My Dissertation Title

HONORS THESIS

Presented in Partial Fulfillment of the Requirements for Graduate
with Research Distinction in the Honors College of Arts and Sciences
of The Ohio State University

By

Matthias Heinz

Undergraduate Program in Physics

The Ohio State University

2018

Thesis Committee:

Professor Richard Furnstahl, Advisor

Prof. Richard Furnstahl

Prof. Robert Perry

Prof. P Sadayappan

© Copyright by

Matthias Heinz

2018

DRAFT MARCH 16, 2018

Abstract

An abstract goes here. It should be less than **500 words**.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Nuclear theory, which attempts to model the atomic nucleus and more generally nuclear matter, has been studied since the discovery of the neutron in 1932 (source? Dainton thesis). Nuclear matter is made of positively-charged protons and uncharged neutrons, which are generally referred to as nucleons. Nucleons interact primarily through the strong interaction, which ensures that stable and unstable nuclei are bound. Nuclear theory seeks to answer open questions in four areas: how do nuclei work, what are the properties of nuclear matter in astrophysical systems, what can be learned about beyond standard model (BSM) physics from the nucleus, and how can more accurate models of nuclear systems be leveraged in a variety of applications, ranging from national security and energy to medicine.

Nuclear theory faces two major challenges when trying to model systems of nucleons at low energies. The first is that, even for modest nucleus sizes, this is a quantum many-body problem. Quantum many-body theory is relevant to many different fields, including quantum chemistry and condensed matter theory. Many-body problems quickly become intractable when approached naively due to the combinatorial growth of the size of the problem with respect to the number of particles. The second challenge of nuclear theory is that, since the strong interaction is an interaction between partons (quarks and gluons which make up nucleons), a closed form of it between nucleons does not exist. These challenges forced nuclear physics in the past to rely on phenomenological models, both for the form of the strong interaction and the treatment of the many-body problem. These models were typically fit to experimental data for certain nuclei and used to predict the properties of nearby nuclei. However, their predictive ability did not extend far outside the domain in which they were fit, limiting their range of applicability.

Recent advancements in experimental nuclear physics, with facilities like the Facility for Rare Isotope Beams (FRIB) and the Laser Interferometer Gravitational-Wave Observatory (LIGO), have generated demand for more accurate calculations of already-calculated nuclear observables (need to introduce this concept somewhere) as well as new calculations of untouched territories with larger, more exotic nuclei. Moreover, accurate nuclear models are essential to understanding dense stars, supernovae, and certain astrophysical events, such as the collisions of neutron stars, which are hypothesized to be a location for the synthesis of heavy elements, such as gold and silver. Improved nuclear models are also essential in predicting the rate of certain nuclear decays, such as neutrino-less double-beta decay, which is currently being searched for by many experiments to answer whether neutrinos are Majorana particles, which means they are their own anti-particle, or not.

In recent years, the improvements in computational hardware and the development of new computational methods has brought about a rapid expansion in the range of nuclear isotopes able to be modeled via ab-initio calculations, or calculations from first principles (explain QCD?). The improvements in computational hardware have come through the development and proliferation of high-throughput devices (such as GPUs) and the assembly of highly parallel systems. Current trends suggest that a machine with exa-FLOP (floating-operations per second) throughput will exist by 2020 and such systems will be commonplace soon after. Much work is being done to ensure that the processing, networking, and power-consumption of these systems will continue to scale as it has for the past two decades.

The computational methods used in modern low-energy nuclear theory come in three classes: interaction models, renormalization group (RG) methods, and effective field theory (EFT) methods. Modern interaction models, like no-core shell model (NCSM) and lattice quantum chromodynamics (lattice QCD), seek to model nuclei from first principles and work in conjunction with RG and EFT methods to make these calculations feasible. RG methods modify interactions to match the resolution relevant to the problem at hand, which for low-energy nuclear physics means framing the problem in terms of low-energy nucleon-nucleon interactions, as opposed to the parton-parton interaction of the strong force. EFT methods offer systematically improvable, model-independent ordering of components of some interaction that allows for importance truncation and uncertainty quantification. These methods have already been used to offer both theoretical predictions that improve on previous calculations and theoretical predictions for nuclei that were unable to be modeled previously. The use of these methods in low-energy nuclear calculations is the state of the art. As a result, questions regarding their application are open problems and the focus of a lot of research.

The similarity renormalization group method (SRG) is an interaction softening method from the class of RG methods whose use in nuclear theory was first explored in a 1-dimensional setting at OSU by Eric Jurgenson in 2009 (citation?). It is a class of continuous unitary transformations that decouple large off-diagonal matrix elements in the interaction Hamiltonian, softening the potential as a result. The evolution of the Hamiltonian to a decoupled form allows a truncated subspace of the original basis to be used in later calculations without affecting the lowest energy eigenvalues which are the aspect of the operator that determine the calculated observables. This basis truncation offers a significant reduction in the size of the problem. SRG is frequently used in modern nuclear theory calculations to soften interactions and extend the range of certain calculations to larger systems. In our work, we return to a simple 1-dimensional setting to study features of SRG and seek to understand how to optimize its use in many-body calculations.

1.1 Contributions

Our main contributions are:

- We have developed an open-source Python library with easy to use abstractions for the SRG method for use by others to test SRG in a simple setting.
- We verify the results published in the 2009 Jurgenson paper, suggesting the correctness of our implementation. We also provide a fairly comprehensive suite of tests for the library, verifying that it fails gracefully when misused and correctly calculates the results simple analytically solvable cases when used correctly.
- We explore some alternative transformation generators and seek to quantify their performance relative to the standard generator that is currently used. We discuss the results of these calculations and offer some preliminary analysis.

1.2 Outline

The rest of the thesis is as follows:

- In chapter 2, we review the matrix representation of quantum operators. We then discuss the details of two different bases used, relative Jacobi momentum coordinates and 1-dimensional symmetrized harmonic oscillator states. We then discuss the details of the SRG method and explain the need for a careful revisit of SRG in a 1-dimensional setting.

- In chapter 3, we explain the design of the Python library. Much of the effort on this project went into making the library design logical and simple-to-use, so this chapter will explain the abstractions made and the reasoning behind those decisions.
- In chapter 4, we discuss ???
- In chapter 5, we discuss ???

Chapter 2
Similarity Renormalization Group Formalism

2.1 Quantum Mechanics Operators

2.1.1 Jacobi Coordinates

2.1.2 Harmonic Oscillator States with Proper Symmetry

2.2 Similarity Renormalization Group

The similarity renormalization group (SRG), whose use in nuclear physics was initially explored at OSU, is one method of reducing the computational complexity of low-energy nuclear calculations. The idea behind it is to continuously unitarily transform the operator of interest (for example, the Hamiltonian) into a simpler form. This simpler form is chosen to allow the large basis to be truncated without affecting the operator eigenvalues, which are essential to the truncated operator's utility in later calculations.

Chapter 3
Python Library

A large component of my research has been the development of a 1-D SRG Python library. The goal of this library is to offer a logical abstraction to the details of the SRG method to allow others to test features of SRG in one dimension. It aims to be as permissible as possible, keeping in line with the Python duck typing philosophy, while offering useful feedback for unsupported or obviously incorrect usage. A comprehensive set of unit tests for all parts of the library attempts to instill confidence that the library achieves this goal.

3.1 Design

The class-based abstractions offered by the library offer consistent representations of different logical classes of objects present in any SRG run. The goal is to make code for SRG runs clean and clear to any reader, and to encapsulate certain consistency conditions to make it difficult for the programmer to violate them and write obviously incorrect code. The SRG Solver class is logically different from the other classes in that is not a representation of some data, but rather the abstraction of a state machine of sorts, similar to the typically implementation of numericall differential equation solvers. Instances of the other classes represent concrete objects that have unambiguous logical representations.

IDEA: Tree graph explaining the dependencies between classes: ie, basis consists of an ordered set of states, operator couples a matrix to a specific basis, srg evolves an operator, etc.

NOTES:

- Need to work on OOP terms and decide on what specific parts are
- Is the term ̈dependencies ̈for classes correct?

3.1.1 State

The state class is one of two classes without dependencies. It offers an abstraction for an n-body harmonic oscillator wavefunction. The basic interface consists of a constructor to create an n-body state, and a `val` function to compute the value of the wavefunction at an $n-1$-tuple of momenta. This abstraction is necessary for the transformation from a momentum basis to a harmonic oscillator basis, as the transformation involves the evaluation of the wavefunction at every discretized momentum in the momentum basis.

There are two variants of the state class. The first is the basic 2-body harmonic oscillator state, which abstracts the basic 2-body harmonic oscillator wavefunction. The 2-body harmonic oscillator state is uniquely defined by its harmonic oscillator number. The `val` method returns the value of this single wavefunction, which involves the evaluation of the n-th Hermite polynomial, with n being the harmonic oscillator number.

The second variant is the general n-body (a-body?) state, which abstracts a normalized linear combination of $n-1$-body-2-body states. Every state has an additional condition for consistency: every product state in the linear combination must have the same total harmonic oscillator number. This prevents, among other things, a linear combination of 2-body states from being a valid state, as unique states have different total harmonic oscillator number. The `val` method returns the linear combination of the `val` method results for the 2-body and $n-1$-body states that make up the state. The recursive definition of the n-body state ensures that any n-body state created will be made up of states that are guaranteed to fulfill the consistency conditions.

3.1.2 Basis

The basis class offers an immutable, iterable, indexable, ordered collection of state objects. The basic interface implemented by all subclasses consists of an n-body property and a type property, to allow for the user to understand the basic qualities of the basis. The basis object is intended to be used to generate transformations and to be coupled with data matrices to give meaning to the data.

The first subclass of the basis class is the p-basis, a quick representation of a momentum basis. The data representation for an n-body p-basis is a list of n-tuples, with the first entry in the tuple being the weight of that node and the remaining $n-1$ entries being the Jacobi momenta for that node. The construction of the momentum basis supports the generation of an even or weighted grid (Gaussian quadrature) in single momentum space, then constructing the n-body space by generating all possible tuples of $n-1$ momenta from the single momentum space. Currently, this has only been used for 2-body momentum bases, as the potential used for my research is a 2-body potential.

The second subclass of the basis class is the HO-basis, a list of state objects. The construction of the n-body HO-basis generates a complete list of appropriatedly symmetrized states up to some user-defined maximum total harmonic oscillator number. As explained in the introduction (where?), the HO-basis offers a more general representation of the data that allows SRG to be applied easily to larger systems, unlike the p-basis. Thus, the generation of correctly symmetrized general n-body HO-bases is essential to testing SRG for larger systems.

3.1.3 Operator

3.1.4 SRG Solver

3.2 Implementation

3.2.1 Libraries Used

3.2.2 Handling Undefined Behavior

3.3 Testing

3.3.1 Framework

3.3.2 Code Coverage

Chapter 4
Exploring SRG in 1-D

4.1 Objectives

4.1.1 Verify Many-body Force Induction

4.1.2 Test Alternative Generators

4.2 Methods

4.2.1 2-Body Tests

4.2.2 3-Body Tests

Chapter 5
RESULTS

5.1 Many-Body Forces Induced

5.2 Alternative Generators