Reeb graph
v+=23, v-=11, a=33
Contour Tree
v=99, e=99
Max: 3.8363
Current: 0.6505
Min: -0.8431
1.
2.
config.b190k1680mu0000j00s12t24 /conp0076 /Topological charge density /cool40 [z=3]
- Cools: 40 +
Topological charge density
- Slice z: 3 +
Data topology
Reeb graph
v+=21, v-=9, a=30
Contour Tree
v=99, e=99
Max: 0.8208
Current: -0.3131
Min: -2.2182
1.
2.
RAM: 0.00GB / 0.00GB / 0.00GB VM: 0.00GB / 0.00GB / 0.00GB

Figure 10.11: Left: Two objects crossing on the periodic x and y boundaries are joined by two non-periodic objects in a time-slice of topological charge density. Right: An object crossing a periodic boundary on the t axis, a second object crosses on the y boundary. Both visualisations are shown alongside their Reeb graph (1.) and the contour Tree (2.) representations to illustrate the difference in these topological structures when dealing with periodic boundaries.

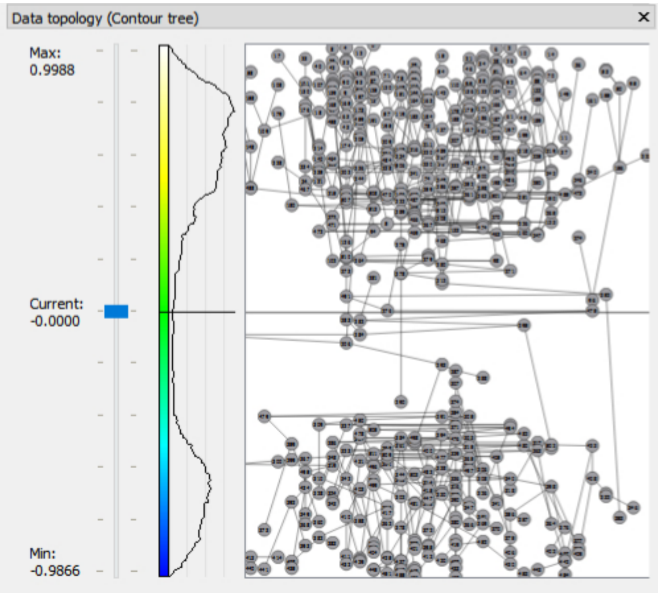


Figure 10.12: Contour trees and Reeb graphs are presented alongside a histogram of isovalues.

Orthographic projections of surfaces (Fig. 10.13) We also present surfaces existing in the scalar field using orthographic projections from each 2D plane. These have a fixed footprint with the scale adjusted according to the range of each axis, with the temporal extent of the data often different to that of the spatial axes. This is because in lattice QCD the temperature is given as an inverse function of the number of time steps. In situations where the physicist prefers not to use these projections they can be hidden.

Orthographic projections can be a useful tool for physicists when tracking objects in specific fields, for instance (anti-)instantons in the topological charge density field, and is best suited to verifying and analysing phenomena identified by the cooling code. As part of the cooling process the position of minima and maxima in the action and topological charge density fields is output by the program. By identifying these objects using the orthogonal views their structure can visualised in various 3D configurations of the 4D hypervolume. Occlusion from overlapping surfaces is rarely a problem in this view, as the objects of most interest appear in sparse areas of the scalar topology.

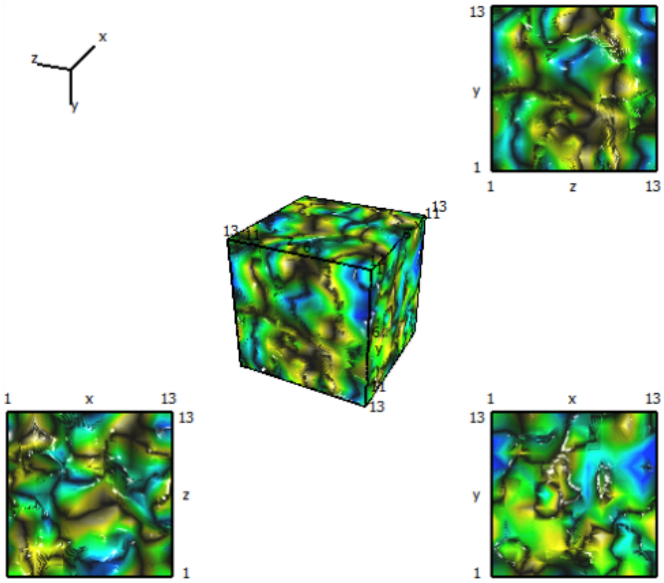


Figure 10.13: The main flexible isosurface view of the data is surrounded by orthographic projections of the data on each axis.

Flexible isosurface view (Fig. 10.13) The core visual interface is a 3D rendered view of the selected scalar field. Use of the contour tree algorithm enables the transition from a basic isosurface to representation of connected contours in the data. This has two benefits; first, optimized visualisation techniques can be applied to the data. Second, properties can be calculated on distinct contours as opposed to the level set as a whole, allowing the calculation of *physical* measurements of objects existing in the data.

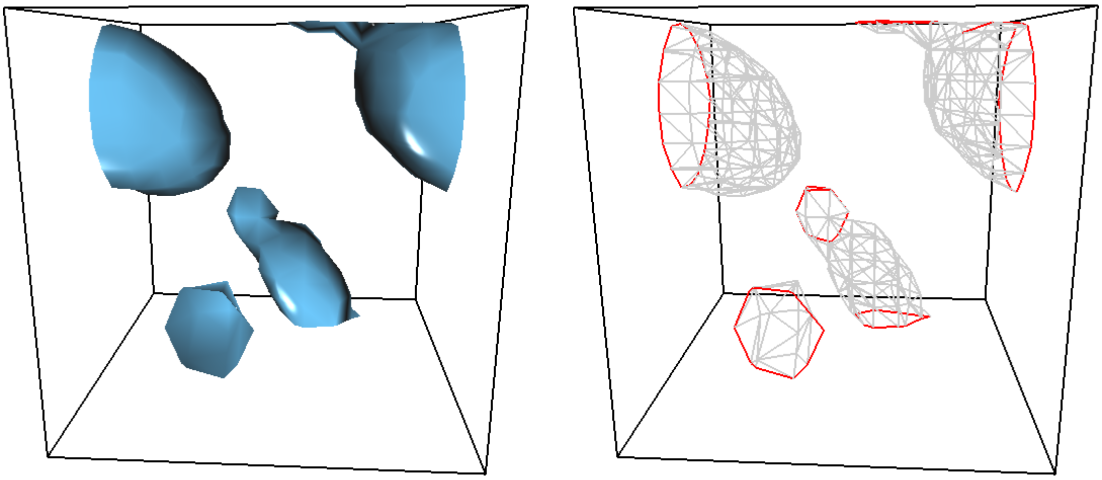
Each arc in the contour tree is a representative of a unique object, the mesh for which is generated using a modified form of the *flexible isosurface* algorithm [76]. Each isosurface can be continuously deformed between upper and lower bounds of the contour tree arc. This offers the possibility for parallel computation of complex contours.

Properties of topological objects The triangle meshes created for rendering also allow us to query properties of the objects that are beyond the statistical mechanics features usually available to physicists. The simplest of these is evaluation of the surface area and volume of the mesh.

To calculate more advanced properties we use the mathematical concept of moments — using methods initially implemented by Zhang and Chin [156]. The zeroth moment represents the signed volume of a contour; the first moment, the centre of mass; and the second representing moments of inertia. A pre-requisite for correct moment calculations is for a mesh to be closed. An initial step to determine this is to calculate the Euler-Poincaré characteristic ($\chi = V - E + F$) of a mesh. Following this approach it is possible that a contour which spans two or more boundaries can be incorrectly identified as an n-torus. To counteract this, objects are also tested to see if vertices lie on the height field boundaries. A more in-depth discussion of the use of moments in visualisation can be found in [157] and [158]. We calculate the centre of mass using the three first-order moments, however we only show centre of mass and moment of inertia for closed contours.

Centre of mass requires the calculation of three first-order moments by computing a weighted average in the x, y, and z planes of each triangle. This is scaled by its contributing volume, calculated by placing an additional vertex at the origin to make a tetrahedron. The final centre of mass for the object is calculated as a positional vector that must be scaled by the entire objects volume. We only show the centre of mass for closed contours in the main display by a marker. The moment of inertia requires the computation of the eigenvectors of the 2^{nd}-order tensor. First-order mixed moments, which take into account averages on the plane, need to be calculated along with second-order moments. Eigenvalues and eigenvectors of the inertia tensor are computed, normalised, and sorted in order of magnitude of their derived eigenvalue with the largest being the principle component axis. Comparison of the relative magnitudes of the eigenvalues, and orientation of the corresponding eigenvectors with respect to the t-axis, reveals any tendency for instanton distortion, either oblate or prolate, as μ is increased.

Periodic boundary conditions (Fig. 10.14). A substantial challenge is posed by the periodic nature of the data being handled in this application. To ensure translational invariance of quantum averages, lattice QCD simulations are performed with periodic boundary conditions for the gauge variables in the time and three spatial dimensions. Hence, the extreme vertices of each dimension of the data are considered as being neighbours with a cell existing between them, as shown by Fig. 10.14. For the purposes of generation this gives the illusion of an infinite universe but complicates computation of the contour tree. The Reeb graph representation, however, does not suffer from this limitation. We therefore supply the user with both representations to further support the analysis. In Figure 10.11 the comparison of the Reeb graph and contour tree for each view shows that the Reeb graph correctly sees the object as one distinct superarc (topological object). However, in the contour tree the surfaces are represented by two separate superarcs. Periodicity of data is a feature not limited to QCD datasets but also appears in other fields of science including computational chemistry [154] and computational biology [170].

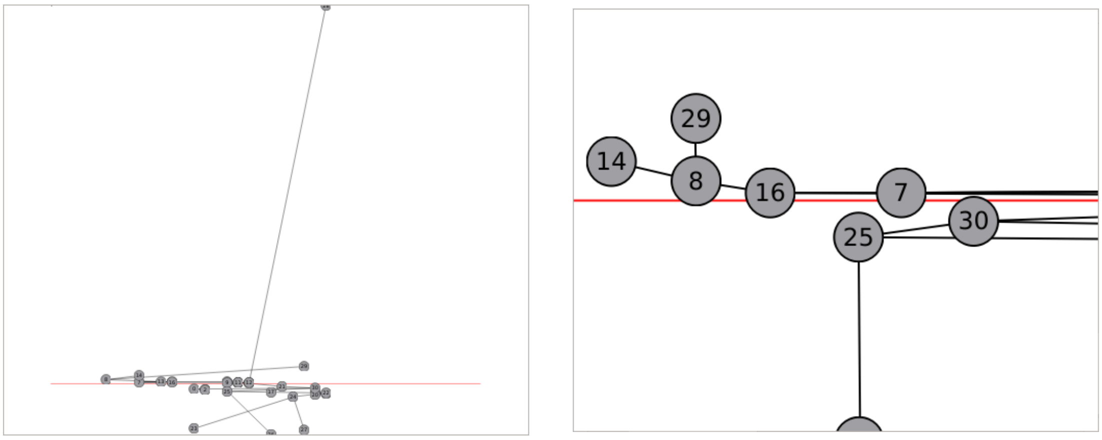


Object rendered as a surface.

Meshes join at red boundaries.

Figure 10.14: A single object spanning multiple boundaries.

Interactive topological graph diagrams (Fig. 10.15). The contour tree and Reeb graph are difficult structures to present clearly due their non-planar form [162]. In the case of the Reeb graph things become more complicated by the presence of loops. In order to assist the user in viewing the structure of these graphs we have provided customised widgets for the task (Fig. 10.15). Using these widgets the user can scroll and zoom the data, and also change the position of node objects in the *x* axis to allow clearer presentation. This is particularly useful for viewing uncooled data where the graphs are most complicated, but is often required even for highly cooled data where clustered nodes are common.



Full Reeb graph.

Zoomed with manual layout.

Figure 10.15: Zooming on interesting regions of the Reeb graph. Nodes can move on the *x* axis.

Nested contours (Fig. 10.16). Whilst it can be beneficial from an exploratory point-of-view to view contours generated by specific isovalues, we also allow the user to view multiple nested contours. This was found to be a useful addition to the system, allowing the user to gain an overview of the data. Using this knowledge they could then identify isovalue ranges of interest to be further examined using the scrollable isosurface view. We allow the user to specify the number of contours required for the visualisation; this can be defined across the entire isovalue range or limited to those values specific to user selected objects.

Multiple nested contours across the entire scalar range of the field.

Nested periodic contours.

Figure 10.16: Using nested contours to better understand field structures.

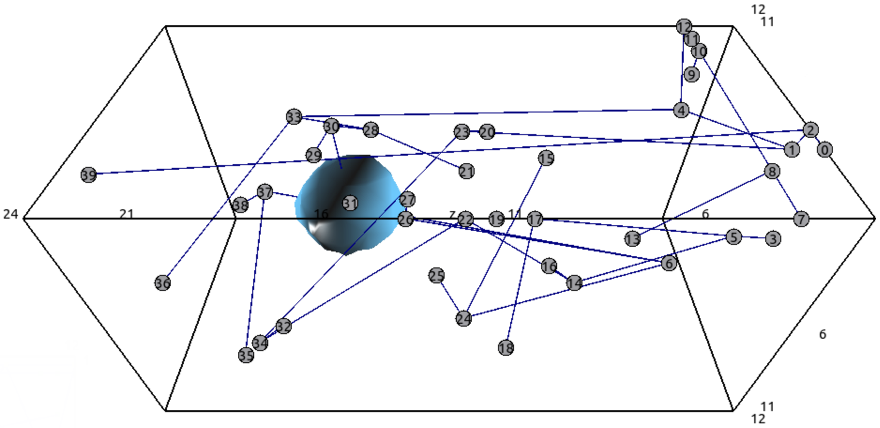


Figure 10.17: The Reeb graph can be shown alongside surfaces in the main view. Each vertex of the Reeb graph is drawn at the position of a critical point in the scalar field.

Embedding of Reeb graph in flexible isosurface view (Fig. 10.17). To help the user understand the link between the topological structure captured by the Reeb graph and the contour models, we allow the Reeb graph to be embedded in the 3D view. When displayed as an overlay to nested contour surfaces it provides a method of giving a spatial representation of the Reeb graph across the entire isovalue range. An alternative method is to overlay the Reeb graph on the scrollable contour view; this is a powerful way of demonstrating how changes in the isovalue affect topological structures in the scalar field.

Generation of joint contour slabs. As the project progressed from working with univariate scalar fields to bivariate data it becomes desirable to be able to load these representations of the data into the core application. Initially, this was used as a tool for understanding the relation between univariate (Reeb graph) and multivariate (Reeb space) topological structures. Using this approach a direct visual comparison of the slab structures of the JCN can be made against the univariate equivalent — the contour. In order to facilitate this functionality the user can select any two fields from the scalar field selection widget (Fig. 10.7.1) and request JCN analysis of the data. The user is able to select parameters of quantisation (slab size) for the two fields in powers of two, allowing a wide range of slab sizes to be covered in a simple interface. Other global parameters can be selected including the ability to switch periodic boundary conditions on or off.

Viewing joint contour slabs (Fig. 10.18). Later in the project the slab view was used to help make informed interpretations of the outcome of the various multivariate studies. We present the slab data from the JCN as three separate visualisations; two smaller slab visualisations are used to show the two input fields as discrete contour slabs alongside the main bivariate joint contour slabs. When viewed alongside the flexible isosurface representations of the same data this assists the user in choosing appropriate level of resolution in terms of slab size. Setting the slab size too high will mean that most features will be undetected in the JCN output; too low will result in slow computations and capturing of noise in the scalar field by the algorithm.

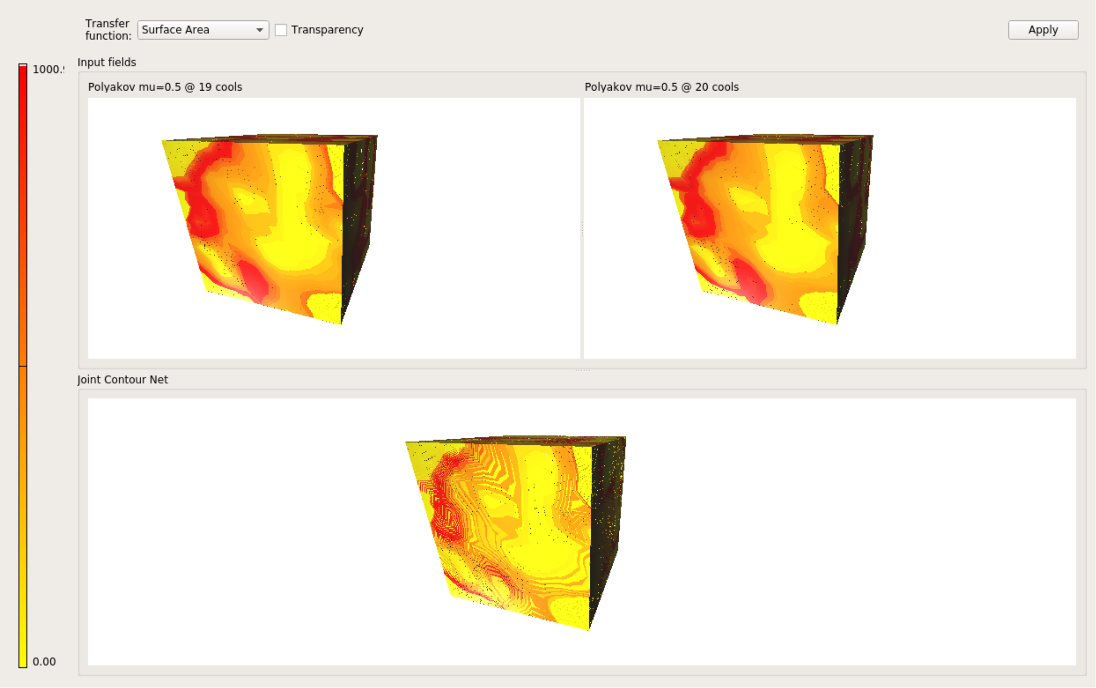
Comparison of multivariate persistence measures. Due to the bivariate nature of the data being visualised, there is no direct way of mapping colour to individual topological objects, as with the established isovalue-to-colour transfer function used for contours. Hence, we found the most logical method for assigning colour to objects was through user selectable persistence measures. We found that using persistence measures was a good method of highlighting boundaries between regions captured by the JCN.

10.2.3 Command line tools

The main purpose of these applications is to form part of an analytical toolchain that can evaluate the topology of multiple configurations in parallel on a desktop or cluster system. Here we give an overview of each of the applications developed.

10.2.3.1 Data slicer

This tool is used to reduce the dimensionality of the space-time hypervolume scalar fields by slicing along a user defined set of axes. Output files can represent a three dimensional volume



Transfer function: Surface Area

□ Transparency

Figure 10.18: The user examining Polyakov loop data at neighbouring levels of cooling. The two top visualisations show the input fields as quantised contours, and the bottom visualisation shows the combined output as joint contour slabs.

or two dimensional plane through the data. As lattice QCD treats all space-time dimensions as equivalent it is perfectly valid to slice and visualise the data on a non temporal axis.

10.2.3.2 Reeb graph computations

This command line application is used to generate a Reeb graph of a specified 3D scalar field using the VTK library [160]. Four dimensional fields must first be reduced to a three dimensional form using the slicing tool, described in Section 10.2.3.1. Building the algorithm into a stand alone application, rather than the core QCDVis visualisation module, allows the process to be easily automated for large ensemble studies. When the user intends to create Reeb graphs from within the main application, the process is executed as a background process.

A specialised version of the *vtkUnstructuredGrid* is used internally to allow the periodic boundary conditions to be respected for lattice QCD applications. In addition to computing the Reeb graph, which is written to file as a set of nodes and arcs with scalar values attached, we augment the Reeb with some additional data. In order to allow the Reeb graph to be embedded in the scalar view, we also attach the geometric position of each supernode, stored as a vector. The normalised height of each critical point is also attached for the purposes of analysis, for example using a Python script to parse the output in an ensemble study.

10.2.3.3 Reeb graph statistics

This application forms part of the automated tool chain for analysis of topological invariants in the context of an ensemble. It processes output from the Reeb graph computation tool (Sec. 10.2.3.2) and computes statistical distributions of objects.

When sweeping through the isovalue range the application computes the distribution of objects using two techniques; the number of objects at an exact isovalue, or alternatively the number of arcs in specified upper and lower bound. The user is able to specify exact limits of the distribution, along with the resolution across the height range. Information from this output can then be further analysed in context with output from other configurations or graphed using Python scripts. To make further analysis and graphing simpler the output in these files is designed to be easily read into NumPy [171] arrays.

A secondary use of the application is to perform a high level analysis of persistence from the Reeb graph. This information is written to a file designed for simple parsing using regular expressions. For each superarc in the Reeb graph we compute the function height range (persistence), we also output a number of measurements for the supernodes in the Reeb graph:

- Position in the scalar heightfield: *For computing proximity to other topological events.*
- The in and out degree: *Measures connectivity between critical vertices (supernodes).*
- The number of supernodes and superarcs existing above this supernode in the graph: *Used to predict how complex the field structure is above a particular object in the topological structure.*
- The number of supernodes and superarcs existing below this supernode in the graph: *Used to predict how complex the field structure is below a particular object in the topological structure.*
- The accumulated function height at the critical nodes above and below this supernode: *Used to give a very rough estimate of persistence.*

These values are designed to extract information about the connectivity of objects within a scalar volume. We found this to be particularly interesting in the context of the Polyakov loop data, which is predicted to have a roughly symmetric distribution when the lattice is in a confined state. In the case of deconfinement it is predicted the topology and distribution of object will see a breaking of symmetry.

10.2.3.4 JCN computations

This tool initially began as a translation, from Python to C++, of sample code included in META project library[163] for multivariate topological analysis. Over the duration of this project the code was modified to allow use as part of the ensemble tool chain by adding new features for the purposes of analysing lattice QCD ensemble data.

An initial modification to the tool was the ability to specify individual axes in the data as being periodic. Input fields to the data must be three dimensional; hence, they may require preprocessing via the slice tool (Sec. 10.2.3.1). The user may provide any number of inputs provided they are defined upon scalar fields of equal dimensionality. In the case that the user provides a single input the output JCN will be a discrete version of the Reeb graph, with a resolution defined by the given slab size.

We added a custom file format to save the structure of the JCN, allowing it to be analysed externally using regular expressions. In the header of the output we specify the input fields and their respective slab sizes, allowing the effects of quantisation to be analysed. For each vertex of the JCN we record the degree of connectivity, as this can be used a method of locating interesting features in the multi-field. For each field in the output we also extract the Reeb graph in its quantised form, allowing it to be further analysed using Python scripts.

Beyond the basic structure of the JCN we also allow the user to record other properties of the Reeb space. The number of Jacobi nodes are recorded for further analysis in the context of an ensemble study. To aid studies in multi-dimensional persistence the slab geometry can be saved as individual *Wavefront .obj* files to allow them to be visualised in QCDVis or analysed externally. As the number of individual slabs quickly becomes unmanageable, we offer the ability to compute statistical measure of the meshes as part of the analysis. For each JCN computation we can summarise the topology using the following invariants:

- Total number of slabs: *Each slab makes up part of the multi-field topology where the function gradients become equal.*
- Total number of polygons: *Slabs are closed meshes made up of polygon faces of various forms.*
- Total number of triangles: *We compute the total number of triangles needed to build the slab structures if the meshes are reduced to triangle meshes.*
- Total surface area: *We compute the net surface area when considering all of the slabs in the volume.*
- Slab mesh average: *We compute the average number of polygons, triangles, and surface area across the entire multi-field.*

10.3 User feedback

The QCDVis framework was developed following a nested model approach [172]. Close collaboration with physicists was crucial throughout the four stages of characterisation, abstraction, and design. The model was merged with a standard iterative design with each stage revised at the end of each cycle together with the domain experts. The approach we adopted allowed the domain experts to gradually familiarise with the framework raising suggestions, questions, and proposal of new directions for development of novel or alternative features.

In order to demonstrate possible uses of the QCDVis framework, we carried out user testing by implementing two case studies. The two tasks were designed in conjunction with the domain experts to reflect an optimal work flow for identifying topological objects within existing lattice QCD data sets. Data sets are obtained from the UK based DiRAC facility which computes and maintains a number of lattice QCD projects. Data is pre-cooled using code supplied by Hands and Kenny [173] — the process is beyond the scope of this thesis.

Here we first present an overview of user feedback, along with a more detailed description of the two case studies. A copy of the case study instructions can be found in Appendix C.

10.3.1 Overview and observations

Feedback gathered during the hands-on case studies sessions was largely positive with the framework welcomed as a new way of viewing and understanding lattice QCD data. Domain experts were taken aback by the possibility of visualising instanton shapes and computing physical variables such as surface areas and volumes, features they had never been able to visually inspect and measure before. Particularly in the case where an anomaly was detected the visual feedback compelled a new way of interaction with the data.

New suggestions on how to enhance existing visualisations were also brought forward; for example, the ability to view ensemble average isovalue histograms in addition to configuration histograms on the global isovalue slider. It was felt that this enhancement could further steer users towards interesting features in the data that were previously beyond the reach of physicists who were unable to calculate the topology of their data.

One domain expert was particularly interested in the possibility of using the program to view data from other simulations belonging to research projects he is currently involved. Their work focuses upon 2 colour configurations generated by the MILC collaboration [174] using a variation of the cooling algorithm. They believed that the tool represents an interesting method for visually examining the difference between differing cooling algorithms. The existing modular design of QCDVis should allow the relevant file format to be incorporated into the program with minimal effort.

Additionally, we noticed a difference in the ability to make full use of the contour tree or Reeb graph between physicists that had been involved in the development of the framework from the very beginning. Experts with less prior experience of using the contour trees tended to focus more on the isosurface representations of the data. However, as the interaction with the framework progressed, they felt that the contour tree and Reeb graph were offering interesting tools for obtaining an overview and navigating through the data.

10.3.2 Case study: Locate and visually inspect an instanton within a configuration

This task demonstrates the core purpose of QCDVis by enabling visual inspection of lattice observables. A $16^3 \times 32$ ensemble from the Edinburgh DiRAC server is provided for the task with μ set to 0.7. The ensemble has been cooled for 30 iterations and is known to contain (anti-)instantons; the task is to locate the (anti-)instantons and view them as objects in the topological charge density and space-like/time-like plaquette fields. Existing methods inherited from statistical physics are able to identify the location of minima and maxima within the Wilson energy and topological charge density fields. However, they are unable to reveal much about the structure of these observables or the surrounding features within the scalar fields at these points.

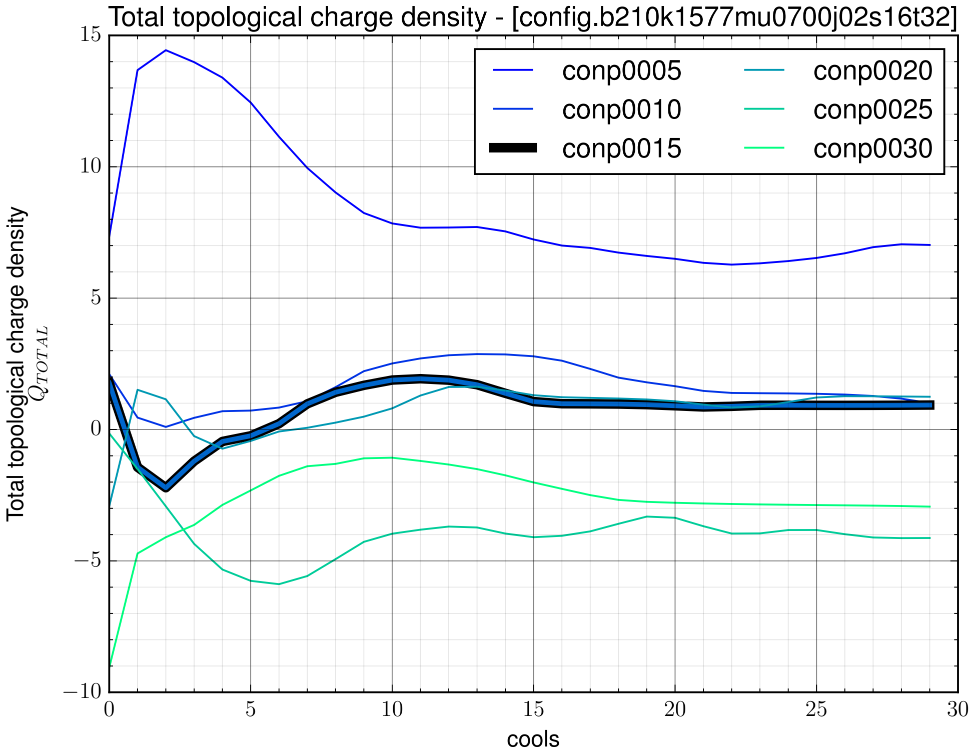


Figure 10.19: A flattening in the topological charge density indicates a region of stability that can be cross referenced against the peak action graph to select interesting levels of cooling for study.

Upon selecting an ensemble, an established first step for QCD researchers is to plot the action and topological charge density quantities for each configuration in the ensemble across the cooling domain. This allows them to select a level of cooling at which they would typically conduct a study using ensemble data. This is achieved by looking for signatures in the plots; of greatest interest is that of topological charge density, where flat areas of the graph indicate a stabilisation of the number of topological objects on the lattice (Fig. 10.19). Cross-consulting the action max graph (Fig. 10.20) makes it possible to observe that the topological charge objects gradually increase in size within the region of stability, before merging at around 21 cools. QCDVis supports physicists in this task by automating the task of generating each graph.

To conduct a study on the physical form of the objects in the data, a configuration with a pronounced flat region was chosen. Using the **ensemble module** (Sec. 10.2.1) participants were able to select 3 levels of cooling for the desired configuration. Field variables, in both the topological charge density and action fields, are important in this case study. To understand the structure of the data, the researchers chose to represent the action density in the time-like and space-like configurations. This enables them to look for obvious similarities in both variations, along with in the topological charge density field.

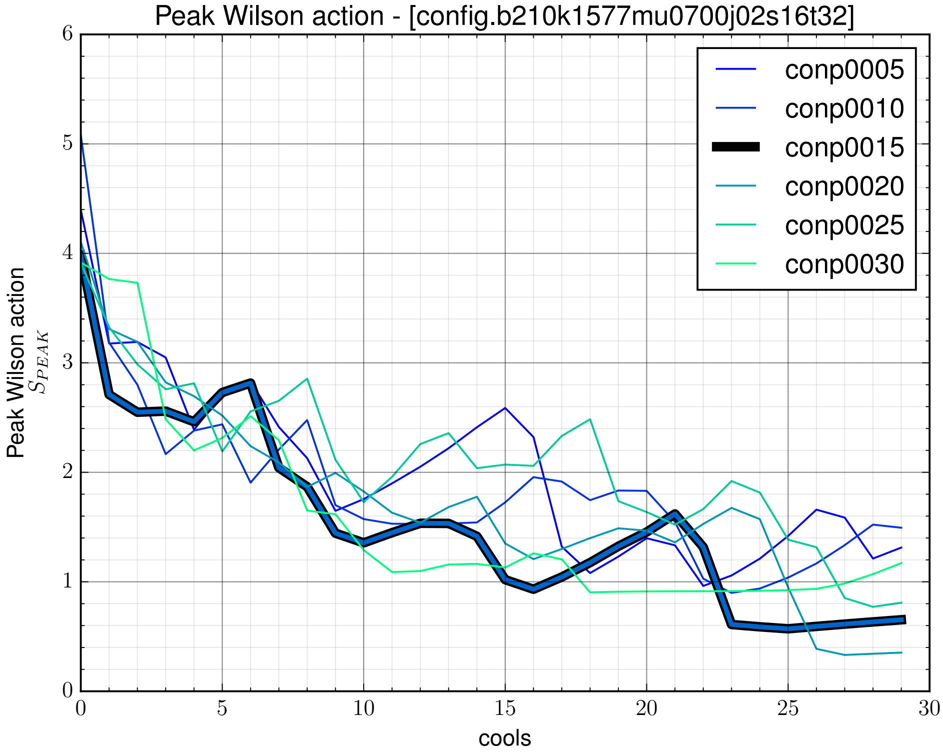


Figure 10.20: Spikes in the action graph indicate the spreading and smoothing of an instanton, followed by a merging of two or more distinct objects.

After computing desired field variables, focus switches to the **visualisation module** (see Sec. 10.2.2). Each field variable is loaded concurrently into the tool and the hyper-volume data cut into 3D configurations. Feedback about the location of minima and maxima steers the researchers toward areas of interest for each field variable, observable in the contour tree. Researchers tended to favour the 3D volumes split by time to locate variables in each field. The global isovalue slider allows them to compare the relative similarities in each field in parallel. Additionally, researchers study the location and structure of objects in the time-like 3D volumes (*xyt*, *yzt*, *xzt*) to validate that the objects coincide in all view configurations (Fig. 10.21).

10.3.3 Case study: View an irregularity within a cooled configuration

This study seeks to resolve an abnormality found within the data and demonstrates how QCD-Vis can be used to view and understand the cooling process in greater detail. For this task a $16^3 \times 32$ ensemble from the Edinburgh DiRAC server is provided; however, in this ensemble μ is set to 0.45. The data has been cooled for 100 iterations to try to emphasise the phenomena of observables dropping through the lattice. An unexpected irregularity is found within one of the configurations, where a positive maxima becomes a negative minima in successive cooling

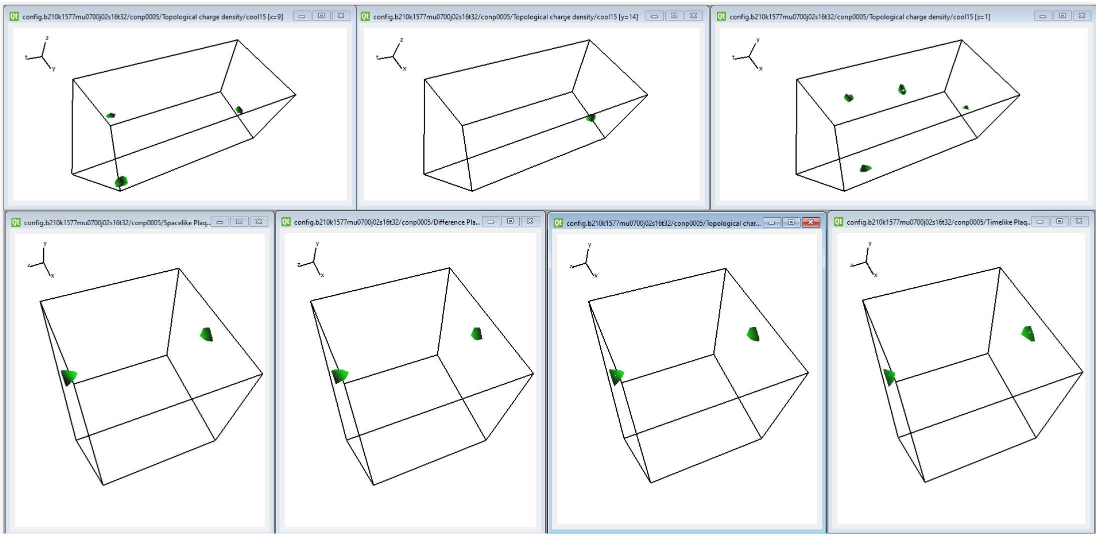

config.b210k1577mu0700j02s16t32/conp0005/Topological charge density/cool15 [x=9]
z
t
y
config.b210k1577mu0700j02s16t32/conp0005/Topological charge density/cool15 [y=14]
z
t
x
config.b210k1577mu0700j02s16t32/conp0005/Topological charge density/cool15 [z=1]
y
t
x
config.b210k1577mu0700j02s16t32/conp0005/Spacelike Plaq...
config.b210k1577mu0700j02s16t32/conp0005/Difference Pla...
config.b210k1577mu0700j02s16t32/conp0005/Topological char...
config.b210k1577mu0700j02s16t32/conp0005/Timelike Plaq...
y
z
x

Figure 10.21: Top: user examining the position of topological charge density minima (anti-instanton) in the *xyt*, *xzt* and *yzt* sub-volumes. Bottom: comparing the location of peak Wilson action and topological charge density minima in *xyz* sub-volumes.

cycles. The task intends to visually inspect this happening and assign reason to it.

Whilst 100 cools represents a vast over exaggeration of how cooling would be applied in a real study, it promotes the possibility of witnessing an instanton *falling through the lattice*. This occurs when an instanton is shrunk to exist at a single point, whereas typically, instantons exist across multiple neighbouring sites. In a typical QCD study this can only be detected using plots of topological charge and action by looking for known signatures.

As with the first case study, upon loading a configuration the first thing the users do is to consult the topological charge and peak action graphs. However, in this scenario the focus is a gradual increase followed by a fairly instantaneous drop in the peak action graph (Fig. 10.22). Usually this will indicate one of two events; first, an annihilation of an instanton/anti-instanton pair; or second, an indication that an object has *fallen through the lattice*. Features in the maximum topological charge graph (Fig. 10.23) suggest that rather than an annihilation, as would be signified by a distinct drop, something else is happening.

Further investigation takes the form of computing the topological charge density at 3 consecutive levels of cooling. The field data is loaded into the visual tool so that it can be investigated in parallel. Again, physicists instinctively open the space-like configuration (*xyz*) to search for highlighted minima and maxima. Upon opening the relevant time slice for all three cooling slices, the most immediate feature is a distinct change in the contour tree for each volume. The slices before the peak show a relatively large maxima beyond the noise at the centre of the data set; however, the data on the downward slope instead shows that the maxima has disappeared to be replaced by a minima of roughly equal magnitude.

Visual inspection allows the physicists to look at the phenomena from different angle and formulate hypothesis about its provenance and nature. As can be seen in Fig. 10.24, this irregularity is characterised by a continual deformation of the object relating to a peak in topological

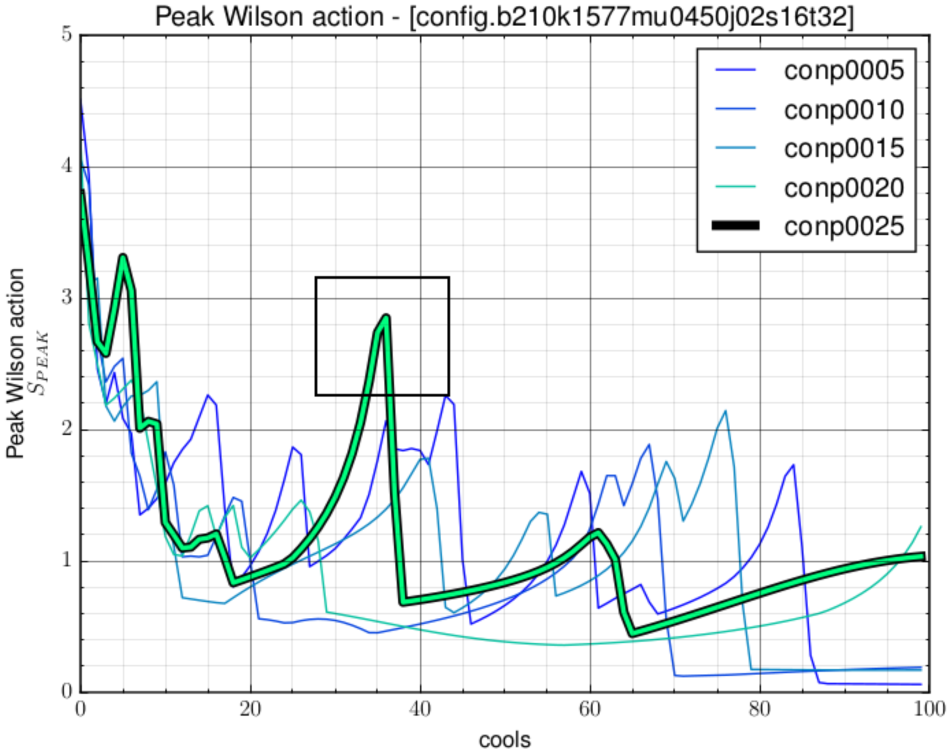


Figure 10.22: A plot showing Peak Wilson action on the *y*-axis against cooling. When choosing a configuration and level of cooling for evaluation, physicists look for a flattening in action.

charge density. Additionally, the minima in the field appears to exist in the same position on the lattice, whereas there is no objects of the relevant isovalue in the earlier slices. An exact reason for this is not yet known, but physicists that have witnessed it were able to reflect upon and develop several possible hypothesis.

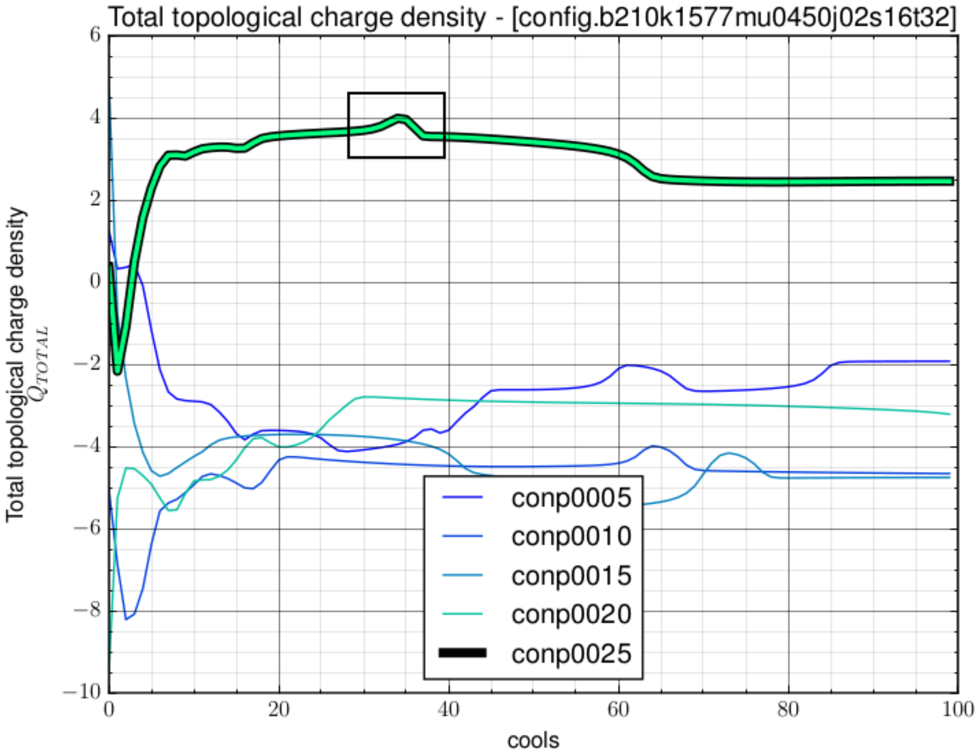
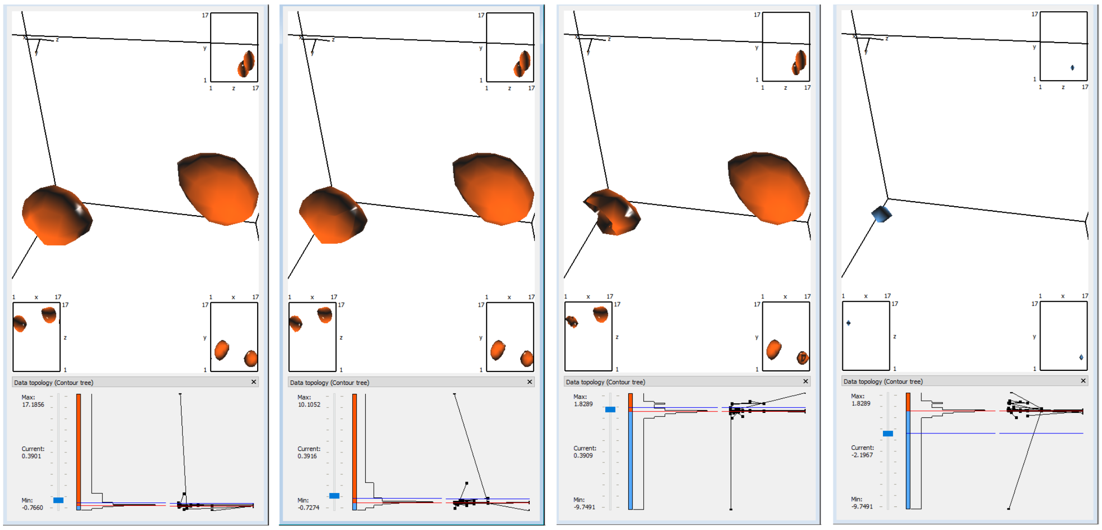


Figure 10.23: A plot showing total topological charge on the *y*-axis against cooling. When choosing a configuration and level of cooling for evaluation, physicists look for a flattening of the topological charge.



Data topology (Contour tree)
Max: 17.1856
Current: 0.3901
Min: -0.7660

Data topology (Contour tree)
Max: 10.1052
Current: 0.3916
Min: -0.7274

Data topology (Contour tree)
Max: 1.8289
Current: 0.3909
Min: -9.7491

Data topology (Contour tree)
Max: 1.8289
Current: -2.1967
Min: -9.7491

Figure 10.24: Top: Cooling slices 35 and 36, shown at the same isovalue. Bottom: At step 37 the large object on the left starts to deform and disappears, being replaced by a global minima. This phenomena can also be observed in the contour trees for each cooling phase.

10.4 Summary

In this chapter we have discussed in detail the software developed for visualisation of lattice QCD data. The software was used extensively throughout the entire project, with its role continually evolving. Initially, it was used as a tool to allow physicists to explore their data. It was later used to analyse anomalies and features present in the data in order to gain a greater understanding of the data — particularly when subjected to cooling. In later stages of the project it was used to help interpret results of ensemble case studies.

This chapter featured the following contributions:

- A fully interactive user interface was presented that enabled users to explore lattice QCD data sets.
- We discussed how the framework uses topological visualisation techniques to visualise and analyse various lattice fields.
- We presented the two main parts of the user interface; a tool for choosing potentially interesting lattice fields, and an interface supporting topological visualisations of univariate and bivariate lattice fields.
- A case study was carried out and presented as part of the collaborative work with lattice QCD scientists.

Chapter 11

Conclusion

Contents

This thesis discusses the use of topological visualisation techniques to display and compute properties of lattice QCD data sets. In pursuit of the objective we presented several different approaches — with a focus on topology based algorithms. The work was carried out in close collaboration with domain scientists ensuring the presented solutions suited the requirements of a typical lattice QCD researcher. Beyond simplifying the visualisation process, we believe that topology presents a number of new and valuable techniques for domain scientist to use when analysing their data.

In Chapter 2 we presented a detailed overview of volume visualisation techniques. This began with a brief introduction to the subject, highlighting two different approaches to displaying scalar data. The first, direct volume rendering, creates images directly mapping scalar values to properties such as colour and opacity using transfer functions. The second approach, indirect volume rendering, creates a mathematical model that is used to manipulate and display the data. It is this approach that is further investigated by examining one of the core algorithms, marching cubes, in depth. We also discuss the use of topology to capture information about connectivity of the models created in indirect volume rendering. Different algorithms used in topology driven visualisation are also discussed in Chapter 2 where we examine the uses and limitations of each. Finally, we discussed generalisation of those methods into higher dimensional data sets and approaches for viewing multivariate data.

An introduction to Quantum Chromodynamics was given in Chapter 3. We discussed why QCD research is important and what domain scientists expect to gain by studying it. A discrete version of the theory, suitable for use on computers, was presented that uses a 4D hypercubic lattice to model the theory. We discussed the need for multiple configurations to be computed as part of a ensemble studies using predefined parameters. In the conclusion of Chapter 3 we showed the existing state-of-the-art in lattice QCD visualisation.

The following bulleted list restates the main contributions of each chapter. Section 11.1

gives a more detailed discussion of applying topological visualisation techniques to lattice QCD and in Section 11.2 we discuss future avenues of research.

- Chapter 4 discussed the approach used to parse existing lattice QCD configurations; validation played a key role in this process and was discussed in detail. We also introduced the lattice field calculations used to generate each of the fields studied in this thesis. We then presented initial visualisations of each field using general purpose visualisation software as previously used in lattice QCD research. This demonstrated how using isosurface based representations of the fields, as opposed to direct volume rendering techniques, enabled us to observe the presence of discrete lattice objects.

- Chapter 5 presented details how about the isosurface visualisations of the previous chapter influenced our decision to use topology based algorithms in our research. Several topological visualisation algorithms were discussed that utilise graph structures to capture the topology of scalar fields. We also discussed one of key issues we faced whilst completing this work — the effect of periodic boundaries and showed how using the Reeb graph, instead of the contour tree, addressed the problems encountered. Chapter 5 is concluded by a discussion of the potential uses for topology in lattice QCD which is demonstrated by the computation of physical properties of individual lattice observables.

- Chapter 6 further demonstrated the use of topological algorithms presented in Chapter 5 by performing an in-depth case study of the topological charge density field. Through the use of topological structures we were able to visually inspect a configuration as multiple distinct objects. We were also able to show how the Reeb graph could be used to summarise a single lattice configuration and quantify properties of the configuration as a whole. The case study also showed how the structure of key observables, (anti-)instantons, varied under the influence of the cooling algorithm. Basic persistence measures computed directly from the Reeb graph were able to capture this information in a format suited to use in larger studies. However, we pointed out limitations of performing these studies in 3 dimensions rather than the native 4 dimensions.

- Chapter 7 used an alternative algorithm for capturing topology that is able to generalise to multivariate and temporal inputs. We evaluated the use of the JCN in a case study that compared multiple time slices of data from the same input configuration. Using this approach we were able to track the continual evolution of an anti-instanton as the time index of the data was varied. We were surprised to see that the object persisted in the data for several more time steps than we had initially expected. Meshes of the object computed via the JCN also showed visual similarity between time steps — highlighting that the object retained much of its structural integrity despite the relative changes in energy. Multivariate persistence measures allowed statistics to be computed about the structure of the object as the time parameter varied. We showed how use of the JCN allowed to us to make predictions about the point at which the object appeared and disappeared as a unique entity on the lattice. Finally, we showed an experimental use of the algorithm where we used eight time-slice inputs to compute a summary of the topology in four dimensions.

- Chapter 8 demonstrated a different use for the JCN where it was used to extract interesting features existing on the lattice as combinations of two fields. Comparison of multiple bivariate fields enabled an (anti-)instanton to be observed as a function of many bivariate field combinations. We made some unexpected observations about how the object appeared; in particular we found variations in the topology of fields with space-like and time-like plaquette inputs gave interesting results. Consultation with domain scientists suggested that studying these fields as thermodynamic control parameters were varied could potentially give interesting new insights into the reaction of lattice observables to changes. We also introduced additional topological structures in the form of the Reeb skeleton which summarised information in the JCN in a simplified format. This allowed additional persistence measures to be captured in the data and presented the ability to perform simplification of the data.

- Chapter 9 used knowledge gathered in the proceeding chapters to perform a study of lattice topology in the setting of an ensemble study. The case study is an example of how topological visualisation techniques could be used by domain scientists where large quantities of data must be analysed. The JCN is used to evaluate the effect of cooling on the lattice by evaluating differences in the Polyakov loop field as the algorithm is iteratively applied. We show how quantities taken directly from the JCN graph and through the use of bivariate persistence measures were able to differentiate properties of the lattice as temperature and chemical potential are varied.

- Chapter 10 gives a detailed overview of a visualisation framework developed in collaboration with domain experts. The software is used to load existing lattice configurations, allowing the user to employ domain knowledge to select interesting configurations for computation of various lattice fields. Each of the fields can be viewed in the visualisation module as separate scalar fields, or in bivariate combinations, alongside the output of various topological algorithms. We discuss how the software was designed and implemented as a continual cycle of rapid development and user feedback. The effectiveness of using topological visualisation techniques was reviewed in the form of case studies. Feedback from these showed that physicists saw the tool as a potential way of asking new questions of their data — for example to compare the effect of different cooling algorithms.

11.1 Summary

Lattice QCD is a field of science that can benefit greatly by the adoption of topological visualisation techniques. The model based approach used to generate the visualisations also provides an interesting method for computing previously unattainable properties of the data as thermodynamic control parameters are varied. Whilst the huge quantities of data make visual analysis of all configurations within an ensemble an infeasible task, we were able to show how initial exploratory visualisations can help define a set of properties within the topology that can be tested at the ensemble level.

To help facilitate this proposed work flow we have presented a visualisation framework in Chapter 10 that we feel could help lattice QCD physicists design new experiments in the future. Previous tools such as those presented by Leinweber [137] and Di Pierro [140] offered features commonly used by physicists, combined with visualisations to statically display the data. We were instead able to use topological visualisation techniques to segment the data and allow the user to interact with it dynamically. The ability to view data from multiple fields in parallel within the application allows the user to make observations about correlations between lattice fields. This was found to be particularly useful when examining the plaquette fields and the topological charge density in parallel as the values in the plaquette fields contribute to topological charge density. Viewing surfaces extracted from the data alongside the topological structure captured by the Reeb graph also allows new understandings to be formed in less ordered fields such as the Polyakov loop.

Use of the Reeb graph in the visual analysis greatly enhanced understanding of the diverse array of scalar fields present in lattice QCD. A number of unexpected phenomena were witnessed in the data that had not been predicted prior to the visualisations. Isolating the core observables using the Reeb graph and contour tree demonstrated that the shape of lattice observables are sensitive to the proximity to nearby objects.

Gaining a better understanding of the cooling process is also a possibility through use of topological visualisation techniques. Whilst performing the user case studies we witnessed what we believe to be an object “falling through the lattice” as an effect of over cooling. This was observable in the Reeb graph as the persistence measures of the object gradually shrank as it merged into the surrounding scalar field before finally disappearing. Cross referencing the behaviour present in the Reeb graph with rendered contours showed that the object structure gradually warped before splitting into two distinct object prior to disappearing from the lattice at higher levels of cooling. The cooling algorithm showed complex behaviour when viewing the Reeb graph in combination with a view of nested contours across the isovalue range. Fluctuations in the number of topological objects close to isovalue zero showed a turbulent behaviour where topological objects split into smaller pieces before re-merging under additional cooling.

In this thesis we also showed two different potential uses for multivariate topology in lattice QCD; understanding the effects of an evolutionary component such as time or cooling, and finding correlations between different fields. The JCN provides a graph structure for representing the quantised Reeb space of multiple variables as a number of regions with common isovalues. The Reeb skeleton provides a simpler overview of the data, also taking multivariate persistence measures into consideration. A number of other observations were made throughout the studies into how lattice QCD data is captured by the JCN. We found that all JCNs typically featured one very highly connected vertex at the centre of the graph. Visual inspection of the slab geometry represented by the vertex showed that it was used to capture the region of peculation around isovalue zero in each field. The degree of connectivity of this vertex gave a basic measure of the number of distinct objects existing on the lattice, both positive and negative, as typically most objects would branch from this region of the scalar field.

Additionally, we found the Reeb skeleton to be a useful tool for summarising the data; however, in many cases the core observables were removed due to a general, rather than domain specific, definition of interesting phenomena in the simplification algorithm. This effect was made much worse when simplification was applied to the Reeb skeleton to remove noise. Hence, we often directly manipulated the JCN in order to select interesting features using the additional context of isovalues relayed using coloured glyphs. When the Reeb skeleton was used to view objects, it often separated the object from the surrounding lattice data at a higher isovalue than manual selection. This leads us to question how to define an absolute isovalue to be used as a cut off in separating an (anti-)instanton from the surrounding lattice field.

We also demonstrated the use of persistence measures on lattice QCD objects in order to make predictions of their physical properties. Encouragingly when applying these techniques the values appeared to agree with predictions made by lattice QCD theory. In particular, the volume of the instantons remained more-or-less constant under cooling, until disappearing from the lattice. The persistence measures also showed that the topological fluctuations appear to have little effect on the core observables in lattice QCD.

An optimal solution for simplification of multivariate lattice QCD data may instead require using domain specific knowledge. For the purposes of isolating (anti-)instanton structures a filter that took self-duality into account during the simplification process would provide specialised focus on observables. This method of simplification relies on knowledge that the principle observables of lattice QCD exist as minima and maxima in the topological charge density fields. These details in combination with persistence measures, such as volume, could provide a more robust simplification technique.

We have presented a potential use of multivariate topology for analysing data from lattice QCD ensemble data sets. The quantity of data required for analysis due to the effects of quantum mechanics means that visual inspection becomes an unfeasible task. Instead we have shown how a number of measures taken directly from the JCN can be displayed as thermodynamic control parameters are varied. Through the use of ensemble averages it is possible to understand if patterns present in the multivariate topology share a correlation with existing statistical physics predictions.

Finally, we showed how an experimental use of the JCN algorithm was able to capture an instanton observable in a multivariate field defined over eight scalar fields. The JCN was able to capture the instanton and various other lattice objects as individual branches in the topology. The algorithm took longer to compute a JCN for this data, as expected; however, the computation time was in the range of minutes. It would be expected that the ability to use the JCN effectively in this way would be limited as data sets grow, as the memory and time constraints are linked to the number of fragments generated by the algorithm [11].

11.2 Future work

Based upon our experiences of working with lattice QCD data and the topological algorithms discussed in this thesis we would like to recommend the following future avenues of research.

High dimensional inputs. Many lattice fields in lattice QCD are defined in four space-time dimensions; the Reeb graph algorithm present in VTK is limited to inputs defined in up to three dimensions. In order to visualise the data we utilise slicing, allowing each field to be viewed in three dimensions from different points-of-view. Output from the Reeb graph algorithms does not provide a consistent ordering for the superarcs or supernodes, making matching graph matching between time slices a difficult task. Existing research has been carried out in this area, using the merge trees [7] and the Reeb graph [8] but this is further complicated by changes in connectivity, such as merging and splitting of objects, between slices. However, in this work we used the JCN to evaluate connectivity between individual time steps — a technique for evaluating time dependent scalar data proposed by Edelsbrunner et al. [175].

An alternative approach is to sample the data directly on a four dimensional input field. Whilst theoretically this is a simple prospect, in reality the complexity the simplicial decomposition of hypercubic cells, makes this a non-trivial problem. The simplest solution sees hypercubes decomposed into $n!$ cells making computation increasingly expensive as dimensionality is increased. As far as we are aware there are currently no openly available libraries that allow the Reeb graph to be computed for higher dimensional inputs.

Periodic boundary conditions. The Reeb graph is able to compute the topology of scalar fields with non-simple (periodic) boundaries. However, the flexible isosurface algorithm [76] uses the contour tree as its internal data structure — meaning that it is unable to correctly handle the boundary conditions of lattice QCD for the purpose of generating surfaces. Attempting to match open meshed objects together at run-time is a non-trivial task as the objects represented by the superarcs continually change shape and connectivity, making simple polygon matching a difficult and unreliable process. Creating flexible isosurfaces using the Reeb graph would require modification of the algorithm but also would present new challenges in how to present data spanning a boundary as a closed manifold.

Comparison of cooling algorithms. At present the system has been used to view data from a specific set of 2 colour QCD experiments, which it is able to handle with a satisfactory computation time. As a result feedback from the case studies, we have also had interest in using the tool on two colour simulations from the MILC QCD project. These differ from the existing configurations in using a continual form of cooling. We therefore envisage the use of topology to help physicists evaluate the relative merits and side effects as a result of the choice of cooling algorithm.

Computation of multivariate topology without slab size parameters. We feel that some of the results we collected were impacted upon by the slab size selection. Often objects, such as (anti-)instantons, would be represented by multiple small fragments as a side effect of aliasing due to slab size. Since the case studies were completed an improved algorithm has been proposed by Tierney and Carr [176] which computes the Reeb space without the need for quantisation parameters.

Appendix A

Glossary of terms

n-**manifold** A closed surface existing in *n* space-time dimensions. A list of common *n*-manifolds is given in Table 2.4.

2-**skeleton** A mesh defined using *simplices* in dimensions 0 to 2; these are vertices, edges, and triangles.

Augmented Contour Tree Graph like structure representing topology with additional information attached to vertices and edges. For example: *path seeds* or *Betti numbers*.

Adjoint matrix A complex matrix generated by transposing the input and taking the *complex conjugate* of the entries.

(Anti-)instanton A four dimensional pseudo-particle characterised by minima / maxima points in the *topological charge density* field.

Betti number Set of invariants used to define properties of a topological object. b_0 represents the number of connected components, b_1 the number of holes, and b_2 the number of voids in an object.

Chemical Potential (μ) A thermodynamic control parameter in lattice QCD simulations believed to induce changes in the instanton / anti-instanton balance.

Cofactor Matrix A square matrix where each (i, j) element is a cofactor, obtained by multiplying the*first minor* by $(-1)^{i+j}$.

Complex conjugate A complex number produced by negating the sign on the imaginary part of the input, the real part is unchanged (Fig. 3.9).

Configuration A unique lattice representing the gluon field at a specified point within the generating Markov chain.

Contour A subset of connected components within an isosurface. An isosurface may consist of multiple disjoint contours.

Contour tree A special variation of the *Reeb graph*. The contour tree is only defined for simply connected domains.

Cooling Iterative algorithm used by physicists to remove noise and stabilise output from lattice QCD simulations.

Critical point A sample where the differential of the function is zero.

Degenerate critical point A critical point where the second order partial differentials (the Hessian matrix) is a matrix with the determinant zero.

Ensemble A collection of configurations created using predefined thermodynamic control parameters.

Euclidean space-time Four-dimensional representation of 3D Euclidean space with an additional simplified time parameter.

Fibre A generalisation of level sets to multiple variables. They can be, but are not limited to being, connected regions of the multi-field.

Fibre component These are multivariate equivalents to connected components in univariate topology.

First minor The determinant of a submatrix formed by deleting the i^{th} row and j^{th} column of a square matrix.

Fragment A section of a *joint contour slab* existing in a single cell of the input field.

Gauge compression A method used to reduce the size of lattice configurations by reconstructing $\mathsf{SU}(n)$ matrices using the complex conjugate of elements.

Hypercubic lattice A lattice existing in four dimensions made of hypercubic cells (see Fig. 3.6).

Jacobi set A set of critical points for m functions where the function gradient are parallel.

Jacobi structure The Jacobi set projected into the Reeb space. Critical vertices in the Multi Dimensional Reeb Graph represent the Jacobi structure.

Join tree Tracks connectivity in the *sub-level set* as the isovalue is increased.

Joint Contour Net (JCN) A graph structure that summarises the discrete Reeb space by assigning vertices to regions. Adjacency between regions is indicated by undirected edges.

Joint contour slab A surface existing in the Reeb space of n functions where the discrete isovalues of each remain constant.

Layered Reeb graph An alternative approach to capturing the Reeb space — similar in implementation to the Multi Dimensional Reeb Graph.

Level set A set of points in $\mathbb{R}^n$ space that have a common isovalue c. Defined as $L_c(f) = \{(x_1,...,x_n) \,|\, f(x_1,...,x_n = c)\}$.

Link variable An $\mathsf{SU}(n)$ matrix representing the gluon potential emerging from a lattice site along a specified direction.

Merge tree The generic name given to split and join trees. Both are related structures.

Minimal seed set A set of seed cells guaranteed to intersect each contour at least once.

Morse function A function sampled at discrete points where all *critical points* are *non-degenerate*.

Multi Dimensional Reeb Graph (MDRG) An alternative approach to capturing information about the Reeb space using a hierarchical structure of individual Reeb graphs restricted to the values of those proceeding it.

Path seed Seeds extracted directly from the contour tree. Typically the seeds represent an edge in a cell intersecting a contour.

Periodic boundary A method for implementing a translational invariant axis where the boundaries are connected to form a loop.

Plaquette An elementary square loop around links (edges) on the lattice from an origin site.

Polyakov loop A 3D lattice observable computed for each spacial site on the lattice (see Section 4.3.3).

Rank invariant A generalisation of the univariate *persistence barcode* to multivariate data. Used to capture a number of persistence invariants similar in form to the Betti numbers.

Reeb graph A graph structure that captures the topology of an n-dimensional manifold using vertices to represent *critical points*.

Reeb skeleton A simplified method of representing the *joint contour net* where vertices with trivial topology are collapsed into a single vertex.

Simplex A generalisation of the triangle in n-dimensions.

Simply connected domain Examples of simply connected domains include the sphere. A torus is non-simply connected because it contains a hole at its centre.

Site Name commonly given to a vertex on the lattice represented by a coordinate 4-tuple (x, y, z, t).

Space-like plaquette A plaquette existing in the XY, XZ or YZ plane (see Fig. 4.6).

Special Unitary group $\mathsf{SU}(n)$ A group of unitary square matrices with determinant 1.

Split tree Tracks connectivity in the *super-level set* as the isovalue is decreased.

Sub-level set The set of points in $\mathbb{R}^n$ space with an isovalue less than or equal to a constant c. Defined as $L_c^-(f) = \{(x_1, ..., x_n) \,|f(x_1, ..., x_n \leq c)\}$.

Super-level set The set of points in $\mathbb{R}^n$ space with an isovalue greater than or equal to a constant c. Defined as $L_c^+(f) = \{(x_1, ..., x_n) \,|f(x_1, ..., x_n \geq c)\}$.

Time-like plaquette A plaquette existing in the XT, YT or ZT plane (see Fig. 4.6).

Topological Charge Density A four dimensional observable computed for each site on the lattice (see Section 4.3.2).

Wilson loop An arbitrary closed loop computed by a path ordered traversal of links from a site on the lattice.

Appendix B

3D graphics techniques

A 2006 survey of techniques used in interactive rendering suggested how the emergence of programmable pipeline graphics methods could be used to simplify and increase the performance of real-time visualization applications [177]. Shader languages, including OpenGL's GLSL and DirectX's High Level Shading Language (HLSL), offer the programmer a greater deal of control by allowing them to programmatically state how parameters including lighting and shading are handled by the hardware. Complex lighting is now easily calculated entirely on the graphics card using custom written shader applications, able to fully utilise the hugely parallel nature of the GPU. Previously this would typically be handled in a serial manner on the CPU or via algorithms provided by the graphics library vendor.

B.1 Shading and lighting algorithms

Computer generated images are largely dependent on high quality lighting techniques in order to produce realistic output. This is especially apparent in three-dimensional projections where shading and reflection are used to relay a feeling of depth to the viewer. Often these techniques require additional information in the form of face or vertex normals and surface characteristics, typically generated at the modelling stage. With the advancement of high level shader languages implementation of different rendering styles becomes a matter of coding new shader programs and swapping with an existing shader, allowing the scene to be rendered in a completely different style.

Early computer graphics techniques were largely subject to limitations in hardware; a trade-off had to be made between producing realistic images and speed. Prior to 1967, 2D projected 3D objects were devoid of any shading characteristics and were rendered using wire-frame overlays to highlight individual polygons. Watkins [178] computed shading on a face-by-face basis that gave meaningful results on discontinuous objects but gave unnatural looks to those that were curved, with individual polygons clearly visible. In 1971, Gouraud [21] used interpolation to calculate shading intensities between vertices. This allowed images to be produced with much improved continuous shading characteristics, whilst impact on overall rendering speed was minimal.

Phong-shading [179] was proposed to take Gouraud shading a step further by complementing shading with surface properties. This allowed specular effects to be calculated, using reflection and surface properties, to produce a lighting effect that is relative to the position of lighting and observer. However, a limitation of Phong-shading is that it gives ambiguous output when a surface transparency element is introduced. In these situations, often the viewer will experience difficulty when making sense of structure from depth characteristics; hence, additional details may be required in the form of surface highlights.

B.2 High level shader languages

Graphics hardware saw dramatic increases in computing power at the turn of the millennium as a result of improvements in silicon manufacturing technology. Graphics cards often feature a number of separate parallel processing nodes, a departure from the centralised CPU model. GPU processing requires a different strategy than is employed in CPU centred projects; hence, specialised frameworks are provided for the task.

NVIDIA was the first GPU manufacturer to provide this with their programmable vertex engine [180]. The language was based upon a set of 17 assembly instructions and forbid certain operations, such as conditional branching, in the interest of hardware simplicity and constant latency. It was felt that traditionally parallel programming had been a specialist task, NVIDIA wanted to change this by bringing parallel processing into the mainstream. The language can be considered as a forerunner to fully featured shader languages, introducing a number of concepts that would appear in later implementations. To follow its basic assembly language shaders, NVIDIA created *Cg*, or C for graphics [181]. This was a high level shader language, syntactically similar to C, that allowed programming of graphics hardware using common data types including floats, vectors and matrices. However, the language was also capable of more general graphics card programming and has since been superseded by NVIDIA's CUDA parallel processing libraries and Microsoft's shader language, HLSL.

HLSL was co-developed by NVIDIA and Microsoft alongside Cg, to form a shader language for DirectX [182]. The architecture of the system defines two distinct elements; the vertex processor and the fragment (or pixel) processor, each working on a single entity per program. This basic architecture of separate vertex and fragment shaders is common to the shader language used in OpenGL; Figure B.1 gives an overview of the processing pipeline. The vertex processor is used to take vertices from object space and transform them into screen space; typically this is achieved by multiplying a vertex by model, view and projection matrices. Output from the vertex processor is then rasterized, before being processed by the fragment processor. This is responsible for calculating the final output colour of a pixel, using lighting information passed through from the vertex shader and surface properties to come to a final result.

OpenGL's shader language, GLSL, was introduced in OpenGL version 2.0 [183]; however, there was also prior support for Cg. By bringing a fully featured high level language to the platform, visual effects could be generated in real time that were previously only available in off-line rendering platforms, such as RenderMan [184]. From the outset, the idea was to create a powerful language that could fully exploit graphics hardware but performed at assembly code speeds [185]. GLSL was built from experience gained in other shader languages; hence, a minimal set of data types were supported consisting of floats, integers and Booleans. Vectors and

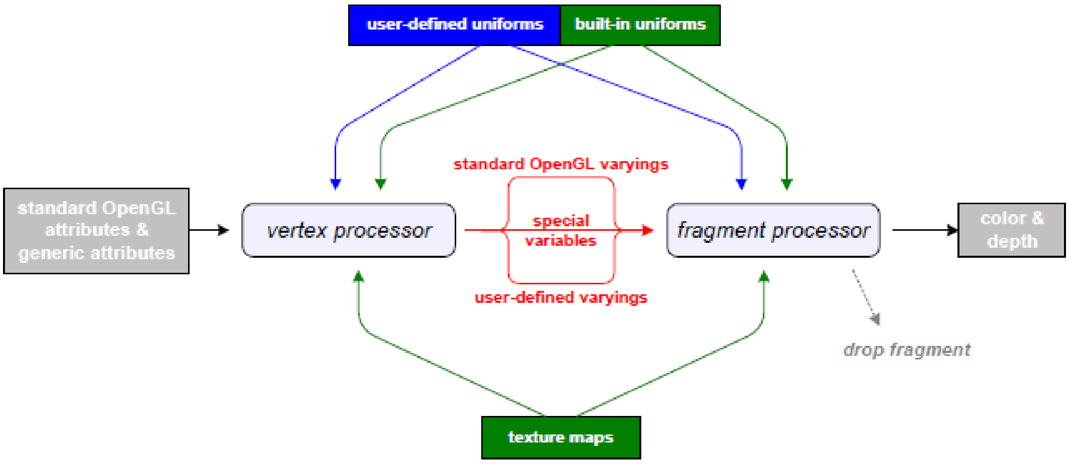


Figure B.1: The programmable pipeline architecture used in OpenGL[185]. The model used in DirectX is very similar but often refers to the fragment processor as the pixel processor.

matrices are built in extensions to the float data type and mathematical functions are natively supported on the GPU.

B.3 Enhancing understanding of volume data

The main issue occurring due to the use of transparency is that surfaces often have sparse features that can help viewers to perceive depth. One proposed method [186] is to use ridge and valley lines to enhance contours by overlaying them on the original surface, as illustrated in Figure B.2. The algorithm combines elements of marching cubes and methods used in ray-casting to try to locate points in the image that are near to a ridge or valley. Generally outlines resulting from identification of ridges and valleys are highlighted in white; however, the shade and opacity can be varied according to the prominence of the edge. Additionally, the silhouette of the object is treated as a special case and is drawn in a thick black outline. A problem that can arise from the algorithm is the tendency to produce too many highlights, complicating the image. This is rectified by only displaying lines with a degree of curvature that indicates an important feature. This allows the creation of highlights that are unobtrusive but informative of depth characteristics.

Within computer graphics, algorithms attempting to mimic technical drawing styles are given the name non-photo-realistic rendering (NPR), an example being the Gooch shading algorithm [187]. Objects rendered in this style combine surface diffuse colour and a hue gradient to produce a final output colour. The hue selected from the gradient runs from blue to yellow according to distance from the viewer; the resultant effect is that objects that are further from the viewer have slightly blue tints, whilst closer objects appear orange. An edge line silhouette and highlighting similar to that used in [186] differentiates the borders of objects from the background. Additionally, areas under intense light receive simplistic off-white highlighting with the intensity calculated using surface reflection characteristics. The muted range of hues

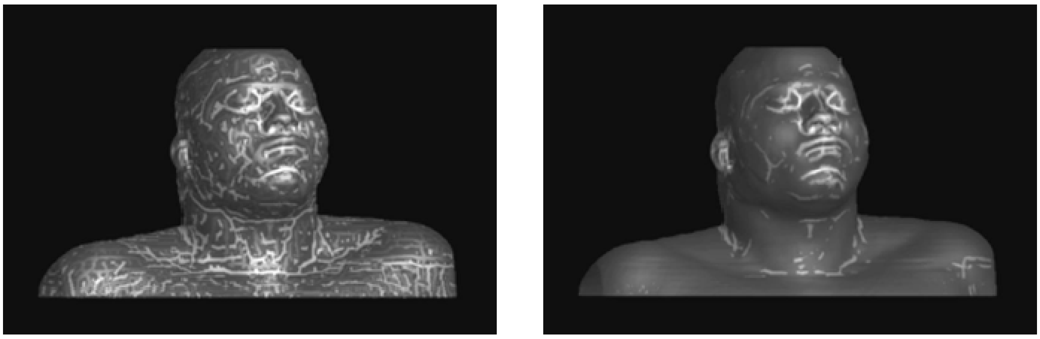


Figure B.2: Examples of image enhancement with ridge and valley lines [186]. Left: all ridges and valleys highlighted. Right: display restricted on curvature.

used to paint the surfaces of objects also reduce the possibility that areas in shadow or intense highlight do not lose features, an issue that is often experienced in traditional photo-realistic renderers.

Appendix C

Case Study instructions

C.1 Introduction

The purpose of this case study is to receive feedback from Lattice Quantum Chromodynamics experts regarding the use of the *QCDVis* software for exploring and visualizing ensemble data. The study is expected to take a maximum of around an hour and requires the participant to follow two predefined tasks.

You will be observed during the study by the lead researcher, ____________________, and may be audibly recorded for the purpose of recalling observations made whilst using the software. If, while using the software, you have any feedback / feature requests it is encouraged that you relay these to the researcher. Additionally, if at any point you are unsure of what to do, please ask for guidance.

C.2 Tasks

This case study takes the form of two predefined tasks that attempt to assess the usability and features of the application. Each case study will use a cooled data set, generated using the SuNtopology FORTRAN code. The program has been modified to output configurations at each stage of the cooling process. Additionally, numerical properties (location of minima / maxima) are available to help guide the visualization process.

C.2.1 Task 1: Locate an instanton within the ensemble data.

The first task seeks to demonstrate the core purpose of QCDVis, that is to visually inspect lattice observables. A $16^3 \times 32$ ensemble from the Edinburgh DiRAC server is provided for the task with μ set to 0.7. The ensemble has been cooled for 30 iterations and is known to contain (anti)instantons within the data. The task is to locate the (anti)instantons and view them as objects in the topological charge density and space/time plaquette fields.

- Using the QCDVis ensemble tool:

1. Select the configuration folder located at *.../QCD Data/Cooled Configurations/Edinburgh/V16X32/J02/MU0700*.
2. Set an output folder at *.../Desktop/QCDVis_case study/<your name>*.
3. Visualize the evolution of the *Action* and *FFDual* values across the cooling domain using the *View Action* and *View FFDual* buttons.
4. Highlight the cooling series data for the *conp0005* configuration in the *Action Max* and *FFDual Total* graphs. To do this:
   a) Click the *green tick* toolbar button.
   b) Select the *Curves* tab.
   c) Choose *conp0005* from the drop-down box.
   d) Set the line width to 5.0
   e) Click the OK button.
5. Compute for configuration *conp0005* at $10, 15$ and 20 cools (using the *Generate fields* button):
   a) Topological charge density
   b) Space-like plaquette
   c) Time-like plaquette
   d) Difference Plaquette (space-like - time-like)
6. If the field data was correctly generated, a folder should appear showing the new data sets.

- In the QCDVis visualization tool:

1. Load the computed topological charge density field in QCDVis, using the desired minima/maxima locations (from the ensemble tool) to locate an (anti)instanton at 10 cools.
2. Load the time-like / space-like plaquette fields using the highlighted maxima to locate regions of peak Wilson action at 10 cools.
3. Compare the position of peak topological charge density and Wilson action.

C.2.2 Task 2: View an irregularity within a cooled configuration.

The second task allows QCDVis to be used to view and understand the cooling process in greater detail. For this task a $16^3 \times 32$ ensemble from the Edinburgh DiRAC server is provided; however, in this ensemble μ is set to 0.45. The data has been cooled for 100 iterations to try to emphasise the phenomena of observables dropping through the lattice. An unexpected irregularity is found within one of the configurations, where a positive maxima becomes a negative minima in successive cooling cycles. The task intends to visually inspect this happening.

- Using the QCDVis ensemble tool:

1. Select the configuration folder located at *.../QCD Data/Cooled Configurations/Edinburgh/V16X32/J02/MU0450*.
2. Set an output folder at *.../Desktop/QCDVis_case study/<your name>*.
3. Visualize the evolution of the *Action* and *FFDual* values across the cooling domain using the *View Action* and *View FFDual* buttons.
4. Highlight the cooling series data for the *conp0025* configuration in the *Action Max* and *FFDual Total* graphs. To do this:
   a) Click the *green tick* toolbar button.
   b) Select the *Curves* tab.
   c) Choose *conp0025* from the drop-down box.
   d) Set the line width to 5.0
   e) Click the OK button.
5. Zoom in on the region of the FFDual Total graph (*conp0025*) at cools 30 to 40. Observe how the value increases until dropping off at around the 36^{th} cooling iteration.
   a) Click the *Crossed arrow* toolbar button.
   b) Hold the left mouse button to pan the region of interest.
   c) Hold the right mouse button to zoom the region of interest.
6. Compute the Topological Charge Density for the *conp0025* configuration at cools $35, 36$ and 37.
7. If the field data was correctly generated, a folder should appear showing the new data sets.

- In the QCDVis visualization tool:

1. Load the computed topological charge density field in QCDVis, using the desired minima/maxima locations (from the ensemble tool) to where q_{max} becomes q_{min} (at cools 36 & 37).
2. Multiple windows can be tiled by selecting *Tile windows horizontally* or *Tile windows vertically* from the *Window* menu.
3. Use the global isovalue slider to explore the 3-configurations concurrently and look for irregularities.
4. A wireframe overlay can be used to give additional clues to the structure of the irregularity. This can be located in the *View > Objects* menu.

List of Figures

List of Tables

Bibliography

[1] K. G. Wilson, “Confinement of quarks,” *Physical Review D*, vol. 10, no. 8, p. 2445, 1974.

[2] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” in *ACM Siggraph Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.

[3] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, “Contour trees and small seed sets for isosurface traversal,” *Proceedings of the thirteenth annual symposium on Computational geometry - SCG ’97*, pp. 212–220, 1997. [Online]. Available: http://portal.acm.org/citation.cfm?doid=262839.269238

[4] S. P. Tarasov and M. N. Vyalyi, “Construction of contour trees in 3D in O (n log n) steps,” in *Proceedings of the fourteenth annual symposium on Computational geometry*. ACM, 1998, pp. 68–75.

[5] C. L. Bajaj, V. Pascucci, and D. R. Schikore, “The contour spectrum,” in *Proceedings of the 8th conference on Visualization’97*. IEEE Computer Society Press, 1997, pp. 167–ff.

[6] H. Edelsbrunner and J. Harer, “Jacobi sets of multiple Morse functions,” *Foundations of Computational Mathematics, Minneapolis*, vol. 8, pp. 35–57, 2002.

[7] K. Beketayev, D. Yeliussizov, D. Morozov, G. H. Weber, and B. Hamann, “Measuring the distance between merge trees,” in *Topological Methods in Data Analysis and Visualization III*. Springer, 2014, pp. 151–165.

[8] U. Bauer, X. Ge, and Y. Wang, “Measuring distance between Reeb graphs,” in *Proceedings of the thirtieth annual symposium on Computational geometry*. ACM, 2014, p. 464.

[9] H. Saikia, H.-P. Seidel, and T. Weinkauf, “Extended Branch Decomposition Graphs: Structural Comparison of Scalar Data,” *Computer Graphics Forum*, vol. 33, no. 3, pp. 41–50, 2014.

[10] H. Edelsbrunner, J. Harer, and A. K. Patel, “Reeb spaces of piecewise linear mappings,” in *Proceedings of the twenty-fourth annual symposium on Computational geometry*. ACM, 2008, pp. 242–250.

[11] H. Carr and D. Duke, "Joint Contour Nets," *IEEE transactions on visualization and computer graphics2*, vol. PP, no. 99, pp. 1–15, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6684144

[12] D. P. Thomas, R. Borgo, and S. Hands, "Topological Visualisation techniques for the understanding of Lattice Quantum Chromodynamics (LQCD) simulations," in *EGUK Computer Graphics & Visual Computing Conference*. Eurographics UK, 2016.

[13] D. P. Thomas, R. Borgo, R. S. Laramee, and S. J. Hands, "QCDVis: a tool for the visualisation of Quantum Chromodynamics (QCD) data," *Computers & Graphics*, vol. 67, pp. 115–126, oct 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0097849317300997

[14] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," in *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. ACM, 1988, pp. 65–74.

[15] G. T. Herman and H. K. Liu, "Three-dimensional display of human organs from computed tomograms," *Computer graphics and image processing*, vol. 9, no. 1, pp. 1–21, 1979.

[16] "Three-Dimensional Display in Nuclear Medicine," *IEEE Transactions on Medical Imaging*, vol. 8, no. 4, pp. 297–303, dec 1989.

[17] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis, "Local Maximum Intensity Projection (LMIP: A New Rendering Method for Vascular Visualization," *Journal of computer assisted tomography*, vol. 22, no. 6, pp. 912–917, 1998.

[18] B. Csebfalvi, A. König, and E. Gröller, "Fast maximum intensity projection using binary shear-warp factorization," in *In Proceedings of WSCG'99*. Citeseer, 1999.

[19] B. Mora and D. S. Ebert, "Low-complexity maximum intensity projection," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 4, pp. 1392–1416, 2005.

[20] H. K. Tuy and L. T. Tuy, "Direct 2-D display of 3-D objects," *Computer Graphics and Applications, IEEE*, vol. 4, no. 10, pp. 29–34, 1984.

[21] H. Gouraud, "Continuous shading of curved surfaces," *Computers, IEEE Transactions on*, vol. 100, no. 6, pp. 623–629, 1971.

[22] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.

[23] A. Wallin, "Constructing isosurfaces from CT data," *IEEE Computer Graphics and Applications*, vol. 11, no. 6, pp. 28–33, nov 1991.

[24] M. W. Jones, "The visualisation of regular three dimensional data," Ph.D. dissertation, University of Wales Swansea, 1995.

[25] M. J. Dürst, "Re: additional reference to marching cubes," *ACM SIGGRAPH Computer Graphics*, vol. 22, no. 5, p. 243, 1988.

[26] G. M. Nielson and B. Hamann, "The asymptotic decider: resolving the ambiguity in marching cubes," in *Proceedings of the 2nd conference on Visualization'91*. IEEE Computer Society Press, 1991, pp. 83–91.

[27] E. V. Chernyaev, "Marching cubes 33: Construction of topologically correct isosurfaces," *Institute for High Energy Physics, Moscow, Russia, Report CN/95-17*, vol. 42, 1995.

[28] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," *Journal of graphics tools*, vol. 8, no. 2, pp. 1–15, 2003.

[29] T. Etiene, L. G. Nonato, C. Scheidegger, J. Tierny, T. J. Peters, V. Pascucci, R. M. Kirby, C. T. Silva, and Others, "Topology verification for isosurface extraction," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 6, pp. 952–965, 2012.

[30] L. Custodio, T. Etiene, S. Pesco, and C. Silva, "Practical considerations on Marching Cubes 33 topological correctness," *Computers & Graphics*, vol. 37, no. 7, pp. 840–850, 2013.

[31] C. Montani, R. Scateni, and R. Scopigno, "A modified look-up table for implicit disambiguation of marching cubes," *The Visual Computer*, vol. 10, no. 6, pp. 353–355, 1994.

[32] G. M. Nielson, "On marching cubes," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 3, pp. 283–297, 2003.

[33] H. Carr, "(No) more marching cubes," in *Proceedings of the Sixth Eurographics/Ieee VGTC conference on Volume Graphics*. Eurographics Association, 2007, pp. 81–90.

[34] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel, "Feature sensitive surface extraction from volume data," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 57–66.

[35] G. M. Nielson, "Dual marching cubes," in *Proceedings of the conference on Visualization'04*. IEEE Computer Society, 2004, pp. 489–496.

[36] A. Bhattacharya and R. Wenger, "Constructing isosurfaces with sharp edges and corners using cube merging," in *Computer Graphics Forum*, vol. 32, no. 3pt1. Wiley Online Library, 2013, pp. 11–20.

[37] C. Bajaj and V. Pascucci, "Progressive isocontouring," *TICAM Report TR*, vol. 36, 1999.

[38] G. M. Treece, R. W. Prager, and A. H. Gee, “Regularised marching tetrahedra: improved iso-surface extraction,” *Computers & Graphics*, vol. 23, no. 4, pp. 583–598, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S009784939900076X

[39] Y. Zhou, W. Chen, and Z. Tang, “An elaborate ambiguity detection method for constructing isosurfaces within tetrahedral meshes,” *Computers & Graphics*, vol. 19, no. 3, pp. 355–364, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/009784939500006X

[40] B. A. Payne and A. W. Toga, “Surface mapping brain function on 3D models,” *IEEE Computer Graphics and Applications*, vol. 10, no. 5, pp. 33–41, sep 1990.

[41] A. Gueziec and R. Hummel, “Exploiting triangulated surface extraction using tetrahedral decomposition,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 4, pp. 328–342, dec 1995.

[42] S. L. Chan and E. O. Purisima, “A new tetrahedral tesselation scheme for isosurface generation,” *Computers & Graphics*, vol. 22, no. 1, pp. 83–90, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S009784939700085X

[43] H. Carr, G. Zhao, J. Tierny, A. Chattopadhyay, and A. Knoll, “Fiber Surfaces: Generalizing Isosurfaces to Bivariate Data,” in *Computer Graphics and Visual Computing (CGVC) 2015*, 2015, pp. 63–64.

[44] S. Bachthaler and D. Weiskopf, “Continuous scatterplots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1428–1435, 2008.

[45] N. Kotava, A. Knoll, M. Schott, C. Garth, X. Tricoche, C. Kessler, E. Cohen, C. D. Hansen, M. E. Papka, and H. Hagen, “Volume rendering with multidimensional peak finding,” in *Visualization Symposium (PacificVis), 2012 IEEE Pacific*. IEEE, 2012, pp. 161–168.

[46] H. Edelsbrunner, J. Harer, and A. Zomorodian, “Hierarchical Morse complexes for piecewise linear 2-manifolds,” in *Proceedings of the seventeenth annual symposium on Computational geometry*. ACM, 2001, pp. 70–79.

[47] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci, “A multi-resolution data structure for two-dimensional Morse-Smale functions,” in *Proceedings of the 14th IEEE Visualization 2003 (VIS’03)*. IEEE Computer Society, 2003, p. 19.

[48] Carr, Hamish and Snoeyink, Jack and van de Panne, Michiel, “Simplifying flexible isosurfaces using local geometric measures,” in *Proceedings of the conference on Visualization’04*. IEEE Computer Society, 2004, pp. 497–504.

[49] H. Edelsbrunner, D. Letscher, and A. Zomorodian, “Topological persistence and simplification,” *Discrete and Computational Geometry*, vol. 28, no. 4, pp. 511–533, 2002.

[50] A. G. Landge, V. Pascucci, A. Gyulassy, J. C. Bennett, H. Kolla, J.-J. Chen, and P.-T. Bremer, "In-situ feature extraction of large scale combustion simulations using segmented merge trees," in *High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for*. IEEE, 2014, pp. 1020–1031.

[51] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," pp. 75–94, 2003.

[52] R. L. Boyell and H. Ruston, "Hybrid techniques for real-time radar simulation," in *Proceedings of the November 12-14, 1963, fall joint computer conference*. ACM, 1963, pp. 445–458.

[53] T. Banchoff and Others, "Critical points and curvature for embedded polyhedra," *Journal of Differential Geometry*, vol. 1, no. 3-4, pp. 245–256, 1967.

[54] H. Freeman and S. P. Morse, "On searching a contour map for a given terrain elevation profile," *Journal of the Franklin Institute*, vol. 284, no. 1, pp. 1–25, 1967. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0016003267905686

[55] V. Pascucci and K. Cole-McLaughlin, "Efficient computation of the topology of level sets," in *Proceedings of the conference on Visualization'02*. IEEE Computer Society, 2002, pp. 187–194.

[56] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, "Contour trees and small seed sets for isosurface traversal," *Proceedings of the thirteenth annual symposium on Computational geometry - SCG '97*, pp. 212–220, 1997. [Online]. Available: http://portal.acm.org/citation.cfm?doid=262839.269238

[57] H. Carr and M. V. Panne, *Topological manipulation of isosurfaces*. The University of British Columbia (Canada), 2004. [Online]. Available: http://www.comp.leeds.ac.uk/scshca/papers/Car04_topologicalManipulation.pdf

[58] H. Carr and J. Snoeyink, "Path seeds and flexible isosurfaces using topology for exploratory visualization," in *Proceedings of the symposium on Data visualisation 2003*. Eurographics Association, 2003, pp. 49–58.

[59] M. de Berg and M. van Kreveld, "Trekking in the Alps Without Freezing or Getting Tired," pp. 306–323, 1997.

[60] Y.-J. Chiang, T. Lenz, X. Lu, and G. Rote, "Simple and optimal output-sensitive construction of contour trees using monotone paths," *Computational Geometry*, vol. 30, no. 2, pp. 165–195, 2005.

[61] S. Takahashi, T. Ikeda, Y. Shinagawa, T. L. Kunii, and M. Ueda, "Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data," in *Computer Graphics Forum*, vol. 14, no. 3, 1995, pp. 181–192.

[62] C. Gueunet, P. Fortin, and J. Jomier, “Contour forests: Fast multi-threaded augmented contour trees,” in *Large Data Analysis and Visualization (LDAV), 2016 IEEE 6th Symposium on*. IEEE, 2016, pp. 85–92.

[63] G. Reeb, “Sur les points singuliers d’une forme de pfaff completement integrable ou d’une fonction numerique [on the singular points of a completely integrable pfaff form or of a numerical function],” *Comptes Rendus Acad. Sciences Paris*, vol. 222, pp. 847–849, 1946.

[64] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, “Loops in Reeb graphs of 2-manifolds,” in *Proceedings of the nineteenth annual symposium on Computational geometry*. ACM, 2003, pp. 344–350.

[65] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, “Robust on-line computation of Reeb graphs: simplicity and speed,” in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 58.

[66] I. Fujishiro, Y. Takeshima, T. Azuma, and S. Takahashi, “Volume data mining using 3D field topology analysis,” *IEEE Computer Graphics and Applications*, no. 5, pp. 46–51, 2000.

[67] Y. Shinagawa and T. L. Kunii, “Constructing a Reeb graph automatically from cross sections,” *IEEE Computer Graphics and Applications*, no. 6, pp. 44–51, 1991.

[68] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, “Surface coding based on Morse theory,” *IEEE Computer Graphics and Applications*, no. 5, pp. 66–78, 1991.

[69] W. Harvey, Y. Wang, and R. Wenger, “A randomized O (m log m) time algorithm for computing Reeb graphs of arbitrary simplicial complexes,” in *Proceedings of the twenty-sixth annual symposium on Computational geometry*. ACM, 2010, pp. 267–276.

[70] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci, “Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1177–1184, 2009.

[71] H. Doraiswamy and V. Natarajan, “Efficient algorithms for computing Reeb graphs,” *Computational Geometry*, vol. 42, no. 6, pp. 606–616, 2009.

[72] Doraiswamy, Harish and Natarajan, Vijay, “Output-sensitive construction of Reeb graphs,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 1, pp. 146–159, 2012.

[73] H. Doraiswamy and V. Natarajan, “Computing Reeb graphs as a union of contour trees,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 2, pp. 249–262, 2013.

[74] G. Wyvill, C. McPheeters, and B. Wyvill, “Data structure for soft objects,” *The visual computer*, vol. 2, no. 4, pp. 227–234, 1986.

[75] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," in *Computer Graphics (Proceedings of SIGGRAPH 87)*, vol. 21, Jul. 1987, pp. 163–169.

[76] H. Carr, J. Snoeyink, and M. van de Panne, "Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree," *Computational Geometry*, vol. 43, no. 1, pp. 42–58, 2010.

[77] H. Carr, D. Brian, and D. Brian, "On histograms and isosurface statistics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1259–1266, 2006.

[78] M. Meyer, C. E. Scheidegger, J. M. Schreiner, B. Duffy, H. Carr, and C. T. Silva, "Revisiting histograms and isosurface statistics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1659–1666, 2008.

[79] Y. Takeshima, S. Takahashi, I. Fujishiro, and G. M. Nielson, "Introducing topological attributes for objective-based visualization of simulated datasets," in *Volume Graphics, 2005. Fourth International Workshop on*. IEEE, 2005, pp. 137–236.

[80] G. H. Weber, S. E. Dillard, H. Carr, V. Pascucci, and B. Hamann, "Topology-controlled volume rendering," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 2, pp. 330–341, 2007.

[81] H. Bunke and K. Riesen, "Graph edit distance: optimal and suboptimal algorithms with applications," *Analysis of complex networks, from biology to linguistics*, pp. 113–143, 2009.

[82] D. M. Thomas and V. Natarajan, "Symmetry in scalar field topology," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2035–2044, 2011.

[83] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci, "Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 1053–1060, 2006.

[84] H. A. Carr, J. Tierny, and G. H. Weber, "Pathological and Test Cases For Reeb Analysis," in *Topology-Based Methods in Visualization 2017 (TopoInVis 2017)*, 2017. [Online]. Available: http://eprints.whiterose.ac.uk/114660/

[85] D. Duke, H. Carr, A. Knoll, N. Schunck, H. A. Nam, and A. Staszczak, "Visualizing nuclear scission through a multifield extension of topological analysis," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2033–2040, 2012.

[86] D. J. Duke, F. Hosseini, and H. Carr, "Parallel computation of multifield topology: experience of Haskell in a computational science application," in *Proceedings of the 3rd ACM SIGPLAN workshop on Functional high-performance computing*. ACM, 2014, pp. 11–21.

[87] D. J. Duke and F. Hosseini, “Skeletons for distributed topological computation,” in *Proceedings of the 4th ACM SIGPLAN Workshop on Functional High-Performance Computing*. ACM, 2015, pp. 35–44.

[88] N. Schunck, D. Duke, H. Carr, and A. Knoll, “Description of induced nuclear fission with skyrme energy functionals: Static potential energy surfaces and fission fragment properties,” *Physical Review C*, vol. 90, no. 5, p. 054305, 2014.

[89] N. Schunck, D. Duke, and H. Carr, “Description of induced nuclear fission with skyrme energy functionals. ii. finite temperature effects,” *Physical Review C*, vol. 91, no. 3, p. 034327, 2015.

[90] Z. Geng, D. Duke, H. Carr, and A. Chattopadhyay, “Visual analysis of hurricane data using joint contour net,” in *Computer Graphics and Visual Computing (CGVC)*, W. Tang and R. Borgo, Eds. The Eurographics Association, 2014.

[91] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, “Local and global comparison of continuous functions,” in *Visualization, 2004. IEEE*. IEEE, 2004, pp. 275–280.

[92] A. Chattopadhyay, H. Carr, D. Duke, and Z. Geng, “Extracting Jacobi Structures in Reeb Spaces,” in *EuroVis-Short Papers*, N. Elmqvist, M. Hlawitschka, and J. Kennedy, Eds. The Eurographics Association, 2014.

[93] A. Chattopadhyay, H. Carr, D. Duke, Z. Geng, and O. Saeki, “Multivariate topology simplification,” *arXiv preprint arXiv:1509.04465*, 2015.

[94] B. Strodthoff and B. Jüttler, “Layered Reeb graphs of a spatial domain,” *Booklet of abstracts of EuroCG*, pp. 21–24, 2013.

[95] A. Zomorodian and G. Carlsson, “Computing persistent homology,” *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[96] G. Carlsson and A. Zomorodian, “The theory of multidimensional persistence,” *Discrete & Computational Geometry*, vol. 42, no. 1, pp. 71–93, 2009.

[97] G. Carlsson, G. Singh, and A. Zomorodian, “Computing multidimensional persistence,” in *Algorithms and computation*. Springer, 2009, pp. 730–739.

[98] P. T. Bremer, E. M. Bringa, M. A. Duchaineau, A. G. Gyulassy, D. Laney, A. Mascarenhas, and V. Pascucci, “Topological feature extraction and tracking,” in *Journal of Physics: Conference Series*, vol. 78, no. 1. IOP Publishing, 2007, p. 12007.

[99] N. Suthambhara and V. Natarajan, “Simplification of jacobi sets,” in *Topological Methods in Data Analysis and Visualization*. Springer, 2011, pp. 91–102.

[100] R. von Bitter Rucker, *Geometry, relativity and the fourth dimension*. Dover Publications, 1977.

[101] S. R. Hollasch, “Four-Space Visualization of 4D Objects,” MSc thesis, Arizona State University, 1991.

[102] Weigle, Chris and Banks, David C, “Complex-valued contour meshing,” in *Proceedings of the 7th conference on Visualization’96*. IEEE Computer Society Press, 1996, pp. 173–.

[103] ——, “Extracting iso-valued features in 4-dimensional scalar fields,” in *Proceedings of the 1998 IEEE symposium on Volume visualization*. ACM, 1998, pp. 103–110.

[104] P. Bhaniramka, R. Wenger, and R. Crawfis, “Isosurfacing in higher dimensions,” *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, 2000.

[105] G. H. Weber, P.-T. Bremer, and V. Pascucci, “Topological landscapes: A terrain metaphor for scientific data,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1416–1423, 2007.

[106] W. Harvey and Y. Wang, “Topological landscape ensembles for visualization of scalar-valued functions,” *Computer Graphics Forum*, vol. 29, pp. 993–1002, 2010. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-77955748404\&partnerID=40\&md5=82804ef36e209ef8d6f31b1f02c97321

[107] S. Hands, “The phase diagram of QCD,” *Contemporary Physics*, vol. 42, no. 4, pp. 209–225, 2001.

[108] M. Creutz, *Quarks, gluons and lattices*. Cambridge University Press, 1983, vol. 8.

[109] M. Yoshida, A. Nakamura, M. Fukada, T. Nakamura, and S. Hioki, “Quantum chromodynamics simulation on NWT,” in *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)*. ACM, 1995, p. 65.

[110] C. R. Allton, S. Ejiri, S. J. Hands, O. Kaczmarek, F. Karsch, E. Laermann, C. Schmidt, and L. Scorzato, “The QCD thermal phase transition in the presence of a small chemical potential,” *Phys. Rev.*, vol. D66, p. 074507, 2002.

[111] F. Belletti, S. F. Schifano, R. Tripiccione, F. Bodin, P. Boucaud, J. Micheli, O. Pene, N. Cabibbo, S. De Luca, A. Lonardo, and Others, “Computing for LQCD: apeNEXT,” *Computing in science & engineering*, vol. 8, no. 1, pp. 18–29, 2005.

[112] R. G. Edwards and B. Joo, “The Chroma software system for lattice QCD,” *Nuclear Physics B-Proceedings Supplements*, vol. 140, pp. 832–834, 2005.

[113] A. Cucchieri, T. Mendes, G. Travieso, and A. R. Taurines, “Parallel implementation of a lattice-gauge-theory code: studying quark confinement on PC clusters,” in *Computer Architecture and High Performance Computing, 2003. Proceedings. 15th Symposium on*. IEEE, 2003, pp. 123–131.

[114] C. Banino-Rokkones, J. r. Amundsen, and E. Smorgrav, “Parallelizing Lattice Gauge Theory Models on Commodity Clusters,” in *Cluster Computing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 1–8.

[115] M. Smelyanskiy, K. Vaidyanathan, J. Choi, B. Joó, J. Chhugani, M. A. Clark, and P. Dubey, “High-performance lattice QCD for multi-core based parallel systems using a cache-friendly hybrid threaded-MPI approach,” in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011, p. 69.

[116] M. A. Clark, “QCD on GPUs: cost effective supercomputing,” *arXiv preprint arXiv:0912.2268*, 2009.

[117] M. A. Clark, R. Babich, K. Barros, R. C. Brower, and C. Rebbi, “Solving Lattice QCD systems of equations using mixed precision solvers on GPUs,” *Computer Physics Communications*, vol. 181, no. 9, pp. 1517–1528, 2010.

[118] R. Babich, M. A. Clark, and B. Joó, “Parallelizing the QUDA library for multi-GPU calculations in lattice quantum chromodynamics,” in *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for*. IEEE, 2010, pp. 1–11.

[119] G. Shi, R. Babich, M. A. Clark, B. Joo, S. Gottlieb, and V. Kindratenko, “The Fat-Link Computation On Large GPU Clusters for Lattice QCD,” in *Application Accelerators in High Performance Computing (SAAHPC), 2012 Symposium on*. IEEE, 2012, pp. 1–10.

[120] S. Hands, I. Montvay, S. Morrison, M. Oevers, L. Scorzato, and J. Skullerud, “Numerical study of dense adjoint matter in two color QCD,” *Eur. Phys. J.*, vol. C17, pp. 285–302, 2000.

[121] R. Giles, “Reconstruction of gauge potentials from Wilson loops,” *Physical Review D*, vol. 24, pp. 2160–2168, 1981.

[122] S. Hands, P. Kenny, S. Kim, and J.-I. Skullerud, “Lattice study of dense matter with two colors and four flavors,” *The European Physical Journal A*, vol. 47, no. 5, pp. 1–7, 2011.

[123] H. J. Rothe, *Lattice gauge theories: an introduction*, 3rd ed. World Scientific, 2005, vol. 74.

[124] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[125] S. Duane and J. Kogut, “The theory of hybrid stochastic algorithms,” *Nuclear Physics B*, vol. 275, no. 3, pp. 398–420, 1986.

[126] G. Parisi, Y.-s. Wu *et al.*, “Perturbation theory without gauge fixing,” *Scientia Sinica*, vol. 24, no. 4, pp. 483–469, 1981.

[127] A. Ukawa and M. Fukugita, "Langevin simulation including dynamical quark loops," *Physical review letters*, vol. 55, no. 18, p. 1854, 1985.

[128] G. Batrouni, G. Katz, A. Kronfeld, G. Lepage, B. Svetitsky, and K. Wilson, "Langevin simulations of lattice field theories," *Physical Review D*, vol. 32, no. 10, p. 2736, 1985.

[129] R. T. Scalettar, D. J. Scalapino, and R. L. Sugar, "New algorithm for the numerical simulation of fermions," *Phys. Rev. B*, vol. 34, no. 11, pp. 7911–7917, dec 1986. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevB.34.7911

[130] S. Gottlieb, W. Liu, D. Toussaint, R. L. Renken, and R. L. Sugar, "Hybrid-molecular-dynamics algorithms for the numerical simulation of quantum chromodynamics," *Phys. Rev. D*, vol. 35, no. 8, pp. 2531–2542, apr 1987. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevD.35.2531

[131] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216–222, 1987.

[132] A. A. Belavin, A. M. Polyakov, A. Schwartz, and Y. S. Tyupkin, "Pseudoparticle solutions of the yang-mills equations," *Physics Letters B*, vol. 59, no. 1, pp. 85–87, 1975.

[133] M. Teper, "Instantons in the quantized su (2) vacuum: A lattice monte carlo investigation," *Physics Letters B*, vol. 162, no. 4, pp. 357–362, 1985.

[134] P. Kenny, "Topology and Condensates in Dense Two Colour Matter," PhD Thesis, Swansea University, 2010.

[135] S. Hands, "Lattice monopoles and lattice fermions," *Nuclear Physics B*, vol. 329, no. 1, pp. 205–224, 1990.

[136] M. Feurstein, H. Markum, and S. Thurner, "Visualization of topological objects in QCD," *Nuclear Physics B - Proceedings Supplements*, vol. 53, no. 1-3, pp. 553–556, Feb. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920563296007165

[137] D. B. Leinweber, "Visualizations of the QCD Vacuum," *arXiv preprint hep-lat/0004025*, 2000.

[138] M. Di Pierro, "Visualization for lattice QCD," *Proceedings of Science*, 2007.

[139] M. Di Pierro, M. Clark, C. Jung, J. Osborn, J. Negele, R. Brower, S. Gottlieb, and Y. Zhong, "Visualization as a tool for understanding QCD evolution algorithms," in *Journal of Physics: Conference Series*, vol. 180, no. 1. IOP Publishing, 2009, p. 12068.

[140] M. Di Pierro, "Visualization Tools for Lattice QCD-Final Report," DOE, Tech. Rep., 2012.

[141] H. Childs, “VisIt: An end-user tool for visualizing and analyzing very large data,” *Proceedings of SciDAC 2011*, 2011.

[142] J. Ahrens, B. Geveci, and C. Law, “ParaView: An End-User Tool for Large Data Visualization,” in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Amsterdam, Netherlands: Elsevier, 2005, ch. 36, pp. 717—-732.

[143] E. Bennett, “Physical and Computational Applications of Strongly-Interacting Dynamics Beyond QCD,” PhD thesis, Swansea University, 2013. [Online]. Available: http://inspirehep.net/record/1336567

[144] M. G. Pérez, “A few basic notions on instantons and confinement,” *Nuclear Physics B-Proceedings Supplements*, vol. 105, no. 1, pp. 52–55, 2002.

[145] T.-P. Cheng and L.-F. Li, *Gauge theory of elementary particle physics*, 1986th ed. New York: Clarendon press Oxford, 1984.

[146] J. W. Negele, “Instantons, the QCD vacuum, and hadronic physics,” *Nuclear Physics B-Proceedings Supplements*, vol. 73, no. 1, pp. 92–104, 1999.

[147] M. Peskin, “Chirality conservation in the lattice gauge theory,” Ph.D. dissertation, Cornell University, 1978.

[148] P. Di Vecchia, K. Fabricius, G. Rossi, and G. Veneziano, “Preliminary evidence for ua (1) breaking in qcd from lattice calculations,” *Nuclear Physics B*, vol. 192, no. 2, pp. 392–408, 1981.

[149] Lawrence-Livermore National Laboratory, “VisIt: a parallel visualization end tool based on Kitware’s VTK,” 2014. [Online]. Available: https://wci.llnl.gov/codes/visit/home.html

[150] L. D. McLerran and B. Svetitsky, “A monte carlo study of su (2) yang-mills theory at finite temperature,” *Physics Letters B*, vol. 98, no. 3, pp. 195–198, 1981.

[151] J. Kuti, J. Polonyi, and K. Szlachanyi, “Monte carlo study of su (2) gauge theory at finite temperature,” *Physics Letters B*, vol. 98, no. 3, pp. 199–204, 1981.

[152] J. Engels, F. Karsch, H. Satz, and I. Montvay, “High temperature su (2) gluon matter of the lattice,” *Physics Letters B*, vol. 101, no. 1, pp. 89–94, 1981.

[153] K. Kajantie, C. Montonen, and E. Pietarinen, “Phase transition of su (3) gauge theory at finite temperature,” *Zeitschrift für Physik C Particles and Fields*, vol. 9, no. 2, pp. 253–254, 1981.

[154] K. Beketayev, “Extracting and visualizing topological information from large high-dimensional data sets,” p. 94, 2014.

[155] H. Coxeter, “Introduction to geometry.” *John Wiley& Sons, New York*, 1969.

[156] C. Zhang and T. Chen, "Efficient feature extraction for 2D/3D objects in mesh representation," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3. IEEE, 2001, pp. 935–938.

[157] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M.-H. Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen, "Moment invariants for the analysis of 2d flow fields," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1743–1750, 2007.

[158] R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzer, "Moment invariants for 2d flow fields using normalization," in *Pacific Visualization Symposium (PacificVis), 2014 IEEE*. IEEE, 2014, pp. 41–48.

[159] G. Guennebaud, B. B. Jacob, and Others, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[160] W. J. Schroeder, B. Lorensen, and K. Martin, *The visualization toolkit: an object-oriented approach to 3D graphics*. Kitware, 2004.

[161] H. Carr, T. Moller, and J. Snoeyink, "Artifacts caused by simplicial subdivision," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 231–242, 2006.

[162] C. Heine, D. Schneider, H. Carr, and G. Scheuermann, "Drawing contour trees in the plane," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 11, pp. 1599–1611, 2011.

[163] H. Carr and D. Duke, "Multifield Extension of Topological Analysis (META) project." [Online]. Available: http://www.engineering.leeds.ac.uk/computing/research/cse/meta/

[164] H. Levkowitz, *Color theory and modeling for computer graphics, visualization, and multimedia applications*. Springer Science & Business Media, 1997, vol. 402.

[165] C. Ware, *Information visualization: perception for design*. Elsevier, 2012.

[166] M. O. Ward, G. Grinstein, and D. Keim, *Interactive Data Visualization: Foundations, Techniques, and Applications*. CRC Press, 2015.

[167] S. Cotter, P. Giudice, S. Hands, and J.-I. Skullerud, "Towards the phase diagram of dense two-color matter," *Physical Review D*, vol. 87, no. 3, p. 34507, 2013.

[168] T. Boz, S. Cotter, L. Fister, D. Mehta, and J.-I. Skullerud, "Phase transitions and gluodynamics in 2-colour matter at high density," *The European Physical Journal A*, vol. 49, no. 7, pp. 1–13, 2013.

[169] K. Shoemake, "Animating rotation with quaternion curves," pp. 245–254, 1985.

[170] N. Alharbi, R. S. Laramee, and M. Chavent, "MolPathFinder: Interactive Multi-Dimensional Path Filtering of Molecular Dynamics Simulation Data," *The Computer Graphics and Visual Computing (CGVC) Conference 2016*, pp. 9—–16, 2016.

[171] S. van der Walt, C. Colbert, and G. Varoquaux, “The NumPy Array: A Structure for Efficient Numerical Computation,” *Computing in Science & Engineering*, vol. 13, pp. 22–30, 2011.

[172] T. Munzner, “A nested model for visualization design and validation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, Nov. 2009.

[173] S. Hands and P. Kenny, “Topological fluctuations in dense matter with two colors,” *Physics Letters B*, vol. 701, no. 3, pp. 373–377, 2011.

[174] A. Bazavov, C. Bernard, N. Brown, C. DeTar, D. Du, S. Gottlieb, U. M. Heller, J. Hetrick, J. Komijani, J. Laiho, C. Monahan, J. Osborn, T. Primer, R. Sugar, D. Toussaint, R. Van de Water, and R. Zhou, “MIMD Lattice Computation (MILC) Collaboration.” [Online]. Available: http://physics.indiana.edu/{~}sg/milc.html

[175] H. Edelsbrunner, J. Harer, A. Mascarenhas, V. Pascucci, and J. Snoeyink, “Time-varying Reeb graphs for continuous space–time data,” *Computational Geometry*, vol. 41, no. 3, pp. 149–166, 2008.

[176] J. Tierny and H. Carr, “Jacobi Fiber Surfaces for Bivariate Reeb Space Computation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 960–969, 2017.

[177] E. Haines, “An introductory tour of interactive rendering,” *IEEE Computer Graphics and Applications*, vol. 26, no. 1, 2006.

[178] G. S. Watkins, “A real time visible surface algorithm,” DTIC Document, Tech. Rep., 1970.

[179] B. T. Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[180] E. Lindholm, M. J. Kligard, and H. Moreton, “A user-programmable vertex engine,” in *SIGGRAPH*. ACM Press, 2001, pp. 149–158. [Online]. Available: http://portal.acm.org/citation.cfm?doid=383259.383274

[181] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard, “Cg: a system for programming graphics hardware in a C-like language,” *SIGGRAPH 03 ACM SIGGRAPH 2003 Papers*, vol. 22, no. 3, pp. 896–907, 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882362

[182] D. Blythe, “The direct3d 10 system,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 724–734, 2006.

[183] M. Segal and K. Akeley, “The OpenGL Graphics System: A specification (Version 2.0),” Khronos Group, Tech. Rep., 2004.

[184] P. Hanrahan and J. Lawson, "A Language for Shading and Lighting Calculations," *SIGGRAPH Comput. Graph.*, vol. 24, no. 4, pp. 289–298, Sep. 1990. [Online]. Available: http://doi.acm.org/10.1145/97880.97911

[185] S. Brumme, "The OpenGL Shading Language," 2003.

[186] V. Interrante, H. Fuchs, and S. Pizer, "Enhancing transparent skin surfaces with ridge and valley lines," *Proceedings Visualization '95*, 1995.

[187] A. Gooch, B. Gooch, P. Shirley, and E. Cohen, "A non-photorealistic lighting model for automatic technical illustration," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 447–452.

[188] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995. [Online]. Available: http://www.cs.up.ac.za/cs/aboake/sws780/references/patternstoarchitecture/Gamma-DesignPatternsIntro.pdf

[189] M. Creutz, *Quarks, gluons and lattices*, L. P.V., D. Weinberg, W. Sciama, and M. W.H., Eds. Cambridge: Cambridge University Press, 1985.

[190] J. Smit, *Introduction to quantum fields on a lattice*. Cambridge University Press, 2002.

[191] B. Whitlock, "Getting Data Into VisIt (version 2.0.0)," Lawrence Livermore National Laboratory, Livermore, Tech. Rep., 2010.

[192] B. Whitlock, J. van Bethlehem, and H. Childs, "VisIt Python Interface Manual (version 2.5.2)," Lawrence Livermore National Lab, Livermore, Tech. Rep., 2011. [Online]. Available: http://vis.lbl.gov/{~}hrchilds/VisItPythonManual.pdf

[193] Summerfield, Mark, *Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming*. Upper Saddle River, New Jersey: Prentice Hall, 2009.

[194] ——, *Advanced Qt Programming: Creating Great Software with C++/Qt 4*, 1st ed. Upper Saddle River, New Jersey: Prentice Hall, 2010.

[195] J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2008.

[196] B. Stroustrup, *The C++ programming language*, 4th ed. Boston: Addison Wesley, 2013.

[197] Meyers, Scott, *Effective C++: 55 specific ways to improve your programs and designs*, 3rd ed. Boston, MA, USA: Addison Wesley, 2005.

[198] ——, *Effective STL: 50 specific ways to improve your use of the standard template library*, 1st ed. Boston, MA, USA: Addison Wesley, 2001.

[199] S. Meyers, *Effective modern C++: 42 specific ways to improve your use of C++ 11 and C++ 14*, 1st ed. Sebastopol, California: O'Reilly Media, Inc., 2014.

[200] D. Vandevoorde and N. M. Josuttis, *C++ Templates: The Complete Guide*, 1st ed. Boston, MA, USA: Addison-Wesley Publishing Company, 2002.

[201] A. Alexandrescu, *Modern C++ design: generic programming and design patterns applied*, 1st ed. Boston: Addison-Wesley, 2001.

[202] R. Forman, "A user's guide to discrete Morse theory," *S{é}m. Lothar. Combin*, vol. 48, p. 35pp, 2002.

[203] G. Rote and G. Vegter, "Computational topology: An introduction," in *Effective Computational Geometry for curves and surfaces*. Springer, 2006, pp. 277–312.