A Wonderful Piece of Scientific Research

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

by

Stuart Thomas

Advisor: Prof. Christopher J. Monahan

Prof. Todd Averett

Prof. Andreas Stathopoulos

Williamsburg, Virginia
May 2021

Contents

Acknowledgments

I would like to thank lots of people, and this is where I will do it...

List of Figures

List of Tables

Abstract

Abstract

Chapter 1

Introduction

Chapter 2

Theory

2.1 Gradient Flow

To remove ultraviolet fluctuations, we introduce a momentum cutoff, effectively re-ducing the predictive power of our theory by ignoring very high momentum states. In a gauge theory, such as QCD, a hard cutoff violates gauge invariance. Therefore, renormalization techniques must be more nuanced in the reduction of high-frequency divergences.

One such technique is "smearing," a local averaging of a gauge field in order to remove diverging fluctuations [1]. In this study, we use a technique known as the "gradient flow" [2] which introduces a new half-dimension called "flow time". The flow time parameterizes the smearing; an evolution in flow time corresponds to suppressing ultraviolet divergences. In a 2D ϕ^4 scalar field theory, given by the action

$$S_\phi[\phi] = \frac{1}{2} \int d^2x \left[(\partial\phi)^2 + m^2\phi^2 + \frac{\lambda}{2} \right], \tag{2.1}$$

we can define the evolution in flow time as

$$\frac{\partial\rho(\tau, x)}{\partial\tau} = \partial^2\rho(\tau, x) \tag{2.2}$$

where ∂^2 is the Laplacian in 4-D Euclidean spacetime and τ is the flow time. Here, ρ is the field evolved into a nonzero flow time, bounded by the condition $\rho(\tau =$

$0, x) = \phi(x)$. When solved exactly, this function forms a Gaussian which forces operators to converge. Therefore, in all orders of pertubation theory, any correlation function in the gauge field that has been evolved in flow time becomes immune to these divergences[3].

Generally, we can choose any flow time equation that drives the field towards a classical minimum. Beyond the ϕ^4 model, we need flow equations that incorporate different types of fields. In the nonlinear σ model, it has been shown that an appropriate manifestation of the flow time can resolve divergencies [3]. Therefore, it is possible to define a different flow equation for this model as well. Our goal is to understand this renormalizability in the nonlinear σ model using the gradient flow. Specifically, we seek to define vacuum condensates, or nonpertubative vacuum expectation values.

To explore this technique, we first plan to produce numerical calculations using a nonlinear σ model. Using a Monte Carlo method in 1+1 dimensions, we can calculate the correlation functions of quantum field theories. With more computing power, expanding this simulation to 2+1 dimensions would provide an analysis more relevant to quantum chromodynamics. Following this numerical calculation, we seek to find and understand an implementation of the gradient flow in this model using analytic methods. This understanding can then be verified by the numerical solution.

2.2 Field Theories on the Lattice

2.3 Non-linear σ model

The nonlinear σ model is a prototypical theory for many physical phenomena, including applications in string theory. As a simple nonperturbative model, it provides an ideal starting point for lattice QCD studies. Specifically, the nonlinear σ model exhibits many properties shared by Yang-Mills gauge theories, such as a mass gap,

asymptotic freedom and $O(2)$ renormalizability. Furthermore, it has an exact application to Heisenberg ferromagnets in condensed matter. The theory is defined by the Euclidean action

$$S_E = \frac{1}{2g^2} \int d^D x \partial^\mu \vec{\phi} \cdot \partial_\mu \vec{\phi} \qquad (2.3)$$

subject to the constraint that $\vec{\phi} \cdot \vec{\phi} = 1$. In this report, we primarily consider $O(3)$, where $\vec{\phi}$ is a 3 dimensional vector.

Following [4], we can define the gradient flow in this model via the differential equation

$$\partial_\tau \vec{\rho}(\tau, x) = \left(1 - \vec{\rho}(\tau, x)\vec{\rho}(\tau, x)^T\right) \nabla^2 \vec{\rho}(\tau, x), \qquad (2.4)$$

where ∇^2 is the Laplacian operator in Euclidean space. We solve this equation numerically using the boundary condition $\vec{\rho}(0, x) = \phi(x)$.

In order to use this theory on the lattice, we discretize the action. We replace the derivatives with a simple difference and impose periodic boundary conditions on the system. For a more in-depth discussion, see Section **??**.

2.4 Topological Charge

For every configuration of the non-linear sigma model, there is an associated integer quantity known as the topological charge. By viewing the field as a mapping from periodic real space to the sphere S^{N-1}, we can can associate each configuration with a member of the homotopy group $\Pi_2(S^2) = \mathbb{Z}$. This integer value is the topological charge of such a configuration.

Chapter 3

Methods

Our study of the gradient flow in the nonlinear σ model consists of a computational part and an analytical part. We begin by outlining a numerical Monte Carlo method to simulate the lattice in 2 and 3 dimensions. We verify our program with the well-studied ϕ^4 scalar field theory. We then generalize our model to a vector field to simulate the non-linear sigma model. This simulation system provides data on vacuum states with which we study the gradient flow's effect on topology.

We implement these two algorithms first in Python for the ϕ^4 model. Afterwards, we transition to C/C++ code due to the increased speed for more complicated theories.

3.1 Monte Carlo Simulations

We implement a Markov Chain Monte Carlo method following Schaich's thesis [5]. This implementation utilizes a "random walk," i.e. a set of random steps through phase space, to determine statistical values such as correlation functions across the lattice. By the definition of the Markov chain, the probability of adoption of each state, and therefore its inclusion in the Monte Carlo calculation, depends only on the current state and the proposed state. This probability is denoted as $P(\mu \to \nu)$ where μ and ν are the existing and proposed lattice configurations respectively. We use a

combination of the Metropolis and Wolff algorithms to determine this value.

3.1.1   Metropolis Algorithm

We primarily use the Metropolis algorithm for the calculation of new Markov chain configurations. We begin with a so-called "hot start," where each field value at each lattice site is randomly selected. Then we propose a new value for a single lattice point, which is accepted with a probability

$$P(\mu \rightarrow \nu) = \begin{cases} e^{-(S_\nu - S_\mu)} & S_\nu > S_\mu \\ 1 & \text{otherwise} \end{cases} \tag{3.1}$$

where S_ν and S_μ are the lattice actions of configuration before and after the change. This process is performed for each point on the lattice, making up a "sweep." Repeating this sweep many times pushes the lattice toward the action minimum.

3.1.2   Wolff Cluster Algorithm

Though the Metropolis algorithm will slowly find the absolute minimum of the theory, the presence of local minima can greatly prolong the convergence. Both the ϕ^4 model and the nonlinear σ model feature "kinetic" terms with gradients of ϕ. Therefore, the presence of large similarly-valued regions in the lattice can lead to a local minimum. The Wolff algorithm helps reduce the presence of these clusters through two steps: identifying a cluster and flipping it along some arbitrary vector. In the case of ϕ^4 theory, this flipping takes the form of a simple sign change. In the non-linear sigma model we choose a random unit vector $\vec{r}$ and consider the projection of the field on this vector. When the cluster flips, each site is flipped along this direction. [1]. To identify the cluster, the algorithm uses a recursive algorithm defined by the probability of adding a new site. In the non-linear σ model, this value is derived as

$$P_{add} = 1 - e^{-\beta \vec{e}_a \vec{e}_b}. \tag{3.2}$$

---

[1]citation needed

3.1.3 Checkerboard algorithm

In order to parallelize the Metropolis algorithm, we use a checkerboard algorithm. Since the Lagrangian density at each site does not depend on any sites diagonal to it, the lattice can be split into "white" sites and "black" sites, like the tiles on a checkerboard. Each white site is independent of every other white site and likewise with black sites. Therefore, we can split the sites of each color into separate parallel processing nodes and independently run the Metropolis algorithm, ensuring that no site affects the Lagrangian density at any other site. We use this method to parallelize the code through the Message Passing Interface (MPI).

3.1.4 Autocorrelation times and thermalization

One important aspect is the correlation between different states of the Markov chain. Ideally, each sample of the lattice would be completely independent, but this is not the case. Therefore, we must sweep the lattice sufficiently to produce an effective Monte Carlo result. We use Wolff's automatic windowing procedure [6] and the magnetic susceptibility χ_m to estimate the autocorrelation and determine an appropriate number of sweep between measurements for each theory. Furthermore, the initial values of our Markov chain will be highly correlated with the initial random start. Therefore, we also wait a set number of sweeps before taking any measurements so that the lattice can approach a minimum of the action.

3.2 Gradient Flow

On the lattice, the definition of the gradient flow (Eq. 2.4) becomes

$$\partial_\tau \vec{e}(\tau)_x = \left(1 - \vec{e}(\tau)_x \vec{e}(\tau)_x^T\right) \nabla^2 \vec{e}(\tau)_x, \tag{3.3}$$

where the Laplace operator ∇^2 is defined as

$$\nabla^2 \vec{e}(t)_x = \sum_{\mu=1}^{d} \left(\vec{e}(t)_{x+\hat{\mu}} - \vec{e}(t)_{x-\hat{\mu}} \right) - 2d\vec{e}(t)_x.$$

3.2.1 Runge-Kutta Algorithm

In order to calculate the gradient flow, we numerically solve the differential equation using a fourth-order Runga-Kutta approximation. This algorithm refines the Euler method

$$\vec{e}(\tau + h)_x \approx \vec{e}(\tau)_x + hf(\vec{e}(\tau)_x).$$

where $f(\vec{e})$ is defined for convenience as

$$f(\vec{e}) = \partial_\tau \vec{e}(\tau)_x = \left(1 - \vec{e}(\tau)_x \vec{e}(\tau)_x^T \right) \nabla^2 \vec{e}(\tau)_x, \tag{3.4}$$

following from Eq. 3.3. To the fourth order, this approximation becomes

$$k_1 = hf(\vec{e}(\tau)_x) \tag{3.5}$$

$$k_2 = hf(\vec{e}(\tau)_x + \frac{k_1}{2}) \tag{3.6}$$

$$k_3 = hf(\vec{e}(\tau)_x + \frac{k_2}{2}) \tag{3.7}$$

$$k_4 = hf(\vec{e}(\tau)_x + k_3) \tag{3.8}$$

$$\vec{e}(\tau + h)_x = \vec{e}(\tau)_x + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5). \tag{3.9}$$

This method is usually superior to Euler's method and the midpoint method [7].

To increase the efficiency of this algorithm, we implement the step-doubling algorithm to adaptively adjust h. If the error of a Runge-Kutta step is greater than the tolerance, the same step is repeated with half the step size. Alternatively, if the error is less than half of the tolerance, the step size is doubled for the next calculation. Finally, if the step size is greater than the distance to the next measurement, that distance is used as the step size, using the normal value afterwards. Otherwise, the algorithm proceeds with the consistent step size.

Chapter 4

Results

Chapter 5

Statistical Analysis

Chapter 6

Conclusion & Outlook

Appendix A

My Great Computer Program

A.1   Code sample

Bibliography

[1] Stefan Solbrig, Falk Bruckmann, Christof Gattringer, Ernst-Michael Ilgenfritz, Michael Müller-Preussker, and Andreas Schäfer. Smearing and filtering methods in lattice QCD - a quantitative comparison. *arXiv:0710.0480 [hep-lat]*, October 2007.

[2] Christopher Monahan and Kostas Orginos. Locally smeared operator product expansions in scalar field theory. *Phys. Rev. D*, 91(7):074513, April 2015.

[3] Hiroki Makino and Hiroshi Suzuki. Renormalizability of the gradient flow in the 2D $O(N)$ non-linear sigma model. *Progress of Theoretical and Experimental Physics*, 2015(3):33B08–0, March 2015.

[4] Wolfgang Bietenholz, Philippe de Forcrand, Urs Gerber, Héctor Mejía-Díaz, and Ilya O. Sandoval. Topological Susceptibility of the 2d O(3) Model under Gradient Flow. *Phys. Rev. D*, 98(11):114501, December 2018.

[5] David A Schaich and William Loinaz. *Lattice Simulations of Nonperturbative Quantum Field Theories*. PhD thesis, Amherst College, May 2006.

[6] Ulli Wolff. Monte Carlo errors with less errors. *Computer Physics Communications*, 176(5):383, March 2007.

[7] William T Vetterling, William T Vetterling, William H Press, William H Press, Saul A Teukolsky, Brian P Flannery, and Brian P Flannery. *Numerical Recipes: Example Book c.* Cambridge University Press, 1992.