

THEORETICAL AND COMPUTATIONAL IMPROVEMENTS TO THE IN-MEDIUM
SIMILARITY RENORMALIZATION GROUP

By

Jacob Davison

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Physics—Doctor of Philosophy
Computational Mathematics, Science and Engineering—Dual Major

2023

ABSTRACT

There has been much progress in *ab initio* nuclear many-body theory in recent years, including a first attempt on an *ab initio* based mass table and a calculation of the neutron skin of ^{208}Pb based on two-plus three-nucleon interactions from chiral Effective Field Theory, with a sophisticated statistical uncertainty quantification. Going forward, there is a need to continue extending the capabilities of *ab initio* nuclear many-body theory to observables in medium-mass and heavy open-shell nuclei, enhance the description of exotic nuclei, and to build on theoretical and computational advances in nuclear structure to tackle nuclear dynamics. These advances in *ab initio* nuclear many-body theory, along with new methods for carrying controlled uncertainties into the relevant calculations, will provide new insight into nuclear forces and phenomena, as well as important input for searches for Beyond-Standard Model physics or nuclear astrophysics.

All of these goals requires theoretical and computational improvements of current nuclear many-body methods in order to enhance the stability of predictions, determine proper theoretical error bars, and to provide greater computational efficiency. In this body of work, we pursue these directions for the In-Medium Similarity Renormalization Group (IMSRG), which has become an important tool in *ab initio* nuclear many-body theory over the past decade.

We will discuss three major contributions in these directions. First, we present a model for the future infrastructure of IMSRG production codes using tensor network architectures, and show that it can soften the memory requirements and favorably improve the efficiency of IMSRG calculations. Second, we present a methodological improvement, the so-called reference state ensemble, that allows us to mitigate truncation errors in the IMSRG flow without performing calculations at a higher truncation rank. We show that the reference state ensemble improves the stability and accuracy of the IMSRG flow by “informing” the underlying operator basis about the features of the many-body system’s excitations. Last but not least, we use a data-driven technique called Dynamic Mode Decomposition (DMD) for emulating the IMSRG solution which reduces the cost of complete integration to convergence from hundreds or thousands of iterations to only tens of iterations. We also present a parametric emulation technique, powered by DMD and trained on

IMSRG data, which we show can robustly predict the IMSRG results for Hamiltonians, including chiral two- plus three-nucleon Hamiltonians with more than 20 parameters. We show that the parametric emulator can produce a global sensitivity analysis of statistical significance in only a few minutes, where full IMSRG(2) calculations would take thousands of years—improving the feasibility of uncertainty quantification for the IMSRG.

Copyright by
JACOB DAVISON
2023

To my wife, Sophie.
To my family.
To my friends.
To my advisors, my mentors, and my teachers throughout my career as a student.

“Serious students are not commonly plunged into fits of despair on turning the pages of a textbook and discovering that some further topic is known to the author but not to the student. Usually the students struggle a little, acquire the new knowledge, and, following an ancient human tradition, continue to turn the pages.”

– Carl Sagan, *Pale Blue Dot: A Vision of the Human Future in Space*

ACKNOWLEDGEMENTS

Foremost, I would like to acknowledge my advisor, Heiko Hergert. Without his support, encouragement, patience, and gracious flexibility, I would have been quite lost on my foray into graduate physics. Heiko is a veritable font of knowledge of all things many-body physics, and more. Thank you, Heiko, for guiding me on this journey.

I would also like to thank Scott Bogner, Alexei Bazavov, Longxiu Huang, and Jaideep Singh, for their commitment to serve on my guidance committee.

I want to thank the collaborating members of what I affectionately refer to as the “Heiko group,” to whom I presented my research progress every week and received thought-provoking advice and fruitful discussions. These members are Jiangming Yao, Boyao Zhu, and additionally Roland Wirth.

When I first arrived at MSU in the summer of 2018, I joined a rather large, exciting collaboration of three professors, Heiko Hergert (who, of course, would go on to be my direct advisor), Scott Bogner, and Morten Hjorth-Jensen, who were running a kind of “summer school” for incoming graduate students interested in their research. Their excitement for this field was infectious, and this introduction was absolutely critical and foundational for my success in the program, and eventually my own research. I want to thank Heiko, Scott, and Morten for being great advisors together. I also want to thank Justin Lietz for his mentorship in this initial summer introduction, and for being my WaMPS peer mentor when I first arrived as a new graduate student. Finally, I thank fellow graduate students Jane Kim, Julie Butler, and Yani Udiani for fruitful discussions and help in classes.

The main reason I pursued this doctoral program was my experience as an undergraduate in physics at Central Michigan University. I want to thank the entire CMU Department of Physics for providing an enriching undergraduate experience. I especially want to thank Georgios Perdikakis for his mentorship, as well as Alan Jackson, Christopher Tycner, Aaron LaCluyzé and Marco Fornari for helping me decide how to proceed in my research career.

I want to thank the entire MSU Department of Physics, although much larger than CMU, for providing an enriching graduate experience. Thanks to Kim Crosslan for navigating the

bureaucracies of the graduate school. I also want to thank the Theory Alliance here at the Facility for Rare Isotope Beams for their resources and support for aspiring nuclear theorists. There is a truly amazing collection of minds here at MSU; what a privilege to have learned from them.

My family is important to me. Without them, I certainly would not be where I am. This list is not exhaustive, but thanks to my mom Mona and my sister Jenna, my dad Jamie, Grandpa Ray and Grandma Linda, Uncle Stein and Aunt Emilie, Uncle Dave and Aunt Lina, Uncle Joe, my cousins Madelyn, Linnea, Gibson, and Sam. Thanks to my father-in-law Joe and my mother-in-law Tracy. Thanks to my dog Bean. Your unconditional encouragement has been a sail in the wind and a rudder in the sea. Love you all!

Thanks to my best friends, Austin, Austin, Alan, Ben, Brady, Brian, Jake, Peter, and Zach, all of whom I have known, in some capacity, since kindergarten. We did it boys.

Thanks to my wife Sophie, who I met at Central Michigan University. She was a chemistry major and I was a physics major. A love story for the ages. Since we met, we have completed every progression of our professional careers together. We pursued and accomplished an undergraduate research internship together, we finished our Bachelor's degrees together, we attended MSU together for our graduate degrees, and now I like to think we are finishing this dissertation together (although, I assure you that I wrote this dissertation myself). Love you, Sophie.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS	x
CHAPTER 1 OVERVIEW	1
BIBLIOGRAPHY	6
CHAPTER 2 MANY-BODY THEORY IN SECOND QUANTIZATION	8
2.1 Introduction	8
2.2 Many-Body Basis	8
2.3 Annihilation and Creation Operators	10
2.4 Normal-Ordering and Wick Contractions	12
2.5 Wick's Theorem	15
2.6 Correlated Reference States and Irreducible Density Matrices	16
2.7 Normal-ordering the Two-body Hamiltonian	17
2.8 Density Matrix Normal-ordering	20
2.9 Slater Determinant Occupation Number Representation	20
2.10 Full Configuration Interaction Theory to Solve Model Hamiltonians	21
BIBLIOGRAPHY	23
CHAPTER 3 IN-MEDIUM SIMILARITY RENORMALIZATION GROUP	24
3.1 Introduction	24
3.2 Similarity Renormalization Group	24
3.3 IMSRG Formalism	25
3.4 Generators for the IMSRG	32
BIBLIOGRAPHY	37
CHAPTER 4 MODEL HAMILTONIANS	38
4.1 The Pairing Model Plus Particle-Hole Model	38
4.2 Chiral Effective Field Theory for Nuclear Interactions	42
BIBLIOGRAPHY	49
CHAPTER 5 BUILDING AN IMSRG MANY-BODY CODE USING TENSOR NETWORKS	50
5.1 Introduction	50
5.2 Tensor Network Diagrams	51
5.3 Factorization with Tensor Train Decomposition	52
5.4 Tensor Networks in the IMSRG	58
5.5 Tensor-factors IMSRG: Python Implementation for the P3H Model	61
5.6 Tensorized C++ IMSRG: C++ Implementation for the P3H Model	64
BIBLIOGRAPHY	67
CHAPTER 6 OPTIMAL REFERENCE STATE ENSEMBLES FOR THE IN-MEDIUM SIMILARITY RENORMALIZATION GROUP	69
6.1 Introduction	69

6.2	Truncation Error in the IMSRG	69
6.3	Formulation of the Reference State Ensemble	70
6.4	Ensemble Optimization Scheme	72
6.5	Regularization of Generator Divergences	73
6.6	Improving the Eigenspectrum	74
6.7	Ensemble Consistency with Non-Targeted Observables	77
6.8	Using Exact Eigenstates as References	79
6.9	Outlook	83
	BIBLIOGRAPHY	85
CHAPTER 7	KOOPMAN OPERATOR FOR THE IN-MEDIUM SIMILARITY RENORMALIZATION GROUP	86
7.1	Introduction	86
7.2	The Koopman Operator	86
7.3	Dynamic Mode Decomposition for IMSRG	87
7.4	Emulating the IMSRG Flow	90
7.5	Emulating the Parametric IMSRG Flow	97
7.6	Interpolating DMD using RBFs	99
7.7	Optimized Training for Large Bases using Streaming SVD	106
7.8	Outlook	110
	BIBLIOGRAPHY	112
CHAPTER 8	GLOBAL SENSITIVITY ANALYSIS USING EMULATION OF THE IMSRG	114
8.1	Introduction	114
8.2	GSA with a Chiral N ³ LO Nucleon-Nucleon Interaction	115
8.3	GSA with Chiral NNLO Two- Plus Three-Nucleon Interactions	116
8.4	Discussion and Outlook	123
	BIBLIOGRAPHY	125
CHAPTER 9	CONCLUSIONS AND OUTLOOK	127
APPENDIX A	TENSOR TRAIN DECOMPOSITION OF THE OCCUPATION FAC- TORS	130
APPENDIX B	MULTI-REFERENCE IMSRG(2) FLOW EQUATIONS	132
APPENDIX C	CONTRIBUTED CODE PROJECTS	135

LIST OF ABBREVIATIONS

IMSRG	In-Medium Similarity Renormalization Group
DMD	Dynamic Mode Decomposition
IE	Interpolation Engine
IEM	Interpolation Engine Model
rIEM	Reduced Interpolation Engine Model
UQ	Uncertainty Quantification
FCI	Full Configuration Interaction
IBCs	Irreducible Brillouin Conditions
SD	Slater determinant
LECs	Low-energy constants
QCD	Quantum chromodynamics
LO	Leading order
NLO	Next-to leading order
NNLO	Next-to-next-to leading order
N³LO	Next-to-next-to-next-to leading order
TT	Tensor Train
IDMs	Irreducible Density Matrices
SVD	Singular Value Decomposition
rKOI	Reduced Koopman Operator Interpolation
RBF	Radial Basis Function
SKL	Sequential Karhunen-Lo��ve
GSA	Global Sensitivity Analysis
PCA	Principal Component Analysis

CHAPTER 1

OVERVIEW

There has been much progress in *ab initio* [3] nuclear many-body theory in the past decade, which is illustrated in Figure 1.1. For example, there has been a first attempt on an *ab initio* based mass table [11] and the neutron skin of ^{208}Pb has been computed based on two- plus three-nucleon interactions from chiral Effective Field Theory [8, 9], with a sophisticated statistical uncertainty quantification [6].

The next tasks of *ab initio* nuclear theory are (i) to extend its capabilities into the domain of medium-mass and heavy open-shell nuclei, which exhibit strong collective correlations associated with intrinsic deformation, (ii) to enhance the description of exotic nuclei, where weak-binding and the coupling to continuum degrees of freedom lead to subtle new phenomena, and (iii) to build on the theoretical and computational advances in nuclear structure in order to tackle nuclear dynamics.

In this way, *ab initio* nuclear many-body calculations with controlled uncertainties will provide crucial new input to tackle grand scientific challenges, like the search for Beyond-Standard Model physics [2], or the improvement of our understanding of supernovae and neutron star mergers, and their detailed role in nucleosynthesis.

All of this requires theoretical and computational improvements of current nuclear many-body methods, in order to enhance the stability of predictions, determine proper theoretical error bars, and to provide much greater computational efficiency. In the present work, we pursue next steps in these directions for the In-Medium Similarity Renormalization Group (IMSRG), which has become an important tool in *ab initio* nuclear many-body theory over the past decade.

Chapter 2 introduces some essential aspects of nuclear many-body theory theory in second quantization that are needed for the discussion in this work; following that is Chapter 3, which is dedicated to a detailed presentation of the IMSRG. The IMSRG is an *ab initio*, or first principles, method for solving the nuclear many-body problem. Importantly, the IMSRG can also be used to pre-process interactions and operators for use in other methods (Valence-Space IMSRG [12],

IM-GCM for Double Beta Decay [13], IN-NCSM [4]). Functionally, the IMSRG decouples a target *reference state*, e.g. the ground state, from all other excitations in the many-body basis via a continuous, unitary transformation that effectively block-diagonalizes the Hamiltonian (Chapter 4 discusses the model Hamiltonians used for the analyses in this work). Excitations are gradually suppressed according to a particular decoupling scheme which separates diagonal elements from off-diagonal elements in a chosen operator basis. Typically, the most convenient operator basis is generated through a process called *normal-ordering* in the language of second quantization (Chapter 2). Then, instead of calculating the IMSRG transformation explicitly, we solve for the transformation implicitly through integration of the IMSRG flow equations (Chapter 3.3), which involves a commutator of the Hamiltonian and an object called the generator; this object *generates* the desired behavior of off-diagonal elements of the Hamiltonian over the integration range. The commutator itself generates terms of increasingly higher-body interactions, and tracking all of them is not computationally feasible. Thus, we must introduce a truncation scheme. In the IMSRG(2), for example, we truncate all operators in the chosen operator basis at the two-body level. The IMSRG is favorable due its polynomial scaling in basis size. Additionally, the IMSRG is built in a framework which allows systematic improvement toward exact solutions via the increase in truncation rank, which introduces increasingly higher-fidelity physics information.

Chapter 5 introduces a model for the future infrastructure of the IMSRG production codes, the motivation for which is built on the exchange of tensor libraries on the backend. We also show how the Tensor Train decomposition (Chapter 5.3) can be used to rewrite particular objects in the flow equations, which reduces the memory demand of those objects by several orders of magnitude in extreme cases. This technique holds the key for efficiently leveraging the ever-evolving hardware landscape, and lets us benefit from the work of library vendors who know best how to adapt tensor calculations to new platforms.

Chapter 6 discusses the use *reference state ensembles* for the IMSRG, which is a technique for potentially mitigating truncation errors while avoiding the computational cost of performing IMSRG at a higher truncation rank. We show that error due to truncation at two-body operators

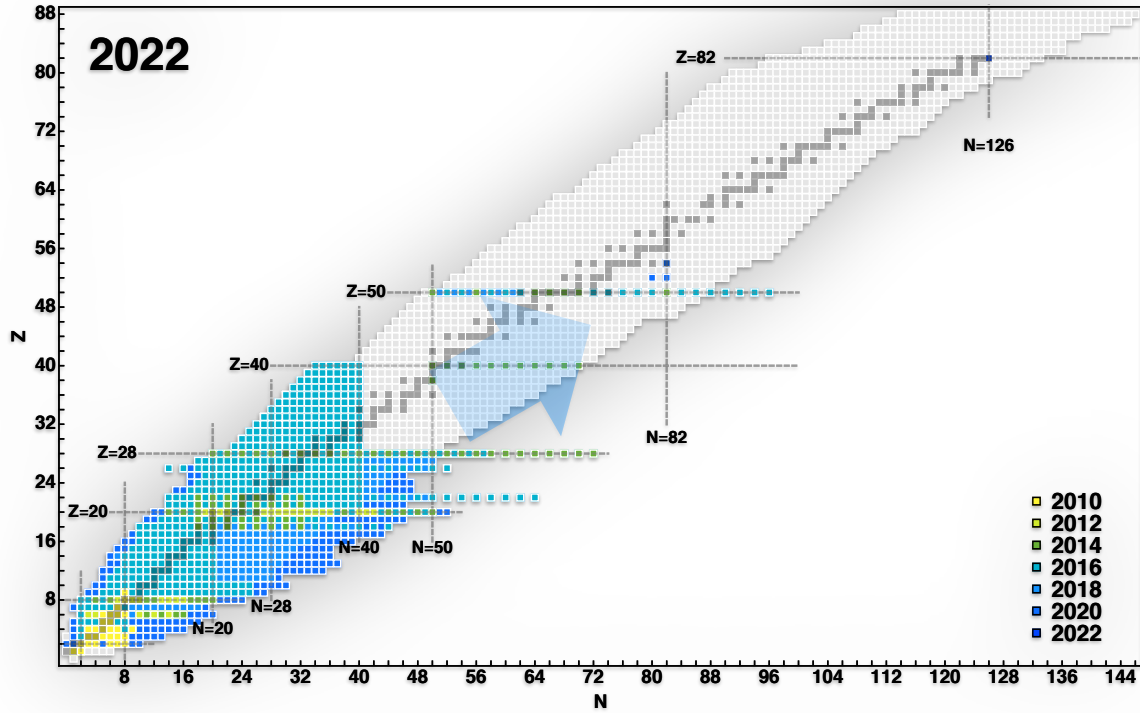


Figure 1.1 Figured updated from [5]. Each filled square represents a nucleus that has been successfully calculated using IMSRG.

manifests as a loss of unitarity in the IMSRG(2) transformation. The severity of this loss depends on the choice of operator basis for the particular problem we study, and is related to the importance of induced terms which have been truncated. The goal of the *reference state ensemble* is to introduce additional physics information to the IMSRG(2) via correlations in the operator basis. In the reference state ensemble scheme, we construct a reference state from statistically-mixed many-body basis configurations (e.g. approximate ground- and excited-state configurations), and then use *that* reference as a starting point for the IMSRG(2). In this way, we introduce more descriptive physical information to the standard operator basis of the IMSRG. We show that the *reference state ensemble* is a controllable method for improving divergences in the IMSRG(2) related to truncation error, as well as improving accuracy in the evolved energy spectrum.

Chapter 7 presents a method for emulating the IMSRG solution using an approach called Dynamic Mode Decomposition (DMD), which is based on Koopman operator theory. The Koopman operator is a linear operator which advances measurement functions of a nonlinear dynamical system forward in time [7, 1]. The eigendecomposition of the Koopman operator characterizes the dynamics of the system, as expressed through measurements of observables that depend on the system’s state variables. Thus, it can be understood as shifting the description of the system to coordinates where the nonlinear dynamics look linear, analogous to canonical transformations in the phase-space formulation of classical mechanics. DMD is a *finite-measurement, discrete-time* method for approximating the Koopman operator. The DMD operator is formulated as the best-fit linear operator which propagates the dynamical system forward one step in time [10]. The DMD is informed by direct measurements of the evolving system. We show that the DMD operator, built from IMSRG(2) observations and informed by simple physical constraints on the predicted dynamics, offers a method by which we may construct a true emulator for the IMSRG(2) solution. DMD emulation allows us to forecast the IMSRG(2) evolution for a particular Hamiltonian with high accuracy and short computational wall-time, at the cost of a few polynomial-time iterations of the IMSRG(2) solution which serve as “training points” for the emulator. We also present an interpolation engine (IE) for the IMSRG, constructed from DMD, which we use to train an interpolative model of the parametric IMSRG, i.e. a model where the inputs include Hamiltonian coupling parameters as well as the dynamical variable. We show that fitted IE model (IEM) produces predictions with robust accuracy for up to 24 coupling parameters in the Hamiltonian.

The parametric emulation of the IMSRG(2) offer a strategy for undertaking the large volumes of calculations required for uncertainty quantification (UQ) of the IMSRG, so that we may accurately assign theoretical constraints on the output. Chapter 9 demonstrates this capability by performing sensitivity analyses of observables computed in IMSRG(2) to the parameters of the underlying chiral interactions. We show that a statistically significant number of calculations can be performed in just a few minutes using the parametric emulator, whereas the same number of full-scale IMSRG(2) calculations would require thousands of years. The feasibility of these sensitivity analyses, using

parametric emulation, is a major step toward feasibility of UQ for the IMSRG.

BIBLIOGRAPHY

- [1] Steven L. Brunton et al. *Modern Koopman Theory for Dynamical Systems*. 2021. arXiv: 2102.12086 [math.DS].
- [2] Vincenzo Cirigliano et al. *Neutrinoless Double-Beta Decay: A Roadmap for Matching Theory to Experiment*. 2022. arXiv: 2203.12169 [hep-ph].
- [3] A. Ekström et al. “What is ab initio in nuclear theory?” In: *Frontiers in Physics* 11 (2023). ISSN: 2296-424X. DOI: 10.3389/fphy.2023.1129094. URL: <https://www.frontiersin.org/articles/10.3389/fphy.2023.1129094>.
- [4] Eskendr Gebrerufael et al. “Ab Initio Description of Open-Shell Nuclei: Merging No-Core Shell Model and In-Medium Similarity Renormalization Group”. In: *Phys. Rev. Lett.* 118 (15 Apr. 2017), p. 152503. DOI: 10.1103/PhysRevLett.118.152503. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.118.152503>.
- [5] Heiko Hergert. “A Guided Tour of ab initio Nuclear Many-Body Theory”. In: *Frontiers in Physics* 8 (2020), p. 379.
- [6] Baishan Hu et al. “Ab initio predictions link the neutron skin of 208Pb to nuclear forces”. In: *Nature Physics* 18.10 (Aug. 2022), pp. 1196–1200. DOI: 10.1038/s41567-022-01715-8. URL: <https://doi.org/10.1038/s41567-022-01715-8>.
- [7] B. O. Koopman. “Hamiltonian Systems and Transformation in Hilbert Space”. In: *Proceedings of the National Academy of Sciences* 17.5 (1931), pp. 315–318. DOI: 10.1073/pnas.17.5.315. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.17.5.315>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.17.5.315>.
- [8] R. Machleidt and D.R. Entem. “Chiral effective field theory and nuclear forces”. In: *Physics Reports* 503.1 (June 2011), pp. 1–75. DOI: 10.1016/j.physrep.2011.02.001. URL: <https://doi.org/10.1016/j.physrep.2011.02.001>.
- [9] Maria Piarulli and Ingo Tews. “Local Nucleon-Nucleon and Three-Nucleon Interactions Within Chiral Effective Field Theory”. In: *Frontiers in Physics* 7 (2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2019.00245. URL: <https://www.frontiersin.org/articles/10.3389/fphy.2019.00245>.
- [10] Peter J. Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28. DOI: 10.1017/S0022112010001217.
- [11] S. R. Stroberg et al. “Ab Initio Limits of Atomic Nuclei”. In: *Phys. Rev. Lett.* 126 (2 Jan. 2021), p. 022501. DOI: 10.1103/PhysRevLett.126.022501. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.126.022501>.

- [12] S. Ragnar Stroberg et al. “Nonempirical Interactions for the Nuclear Shell Model: An Update”. In: *Annual Review of Nuclear and Particle Science* 69.1 (2019), pp. 307–362. DOI: 10.1146/annurev-nucl-101917-021120. eprint: <https://doi.org/10.1146/annurev-nucl-101917-021120>. URL: <https://doi.org/10.1146/annurev-nucl-101917-021120>.
- [13] J. M. Yao et al. “Ab Initio Treatment of Collective Correlations and the Neutrinoless Double Beta Decay of ^{48}Ca ”. In: *Phys. Rev. Lett.* 124 (23 June 2020), p. 232501. DOI: 10.1103/PhysRevLett.124.232501. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.124.232501>.

CHAPTER 2

MANY-BODY THEORY IN SECOND QUANTIZATION

2.1 Introduction

Much of the nuclear theory toolbox leveraged by this body of work is constructed within the “second quantization” formalism for many-body quantum theory. Second quantization is concerned with modeling a space which contains multiple quantum systems, e.g. the nucleus of an atom for which the number of particles is not constant. Second quantization, which exists inside of Fock space, provides tools which allow us to navigate between individual Hilbert spaces, and is the most convenient framework for quantum many-body theory.

The purpose of this chapter is to provide context and definitions for these tools which are used in following chapters. Much of the information here is compiled from Shavitt and Bartlett [6] and may take on the flavor of quantum chemistry as a result; however, the notation and techniques are general enough to be understood in the context of nuclear theory.

2.2 Many-Body Basis

The “many-body basis” may be understood as a chosen collection of states, varying particle number, which exist inside the Fock space. Formally, the *fermionic* Fock space is the *direct sum* over all antisymmetrized tensor products of Hilbert spaces of quantum states up to N particle numbers,

$$\mathcal{F}(H) = \mathbb{C} \oplus H \oplus \mathcal{A}(H \otimes H) \oplus \dots \mathcal{A}(H^{\otimes N}), \quad (2.1)$$

where $\mathcal{A}$ is an operator which generates the antisymmetrization. Equivalent notation which compresses the definition reads

$$\mathcal{F}_A(H) = \bigoplus_{n=0}^N \mathcal{A}H^{\otimes n}. \quad (2.2)$$

The Fock states are acted upon by operators which also exist inside the Fock space. For the purposes of this work, we are interested in *fermionic* Fock space because the techniques we apply are meant to model nuclear systems. Fermionic Hilbert spaces are *antisymmetric* under the exchange of particles. The complete *single particle basis* spans the Hilbert space of a single

particle, and is uniquely characterized by a set of quantum numbers that are associated with the static and dynamical properties of that particle. [6]. The Hilbert space for an A -body system is constructed as a product of single-particle Hilbert spaces, which is spanned by tensor products of the single-particle states. We notate the single particle basis ϕ_p , which encodes the set of all possible single particle states indexed by $p = 1, 2, 3, \dots$. We assume that the single particle basis is orthonormal. Schematically, we might write the Hilbert space for an A -body system as,

$$H(A) = \bigotimes_{n=0}^A \phi_n. \quad (2.3)$$

The antisymmetrizer $\mathcal{A}$ enforces that condition that product states, elements of the A -body Hilbert space in Equation 2.3, be *antisymmetric* under the exchange of particle, e.g. for a two-particle product state,

$$\phi_q^{(1)} \otimes \phi_p^{(2)} = -\phi_p^{(1)} \otimes \phi_q^{(2)}. \quad (2.4)$$

The antisymmetrizer $\mathcal{A}$ generates the so-called *Slater determinant*, such that

$$\begin{aligned} \Phi &= \mathcal{A} \left(\phi_p^{(1)} \otimes \phi_q^{(2)} \right) = \sqrt{\frac{1}{2}} \left(\phi_p^{(1)} \otimes \phi_q^{(2)} - \phi_q^{(1)} \otimes \phi_p^{(2)} \right) \\ &= \sqrt{\frac{1}{2}} \begin{vmatrix} \phi_p^{(1)} & \phi_p^{(2)} \\ \phi_q^{(1)} & \phi_q^{(2)} \end{vmatrix}. \end{aligned} \quad (2.5)$$

For brevity, we will often write a Slater determinant Φ as a ket of single particle states, or maybe single particle indices,

$$\Phi = |\phi_p \phi_q \phi_r \dots\rangle, \quad (2.6)$$

where, intuitively, the single particle states collected in this string are *occupied* within this particular state. The antisymmetrizer has been implicitly absorbed into the notation. Note that e.g. $|\phi_p \phi_q\rangle = -|\phi_q \phi_p\rangle$. In principle, there an infinite number of *unoccupied* states which we could write for this particular state, but there is no reason to write all of them. Additionally, we might choose to drop the ϕ for brevity, settling on raw indices $p, q, r, \dots$ to represent these single particle states which are occupied in the Slater determinant.

Of course, the order in which the single particle states appear in this string is deliberate, and the fermionic nature of the Slater determinant guarantees that permuting the order of these states results in a change of phase. Written succinctly [6],

$$\hat{P} |pqr \dots\rangle = (-1)^{\sigma(\hat{P})} |pqr \dots\rangle. \quad (2.7)$$

Permuting the indices of Slater determinant $|pqr \dots\rangle$ via $\hat{P}$ results in a phase change of order $\sigma(\hat{P})$, which we must track whenever we perform operations on the Slater determinants in our relevant bases. This *signature of the permutation* counts the minimum number of two particle exchanges that is required to produce the final ordering.

The Slater determinants span a complete orthonormal basis of the fermionic subspace of H_A , where

$$\langle pqr \dots | pqr \dots \rangle = 1, \quad (2.8)$$

when the indices of the bra are in the same ordering as the indices of the ket. Note that indices in bras or kets can always be permuted to match the ordering, at the cost of a phase.

2.3 Annihilation and Creation Operators

In the context of second quantization, the annihilation and creation operators allow us to add or remove (respectively) occupied states in a particular Slater determinant. Interpreted a different way, the annihilation and creation operators connect states of differing number of particles, within the Fock space. We denote the annihilation operator by a_p and its Hermitian adjoint, the creation operator, as $a_p^\dagger$ [6]. The operators are defined by their action on the Slater determinant, in the following way:

$$a_p |pqr \dots\rangle = |qr \dots\rangle, \quad a_p |qr \dots\rangle = 0, \quad (2.9)$$

$$a_p^\dagger |qr \dots\rangle = |pqr \dots\rangle, \quad a_p^\dagger |pqr \dots\rangle = 0. \quad (2.10)$$

Here we show that the annihilation operator a_p removes a particle from the occupied single-particle state associated with p in the target Slater determinant. Similarly, the creation operator $a_p^\dagger$ adds a new particle to the single-particle state p . Attempting to annihilate a particle in a state which

is already unoccupied, or attempting to create a particle in a state which is already occupied, is forbidden. Note that this zero result is different from the vacuum state $|0\rangle$, which is defined as the Slater determinant with no occupied single-particle states. It is clear that any application of an annihilator to this vacuum leads to a zero result, i.e.

$$a_p |0\rangle = 0 = \langle 0| a_p^\dagger. \quad (2.11)$$

The fermionic creation and annihilation operators obey the following anticommutation relations [6], which will be important for our subsequent discussion:

$$\{a_p, a_q\} = \{a_p^\dagger, a_q^\dagger\} = 0; \quad \{a_p^\dagger, a_q\} = \delta_{pq}. \quad (2.12)$$

Note that any A -body Slater determinant can be constructed by a string of creation operators acting on the vacuum [2],

$$\prod_{k=1}^A a_{p_k}^\dagger |0\rangle = |p_1 p_2 \cdots p_A\rangle. \quad (2.13)$$

In this representation, we could compute the matrix element associated with two Slater determinants (sometimes called the *overlap*) via

$$\langle q_1 q_2 \cdots q_B | p_1 p_2 \cdots p_A \rangle = \langle 0 | \prod_{l=1}^B a_{q_l} \prod_{k=1}^A a_{p_k}^\dagger | 0 \rangle. \quad (2.14)$$

The result depends on the number of operators A and B , as well as the ordering of indices. If $B = A$ and indices p_k, q_l have matching order, then the result is 1. In this work, we will only concern ourselves with *particle number conserving operators*, so we can assume that $B = A$ unless stated otherwise.

In practical applications within nuclear theory, the vacuum state is an inefficient reference point from which to define the many-body basis; this is because of the resolution scale of the nuclear interaction [2]. More importantly, the vacuum state contains no information about the energy scales and occupation density of the nuclear matter we wish to describe. As a result, we may also define the creation and annihilation operators with respect to a particular reference state $|\Phi\rangle$ (called the *Fermi vacuum* for a Slater determinant reference state), which is often chosen as an approximation

to the ground state. In this case, we would define *particle* and *hole* operators, which are distinct in the kind of single particle state index on which they operate. The hole operators, usually indexed by $i, j, k, \dots$, act on states which are *occupied* in $|\Phi\rangle$. The particle operators, usually indexed by $a, b, c, \dots$, act on states which are *unoccupied* in $|\Phi\rangle$. We can construct excited Slater determinants by acting on the reference state with combinations of creation and annihilation operators, e.g.

$$a_a^\dagger a_i |\Phi\rangle = |\Phi_i^a\rangle, \quad (2.15)$$

where the operator pair $a_a^\dagger a_i$ annihilates a particle in the occupied (hole) state i from the reference and creates a particle in the unoccupied (particle) state a . This action can be interpreted as a *one-particle excitation* relative to the reference state, and we may define a complete many-body basis as the set of all possible one-, two-, up to A -particle excitations for an A -body reference state.

2.4 Normal-Ordering and Wick Contractions

The process of subtracting off *contractions* in the product of operators is called *normal-ordering*. The normal-ordered product of two operators $\hat{A}\hat{B}$ reads

$$n [\hat{A}\hat{B}] = \hat{A}\hat{B} - \overline{\hat{A}\hat{B}}, \quad (2.16)$$

where the contraction, in this context also called the Wick contraction, $\overline{\hat{A}\hat{B}}$ is *defined* as the expectation value of $\hat{A}\hat{B}$ in the chosen reference state (e.g. the vacuum, in this case),

$$\overline{\hat{A}\hat{B}} \equiv \langle 0 | \hat{A}\hat{B} | 0 \rangle. \quad (2.17)$$

By construction, this definition of the contraction leads to the consequence that the expectation value of the normal-ordered operator $n [\hat{A}\hat{B}]$ is *zero*, in the reference state for which the contraction was defined:

$$\overline{\hat{A}\hat{B}} = \langle 0 | \hat{A}\hat{B} | 0 \rangle - \langle 0 | n [\hat{A}\hat{B}] | 0 \rangle \quad (2.18)$$

$$\implies \langle 0 | n [\hat{A}\hat{B}] | 0 \rangle = 0 \quad \text{by Eq. 2.17.} \quad (2.19)$$

These definitions extend to any length string of operator products, but will not be derived here (refer to Shavitt and Bartlett [6] and Hergert et al. [2]). These definitions also extend to any general reference state in which we define the Wick contraction.

We can exchange any two operators inside the normal-ordered product at the cost of a sign change, e.g.

$$n [\hat{A}\hat{B}] = -n [\hat{B}\hat{A}]. \quad (2.20)$$

Note that the contractions induced by permuting operators in the operator string have been taken care of by the expansion in terms generated by the normal-ordering process.

Nuclear many-body theory primarily deals with pairs of annihilator/creator operators, e.g. $a_p^\dagger a_q$ or $a_p a_q^\dagger$. The Wick contraction of operator pair $a_p a_q^\dagger$ may be written, by virtue of Equation 2.16, as

$$\overline{a_p a_q^\dagger} = a_p a_q^\dagger - n [a_p a_q^\dagger] = \{a_p, a_q^\dagger\} = \delta_{pq}, \quad (2.21)$$

where we have written the only operator pair which gives a nonzero contraction, with respect to the vacuum. In this case, a particle in state q is created in the vacuum and a particle in state p is annihilated — thus, the contraction is 1 for $p = q$ and zero otherwise. Interpreted a different way, this contraction is the expectation value of the operator $a_p a_q^\dagger$ in the vacuum.

In a similar fashion, the Wick contraction of operator pairs with respect to the general reference is defined as

$$\begin{aligned} \overline{a_a a_b^\dagger} &= \delta_{ab} \\ \overline{a_i^\dagger a_j} &= \delta_{ij}. \end{aligned} \quad (2.22)$$

We can interpret the results from the perspective of occupied and unoccupied states in the reference. The particle operator contraction is the expectation value $\langle \Phi | a_a a_b^\dagger | \Phi \rangle$, and is only nonzero when $b = a$. The hole operator contraction is the expectation value $\langle \Phi | a_i^\dagger a_j | \Phi \rangle$, and is only nonzero when $i = j$.

For a general reference state, the expectation value $\langle \Phi | a_p^\dagger a_q | \Phi \rangle$ defines the one-body density matrix:

$$\rho_{qp} = \langle \Phi | a_p^\dagger a_q | \Phi \rangle = \overline{a_p^\dagger a_q} \langle \Phi | \Phi \rangle + \langle \Phi | N [a_p^\dagger a_q] | \Phi \rangle. \quad (2.23)$$

Note that the reference state is normalized so that $\langle \Phi | \Phi \rangle = 1$. The definition of the Wick contraction implies the expectation value of the normal-product $\langle \Phi | N [a_p^\dagger a_q] | \Phi \rangle$ must vanish [2]. It follows

that the reference expectation value of any normal-product, with respect to that reference state, must vanish (analogous to Equation 2.19):

$$\langle \Phi | N [\dots] | \Phi \rangle = 0. \quad (2.24)$$

Finally, note that we can drop the index distinction in Equation 2.22 altogether using the anticommutation relations in Equation 2.12. Starting with the definition of the contraction between hole operators and allowing the indices to span the entire single particle basis, we can define a new contraction $\overline{a_p a_q^\dagger}$ which is related to $\overline{a_p^\dagger a_q}$ in the following way:

$$\begin{aligned} \langle \Phi | a_p^\dagger a_q | \Phi \rangle &= \overline{a_p^\dagger a_q} = \rho_{qp} \\ \langle \Phi | a_q a_p^\dagger | \Phi \rangle &= \overline{a_q a_p^\dagger} = \langle \Phi | (\delta_{pq} - a_p^\dagger a_q) | \Phi \rangle = \delta_{pq} - \rho_{qp}. \end{aligned} \quad (2.25)$$

These two contractions automatically encode the behavior of expectation values of hole/particle pairs of operators in the reference state.

For a single Slater determinant reference state, the elements of $\overline{a_p^\dagger a_q}$ are nonzero only where both p, q indices are occupied states in the reference. If either p or q (or both) are unoccupied states, the element is zero for one of two reasons: 1) the element vanishes due to Slater determinant orthogonality or 2) the action of a particle-index annihilator on the reference yields a zero result. Therefore, for a single Slater determinant reference state, the contraction $\overline{a_q a_p^\dagger}$ yields elements which are 0 where the one-body density matrix is nonzero, and nonzero where the one-body density matrix is 0.

For brevity and convenience, we usually work in the eigenbasis of the one-body density matrix:

$$\rho_{pq} = n(p) \delta_{pq}, \quad (2.26)$$

where the factor $n(p)$ encodes the occupation of the single particle state p , i.e.,

$$n(p) = \begin{cases} 0 & \text{if } p \text{ is unoccupied in ref. (particle state)} \\ 1 & \text{if } p \text{ is occupied in ref. (hole state)} \end{cases} \quad (2.27)$$

Analogously to Equation 2.25, we define a *particle* occupation factor $\bar{n}(p)$, such that

$$\bar{n}(p) = 1 - n(p). \quad (2.28)$$

These *occupation factors* are convenient for expressing occupation information in a general single particle basis because the diagonality of the one-body density ρ eliminates summations in expansions of operators (e.g., refer to the IMSRG(2) flow equations later in Chapter 3, Equation 3.18).

2.5 Wick's Theorem

Whether with respect to the vacuum or a reference state, Wick's theorem states that any (particle-number conserving) string of creation and annihilation operators can be written as the sum of all possible normal-product contractions [2, 1, 6]:

$$\begin{aligned}
& a_{p_1}^\dagger \cdots a_{p_A}^\dagger a_{q_A} \cdots a_{q_1} \stackrel{\text{WT}}{=} N [a_{p_1}^\dagger \cdots a_{p_A}^\dagger a_{q_A} \cdots a_{q_1}] \\
& + \overline{a_{p_1}^\dagger a_{q_1}} N [a_{p_2}^\dagger \cdots a_{p_A}^\dagger a_{q_A} \cdots a_{q_2}] - \overline{a_{p_1}^\dagger a_{q_2}} N [a_{p_2}^\dagger \cdots a_{p_A}^\dagger a_{q_A} \cdots a_{q_1}] + \text{single contractions} \\
& + \left(\overline{a_{p_1}^\dagger a_{q_1}} \overline{a_{p_2}^\dagger a_{q_2}} - \overline{a_{p_1}^\dagger a_{q_2}} \overline{a_{p_2}^\dagger a_{q_1}} \right) N [\cdots] + \text{double contractions} \\
& + \text{triple contractions} + \cdots + \text{full contractions}.
\end{aligned} \tag{2.29}$$

Notice that the process of removing the contraction from the normal-product results in an appropriate phase change on the normal-product operator string. The resulting sum of operators assumes the general form [2],

$$A^M B^N = \sum_{k=|M-N|}^{M+N} C^{(k)}, \tag{2.30}$$

where the lower bounds exist because we are not able to fully contract all indices unless $M = N$.

The power of Wick's theorem is that we can take any many-body operator product and “decompose” the product into a sum of normal-ordered products. In this expansion, we need only worry about phases when permuting the operators in the normal-ordered string, because contributions from the δ in the anticommutator have vanished according to the contractions. Additionally, when computing an expectation value with this theorem, only full contractions survive by Equation 2.24.

We may leverage Wick's theorem to define the two-body density matrix. Analogously to the one-body hole density matrix in Equation 2.23, the two-body hole density encode two-body

correlations in the reference state. We write,

$$\rho_{rspq} = \langle \Phi | a_r^\dagger a_s^\dagger a_q a_p | \Phi \rangle \stackrel{\text{WT}}{=} \rho_{rp} \rho_{sq} - \rho_{rq} \rho_{sp}, \quad (2.31)$$

where we have used Wick's theorem to further decompose the two-body operator string. Using Wick's theorem, we have shown that the two-body hole density matrix, for a single Slater determinant reference state, can be reduced to a product of one-body density matrices. The three-body density matrix factorizes analogously,

$$\begin{aligned} \rho_{stupqr} &= \langle \Phi | a_s^\dagger a_t^\dagger a_u^\dagger a_r a_q a_p | \Phi \rangle \\ &\stackrel{\text{WT}}{=} \rho_{sp} \rho_{tq} \rho_{ur} - \rho_{sq} \rho_{tp} \rho_{ur} + \rho_{sq} \rho_{tr} \rho_{up} \\ &\quad - \rho_{sr} \rho_{tq} \rho_{up} - \rho_{sp} \rho_{tr} \rho_{uq} + \rho_{sr} \rho_{tp} \rho_{uq}. \end{aligned} \quad (2.32)$$

2.6 Correlated Reference States and Irreducible Density Matrices

In Chapter 2.5, we discussed the one-body and two-body density matrices for a single, *uncorrelated* Slater determinant reference state. For a *correlated* reference state, the density matrices include irreducible terms which encode “pure” correlations in the reference state. In order to define contractions with respect to a general reference state, Wick's theorem must be augmented with additional contractions that reflect the present correlations, using a formalism developed by Kutzelnigg and Mukherjee [4] in the context of quantum chemistry.

The new contractions are given by the so-called irreducible density matrices of the reference state, for reasons that will become clear soon. Following the notation of Hergert [1], we will denote these irreducible density matrices with the symbol λ . In the correlated reference state, the one-body density matrix itself is irreducible:

$$\langle \Phi | a_p^\dagger a_q | \Phi \rangle = \lambda_{qp} = \rho_{qp}. \quad (2.33)$$

The irreducible two-body density matrix is defined as:

$$\lambda_{rspq} = \rho_{rspq} - (\lambda_{rp} \lambda_{sq} - \lambda_{rq} \lambda_{sr}). \quad (2.34)$$

Comparison with Equation 2.34 shows that for an uncorrelated reference state, such as a single Slater determinant, the irreducible two-body density matrix vanishes. In other words, λ_{ijkl} represents pure two-body correlation information, after contributions from independent particles have been removed from the full two-body density matrix [1].

The irreducible three-body density matrix is similarly defined,

$$\lambda_{stupqr} = \rho_{stupqr} - \mathcal{A}(\lambda_{sp}\lambda_{tuqr}) - \mathcal{A}(\lambda_{sp}\lambda_{tq}\lambda_{ur}) \quad (2.35)$$

where $\mathcal{A}(\cdot)$ represents all antisymmetric combinations of its argument (refer to [4, 5] for their complete derivation). In this case, the three-body irreducible term is the pure three-body correlation, after contributions from independent particles, as well as a correlated pair in presence of an independent spectator, have been subtracted from the full three-body density matrix.

2.7 Normal-ordering the Two-body Hamiltonian

Consider a many-body vacuum Hamiltonian which consists of a one-body operator term, corresponding to single-particle kinetic energy, and a two-body operator term,

$$H = \sum_{pq} \langle p|t|q \rangle a_p^\dagger a_q + \frac{1}{4} \sum_{pqrs} \langle pq|v|rs \rangle a_p^\dagger a_q^\dagger a_s a_r, \quad (2.36)$$

where the vacuum two-body coefficients $\langle pq|v|rs \rangle$ are antisymmetrized. Wick's theorem can be applied to H to extract the normal-ordered H_N . We work in the eigenbasis of the one-body operator and begin by applying Wick's theorem to the one-body operator:

$$a_p^\dagger a_q \stackrel{\text{WT}}{=} N [a_p^\dagger a_q] + N [\overline{a_p^\dagger a_q}] = N [a_p^\dagger a_q] + n(p)\delta_{pq}. \quad (2.37)$$

The Wick's theorem two-body operator is written,

$$\begin{aligned} a_p^\dagger a_q^\dagger a_s a_r \stackrel{\text{WT}}{=} & N [a_p^\dagger a_q^\dagger a_s a_r] + \\ & N [\overline{a_p^\dagger a_q^\dagger a_s a_r}] + N [\overline{a_p^\dagger a_q^\dagger a_s} a_r] + \\ & N [a_p^\dagger \overline{a_q^\dagger a_s a_r}] + N [a_p^\dagger \overline{a_q^\dagger a_s} a_r] + \\ & N [\overline{a_p^\dagger a_q^\dagger a_s} a_r] + N [\overline{a_p^\dagger a_q^\dagger} a_s a_r], \end{aligned} \quad (2.38)$$

and evaluating the Wick contractions reveals the two-body dependence on the occupation structure in the reference state,

$$\begin{aligned}
a_p^\dagger a_q^\dagger a_s a_r &\stackrel{\text{WT}}{=} N [a_p^\dagger a_q^\dagger a_s a_r] \\
&\quad - n(p) \delta_{ps} N [a_q^\dagger a_r] + n(p) \delta_{pr} N [a_q^\dagger a_s] \\
&\quad + n(q) \delta_{qs} N [a_p^\dagger a_r] - n(q) \delta_{qr} N [a_p^\dagger a_s] \\
&\quad + n(p)n(q) \delta_{pr} \delta_{qs} - n(p)n(q) \delta_{ps} \delta_{qr}.
\end{aligned} \tag{2.39}$$

Note that the occupation factors $n(p)$ and $n(q)$ restrict the single particle indices to *only occupied states*, or hole indices. Consequently, the summation indices in Equation 2.36 are also restricted where relevant.

We proceed by collecting zero-, one-, and two-body terms into 0B, 1B, and 2B normal-ordered operators which appear on the right-hand side of Wick's theorem applied to the vacuum Hamiltonian in Equation 2.36. Beginning with the 0B normal-ordered operator (fully contracted),

$$0\text{B} = \sum_i \langle i|t|i \rangle + \frac{1}{2} \sum_{ij} \langle ij|v|ij \rangle \equiv E, \tag{2.40}$$

where the sum runs over the occupied (hole) states. This term is referred to as the *reference energy* E .

The 1B normal-ordered operator (one contraction) reads

$$\begin{aligned}
1\text{B} &= \sum_{pq} \langle p|t|q \rangle N [a_p^\dagger a_q] \\
&\quad + \frac{1}{4} \sum_{ipq} N [a_p^\dagger a_q] (\langle ip|v|i q \rangle + \langle pi|v|qi \rangle \\
&\quad \quad - \langle ip|v|qi \rangle - \langle pi|v|i q \rangle) \\
&= \sum_{pq} \langle p|t|q \rangle N [a_p^\dagger a_q] + \sum_{ipq} N [a_p^\dagger a_q] \langle ip|v|i q \rangle \\
&\equiv \sum_{pq} \langle p|f|q \rangle N [a_p^\dagger a_q],
\end{aligned}$$

where the Fock operator f contains the kinetic energy of the particle plus a mean-field contribution from the interaction.

The 2B normal-ordered operator is simply the last remaining two-body term which contains no contractions,

$$2B = \frac{1}{4} \sum_{pqrs} \langle pq|v|rs \rangle N [a_p^\dagger a_q^\dagger a_s a_r]. \quad (2.41)$$

Finally, the normal-ordered two-body Hamiltonian reads

$$H_N = E + \sum_{pq} \langle p|f|q \rangle N [a_p^\dagger a_q] + \frac{1}{4} \sum_{pqrs} \langle pq|v|rs \rangle N [a_p^\dagger a_q^\dagger a_s a_r] \quad (2.42)$$

Wick's theorem shuffles operator contributions into terms of fewer-body character by contracting with the density matrix of the independent particle system.

There are situations where we may want to reverse the normal-ordering process in order to recover the true vacuum operators. To see how we can undo the normal-ordering of the Hamiltonian (or any other operator), we compare coefficients in Equation 2.42 (the normal-ordered Hamiltonian) and Equation 2.36 (the two-body vacuum Hamiltonian). Comparing coefficients where ^{vac} corresponds to vacuum operators and ^{NO} corresponds to normal-ordered operators,

$$\begin{aligned} \langle pq|v|rs \rangle^{\text{vac}} &= \langle pq|v|rs \rangle^{\text{NO}} \\ \langle p|t|q \rangle^{\text{vac}} &= \langle p|f|q \rangle^{\text{NO}} - \sum_i \langle pi|v|qi \rangle^{\text{vac}} \\ H_0 &= E - \sum_i \langle i|t|i \rangle^{\text{vac}} - \frac{1}{4} \sum_{ij} \langle ij|v|ij \rangle^{\text{vac}}, \end{aligned} \quad (2.43)$$

where the H_0 is necessary to recover information lost due to truncation of higher-body operators generated by the normal-ordering process. In the case where no operators were truncated in the normal-ordered operator, H_0 vanishes by Equation 2.40. If normal-ordered contributions which would have been shuffled into the zero-body term in Equation 2.40 are instead truncated off, then the unnormal-ordered H_0 piece will not vanish, and must be included in the total vacuum Hamiltonian, written

$$H^{\text{vac}} = H_0 + H_{1B}^{\text{vac}} + H_{2B}^{\text{vac}}. \quad (2.44)$$

This procedure is useful in the context of IMSRG-evolved Hamiltonians (Chapter 3.3), where we may wish to use vacuum coefficients obtained from the unnormal-ordered evolved Hamiltonian as inputs for exact methods (e.g. Full Configuration Interaction calculations in Chapter 2.10).

2.8 Density Matrix Normal-ordering

When generally correlated reference states are introduced to the normal-ordering, we must modify the normal-ordering procedure to account for occupations which are fractional; in a correlated reference, the occupation factor in Equations 2.27 and 2.28 is no longer an integer, in general.

Normal-ordering with respect to a general density matrix is as simple as inserting the full density matrices, where appropriate, into the calculation of the normal-ordered operator in Equations 2.40, 2.41, and 2.41. Summations over hole indices must then be replaced by sums over the full single-particle basis instead, e.g.

$$\begin{aligned}
\sum_i \langle pi|v|qi \rangle^{\text{vac}} &\longrightarrow \sum_{rs} \langle pr|v|qs \rangle^{\text{vac}} \rho_{sr} \\
\sum_i \langle i|t|i \rangle^{\text{vac}} &\longrightarrow \sum_{rs} \langle p|t|q \rangle^{\text{vac}} \rho_{qp} \\
\frac{1}{4} \sum_{ij} \langle ij|v|ij \rangle^{\text{vac}} &\longrightarrow \frac{1}{4} \sum_{pqrs} \langle pq|v|rs \rangle^{\text{vac}} \rho_{rspq}.
\end{aligned} \tag{2.45}$$

2.9 Slater Determinant Occupation Number Representation

A common strategy for writing and computing Slater determinants, particularly in code, is the occupation number representation. In this scheme, a Slater determinant with number of single states N is written as a string of 1s and 0s of length N , where 1 indicates an occupied single particle state and 0 an unoccupied single particle state. For example, a Fermi vacuum Slater determinant for $N = 8$ states and $A = 4$ particles states can be written as

$$|\Phi_0\rangle = |11110000\rangle, \tag{2.46}$$

where the ordering of the bit digits is convention, but must be consistent within a given application.

In this scheme, the application of creation/annihilation operators replace 0s with 1s and 1s with 0s where appropriate, and the usual rules for antisymmetric exchanges of indices still apply. For example, we can apply an excitation operator $\hat{C}_4^5 = a_5^\dagger a_4$ which excites the particle occupied in single particle index $i = 4$ into the single particle index $a = 5$ above the Fermi surface:

$$\hat{C}_4^5 |11110000\rangle = - |11101000\rangle. \tag{2.47}$$

The Slater determinants in this scheme are still orthonormal, meaning that

$$\begin{aligned}\langle 11110000 | 11110000 \rangle &= 1 \\ \langle 11110000 | \hat{C}_4^5 | 11110000 \rangle &= -\langle 11110000 | 11101000 \rangle = 0.\end{aligned}\tag{2.48}$$

2.10 Full Configuration Interaction Theory to Solve Model Hamiltonians

Sometimes we require an exact method in order to validate results from approximate models. The exact method we use often in this dissertation is called Full Configuration Interaction (FCI) theory. The defining feature of an FCI calculation is that the Hamiltonian is constructed in the *full* basis of Slater determinant *configurations* of particles [3]. In the occupation number representation, we proceed by constructing the Hamiltonian matrix in the full basis of configurations of occupation strings relative to a “ground state” configuration such as the occupation string in Equation 2.46. After the matrix is built, we can apply a diagonalization method to extract the energy spectrum, or other target observables.

This method is prohibitive for practical problems in nuclear theory due to the dimensionality of the model space. For a model with A particles in N possible single particle states, the total number of Slater determinant configurations is given by [3]

$$\binom{N}{A} = \frac{N!}{(N-A)!A!},\tag{2.49}$$

This number quickly explodes with the number of possible single particle states, which may be on the order of 100s or 1000s for typical medium and heavy nuclei.

For a toy model, such as the pairing model (see Chapter 4.1), the model space may be so simplified that the dimensionality is not an issue. In the pairing model, we can typically keep our particle number small (on the order of 10 or fewer), and fill half the number of energy levels, in order to extract the physics we require using our theoretical methods.

An FCI calculation involving a basis of Slater determinant configurations is performed over several steps:

1. Define the number of single particle states n and number of particles N

2. Generate $\binom{n}{N}$ number of Slater determinants in the occupation number representation as the basis (see Chapter 2.9)
3. Compute $\binom{n}{N} \times \binom{n}{N}$ square matrix $\mathbf{H}$ with elements given by $\langle \Phi_i | H | \Phi_j \rangle$, where Φ_i is the i -th Slater determinant in the basis (typically ordered by increasing energy with respect to the ground state, although this ordering may not be known *a priori*)
4. Diagonalize $\mathbf{H}$ to extract the energy spectrum

Since we have access to the eigenstates after the diagonalization, we can in principle compute any other observable of interest in this framework. When calculations are feasible, the energy spectrum extracted from FCI is the ultimate benchmark for validating results from other techniques, which will be presented in later sections.

BIBLIOGRAPHY

- [1] H Hergert. “In-medium similarity renormalization group for closed and open-shell nuclei”. In: *Physica Scripta* 92.2 (Dec. 2016), p. 023002. ISSN: 1402-4896. DOI: 10.1088/1402-4896/92/2/023002. URL: <http://dx.doi.org/10.1088/1402-4896/92/2/023002>.
- [2] H. Hergert et al. “The In-Medium Similarity Renormalization Group: A novel ab initio method for nuclei”. In: *Physics Reports* 621 (Mar. 2016), pp. 165–222. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2015.12.007. URL: <http://dx.doi.org/10.1016/j.physrep.2015.12.007>.
- [3] Morten Hjorth-Jensen, Maria-Paola Lombardo, and Ubirajara van Kolck. *An advanced course in computational nuclear physics: bridging the scales from quarks to neutron stars*. Springer, 2017.
- [4] Werner Kutzelnigg and Debashis Mukherjee. “Normal order and extended Wick theorem for a multiconfiguration reference wave function”. In: *The Journal of Chemical Physics* 107.2 (1997), pp. 432–449. DOI: 10.1063/1.474405. eprint: <https://doi.org/10.1063/1.474405>. URL: <https://doi.org/10.1063/1.474405>.
- [5] Debashis Mukherjee and Werner Kutzelnigg. “Irreducible Brillouin conditions and contracted Schrödinger equations for n-electron systems. I. The equations satisfied by the density cumulants”. In: *The Journal of Chemical Physics* 114.5 (2001), pp. 2047–2061. DOI: 10.1063/1.1337058. eprint: <https://doi.org/10.1063/1.1337058>. URL: <https://doi.org/10.1063/1.1337058>.
- [6] I. Shavitt and R.J. Bartlett. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge Molecular Science. Cambridge University Press, 2009. ISBN: 9780521818322. URL: <https://books.google.com/books?id=SWw6ac1NHZYC>.

CHAPTER 3

IN-MEDIUM SIMILARITY RENORMALIZATION GROUP

3.1 Introduction

In this chapter, we will present the conceptual and mathematical framework for the in-medium similarity renormalization group (IMSRG), which is the main focus for this body of work. We begin by presenting the idea behind Similarity Renormalization Group methods.

3.2 Similarity Renormalization Group

The SRG was conceived by Glazek and Wilson [2] for applications in light-front quantum field theory, and independently developed by Wegner [8] for applications in condensed matter systems. The SRG is constructed such that the Hamiltonian matrix, expressed in an appropriate basis, is driven to a band-diagonal form via a continuous unitary transformation that suppresses off-diagonal elements. In a basis like the harmonic oscillator, these off-diagonal elements encode interactions between states of increasing difference in energy. This process is demonstrated schematically in Figure 3.1. The suppression scales s_0, s_1, s_2 represent characteristic relative momentum “bands” which limit relative momentum according to $s = \lambda^{1/4}$, where λ is in units fm^{-1} . The scale parameter s parameterizes the cutoff Λ in the χEFT theory (see Chapter 4.2). As s increases, high momentum modes are decoupled from low momentum modes.

The flowing SRG Hamiltonian is expressed as

$$H(s) = U(s)H(s=0)U^\dagger(s), \quad (3.1)$$

where $U(s)$ describes the evolution of the initial, unevolved Hamiltonian to the current value of the continuous flow parameter s .

$$\begin{aligned} \frac{dH(s)}{ds} &= \frac{dU(s)}{ds}H(0)U^\dagger(s) + U(s)H(0)\frac{dU^\dagger(s)}{ds} \\ &= \frac{dU(s)}{ds}U^\dagger(s)H(s) + H(s)U(s)\frac{dU^\dagger(s)}{ds}. \end{aligned} \quad (3.2)$$

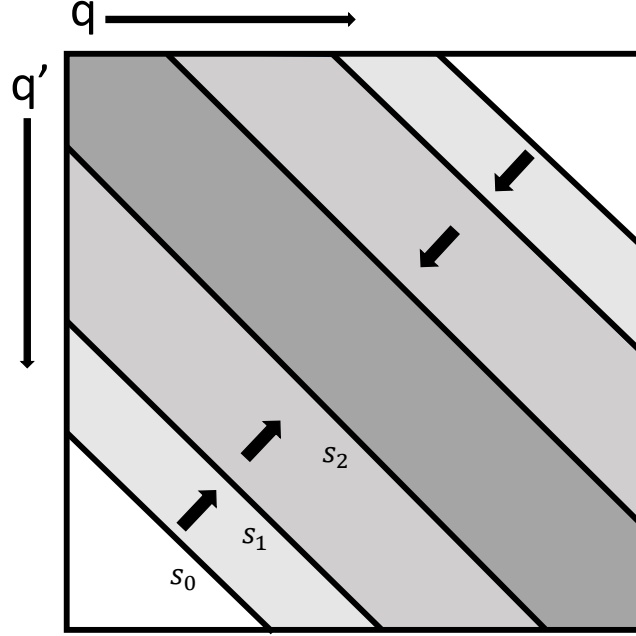


Figure 3.1 Schematic diagram of NN scattering potential, in a momentum basis, showing how modes of increasing relative momentum (energy) are suppressed faster than low momentum modes. Figure adapted from Bogner, Furnstahl, and Schwenk [1].

Next, we define the s -dependent anti-Hermitian generator $\eta(s)$, which is *chosen* in a way to “generate” the transformation we want to achieve (i.e. suppression of off-diagonal elements according to chosen decoupling strategy). The generator $\eta(s)$ is defined as [4, 3, 5, 1]

$$\eta(s) = \frac{dU(s)}{ds} U^\dagger(s) = -\eta^\dagger(s). \quad (3.3)$$

Inserting Equation 3.3 into Equation 3.2, we naturally arrive at the commutator expression for the operator flow equation:

$$\frac{dH(s)}{ds} = [\eta(s), H(s)]. \quad (3.4)$$

3.3 IMSRG Formalism

The computational cost of the SRG, with matrices, in Equation 3.4 scales *exponentially* with the many-body basis dimension. The IMSRG approach moves this description “in-medium” by shifting the flowing Hamiltonian to a normal-ordered operator basis (Chapter 2), instead of the many-body basis [4]. The flow is implemented at the level of operators, using strings of annihilation and

creation operators that are normal-ordered with respect to an appropriate reference state, defining our operator basis. We will see that the IMSRG flow equations are *polynomially* scaling with single particle basis size. Practically, the purpose of the IMSRG is to decouple a target state from excitations relative to that state, by suppressing the interaction terms in the normal-ordered operator expansion of the Hamiltonian.

We start from Equation 3.4, and use the second quantization techniques discussed in Chapter 2.4 to derive the coupled flow equations associated with the IMSRG evolution. We assume that the operator $H(s)$ is in normal-ordered form with respect to a chosen reference state (which we might choose to approximate the Hamiltonian ground state); the normal-ordered *two-body* Hamiltonian is written,

$$H_{\text{IMSRG}}(s) = E(s) + \sum_{pq} f_{pq}(s) N [a_p^\dagger a_q] + \frac{1}{4} \sum_{pqrs} \Gamma_{pqrs}(s) N [a_p^\dagger a_q^\dagger a_s a_r], \quad (3.5)$$

where the f_{pq} and Γ_{pqrs} operator coefficients are derived from the vacuum representation of the Hamiltonian through the normal ordering procedure, and E represents the reference energy, which encodes the expectation value of the vacuum Hamiltonian in the reference state (see Chapter 2.7). We assume that Γ_{pqrs} is antisymmetrized, i.e.,

$$\Gamma_{pqrs} = -\Gamma_{qprs} = -\Gamma_{pqsr} = \Gamma_{qpsr}. \quad (3.6)$$

Note that all operator coefficients in the IMSRG Hamiltonian (3.5) depend on the flow parameters s . In this way, the IMSRG evolves the Hamiltonian within the chosen operator basis toward a desired form which is defined by a chosen decoupling scheme. Evaluation of the commutators in Equation 3.4 induces higher-body terms that are computationally expensive to calculate. In general, we need to truncate the operator basis at a particular level. For example, keeping up to two-body operators throughout the flow defines the IMSRG(2) truncation.

In the IMSRG(2) truncation, the anti-Hermitian generator $\eta(s)$ has the form,

$$\eta(s) = \sum_{pq} \eta_{pq}(s) N [a_p^\dagger a_q] + \frac{1}{4} \sum_{pqrs} \eta_{pqrs}(s) N [a_p^\dagger a_q^\dagger a_s a_r]. \quad (3.7)$$

To proceed, we recall the general form of a product of operators from Equation 2.30. Each product in the commutator of Equation 3.4 generates new operators up to 4-body; however, the symmetry in the commutator eliminates the four-body operators. Again, working in the IMSRG(2) truncation means we keep only up to two-body operators generated by the commutators.

As a demonstration, we will derive the term associated with $\left[N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} \right], N \left[a_{p_2}^\dagger a_{q_2} \right] \right]$, where we have added the sub-indices ₁ and ₂ to differentiate between the source of the operator (Hamiltonian versus generator). The logic we use here can be extended to each additional commutator term to completely derive the full IMSRG flow equations. We start with the definition of the commutator,

$$\begin{aligned} \left[N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} \right], N \left[a_{p_2}^\dagger a_{q_2} \right] \right] = & N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} \right] N \left[a_{p_2}^\dagger a_{q_2} \right] \\ & - N \left[a_{p_2}^\dagger a_{q_2} \right] N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} \right] \end{aligned} \quad (3.8)$$

We may apply Wick's theorem (refer Shavitt and Bartlett [7], Equation 3.194, as well as Chapter 2.5) to a product of normal-ordered operators, which operates similarly to application of Wick's theorem to a vacuum operator. Using the first term as an example (dropping the summations $\sum$ for brevity),

$$\begin{aligned} N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} \right] N \left[a_{p_2}^\dagger a_{q_2} \right] & \stackrel{\text{WT}}{=} N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger a_{q_2} \right] \\ & + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] \\ & + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] \\ & + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] \\ & + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] + N \left[\overbrace{a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger}^{\text{contract}} a_{q_2} \right] \end{aligned} \quad (3.9)$$

Contractions between the operators inside a particular normal-ordered operator string on the left-hand side of Equation 3.9 are already accounted for by the initial normal ordering, hence only contractions *between* operators from string 1 and 2 survive.

Simplifying Equation 3.9 further using the Wick contractions in Equations 2.22 and 2.25, we

find

$$\begin{aligned}
N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1}] N[a_{p_2}^\dagger a_{q_2}] &= N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1} a_{p_2}^\dagger a_{q_2}] \\
&+ \rho_{p_1 q_2} N[a_{q_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1}] - \rho_{q_1 q_2} N[a_{p_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1}] \\
&- (\delta_{s_1 p_2} - \rho_{s_1 p_2}) N[a_{p_1}^\dagger a_{q_1}^\dagger a_{r_1} a_{q_2}] + (\delta_{r_1 p_2} - \rho_{r_1 p_2}) N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{q_2}] \quad (3.10) \\
&- \rho_{p_1 q_2} (\delta_{s_1 p_2} - \rho_{s_1 p_2}) N[a_{q_1}^\dagger a_{r_1}] + \rho_{p_1 q_2} (\delta_{r_1 p_2} - \rho_{r_1 p_2}) N[a_{q_1}^\dagger a_{s_1}] \\
&+ \rho_{q_1 q_2} (\delta_{s_1 p_2} - \rho_{s_1 p_2}) N[a_{p_1}^\dagger a_{r_1}] - \rho_{q_1 q_2} (\delta_{r_1 p_2} - \rho_{r_1 p_2}) N[a_{p_1}^\dagger a_{s_1}].
\end{aligned}$$

The same procedure applied to the product $N[a_{p_2}^\dagger a_{q_2}] N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1}]$ yields a similar result,

$$\begin{aligned}
N[a_{p_2}^\dagger a_{q_2}] N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1}] &= N[a_{p_2}^\dagger a_{q_2} a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1}] \\
&- \rho_{p_2 r_1} N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{q_2}] + \rho_{p_2 s_1} N[a_{p_1}^\dagger a_{q_1}^\dagger a_{r_1} a_{q_2}] \\
&- (\delta_{q_2 p_1} - \rho_{q_2 p_1}) N[a_{q_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1}] + (\delta_{q_2 q_1} - \rho_{q_2 q_1}) N[a_{p_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1}] \quad (3.11) \\
&- \rho_{p_2 s_1} (\delta_{q_2 p_1} - \rho_{q_2 p_1}) N[a_{q_1}^\dagger a_{r_1}] + \rho_{p_2 r_1} (\delta_{q_2 p_1} - \rho_{q_2 p_1}) N[a_{q_1}^\dagger a_{s_1}] \\
&+ \rho_{p_2 s_1} (\delta_{q_2 q_1} - \rho_{q_2 q_1}) N[a_{p_1}^\dagger a_{r_1}] - \rho_{p_2 r_1} (\delta_{q_2 q_1} - \rho_{q_2 q_1}) N[a_{p_1}^\dagger a_{s_1}],
\end{aligned}$$

where we have exchanged operators inside the normal-product to match Equation 3.10. At this point, the terms which vanish from the commutator are straightforward to recognize—the three-body normal-product vanishes, as well as any term which only includes the one-body density matrix ρ . The commutator becomes

$$\begin{aligned}
[N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{r_1}], N[a_{p_2}^\dagger a_{q_2}]] &= \\
&- \delta_{s_1 p_2} N[a_{p_1}^\dagger a_{q_1}^\dagger a_{r_1} a_{q_2}] + \delta_{r_1 q_2} N[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{q_2}] \\
&+ \delta_{q_2 p_1} N[a_{q_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1}] - \delta_{q_2 q_1} N[a_{p_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1}] \\
&+ (-\rho_{p_1 q_2} \delta_{s_1 p_2} + \rho_{p_2 s_1} \delta_{q_2 p_1}) N[a_{q_1}^\dagger a_{r_1}] \quad (3.12) \\
&+ (\rho_{p_1 q_2} \delta_{r_1 p_2} - \rho_{p_2 r_1} \delta_{q_2 p_1}) N[a_{q_1}^\dagger a_{s_1}] \\
&+ (\rho_{q_1 q_2} \delta_{s_1 p_2} - \rho_{p_2 s_1} \delta_{q_2 q_1}) N[a_{p_1}^\dagger a_{r_1}] \\
&+ (-\rho_{q_1 q_2} \delta_{r_1 p_2} + \rho_{p_2 r_1} \delta_{q_2 q_1}) N[a_{p_1}^\dagger a_{s_1}].
\end{aligned}$$

Now we will re-establish the summation $\sum$ over all single particle states for each operator in the commutator. Since this sum covers all single particle indices for every operator in each normal-

product, we can rename and manipulate indices to our advantage. Inside the normal-product, the label of a particular operator is “free” because the indices span the full space. Introducing the appropriate matrix elements associated with each operator reveals that we can combine each term into an associated many-body operator, multiplied by a corresponding matrix element (or expression). Applying this logic,

$$\begin{aligned}
& \left[A^{(2)}, B^{(1)} \right] = \\
& \frac{1}{4} \sum_{p_1 q_1 r_1 s_1 p_2 q_2} A_{p_1 q_1 r_1 s_1} B_{p_2 q_2} \left(-\delta_{s_1 p_2} N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{r_1} a_{q_2} \right] + \delta_{r_1 p_2} N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{q_2} \right] \right. \\
& \quad + \delta_{q_2 p_1} N \left[a_{q_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1} \right] - \delta_{q_2 q_1} N \left[a_{p_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1} \right] \\
& \quad + \left(-\rho_{p_1 q_2} \delta_{s_1 p_2} + \rho_{p_2 s_1} \delta_{q_2 p_1} \right) N \left[a_{q_1}^\dagger a_{r_1} \right] \\
& \quad + \left(\rho_{p_1 q_2} \delta_{r_1 q_2} - \rho_{p_2 r_1} \delta_{q_2 p_1} \right) N \left[a_{q_1}^\dagger a_{s_1} \right] \\
& \quad + \left(\rho_{q_1 q_2} \delta_{s_1 p_2} - \rho_{p_2 s_1} \delta_{q_2 q_1} \right) N \left[a_{p_1}^\dagger a_{r_1} \right] \\
& \quad \left. + \left(-\rho_{q_1 q_2} \delta_{r_1 p_2} + \rho_{p_2 r_1} \delta_{q_2 q_1} \right) N \left[a_{p_1}^\dagger a_{s_1} \right] \right).
\end{aligned} \tag{3.13}$$

For simplicity, we will continue in the eigenbasis of the one-body density matrix such that,

$$\rho_{pq} = n_p \delta_{pq}, \tag{3.14}$$

where the factor n_p is 0 or 1 when the single particle state a is unoccupied or occupied, respectively, in the reference [4]. Note that working in this basis, we assume that the reference state is a single Slater determinant.

We push the associated matrix element $A_{p_1 q_1 r_1 s_1} B_{p_2 q_2}$ through each term in the sum, carefully

removing index dependence that is cancelled by a particular δ ,

$$\begin{aligned}
\left[A^{(2)}, B^{(1)} \right] = & -\frac{1}{4} \sum_{p_1 q_1 r_1 u q_2} A_{p_1 q_1 r_1 u} B_{u q_2} N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{r_1} a_{q_2} \right] + \frac{1}{4} \sum_{p_1 q_1 u s_1 q_2} A_{p_1 q_1 u s_1} B_{u q_2} N \left[a_{p_1}^\dagger a_{q_1}^\dagger a_{s_1} a_{q_2} \right] \\
& + \frac{1}{4} \sum_{q_1 r_1 s_1 p_2 u} A_{u q_1 r_1 s_1} B_{p_2 u} N \left[a_{q_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1} \right] - \frac{1}{4} \sum_{p_1 r_1 s_1 p_2 u} A_{p_1 u r_1 s_1} B_{p_2 u} N \left[a_{p_1}^\dagger a_{p_2}^\dagger a_{s_1} a_{r_1} \right] \\
& + \frac{1}{4} \sum_{q_1 r_1 u v} A_{v q_1 r_1 u} B_{u v} (-n_v + n_u) N \left[a_{q_1}^\dagger a_{r_1} \right] \\
& + \frac{1}{4} \sum_{q_1 s_1 u v} A_{v q_1 u s_1} B_{u v} (n_v - n_u) N \left[a_{q_1}^\dagger a_{s_1} \right] \\
& + \frac{1}{4} \sum_{p_1 r_1 u v} A_{p_1 v r_1 u} B_{u v} (n_v - n_u) N \left[a_{p_1}^\dagger a_{r_1} \right] \\
& + \frac{1}{4} \sum_{p_1 s_1 u v} A_{p_1 u v s_1} B_{u v} (-n_v + n_u) N \left[a_{p_1}^\dagger a_{s_1} \right], \tag{3.15}
\end{aligned}$$

where the new summation indices u, v result from the associated δ restrictions. The four one-body operator terms in Equation 3.15 are equivalent up to a sign, and the associated indices may be permuted in a way to combine all four terms into a single one-body operator term that reads

$$\left[A^{(2)}, B^{(1)} \right]^{(1)} = \sum_{p q u v} A_{v p u q} B_{u v} (n_v - n_b) N \left[a_p^\dagger a_q \right]. \tag{3.16}$$

The four two-body terms can be rearranged into a single two-body operator with a coefficient that consists of a sum over permuted matrix elements, i.e.

$$\begin{aligned}
\left[A^{(2)}, B^{(1)} \right]^{(2)} &= \frac{1}{4} \sum_{p q r s u} N \left[a_p^\dagger a_q^\dagger a_s a_r \right] \times \\
&\quad (A_{p q r u} B_{u s} + A_{p q u s} B_{u r} \\
&\quad - A_{u q r s} B_{p u} - A_{p u r s} B_{q u}) \\
&= \frac{1}{4} \sum_{p q r s u} N \left[a_p^\dagger a_q^\dagger a_s a_r \right] \times \\
&\quad \{ (1 - P_{rs}) A_{p q u s} B_{u r} - (1 - P_{pq}) A_{u q r s} B_{p u} \} \tag{3.17}
\end{aligned}$$

Here, the operator P_{pq} exchanges the indices p, q . We have arrived at the final result for just one commutator that appears in the IMSRG(2) flow equations, with seemingly Herculean effort.

Computing additional commutators requires similar, or greater, effort. Luckily, the fundamental commutators which can be used to compute the remaining terms in the flow equation commutator are catalogued in Appendix A of Hergert et al. [4], and tools are now available that can automate these derivations. The zero-, one-, and two-body terms that are produced by the commutators can now be collected into separate, but coupled, flow equations for the reference energy E , the one-body operator coefficients f , and the two-body operator coefficients Γ . The resulting system of IMSRG(2) flow equations reads[4]

$$\begin{aligned}
\frac{dE}{ds} &= \sum_{pq} (n_p - n_q) \eta_{pq} f_{qp} \\
&\quad + \frac{1}{2} \sum_{pqrs} \eta_{pqrs} \Gamma_{rs pq} n_p n_q \bar{n}_r \bar{n}_s \\
\frac{df_{pq}}{ds} &= \sum_r (1 + P_{pq}) \eta_{pr} f_{rq} \\
&\quad + \sum_{rs} (n_r - n_s) (\eta_{rs} \Gamma_{sprq} - f_{rs} \eta_{sprq}) \\
&\quad + \frac{1}{2} \sum_{rst} (n_r n_s \bar{n}_t + \bar{n}_r \bar{n}_s n_t) (1 + P_{pq}) \eta_{tprs} \Gamma_{rstq} \\
\frac{d\Gamma_{pqrs}}{ds} &= \sum_t \{ (1 - P_{pq}) (\eta_{pt} \Gamma_{tqrs} - f_{pt} \eta_{tqrs}) \\
&\quad - (1 - P_{rs}) (\Gamma_{pqts} \eta_{tr} - \eta_{pqts} f_{tr}) \} \\
&\quad + \frac{1}{2} \sum_{tu} (1 - n_t - n_u) (\eta_{pqtu} \Gamma_{turs} - \Gamma_{pqtu} \eta_{turs}) \\
&\quad - \sum_{tu} (n_t - n_u) (1 - P_{pq}) (1 - P_{rs}) \eta_{uqts} \Gamma_{tpur},
\end{aligned} \tag{3.18}$$

where indices run over the full single-particle basis. Note that the occupation numbers n_p do not depend on s because the reference state is held fixed during the evolution.

Since we must loop over all operator coefficients for each flow, the computational effort for implementing the IMSRG(2) evolution naively scales like $O(N^6)$ where N is the number of single particle states. For Slater determinant references and certain types of IMSRG generators, we can leverage the partitioning of the single-particle basis into particle and hole states to achieve an $O(N_h^2 N_p^4)$ scaling akin to the Coupled Cluster method with Singles and Doubles excitations

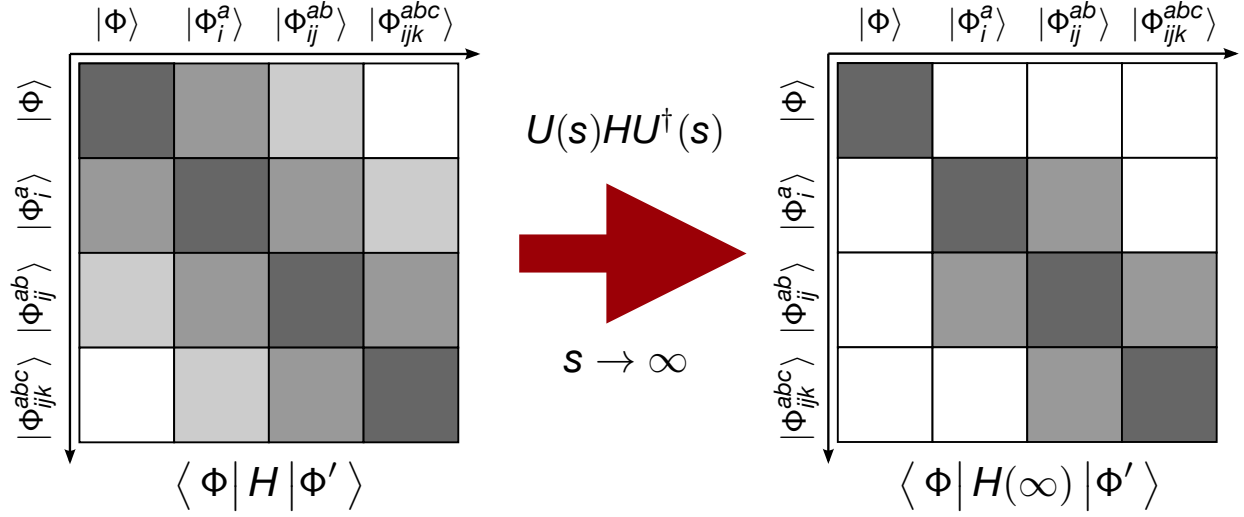


Figure 3.2 Schematic illustration of the evolving structure of the IMSRG(2) Hamiltonian. The reference state $|\Phi\rangle$ is completely decoupled from 1p-1h and 2p-2h excitations, $|\Phi_h^p\rangle$ and $|\Phi_{hh'}^{pp'}\rangle$ as a result of the flow. Figure adapted from Hergert et al. [4].

(CCSD), which aims to describe a comparable amount of many-body correlations [4].

3.4 Generators for the IMSRG

The IMSRG *generator* η is an anti-Hermitian operator that we tailor in to implement a particular decoupling scheme and implicitly drive the Hamiltonian matrix toward its desired form. To this end, we need to identify the “off-diagonal” parts of the Hamiltonian that we would like to suppress through the flow.

3.4.1 Choice of Decoupling Scheme

Here, we adopt the *minimal decoupling* scheme, in which we want to decouple a single eigenstate — usually the ground state — from the rest of the many-body Hamiltonian matrix [4]. Figure 3.2 schematically displays the structure of initial and evolved Hamiltonian in the Ap-Ah basis of excitations relative to a reference state $|\Phi\rangle$. In this scheme, the off-diagonal matrix elements of the evolving Hamiltonian that are targeted for suppression read [4]

$$\begin{aligned} \langle \Phi | H(s) | \Phi_i^a \rangle &= f_{ai}(s) \\ \langle \Phi | H(s) | \Phi_{ij}^{ab} \rangle &= \Gamma_{abij}(s), \end{aligned} \tag{3.19}$$

and the normal-ordered off-diagonal Hamiltonian is constructed from these coefficients in the usual way [4],

$$H^{\text{od}}(s) = \sum_{ai} f_{ai}(s) N[a_a^\dagger a_i] + \frac{1}{4} \sum_{abij} \Gamma_{abij}(s) N[a_a^\dagger a_b^\dagger a_j a_i] + H.c.. \quad (3.20)$$

This definition must be generalized slightly in if we work with a general correlated reference state in the *multi-reference* IMSRG (MR-IMSRG, see Appendix B), or with a *reference state ensemble* (Chapter 6).

3.4.2 White Generator

Using the decoupling scheme and definition of the off-diagonal Hamiltonian from the previous section, we can construct a class of generators that are inspired by the work of Steven White [9]. They are defined as [4],

$$\eta_W(s) = \sum_{ai} \frac{f_{ai}(s)}{\Delta_{ai}(s)} N[a_a^\dagger a_i] + \frac{1}{4} \sum_{abij} \frac{\Gamma_{abij}(s)}{\Delta_{abij}(s)} N[a_a^\dagger a_b^\dagger a_j a_i] - H.c.. \quad (3.21)$$

The anti-Hermiticity enters from the energy denominators Δ , which are inspired by the Epstein-Nesbet (EN) and Møller-Plesset (MP) versions of many-body perturbation theory. The EN denominators read

$$\begin{aligned} \Delta_{ai}^{(\text{EN})} &= f_a - f_i - \Gamma_{aiai} \\ \Delta_{ia}^{(\text{EN})} &= f_i - f_a + \Gamma_{aiai} \\ \Delta_{abij}^{(\text{EN})} &= f_a + f_b - f_i - f_j \\ &\quad + \Gamma_{abab} + \Gamma_{ijij} - \Gamma_{aiai} - \Gamma_{bjbj} - \Gamma_{ajaj} - \Gamma_{bibi} \\ \Delta_{ijab}^{(\text{EN})} &= f_i + f_j - f_a - f_b \\ &\quad - \Gamma_{abab} - \Gamma_{ijij} + \Gamma_{aiai} + \Gamma_{bjbj} + \Gamma_{ajaj} + \Gamma_{bibi}, \end{aligned} \quad (3.22)$$

and the MP denominators are given by

$$\begin{aligned} \Delta_{ai}^{(\text{MP})} &= f_a - f_i \\ \Delta_{ia}^{(\text{MP})} &= f_i - f_a \\ \Delta_{abij}^{(\text{MP})} &= f_a + f_b - f_i - f_j \\ \Delta_{ijab}^{(\text{MP})} &= f_i + f_j - f_a - f_b. \end{aligned} \quad (3.23)$$

For additional details, we refer to Hergert et al. [4].

An alternative formulation to the White generator regularizes the matrix elements via the $\arctan$ function, so that “rotations” of the Hamiltonian are restricted to the interval $]-\frac{\pi}{4}, \frac{\pi}{4}[$ [5]. This formulation proves useful in the reference ensemble (Chapter 6), where this additional constraint also constrains the optimization manifold of correlation parameters.

The White-Atan generator reads

$$\begin{aligned} \eta_{\text{WA}}(s) = & \sum_{ph} \frac{1}{2} \arctan \frac{2f_{ai}(s)}{\Delta_{ai}(s)} N[a_a^\dagger a_i] \\ & + \frac{1}{4} \sum_{abij} \frac{1}{2} \arctan \frac{2\Gamma_{abij}(s)}{\Delta_{abij}(s)} N[a_a^\dagger a_b^\dagger a_j a_i] - H.c.. \end{aligned} \quad (3.24)$$

Finally, we note that the White generator can be implemented for a general, fractional reference state (e.g. the reference ensemble) by multiplying the relevant particle and hole indices with the corresponding particle and hole occupation numbers. This is the same strategy we use for density matrix normal-ordering in Chapter 2.8. For example, the matrix elements relevant to the one-body denominator Δ_{ai} in the White generator may be transformed to this generalized single-particle index scheme via,

$$\begin{aligned} f_i &\longrightarrow f_p n_p \\ f_a &\longrightarrow f_p \bar{n}_p \\ \Gamma_{aia i} &\longrightarrow \Gamma_{pqpq} n_p \bar{n}_q n_p \bar{n}_q. \end{aligned} \quad (3.25)$$

By transforming the matrix elements in this way, we ensure the White generator is compatible with the MR-IMSRG flow equations.

3.4.3 Brillouin Generator

In MR-IMSRG applications, the so-called Brillouin generator has proven to be a very useful and robust choice. It is defined by considering the change of the energy expectation value in the (potentially correlated) reference state under unitary variations. The evolved Hamiltonian is required to satisfy the so-called *irreducible Brillouin conditions* (IBCs) [6, 3], which are the

stationarity conditions for the unitary variation:

$$\begin{aligned}
\langle \Phi | [H(\infty), N [a_p^\dagger a_q]] | \Phi \rangle &= 0 \\
\langle \Phi | [H(\infty), N [a_p^\dagger a_q^\dagger a_s a_r]] | \Phi \rangle &= 0 \\
&\dots \text{(including conjugates)}.
\end{aligned} \tag{3.26}$$

Note that we cannot distinguish particle and hole indices here, since we allow correlated reference states and ensembles. In the limit where $|\Phi\rangle$ is a single Slater determinant, the IBCs reduce to the decoupling conditions in Equation 3.19.

The Brillouin generator is written in second-quantized form as usual,

$$\begin{aligned}
\eta(s) &= \sum_{pq} \eta_{pq}(s) N [a_p^\dagger a_q] \\
&\quad + \frac{1}{4} \sum_{pqrs} \eta_{pqrs}(s) N [a_p^\dagger a_q^\dagger a_s a_r],
\end{aligned} \tag{3.27}$$

where the matrix elements $\eta_{pq}(s), \eta_{pqrs}(s)$ are written in terms of the IBCs in Equation 3.26:

$$\begin{aligned}
\eta_{pq}(s) &\equiv \langle \Phi | [H(s), N [a_p^\dagger a_q]] | \Phi \rangle \\
\eta_{pqrs}(s) &\equiv \langle \Phi | [H(s), N [a_p^\dagger a_q^\dagger a_s a_r]] | \Phi \rangle.
\end{aligned} \tag{3.28}$$

Since these matrix elements vanish when the energy becomes stationary, we can think of the Brillouin generator as a kind of gradient of the energy under unitary variations. The commutators in Equation 3.28 are evaluated according to the generalized Wick contraction discussed in Chapter 2.6. The procedure for evaluating the commutators is similar to the derivation outlined in Chapter 3.3, but additional contractions appear that depend on the irreducible two- and three-body density matrices (see Chapter 2.6). Following Hergert [3], the Brillouin matrix elements are written in

Equations 3.29 and 3.30,

$$\eta_{pq} = (n_q - n_p) f_{qp} - \frac{1}{2} \sum_{rst} (\Gamma_{qrst} \lambda_{prst} - \Gamma_{rspt} \lambda_{rsqt}) \quad (3.29)$$

$$\begin{aligned} \eta_{pqrs} = & \Gamma_{rspq} (\bar{n}_p \bar{n}_q n_r n_s - n_p n_q \bar{n}_r \bar{n}_s) \\ & + \sum_t ((1 - P_{pq}) f_{tp} \lambda_{tqrs} - (1 - P_{rs}) f_{rt} \lambda_{pqts}) \\ & + \frac{1}{2} \left((\lambda \Gamma)_{rspq} (1 - n_p - n_q) - (\Gamma \lambda)_{rspq} (1 - n_r - n_s) \right) \\ & + (1 - P_{pq} - P_{rs}) \sum_{tu} (n_q - n_r) \Gamma_{truq} \lambda_{tpus} \\ & + \frac{1}{2} \sum_{tuv} ((1 - P_{rs}) \Gamma_{rtuv} \lambda_{tpquvs} - (1 - P_{ps}) \Gamma_{tupv} \lambda_{tuqvrs}) . \end{aligned} \quad (3.30)$$

For additional details, we refer to Hergert [3].

For completely general reference states, the use of the Brillouin generator naively increases our storage requirements to $O(N^6)$ because of its dependence on λ_{pqrstu} , and the computational cost increases to $O(N^7)$ because of the summation in the last line of Equation 3.30. Fortunately, we can exploit that the irreducible three-body density matrix is either compact or highly structured for many relevant types of correlated reference states, so that the $O(N^4)$ storage and $O(N^6)$ computational cost of the standard IMSRG(2) approach can be preserved in MR-IMSRG(2) applications.

BIBLIOGRAPHY

- [1] S.K. Bogner, R.J. Furnstahl, and A. Schwenk. “From low-momentum interactions to nuclear structure”. In: *Progress in Particle and Nuclear Physics* 65.1 (2010), pp. 94–147. ISSN: 0146-6410. DOI: <https://doi.org/10.1016/j.ppnp.2010.03.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0146641010000347>.
- [2] Stanisław D. Głazek and Kenneth G. Wilson. “Renormalization of Hamiltonians”. In: *Phys. Rev. D* 48 (12 Dec. 1993), pp. 5863–5872. DOI: 10.1103/PhysRevD.48.5863. URL: <https://link.aps.org/doi/10.1103/PhysRevD.48.5863>.
- [3] H Hergert. “In-medium similarity renormalization group for closed and open-shell nuclei”. In: *Physica Scripta* 92.2 (Dec. 2016), p. 023002. ISSN: 1402-4896. DOI: 10.1088/1402-4896/92/2/023002. URL: <http://dx.doi.org/10.1088/1402-4896/92/2/023002>.
- [4] H. Hergert et al. “The In-Medium Similarity Renormalization Group: A novel ab initio method for nuclei”. In: *Physics Reports* 621 (Mar. 2016), pp. 165–222. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2015.12.007. URL: <http://dx.doi.org/10.1016/j.physrep.2015.12.007>.
- [5] Morten Hjorth-Jensen, Maria-Paola Lombardo, and Ubirajara van Kolck. *An advanced course in computational nuclear physics: bridging the scales from quarks to neutron stars*. Springer, 2017.
- [6] Debashis Mukherjee and Werner Kutzelnigg. “Irreducible Brillouin conditions and contracted Schrödinger equations for n-electron systems. I. The equations satisfied by the density cumulants”. In: *The Journal of Chemical Physics* 114.5 (2001), pp. 2047–2061. DOI: 10.1063/1.1337058. eprint: <https://doi.org/10.1063/1.1337058>. URL: <https://doi.org/10.1063/1.1337058>.
- [7] I. Shavitt and R.J. Bartlett. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge Molecular Science. Cambridge University Press, 2009. ISBN: 9780521818322. URL: <https://books.google.com/books?id=SWw6ac1NHZYC>.
- [8] Franz J. Wegner. “Flow equations for Hamiltonians”. In: *Nuclear Physics B - Proceedings Supplements* 90 (2000). NON-PERTURBATIVE QCD AND HADRON PHENOMENOLOGY, pp. 141–146. ISSN: 0920-5632. DOI: [https://doi.org/10.1016/S0920-5632\(00\)00911-7](https://doi.org/10.1016/S0920-5632(00)00911-7). URL: <https://www.sciencedirect.com/science/article/pii/S0920563200009117>.
- [9] Steven R. White. “Numerical canonical transformation approach to quantum many-body problems”. In: *The Journal of Chemical Physics* 117.16 (Oct. 2002), pp. 7472–7482. ISSN: 0021-9606. DOI: 10.1063/1.1508370. eprint: https://pubs.aip.org/aip/jcp/article-pdf/117/16/7472/10843158/7472_1_online.pdf. URL: <https://doi.org/10.1063/1.1508370>.

CHAPTER 4

MODEL HAMILTONIANS

4.1 The Pairing Model Plus Particle-Hole Model

In this section, we introduce the model Hamiltonian which is used frequently in the body of work—namely, the pairing model. Information in this Chapter is compiled largely from Hjorth-Jensen et al. [5] and Hjorth-Jensen, Lombardo, and Kolck [6], although this model appears elsewhere outside the context of second-quantized many-body physics. The pairing model is a convenient tool for testing computational and theoretical methods in nuclear physics. For one reason, the model is exactly solvable, because the many-body basis dimension is *small enough* that the exponential scaling is not an issue.¹ We may probe the exact solution to the pairing model to validate the results for a particular method. For another reason, the pairing model captures qualitative features of more realistic nuclear interactions, despite its simplicity [5].

The pairing model consists of equidistant energy levels which contain up to two particles per level with opposite spin values, which we label + (up) and – (down) for convenience. The single-particle states are implicitly assumed to be the eigenstates of the one-particle operator, so that the one-body interaction term is diagonal. The two-body interaction term acts on pairs of particles with the same energy but opposite spin. Thus, the pairing model Hamiltonian can be written as

$$H_{pairing} = \delta \sum_{p\sigma} (p-1) a_{p\sigma}^\dagger a_{p\sigma} - \frac{g}{2} \sum_{pq} a_{p+}^\dagger a_{p-}^\dagger a_{q-} a_{q+}, \quad (4.1)$$

where σ runs over all spin values, δ controls the energy level spacing, and g controls the pairing strength. Note that the explicit – sign in the pairing term implies that we have an *attractive* pairing interaction for $g > 0$, i.e., the formation of pairs is energetically favorable. Moreover, the interaction term cannot break pairs by moving the particles to different energy levels, and it does not change the spins of the pair it acts upon. This implies that both the number of pairs in the system and the

¹We will demonstrate how to extract the exact solution via Full Configuration Interaction in a later section.

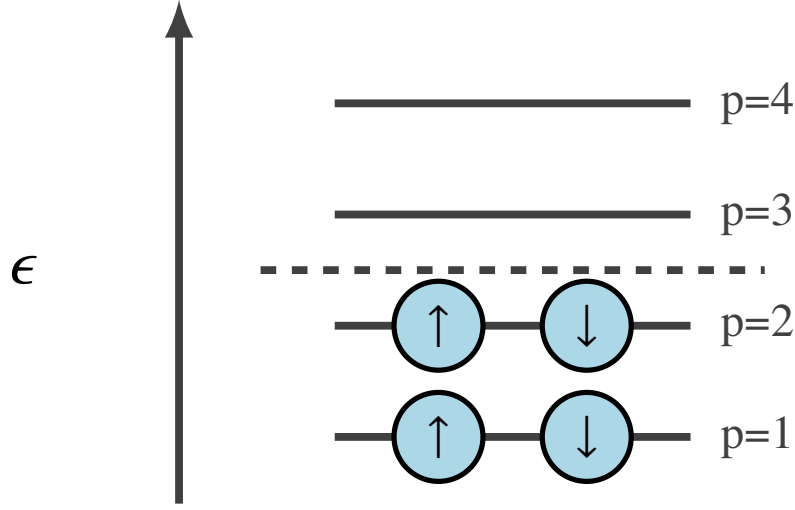


Figure 4.1 Pairing model of four particles at half filling, with two opposite spins per level. The vertical axis indicates increasing energy. The dashed line indicates the Fermi surface.

total spin projection

$$S_z = \frac{\hbar}{2} \sum_{i=1}^A \sigma_z(i) \quad (4.2)$$

are conserved.

To add complexity to our model, we can introduce an additional interaction term that can create one-particle-one-hole excitations and break up pairs. The Hamiltonian of this pairing-plus-particle-hole (P3H) model can be written as

$$H_{P3H} = H_{pairing} - \frac{b}{2} \sum_{pp'q} (a_{p+}^\dagger a_{p-}^\dagger a_{p'-} a_{q+} + a_{q+}^\dagger a_{p'-}^\dagger a_{p-} a_{p+}), \quad (4.3)$$

where b controls the particle-hole / pair-breaking interaction strength. Note that the spin structure of this interaction was chosen such that S_z remains conserved.

Figure 4.1 shows a ground-state configuration for an attractive pairing model with four particles distributed over four energy levels (for a total of eight single particle states). In the P3H model, on the other hand, we expect richer behavior because of the competition between the pairing and

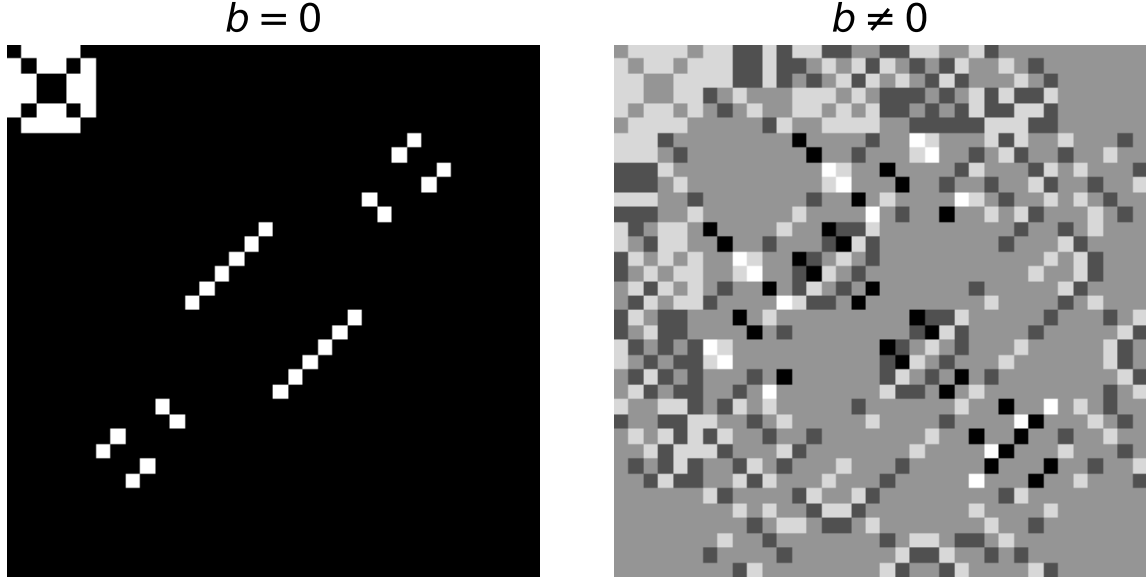


Figure 4.2 The left panel features a P3H Hamiltonian with $b = 0$, i.e. no pair-breaking. Black maps to zero. The right panel features a P3H Hamiltonian with $b \neq 0$. Grey maps to zero. The diagonal elements have been subtracted out to emphasize the off-diagonal structure of each Hamiltonian.

pair-breaking terms, and the configuration shown in Figure 4.1 will most likely not be a good approximation for the ground state.

In Figure 4.2, we show the qualitative structure of off-diagonal elements in the P3H Hamiltonian. We observe that the nonzero pair-breaking term probes new sectors of the Hamiltonian that were not accessible in the pure pairing interaction.

In Figures 4.3 and 4.4, we visualize the eigenstate structure of several P3H Hamiltonians, with no pair-breaking and no pairing, respectively. The qualitative features of these eigenstates are similar, in nature, to wavefunctions in the nuclear shell model, hence the heavy focus on the P3H model in many of this results of this work.

Each P3H Hamiltonian has been prepared in a basis of Slater determinants (SD) in an occupation number representation; therefore, each coefficient in the eigenvectors plotted in Figures 4.3 and 4.4 correspond to a particular SD configuration. For example, the first six coefficients indexed by v_0 through v_5 correspond to *fully-paired* configurations. Coefficients beyond v_5 correspond to *unpaired* configurations, in the spin $S = 0$ sub-block.

By examining the wavefunction structure in the SD basis, we learn how important each SD

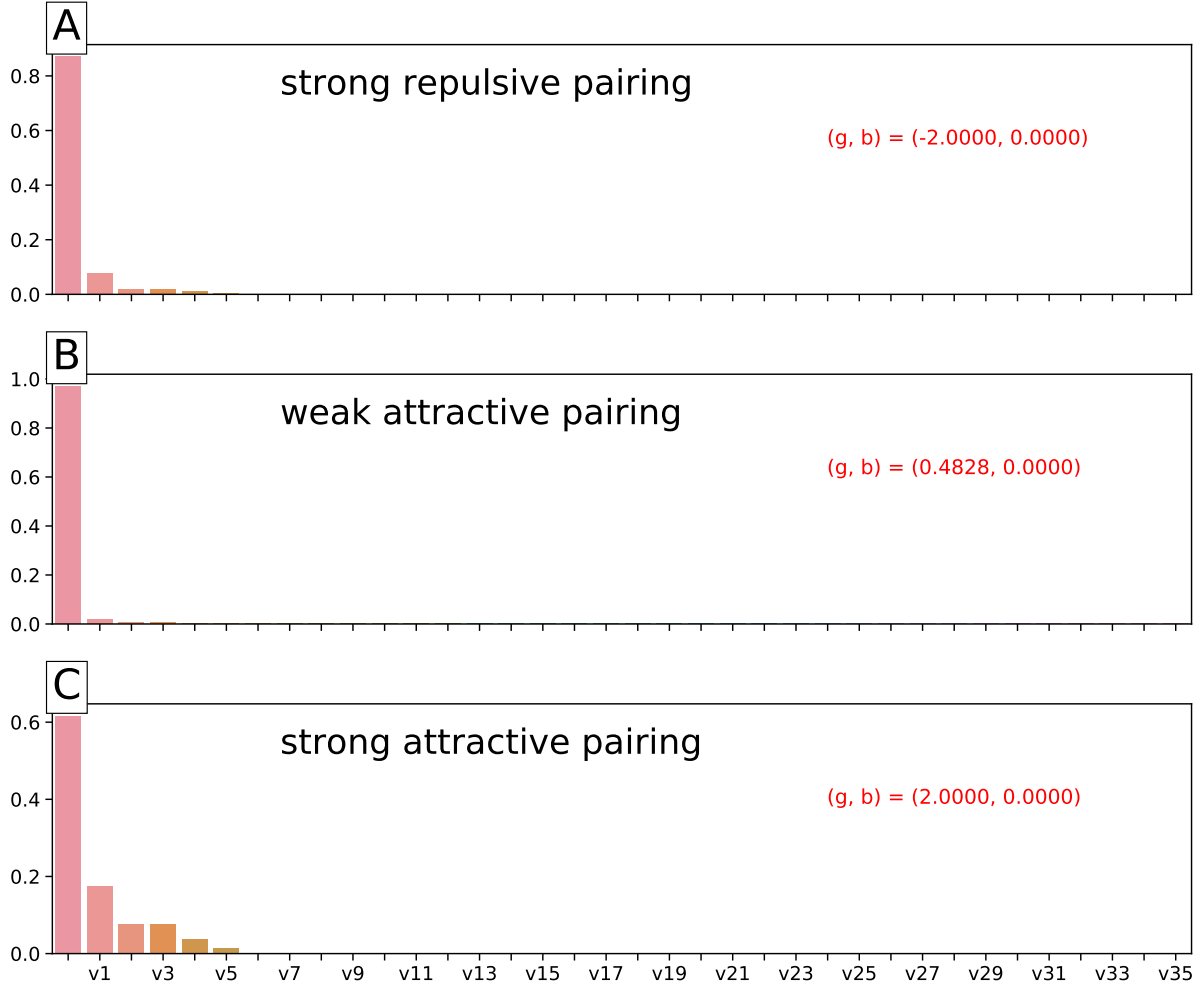


Figure 4.3 Three pairing-only examples of wavefunction structure in the P3H model. The horizontal axis is the index of the eigenvector coefficient. The vertical axis is the corresponding squared eigenvector coefficient.

configuration is in describing a particular eigenstate. For example, all three panels in Figure 4.3 immediately suggest that *unpaired* configurations contribute nothing to the eigenstates of the *pairing-only* P3H model. However, we see that the pairing strength controls *how much each paired configuration* contributes to the structure of the eigenstate. All three panels in Figure 4.4 show that most, or all, SD configurations are important to the eigenstate structure of the pair-breaking-only P3H model. Contributions for paired and unpaired configurations are especially important in the repulsive pair-breaking case.

Understanding the structure of the P3H eigenstates in the SD configuration basis is useful for

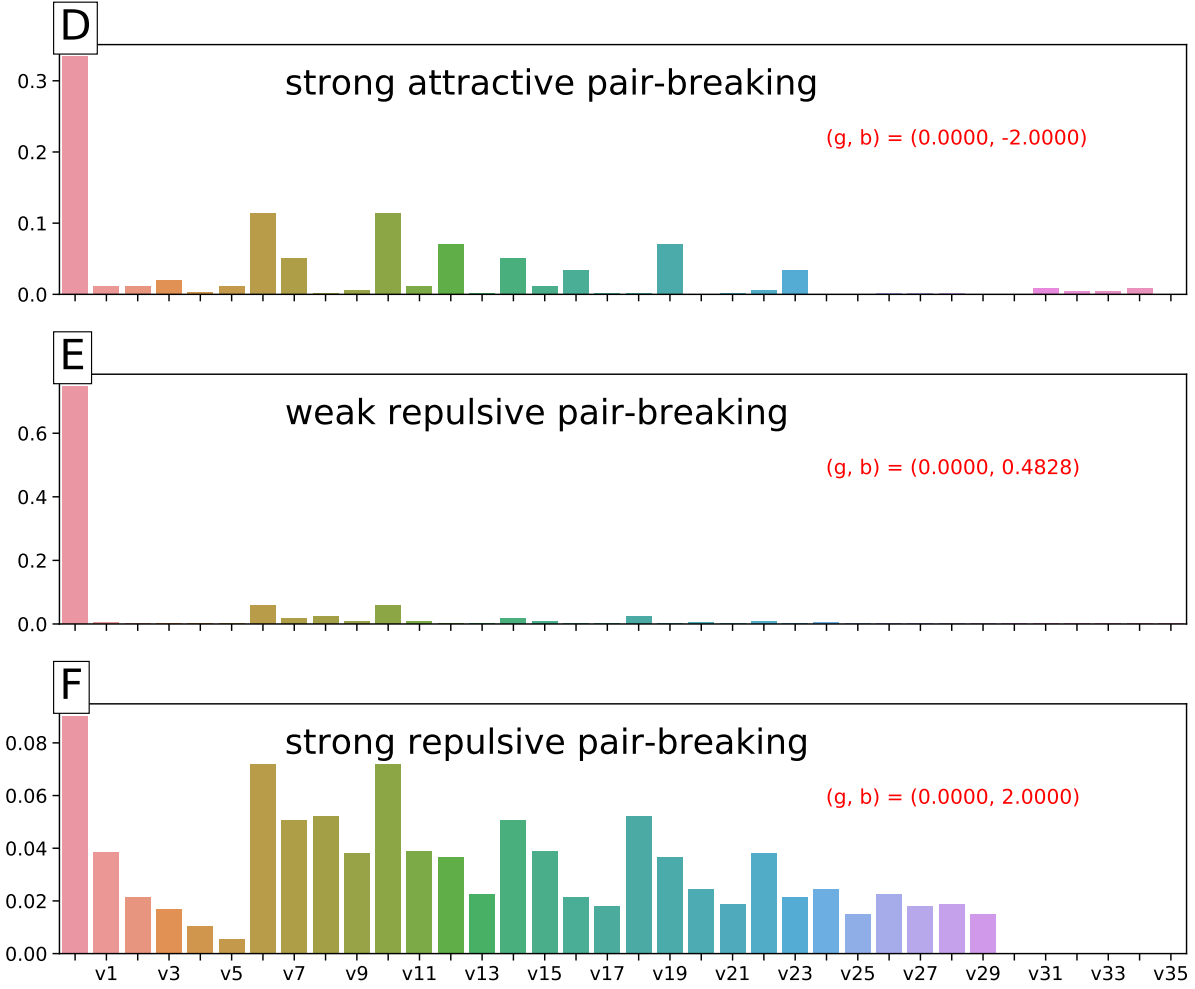


Figure 4.4 Three pair-breaking-only examples of wavefunction structure in the P3H model. The horizontal axis is the index of the eigenvector coefficient. The vertical axis is corresponding squared eigenvector coefficient.

the reference-state ensemble discussions in Chapter 6, because it offers insight on how to best to design an ensemble of SD configurations to capture the most important behavior in a particular eigenstate, or the full eigenspectrum.

The eigenvectors in Figures 4.3 and 4.4 were obtained via FCI calculations of the P3H Hamiltonians, which is the subject of Section 2.

4.2 Chiral Effective Field Theory for Nuclear Interactions

In this section, we provide a brief overview of chiral effective field theory (χ EFT), focusing on breadth rather than depth, in order to build context for the results in Chapters 7 and 8. Machleidt

and Entem [7] present a detailed discussion on the derivation of nuclear interactions from χ EFT, and Piarulli and Tews [8] provides a comprehensive review of local chiral nuclear interactions, in particular. We will skip any rigorous mathematical description of χ EFT, and instead summarize where the low-energy constants (LECs), relevant to Chapters 7 and 8, appear in the theory.

χ EFT is a low-energy, effective formulation of quantum chromodynamics (QCD)—“effective”, in the sense that the most *effective* degrees of freedom are strategically chosen to systems of interacting nucleons at the energy scales relevant to nuclear science experiments. QCD is the fundamental theory of the strong interaction in the Standard Model of Particle Physics, and provides tools to describe interactions at the level of quarks and gluons, which are the constituents of nucleons and other strongly interacting composite particles. Since the description of even a single nucleon at this level is extremely challenging, the description of all but the lightest nuclei is all but infeasible. However, it is not necessary to do so because the quark substructure of individual nucleons is irrelevant for the description of nuclear structure and dynamics. Thus, χ EFT pivots the description to the nucleons themselves, and uses pions as the mediators of the strong interaction at low energies. Essentially, we are lowering the resolution at which we describe the system, “blurring” out the substructure of nucleons and pions so that they can be treated as a point-like particles.

The conceptual framework of any EFT posits that we need to consider *all* interactions that are consistent with the symmetries of the underlying theory. In addition to rotational, translational and Lorentz invariance that are required of fundamental interactions, it is necessary to account for the chiral symmetry of QCD [7, 8]. In principle, we can conceive of infinitely many interactions that respect those symmetries, but a properly constructed EFT provides a systematic framework for classifying their importance and truncating interactions that give negligible contributions to the observables of interest — in the case of χ EFT, this is the power counting explained below. In the χ EFT framework, interactions now fall into two broad categories — long-range interactions that are modeled by the exchange of pions (π) between nucleons (N), and short-range contact interactions between nucleons. The parameters associated with corrections to πN interaction vertices as well as the contact interactions parameterize the QCD physics that is not directly resolved by the EFT

in the most general, unbiased way, so as not to introduce modeling artifacts. Collectively, these parameters are referred to as the *low-energy constants* (LECs) of the theory. In principle, they could be determined from QCD via direct computations or matching results for observables that are accessible in both theories, but this is currently not feasible. Instead, the LECs are fit to low-energy QCD data, such as NN scattering phase shifts, or 3N observables.

In addition to a χ EFT with N and π degrees of freedom, we will also consider the so-called Δ -full χ EFT, a variant that additionally includes the Δ isobar as a degree of freedom. This particle can be thought of as an excited $N\pi$ resonant state. The energy and momenta of this excitation are sufficiently low to overlap with the range we seek to describe with χ EFT, so testing its impact on the description of nuclear observables is of high interest. We do note that the Δ s are only included as *virtual* intermediate states — roughly speaking, this means that while Δ s will not be observed as constituents of a nucleus on any experimental time scale, they can be formed for brief instants of time during an interaction process that might leave an imprint on the description of nuclear observables. In the Δ -less χ EFT, the effects of the intermediate Δ excitations are implicitly included in LECs of the theory.

The interactions relevant to this dissertation have been derived from Weinberg power-counting [9] of the ratio Q/Λ_b , where Q is a typical momentum scale and Λ_b is the breakdown energy scale, or cutoff, beyond which χ EFT is no longer a valid theory for the chosen degrees of freedom. Λ_b is typically chosen in the range 500 – 1000 MeV, either based on the masses of particles that are not explicit degrees of freedom in the χ EFT scheme, or combinations of parameters that are related to physics beyond the χ EFT description. With typical momentum scales Q being of the order of the Fermi momentum $k_F \approx 1.4 \text{ fm}^{-1}$ (corresponding to approximately 280 MeV) or the pion mass $m_\pi = 0.7 \text{ fm}^{-1}$ (approx. 140 MeV), the ratio Q/Λ_b will then roughly be on the order of 1/5 to 1/3, so that higher-order processes are suppressed at a reasonable rate.

The pion exchange and contact interactions can then be organized according to powers Q/λ_χ , leading to the organization scheme shown in Figure 4.5. In the common terminology, the $(Q/\Lambda_b)^{\nu=0} = 1$ interactions are referred to as leading order (LO), $\nu = 2$ as next-to-leading or-

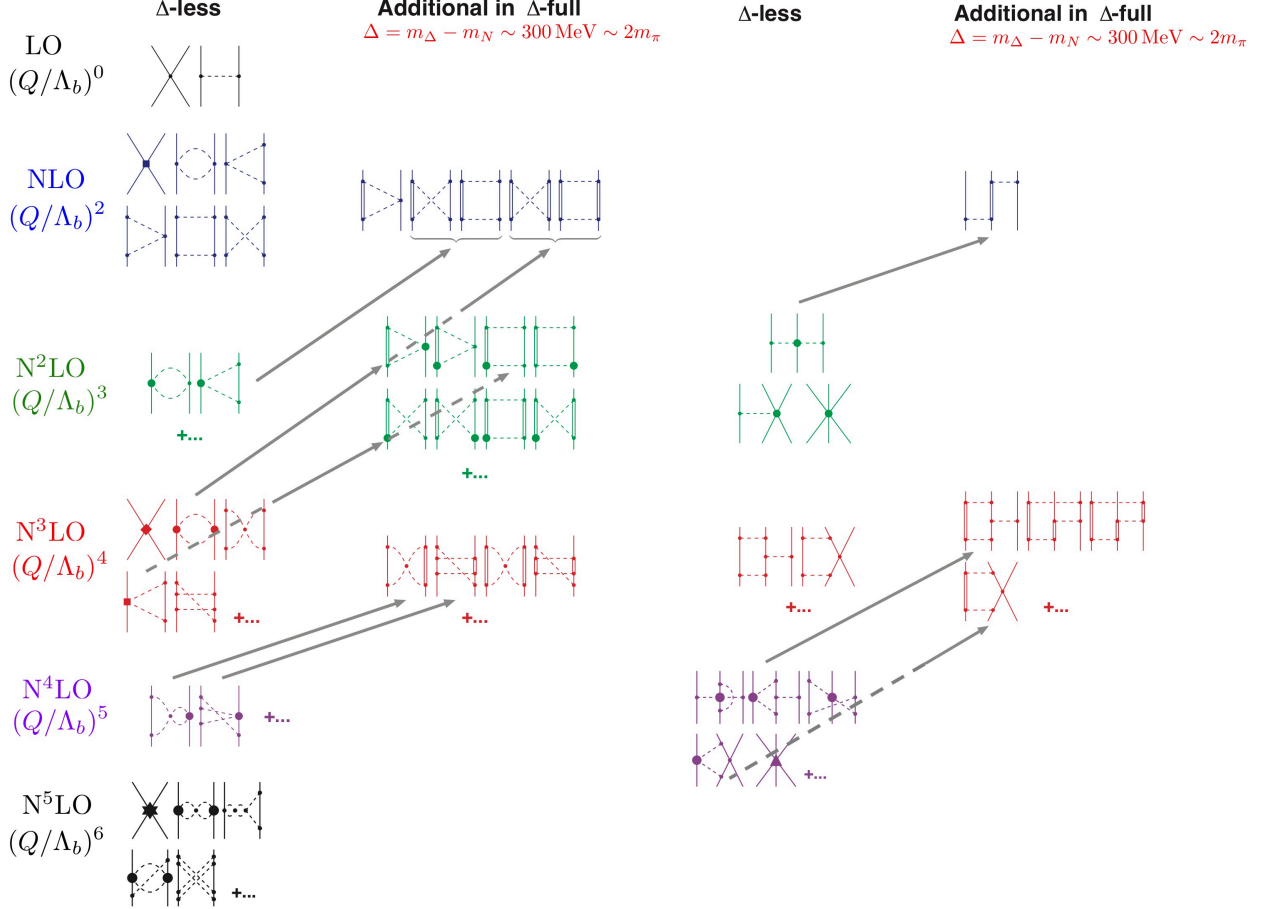


Figure 4.5 Figure taken from Piarulli and Tews [8]. On the left column are the NN force diagrams. On the right column are the 3N force diagrams. Solid lines are nucleons, while dashed lines are pions. Double solid lines are Δ -isobars. Each diagram may be identified with an LEC, which controls the strength of the contact force described by the diagram.

der (NLO), $\nu = 3$ as next-to-next-to-leading order (NNLO) and so on. Note the absence of $\nu = 1$, because parity invariance stipulates that nuclear potentials cannot be linear in momentum [7, 8]. Diagrams with dashed lines involve the exchange of pions, while diagrams with only solid single lines are nuclear contact interactions. Double solid lines indicate Δ isobars in the Δ -full χ EFT. The arrows indicate how Δ -full diagrams emerge from the short-range parameterizations in the Δ -less theory — note that these diagrams are usually *promoted* in order, which suggests that there should be differences in how rapidly observables converge in the two EFT variants, and which order of the expansion is required to achieve a desired precision.

The relative momentum described by NN scattering is decomposed into “partial wave” con-

tributions of differing angular momentum, where the S-wave partial waves corresponds to orbital angular momentum equal to 0, P-wave partial waves corresponds to orbital angular momentum equal to 1, D-wave partial waves corresponds to orbital angular momentum 2, and so on. The orbital angular momentum couples to spin to produce the partial waves, characterized by differing total angular momentum. At LO, only one-pion exchange and S-wave contact forces contribute. At NLO, we have two-pion exchanges plus subleading contact forces in the S wave and P waves. In the delta-full χ EFT, a leading 3N force also appears at this order (also called the Fujita-Miyawazawa force, named for the work of [4] in its derivation). At NNLO, there is no new contact term in the NN interaction, but higher-order corrections to the πN vertex appear: The LECs which control these contributions are c_1, c_2, c_3, c_4 , where c_2 is only relevant in the Δ -full theory. The LECs c_1, c_3 and c_4 also appear consistently in the long-range part of the NNLO 3N interaction, and in addition, a mixed long- and short-range term (proportional to c_D) and a genuine contact (proportional to c_E) contribute. At N³LO contact forces up to D-waves appear in the NN force, adding to the number of parameters, in addition to more complicated but parameter-free 3N forces as well as a leading 4N force. In Tables 4.1 and 4.2 we list the “baseline” LEC sets for two different NNLO interactions (up to 3B forces) and one N3LO interaction (up to 2B forces), respectively, that will be explored in later chapters of this work.

	$\Delta\text{NNLO}_{\text{GO}}(394)$	$\text{NNLO}_{\text{sat}}(450)$
$\tilde{C}_{1S_0}^{(pp)}$	-0.338142	-0.158149
$\tilde{C}_{1S_0}^{(np)}$	-0.339250	-0.159822
$\tilde{C}_{1S_0}^{(nn)}$	-0.338746	-0.159150
$\tilde{C}_{3S_1}^{(pp)}$	-0.259839	-0.177674
$\tilde{C}_{3S_1}^{(np)}$	-0.259839	-0.177674
$\tilde{C}_{3S_1}^{(nn)}$	-0.259839	-0.177674
C_{1S_0}	+2.505389	+2.539368
C_{3P_0}	+0.700499	+1.398366
C_{1P_1}	-0.387960	+0.555959
C_{3P_1}	-0.964856	-1.136095
C_{3S_1}	+1.002189	+1.002893
$C_{3S_1-3D_1}$	+0.452523	+0.600716
C_{3P_2}	-0.883122	-0.802300
c_1	-0.74	-1.121521
c_2	-0.49	—
c_3	-0.65	-3.925006
c_4	+0.96	+3.765687
c_D	+0.081	+0.081
c_E	-0.002	-0.002

Table 4.1 $\Delta\text{NNLO}_{\text{GO}}(394)$ [1] and $\text{NNLO}_{\text{sat}}(450)$ [2] interaction LEC values (including c_E and c_D relevant to 3B force). We refer to these as “baseline” LEC sets. We thank Andreas Ekström for providing NN interaction routines for the relevant LEC values.

NN ³ LO(500)	
$\tilde{C}_{1S_0}^{(pp)}$	-0.145286
$\tilde{C}_{1S_0}^{(np)}$	-0.147167
$\tilde{C}_{1S_0}^{(nn)}$	-0.146285
$\tilde{C}_{3S_1}^{(pp)}$	-0.118972
$\tilde{C}_{3S_1}^{(np)}$	-0.118972
$\tilde{C}_{3S_1}^{(nn)}$	-0.118972
C_{1S_0}	+2.38
C_{3P_0}	+1.487
C_{1P_1}	+0.656
C_{3P_1}	-0.63
C_{3S_1}	+0.76
$C_{3S_1-3D_1}$	+0.826
C_{3P_2}	-0.538
$\hat{D}_{1S_0}$	-2.545
D_{1S_0}	-16.0
D_{3P_0}	+0.245
D_{1P_1}	+5.25
D_{3P_1}	+2.35
$\hat{D}_{3S_1}$	+7.0
D_{3S_1}	+6.55
D_{3D_1}	-2.8
$\hat{D}_{3S_1-3D_1}$	+2.25
$D_{3S_1-3D_1}$	+6.61
D_{1D_2}	-1.77
D_{3D_2}	-1.46
D_{3P_2}	+2.295
$D_{3P_2-3F_2}$	-0.465
D_{3D_3}	+5.66
c_1	-0.81
c_2	+2.8
c_3	-3.2
c_4	+5.4
$d_1 + d_2$	+3.06
d_3	-3.27
d_5	+0.45
$d_{14} - d_{15}$	-5.65

Table 4.2 “Baseline” LEC set for the N³LO interaction with cutoff $\Lambda = 500$ MeV by Entem and Machleidt [3, 7]. Does not include LECs relevant to 3B force.

BIBLIOGRAPHY

- [1] A. Ekström et al. “ Δ isobars and nuclear saturation”. In: *Phys. Rev. C* 97 (2 Feb. 2018), p. 024332. DOI: 10.1103/PhysRevC.97.024332. URL: <https://link.aps.org/doi/10.1103/PhysRevC.97.024332>.
- [2] A. Ekström et al. “Accurate nuclear radii and binding energies from a chiral interaction”. In: *Phys. Rev. C* 91 (5 May 2015), p. 051301. DOI: 10.1103/PhysRevC.91.051301. URL: <https://link.aps.org/doi/10.1103/PhysRevC.91.051301>.
- [3] D. R. Entem and R. Machleidt. “Accurate charge-dependent nucleon-nucleon potential at fourth order of chiral perturbation theory”. In: *Physical Review C* 68.4 (Oct. 2003). DOI: 10.1103/physrevc.68.041001. URL: <https://doi.org/10.1103%5C%2Fphysrevc.68.041001>.
- [4] Jun-ichi Fujita and Hironari Miyazawa. “Pion Theory of Three-Body Forces”. In: *Progress of Theoretical Physics* 17.3 (Mar. 1957), pp. 360–365. ISSN: 0033-068X. DOI: 10.1143/PTP.17.360. eprint: <https://academic.oup.com/ptp/article-pdf/17/3/360/5252121/17-3-360.pdf>. URL: <https://doi.org/10.1143/PTP.17.360>.
- [5] M Hjorth-Jensen et al. “Many-body interactions and nuclear structure”. In: *Journal of Physics G: Nuclear and Particle Physics* 37.6 (Apr. 2010), p. 064035. DOI: 10.1088/0954-3899/37/6/064035. URL: <https://doi.org/10.1088%2F0954-3899%2F37%2F6%2F064035>.
- [6] Morten Hjorth-Jensen, Maria-Paola Lombardo, and Ubirajara van Kolck. *An advanced course in computational nuclear physics: bridging the scales from quarks to neutron stars*. Springer, 2017.
- [7] R. Machleidt and D.R. Entem. “Chiral effective field theory and nuclear forces”. In: *Physics Reports* 503.1 (June 2011), pp. 1–75. DOI: 10.1016/j.physrep.2011.02.001. URL: <https://doi.org/10.1016%5C%2Fj.physrep.2011.02.001>.
- [8] Maria Piarulli and Ingo Tews. “Local Nucleon-Nucleon and Three-Nucleon Interactions Within Chiral Effective Field Theory”. In: *Frontiers in Physics* 7 (2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2019.00245. URL: <https://www.frontiersin.org/articles/10.3389/fphy.2019.00245>.
- [9] Steven Weinberg. “High-Energy Behavior in Quantum Field Theory”. In: *Phys. Rev.* 118 (3 May 1960), pp. 838–849. DOI: 10.1103/PhysRev.118.838. URL: <https://link.aps.org/doi/10.1103/PhysRev.118.838>.

CHAPTER 5

BUILDING AN IMSRG MANY-BODY CODE USING TENSOR NETWORKS

5.1 Introduction

Tensor network theory, in the context of this dissertation, describes the abstraction of multidimensional arrays and mathematical operations on them into a diagrammatic “formalism”, which simplifies the calculations which involve these often complex data structures. The diagrammatic method of tensor networks is somewhat less rigorous, but analogous, to the Goldstone and Hugenholtz diagrammatic notation used in second-quantized many-body theory (see Chapter 4 of Shavitt and Bartlett [11]) to render many-body calculations less cumbersome. As a result, translating operators and working equations in methods like the IMSRG into tensor networks feels natural. In addition, tensor networks and tensor decompositions have been successfully applied to many-body methods in areas adjacent to this work, including Many-Body Perturbation Theory [14, 13], variational Monte-Carlo [1], Tree Tensor Network States [6], Density Matrix Renormalization Group [12], and Tensor Network Renormalization [2].

In this body of work, we use tensor networks to provide an interface for abstraction of the computation of the IMSRG flow equations (e.g. Chapter 3, Equation 3.18). We want to offload the effort of computing the the IMSRG flow equations to a generalized tensor network library, in order to leverage the efficiency and optimization techniques implemented therein. With the computational load abstracted, we may freely exchange tensor network libraries, as well as sync our many-body codes with future improvements in those libraries. Ultimately, we strive for a separation of the workload between many-body physicists and computer scientists, so that the scientific progress benefits from ongoing optimizations of algorithms, without the friction associated with ad-hoc improvements to the many-body codebase. This strategy offers more long-term sustainability for the code relevant to the science, because of changing compiler and hardware landscapes.

In addition, we present two IMSRG codes which were written or contributed to aid the studies that encompass this dissertation. These codes employ key ideas from tensor network theory, and

are discussed in Chapter 5.5 and Chapter 5.6. The purpose of these sections is to explain how these codes operate and the logic in their design. Appendix C features a catalog of these two codes, as well as their dependencies.

Both IMSRG codes were designed with an object-oriented approach, so that the pieces which comprise the IMSRG “engine” are modular in nature. That is, users may e.g. write their own Hamiltonian class, inheriting from the appropriate interfaces, and plug directly into the rest of the IMSRG machine with relative simplicity. Additionally, one may write their own backends for performing the tensor contractions with solve the IMSRG flow equation terms. In principle, the administrative overhead in executing an object-oriented code is negligible compared to a sequential version of the same code; however, there are certainly areas within the codes we introduce that could benefit from a closer look at optimal memory management and code organization. For now, our goal is to provide simple access to validating results via the P3H model, as well as a template for improving the IMSRG production code with tensor libraries.

Chapter 5.4 provides a detailed overview of how tensor networks are integrated into the computation of the IMSRG flow equations. These tensor networks are translated into code by tensor network calculation libraries such as TensorNetwork¹ [9], built in Python, and the Tensor Algebra Compiler² [5], built in C++.

5.2 Tensor Network Diagrams

We may introduce a *tensor*, graphically as in Figure 5.1, as a node with a number of dangling *edges* corresponding to indices of a multidimensional array [9, 7]. Abstractly, the order of the edges does not matter (although they might matter in certain physics applications). As a general rule, dangling edges represent fixed indices, and edges between vertices (or loops) are summed over in an operation called a *contraction*.

Contractions between tensors are performed graphically via connecting two edges. For example, a trace over one index in Figure 5.2 encodes the operation written mathematically,

¹Source: <https://github.com/google/TensorNetwork>

²Source: <https://github.com/tensor-compiler/taco>

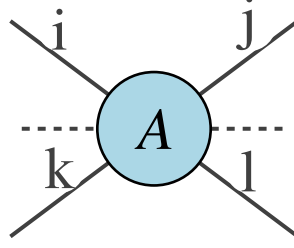


Figure 5.1 A tensor A with some number of dangling edges. Evaluation of indices $i \dots j \dots k \dots l$ gives the value of A at that location.

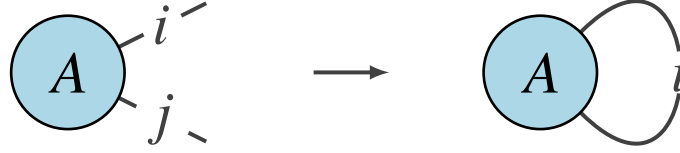


Figure 5.2 The trace operation over two dangling edges of the tensor A .

$$\text{tr}(A) = \sum_i A_{ii}. \quad (5.1)$$

Additional examples of common linear algebra operations, translated to a tensor diagram, are displayed in Figure 5.3. Notice that the resultant tensor is immediately identified by the dangling edges of the network, which remain fixed through the operation. A tensor network, however arbitrarily complex, will always involve dangling edges and/or contraction edges. Based on how the nodes and edges are connected in the diagram, we can always translate the diagram to mathematical form.

5.3 Factorization with Tensor Train Decomposition

In general, the problems involving tensors require computing resources that scale exponentially in the dimension of the problem. Naive tensor storage (that is, a contiguous multidimensional array) is impractical or impossible to implement in large dimensional problems. As a result,

$$\begin{array}{ll}
R = \sum_i A_i B_i & \text{---} \textcircled{A} \text{---} i \text{---} \textcircled{B} \text{---} \\
R_i = \sum_j A_{ij} B_j & \text{---} i \text{---} \textcircled{A} \text{---} j \text{---} \textcircled{B} \text{---} \\
R_{ij} = \sum_k A_{ik} B_{kj} & \text{---} i \text{---} \textcircled{A} \text{---} k \text{---} \textcircled{B} \text{---} j \text{---}
\end{array}$$

Figure 5.3 Depicted are the vector-vector multiplication, matrix vector multiplication, and matrix-matrix multiplication in tensor diagrams.

we seek efficient and effective representations of tensors that, in general, do not require storage of the complete tensor—instead, the tensor is *decomposed* into characteristic *factors* which can be contracted in order to reconstruct the full tensor. The tensor train decomposition (or TT-decomposition) is one such factorization technique introduced by Oseledets [8] as an alternative to the NP-hard problem of the canonical decomposition [8],

$$A(i_1, i_2, \dots, i_d) = \sum_{\alpha=1}^r U_1(i_1, \alpha) U_2(i_2, \alpha) \dots U_d(i_d, \alpha), \quad (5.2)$$

where $A(i_1, i_2, \dots, i_d)$ represents the element of A associated with the indices $i_1, i_2, \dots, i_d$. Note that i_k spans an unspecified range n_k , which may or may not be equal to another index range n_l . For most methods involving the canonical decomposition, the decomposition rank r must be known and the algorithms for computing the low-rank tensor components are not stable [8]. Figure 5.4 illustrates a schematic diagram of the canonical decomposition. There are d canonical factors which correspond to the tensor dimension d . Each decomposition rank features an outer product over all canonical factors in that rank, and the sum over all ranks generate the full tensor A . The decomposition rank r is taken as the minimum number of terms which reproduce A satisfying an error threshold ϵ .

The TT-decomposition seeks a tensor A which approximates a target tensor B such that $A \approx B$.

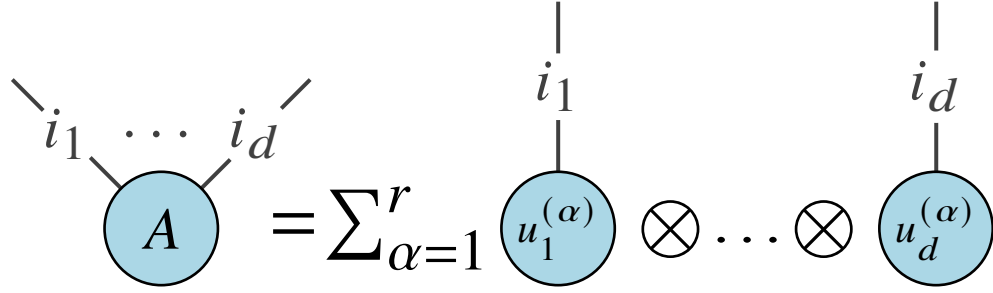


Figure 5.4 Schematic tensor network diagram of the canonical decomposition. The rank 1 tensors (vectors) are referred to as *canonical factors*.

The elements of A , which is characterized by d indices, are computed via

$$A(i_1, i_2, \dots, i_d) = G(i_1)G(i_2) \dots G(i_d), \quad (5.3)$$

where $G(i_k)$ is a *matrix* with dimension $r_{k-1} \times r_k$. The elements of the tensor A are expressed as a matrix product. Thus, instead of storing the number of elements corresponding to A , which might be as large as n^d , we store a number of elements equal to $\sum_{k=1}^d r_{k-1}r_k$ which is considerably more efficient in most cases. Figure 5.5 draws a TT-decomposition diagram. The sum over decomposition ranks does not appear in the TT-decomposition, in contrast to the canonical decomposition in Figure 5.4. The auxiliary indices α which contract between TT cores are fixed. The cores which reproduce the elements of A , which we denote with G , are structurally different than the cores which reproduce the full tensor $\mathbf{A}$ (in bold-face), which we denote in bold-face $\mathbf{G}$. However, we emphasize that $\mathbf{G}$ is a full realization of the parameters that characterize G .

The procedure for generating a TT-decomposition is not unique in general, due to algorithms which minimize a norm deviation, such as the Frobenius error. In these cases minimization may not yield unique results. In this study, we have not used an algorithmic procedure to generate the relevant TT-decomposition, instead relying on *ad hoc* inspection, because the decomposition are simple enough to derive by hand. The inspection method is best explained by example. Particularly helpful explanations can be found in [10]. Consider a rank 2 tensor A with elements given by $A(a, b) = n_a - n_b$, where the right-hand side defines a particular element, corresponding to (a, b) .

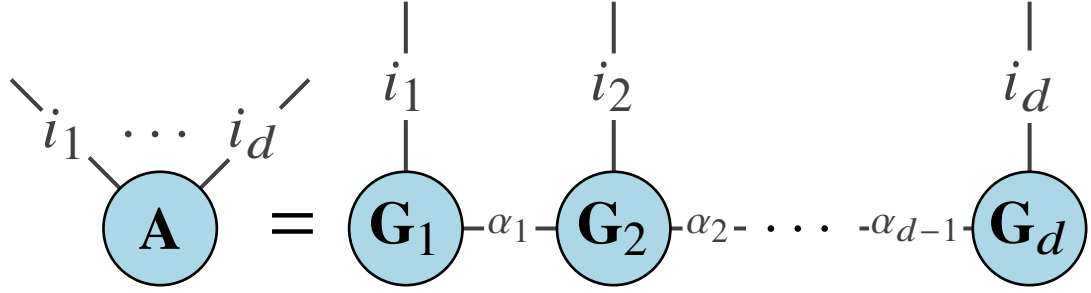


Figure 5.5 Schematic tensor network diagram of the tensor-train decomposition. When drawing the reconstruction of the full tensor, note that the boundary cores appear as rank 2 tensors, while inner cores appear as rank 3 tensors.

The elements n_a and n_b are numbers which come from two distinct (or equivalent) sets. Let the size of these sets be equal to N . Then, the total number of elements stored by the tensor $A(a, b)$ is N^2 .

We can generate a TT-decomposition by inspection from the analytical representation of the tensor elements $A(a, b)$. We need a number of smaller tensors G equal to 2, by Equation 5.3 ($d = 2$). For example,

$$A(a, b) = G(n_a)G(n_b) = \begin{bmatrix} n_a & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -n_b \end{bmatrix} = n_a - n_b. \quad (5.4)$$

In order to evaluate the tensor element at a particular index location, we need only evaluate the multiplication of the smaller tensor “cores” corresponding to that index location. Note that under the TT-decomposition scheme, we need only store the decomposed tensors, which in this case amounts to $2N + 2N = 4N$ total elements instead of N^2 elements to store the full tensor—a reduction in memory scaling from quadratic to linear (refer to Figure 5.6 for an example of how memory scales with dimension). The disadvantage, of course, is the computational cost associated with evaluating each tensor element dynamically. Later we will show that for a sufficient computing environment, this cost is negligible for the TT-decomposition relevant to this body of work. Although, we emphasize that the cost is dependent on the problem for which the TT-decomposition has been applied.

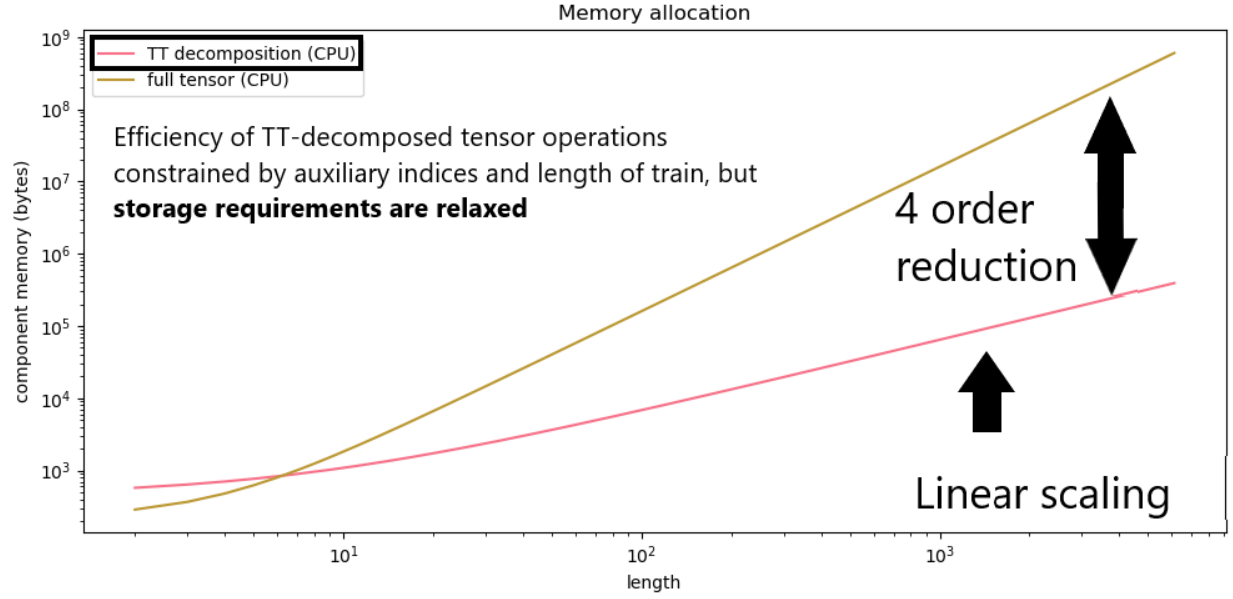


Figure 5.6 Memory scaling for the rank 2 tensor in Equation 5.4, comparing storage of the full matrix to the storage for the tensor train components.

Notice that the TT-decomposition in Equation 5.4 can be evaluated to compute the full tensor $\mathbf{A}$ by building two matrices $\mathbf{G}(n_a), \mathbf{G}(n_b)$ from all possible n_a, n_b in tensor cores $G(n_a), G(n_b)$.

Let $N = 4$, and let the possible $n_a = n_b = [1, 1, 0, 0]$ for $a = 1, 2, 3, 4$, respectively. Then,

$$\begin{aligned}
 \mathbf{G}(n_a)\mathbf{G}(n_b) &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} = \mathbf{A}.
 \end{aligned} \tag{5.5}$$

In this way, the tensor cores can be fully contracted to recover the full tensor $\mathbf{A}$. An important feature of this exact decomposition is that we can use the decomposed tensor cores to perform calculations without ever explicitly constructing the full tensor $\mathbf{A}$. Consider the matrix-vector

operation $\mathbf{A}\mathbf{v}$, where $\mathbf{v}^T = [1, 2, 3, 4]$,

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 7 \\ 7 \\ -3 \\ -3 \end{bmatrix} \quad (5.6)$$

$$\begin{aligned} \mathbf{G}(n_a)\mathbf{G}(n_b)\mathbf{v} &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ -3 \end{bmatrix} = \begin{bmatrix} 7 \\ 7 \\ -3 \\ -3 \end{bmatrix}. \end{aligned} \quad (5.7)$$

This short example is not meant to formally prove this fact, but to provide intuition for why this method works. The outer axes of the decomposition contain information about the elements of the full tensor. These axes directly interact with the multiplying factor, and the auxiliary axes contract only among themselves. Therefore, as long as the full tensor cores are stored for a particular (exact) TT-decomposition, calculations involving the full tensor can be performed by substituting the TT-decomposition. In this idealized example of exact decomposition, the operation in Equation 5.6 scales $O(N^2)$, while the operation in Equation 5.7 scales $O(2 * N + N * 2) = O(4N)$. In this special case, the complexity is actually reduced (not accounting for the associated cost of performing the decomposition, which in this special case was performed *ad hoc* by inspection).

An additional benefit of the TT-decomposition is efficiency in large-scale calculations. Our hope is that the cost of evaluating the TT-decomposition, in tandem with e.g. a matrix calculation, does not affect performance when compared to standard numerical methods; in Figure 5.7 we show that factorization via TT-decomposition of the two input matrices to a random matrix-matrix calculation achieves comparable time scaling to standard, optimized techniques. Note that the

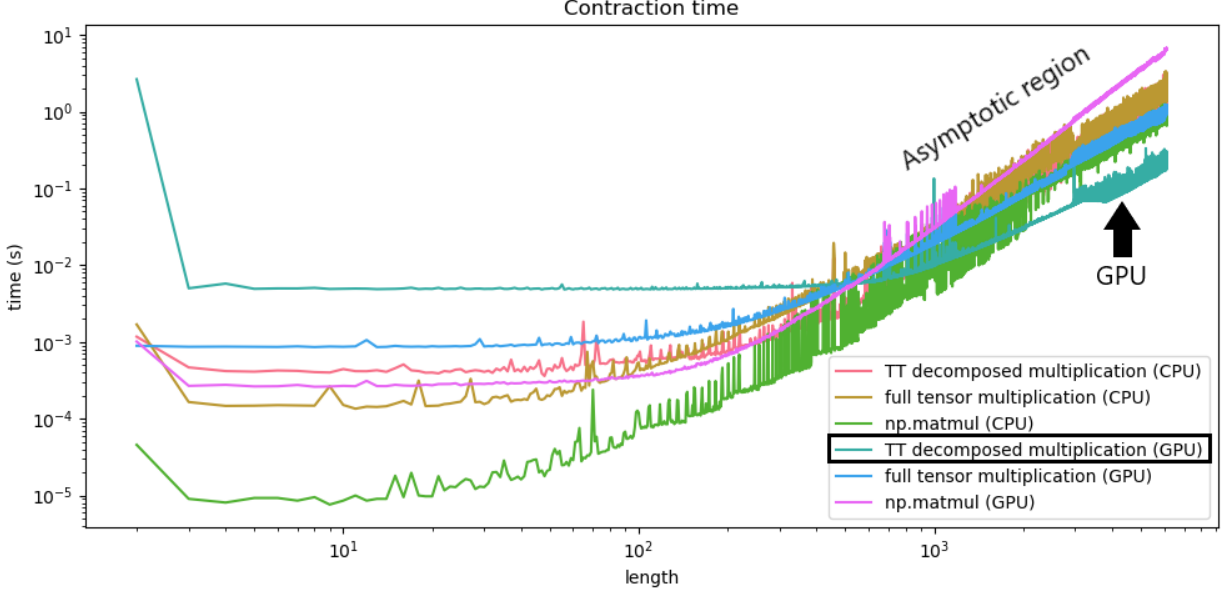


Figure 5.7 Time scaling for performing matrix-matrix multiplication, while comparing several methods. Each curve computes multiplication of two random, dense, square matrices of varying dimension. The calculations were performed in Python using TensorNetwork [9] and numpy.

TT-decomposed calculations seem to scale slightly better than the naive $O(N^3)$, which is likely due to a similar scaling reduction to the example in Equations 5.6 and 5.7.

5.4 Tensor Networks in the IMSRG

As discussed in the Introduction (Chapter 5.1), one of the reasons for viewing the IMSRG through a tensor network lens is to sequester the effort to compute the tensor contractions in Equation 3.18, the IMSRG(2) flow equations, into a separate library, or modular environment, where the optimization can be controlled independently of the many-body code. Additionally, we leverage the TT-decomposition discussed in Chapter 5.3 to bundle the decomposed occupation factors in Equation 3.18 with the IMSRG coefficient tensor contraction operations. In the future, we want to leverage tensor factorization beyond the occupation factors, and apply these factorization ideas to the IMSRG flow terms (Equation 3.18) directly. Zhu, Wirth, and Hergert [15] has shown success in factorizing the χ EFT interactions (see Chapter 4.2) used as inputs for the IMSRG, which motivates further effort in a factorized IMSRG flow.

Based on the discussion in Chapter 5.2, we observe that any term in Equation 3.18 which

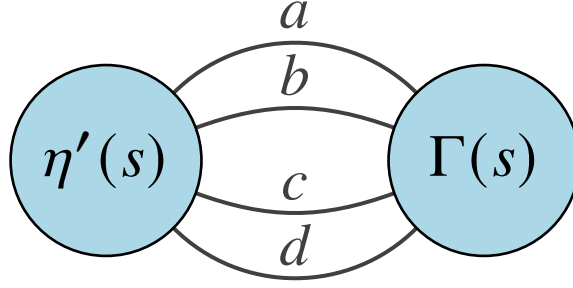


Figure 5.8 Tensor diagram corresponding to the two-body contribution to the reference energy in the IMSRG(2) flow equations in Equation 3.18.

contains a multiplication of two tensors, as well as a sum over a shared index between those two tensors, can be translated to a tensor network diagram. For example, consider the two-body contribution to the reference energy (ignoring numerical factors):

$$\left(\frac{dE(s)}{ds}\right)^{(2)} = \sum_{abcd} \eta_{abcd}(s) \Gamma_{cdab}(s) n_a n_b \bar{n}_c \bar{n}_d. \quad (5.8)$$

While this term actually contains three tensors to contract, we can leverage the fact that the occupation factor in $n_a n_b \bar{n}_c \bar{n}_d$ is just a number independent of s ; we may freely commute the occupation factor among the IMSRG(2) coefficients, and combine with the η factor via elementwise multiplication to obtain

$$\left(\frac{dE(s)}{ds}\right)^{(2)} = \sum_{abcd} \eta'_{abcd}(s) \Gamma_{cdab}(s), \quad (5.9)$$

where $\eta'_{abcd}(s)$ is now

$$\eta'_{abcd}(s) = n_a n_b \bar{n}_c \bar{n}_d \cdot \eta_{abcd}(s). \quad (5.10)$$

The tensor network diagram corresponding to this term might look like Figure 5.8.

Let us express the *elements* of the left hand tensor $\eta'(s)$ in a network diagram, assuming that the occupation factor $n_a n_b \bar{n}_c \bar{n}_d$ has been TT-decomposed (see Appendix A.1, Equation A.5). This particular occupation factor is especially easy to decompose, because the analytical form of the tensor elements is a simple product. Then, the decomposition network would be evaluated for each element of the full tensor $\eta'(s)$. Figure 5.9 illustrates where the TT-decomposition of the occupation factor integrates with the network for this term, which further increases the efficiency of evaluating the full term in a tensor network architecture.

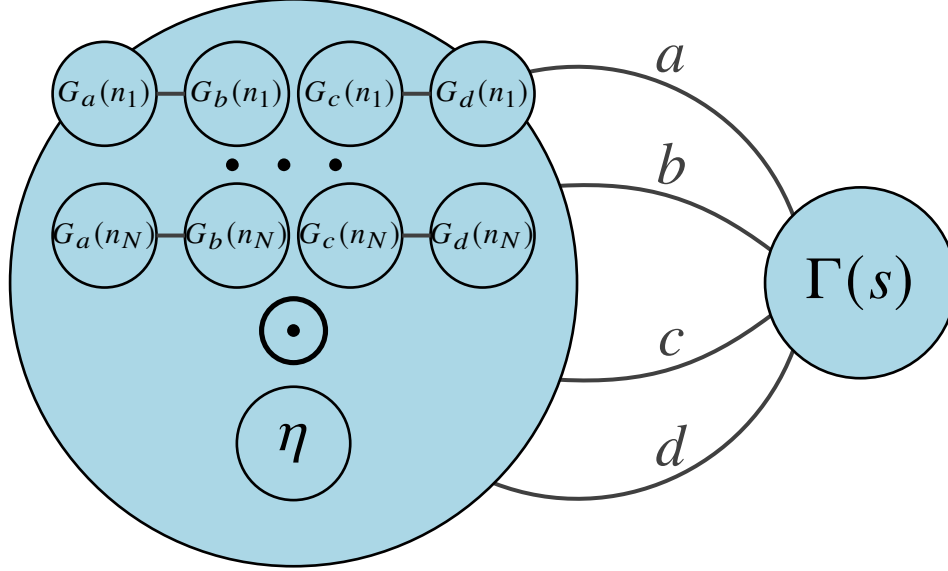


Figure 5.9 Diagram illustrating where the tensor train decomposition of the occupation factor $n_a n_b \bar{n}_c \bar{n}_d$ integrates with the tensor network to compute Equation 5.9. The elements of the occupation factor are generated via repeated evaluations of the corresponding TT-decomposition. Then, each element is multiplied elementwise with the tensor $\eta(s)$ to compute $\eta'(s)$. Finally, $\eta'(s)$ is fully contracted with $\Gamma(s)$ to complete the diagram.

As another example, let us consider the not-permuted piece of the two-body contribution to the one-body flowing operator $f(s)$, written (ignoring numerical factors)

$$\begin{aligned} \left(\frac{df_{pq}(s)}{ds} \right)^{(2)} &= \sum_{abc} (n_a n_b \bar{n}_c + \bar{n}_a \bar{n}_b n_c) \eta_{cpab}(s) \Gamma_{abcq}(s) \\ &= \sum_{abc} \eta'_{cpab}(s) \Gamma_{abcq}(s), \end{aligned} \quad (5.11)$$

where again we have absorbed the s -independent occupation factor into the factor corresponding to η' using elementwise multiplication. In this case, we wish to elementwise-multiply two tensors of different rank. This process amounts to multiplying the elements of the occupation factor $C^{(1)}(a, b, c)$ (refer to Appendix A) in the relevant index range of η_{cpab} , being careful to ensure that the information stored along axis p remains unchanged. We can broadcast $C^{(1)}(a, b, c)$ to a rank 4 tensor, with only identity information stored along the last axis corresponding to p , via a tensor product with the identity operator,

$$C^{(1)}(a, b, c, p) = C^{(1)}(a, b, c) \otimes \mathbb{I}. \quad (5.12)$$

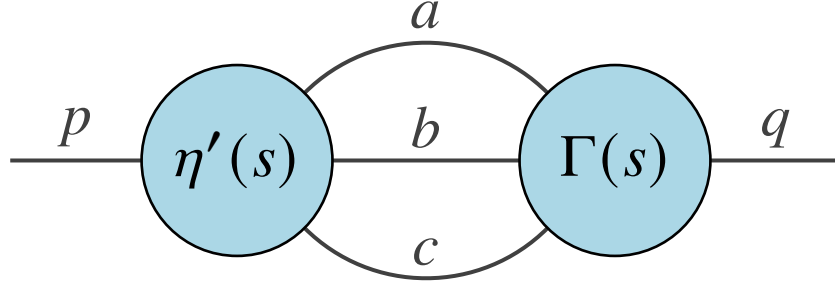


Figure 5.10 Tensor diagram corresponding to the two-body contribution to the one-body flowing operator in the IMSRG(2) flow equations in Equation 3.18.

Finally, we permute the indices of $C^{(1)}(a, b, c, p)$ to align with η_{cpab} and then the elementwise multiplication can be performed to arrive at η'_{cpab} . The tensor network corresponding to Equation 5.11 is captured in Figure 5.10. The dangling edges immediately show the result is a rank 2 tensor which is consistent with the required shape of the one-body operator flow equation.

5.5 Tensor-factors IMSRG: Python Implementation for the P3H Model

The code called Tensor-factors IMSRG, or `tfimsrg`, is a Python implementation of the IMSRG(2) (with developing support for IMSRG(3)) for the P3H model (see Chapter 4.1 and also Appendix C.1). The code was originally developed to demonstrate the feasibility of using Google’s TensorFlow tensor contraction API to express the IMSRG flow equations, interpreting the complicated sums as contractions between tensors. However, the project now uses the `ncon` API of the tensor contraction library TensorNetwork [9], which evaluates the contractions using the `numpy` backend. TensorNetwork provides several additional backends including JAX, TensorFlow, and PyTorch.

The codebase was designed as an importable Python package rather than a standalone executable. Its overall structure (excluding extraneous files and directories) is:

```
-- tfimsrg/
|-- main.py
-- oop_imsrg/
|-- flow.py
|-- generator.py
|-- hamiltonian.py
|-- occupation_tensors.py
```

The primary pieces of the IMSRG flow machine are organized into objects within the

`oop_imsrg/` directory. `Flow.py` computes the IMSRG flow equations in Equation 3.18 (as well as an implementation of the multi-reference IMSRG, see Appendix B). `Generator.py` computes the generators from Chapter 3.4. `Hamiltonian.py` computes the model Hamiltonian, the P3H model, from Chapter 4.1. `Occupation_tensors.py` pre-computes the TT-decomposed occupation factors from Appendix A. These components are bundled and passed to the `scipy` solver, which integrates the solution of the flow equations, in `main.py`.

5.5.1 Performance Study

The performance of the `tfimsrg` code, evaluating the P3H model, was measured relative to the naive implementation of IMSRG(2) in Python using matrix calculations introduced in Chapter 10 of Hjorth-Jensen, Lombardo, and Kolck [4]. Figure 5.11 shows the total wall time for several implementations of the IMSRG(2) code that (approximately) solve the P3H model with zero pair-breaking, including the matrix implementation, occupation factors constructed with loops, occupation factors constructed with contractions of TT-decomposed tensors (see Chapter 5.3 and Appendix A), and a TT-decomposition implementation exported to the GPU. Exporting the calculations to the GPU offers the most advantageous scaling in the context of these Python codes. All implementations that outsource tensor contractions in the IMSRG(2) flow equation to the TensorNetwork library perform faster than the naive implementations using matrix calculations. In Figure 5.12 we show that the tensor library implementation of the IMSRG(2) code captures the weak scaling feature of the single reference IMSRG, when particle and hole states are distinct. Note that the total number of single particle states $N = N_h + N_p$, where N_h is the number of hole states and N_p is the number of particle states. For large bases in realistic calculations, we have $N_p \approx N$. The performance results for this IMSRG(2) implementation are motivation to extend the tensor contraction strategy to a compiled language like C++, as well as modify the tensor contraction approach to take advantage of weak scaling. We want to also leverage this technology to the IMSRG production codes (Appendix C.1) which are written in C/C++.

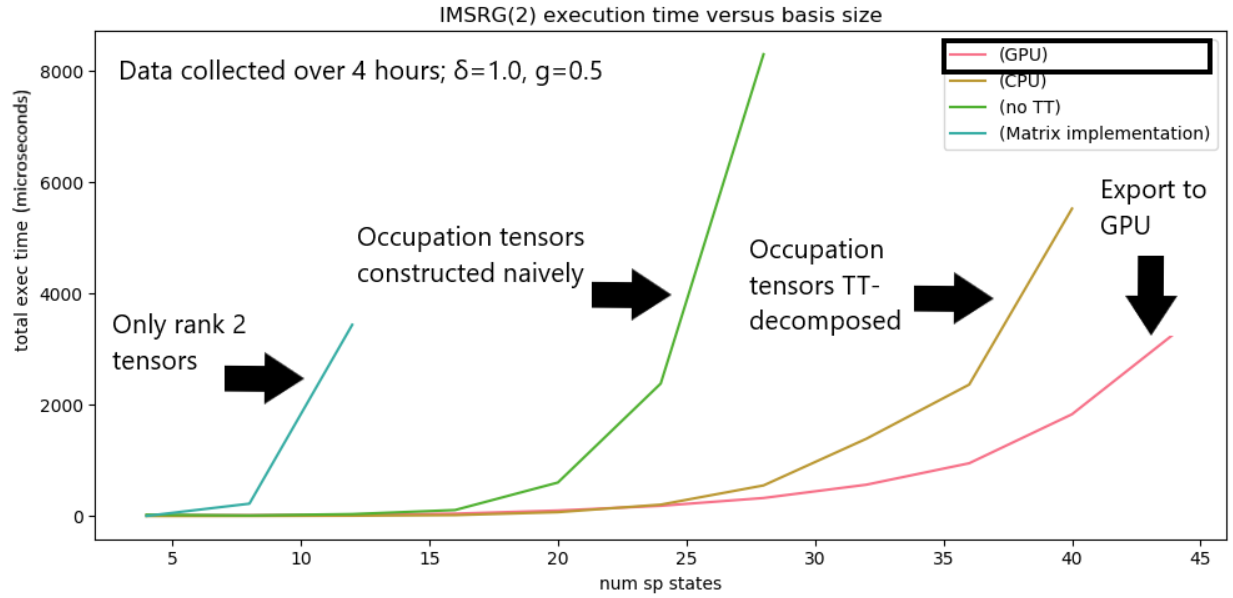


Figure 5.11 Total wall time versus the number of single particle states in the P3H model, excluding the pair-breaking term. Although not depicted here, the relative scaling for a P3H model with nonzero pair-breaking is not appreciably different. In general, the IMSRG(2) scales $O(N^6)$ with the number of single particle states N ; however using the TensorNetwork library allows us to soften the prefactor on that scaling to reach larger basis sizes.

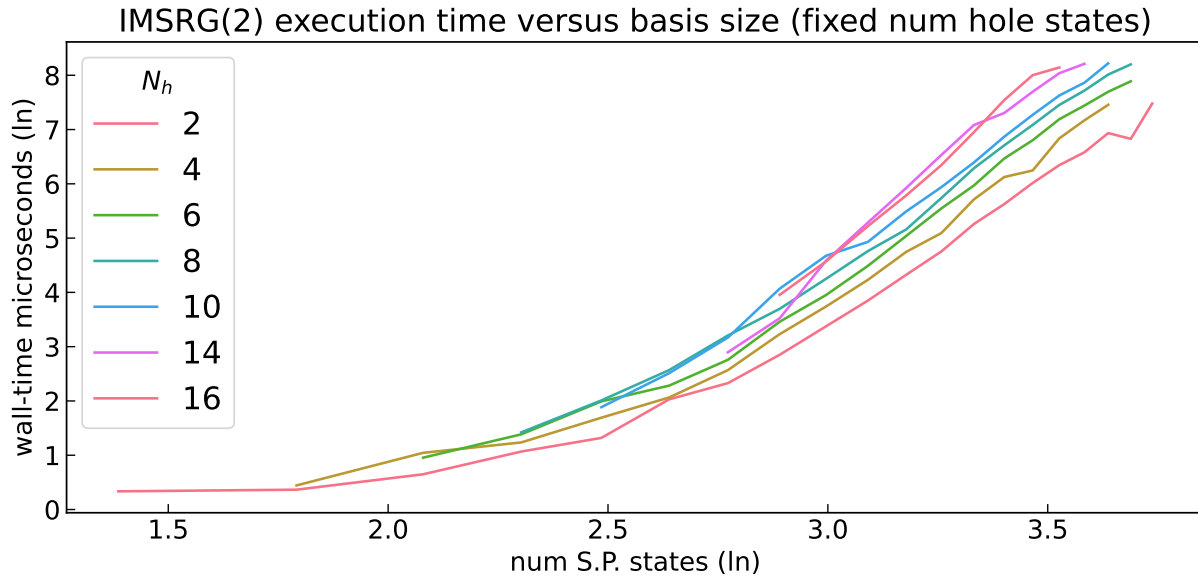


Figure 5.12 Total wall time versus the number of single particle states, for a *fixed* number of hole states. The P3H model parameters for input Hamiltonian are $\delta = 1.0, g = 0.5, b = 0.0$. Although not depicted here, the relative scaling for a P3H model with nonzero pair-breaking is not appreciably different.

5.6 Tensorized C++ IMSRG: C++ Implementation for the P3H Model

The code called Tensorized C++ IMSRG, or `tcimsrg`, is a C++ implementation of the IMSRG(2) for solving the P3H model (cf. Appendix C.1). This code is, more or less, a translation of `tfimsrg` from Python to C++. This code also includes more features for customizing the IMSRG(2) flow at execution time, including an option to use Dynamic Mode Decomposition (DMD) emulation of the IMSRG flow (see Chapter 7.4). This code was designed to run as an executable, with command line options similar to the IMSRG production code [3] (see also Appendix C.1). Again, we have taken an object-oriented approach to keep the code modular for future development and feature addition. The breakdown of objects is similar to `tfimsrg`, but we have tried to structure it according to best practices for C++:

```
-- tcimsrg/
  |-- CMakeLists.txt
  -- build/
    |-- solve_imsrg
  -- include/
    |-- BACKEND.hpp
    |-- DMD.h
    |-- flow_imsrg2.hpp
    |-- generator.hpp
    |-- pairinghamiltonian.hpp
    |-- occupation_factors.hpp
    |-- state_type.hpp
    |-- system.hpp
    |-- imsrg_utils.hpp
  -- src/
    |-- BACKEND_taco.cpp
    |-- BACKEND_ublas.cpp
    |-- DMD.c
    |-- flow_imsrg2.cpp
    |-- white.cpp
    |-- white_atan.cpp
    |-- brillouin.cpp
    |-- pairinghamiltonian.cpp
    |-- occupation_factors.cpp
    |-- system.cpp
    |-- main.cpp
```

The object headers are organized within the `include/` directory, and the object source files

which implement the instructions in their respective header files are organized under the `src/` directory. Running `cmake` within the `build/` directory will build the executable `solve_imsrg`. `BACKEND_taco.cpp` and `BACKEND_ublas.cpp` set up the IMSRG(2) flow equation evaluation using TACO and uBLAS loops, respectively. `DMD.c` sets up the DMD emulation. `Flow_imsrg2.cpp` sets up calculation the IMSRG(2) flow equation terms, based on the user’s specification of the backend. `White.cpp`, `white_atan.cpp`, and `brillouin.cpp` set up the generators specified in Chapter 3.4. `Pairinghamiltonian.cpp` sets up the P3H model Hamiltonian according to Chapter 4.1. `Occupation_tensors.cpp` pre-computes the TT-decomposed occupation factors from Appendix A. `System.cpp` organizes the ODE system which is passed to the Boost library’s `odeint` tool. Finally, the file `main.cpp` combines all relevant objects for computing a full IMSRG(2) calculation into the executable `solve_imsrg`.

5.6.1 Performance Study

Here we discuss the performance comparison between two different implemented calculation backends, naive loops using uBLAS vectors and contractions of tensors using the TACO library. The proceeding results include TT-decomposed occupation factors contracted directly in the network evaluation of the IMSRG(2) flow terms. Figure 5.13 shows the speedup of the TACO implementation versus the naive loops implementation. Again, the tensor library contraction combined with TT-decomposed occupation factors softens the prefactor on the IMSRG(2) computational scaling, allowing us to reach larger basis sizes. The results in Figure 5.13 reflect implementations that are not multithreaded; however, we expect the results would be similar provided both implementations are optimally multithreaded. In Figure 5.14, we show the wall-time associated with increasing single particle basis size, varying the fixed number of hole states (details as in Figure 5.12). Here, the results are consistent even across the number of fixed hole states, in contrast to the Python implementation. This is likely due to differing tensor library implementations. Again, the weak scaling is captured.

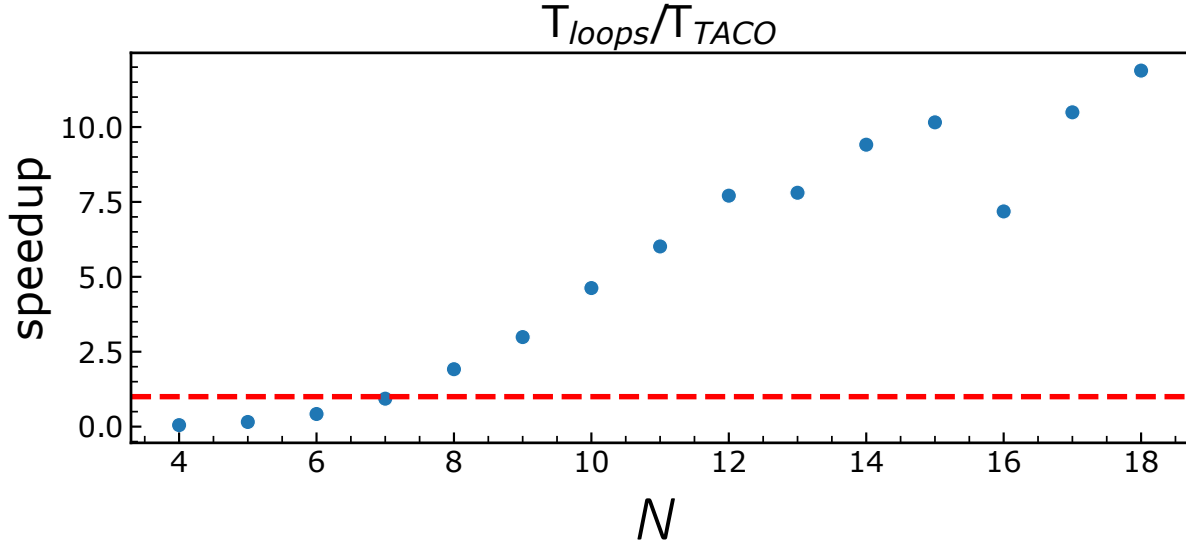


Figure 5.13 Associated speedup of TACO tensor contractions to the naive loop calculations, versus the number of single particle states in the P3H model. The red dotted line marks a ratio equal to 1, where the performance of the two implementations is equal.

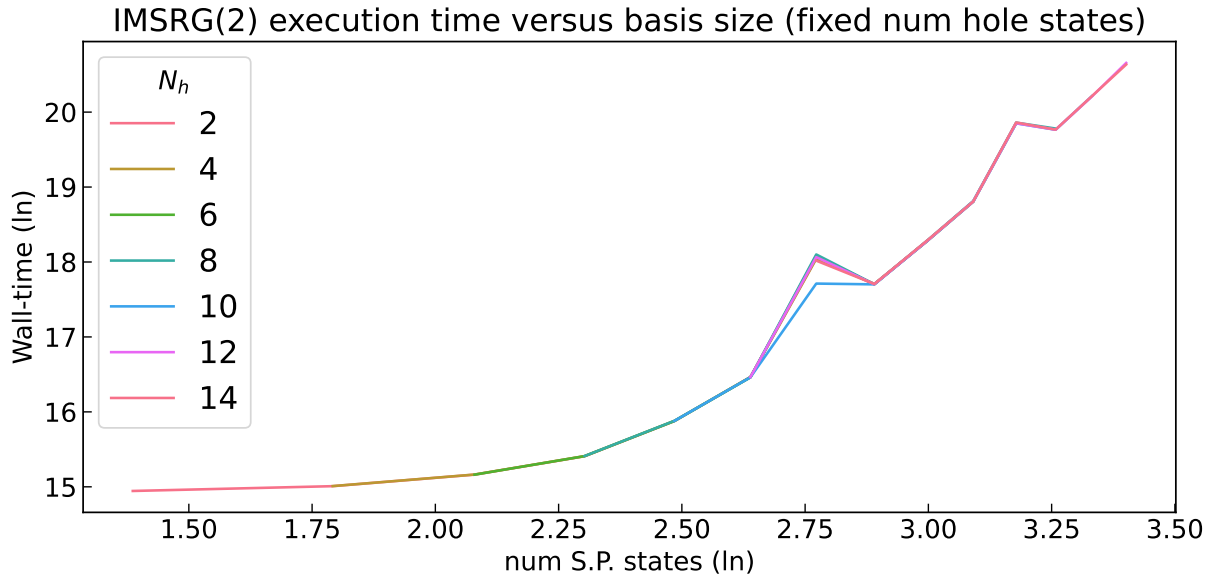


Figure 5.14 Details as in Figure 5.12, except as measured by the TACO C++ implementation. Note that wall-time is longer than the Python implementation due to differing integration strategies between the codes.

BIBLIOGRAPHY

- [1] Stephen R Clark. “Unifying neural-network quantum states and correlator product states via tensor networks”. In: *Journal of Physics A: Mathematical and Theoretical* 51.13 (Feb. 2018), p. 135301. ISSN: 1751-8121. DOI: 10.1088/1751-8121/aaaaf2. URL: <http://dx.doi.org/10.1088/1751-8121/aaaaf2>.
- [2] G. Evenbly and G. Vidal. “Tensor Network Renormalization”. In: *Physical Review Letters* 115.18 (Oct. 2015). ISSN: 1079-7114. DOI: 10.1103/physrevlett.115.180405. URL: <http://dx.doi.org/10.1103/PhysRevLett.115.180405>.
- [3] H. Hergert et al. “The In-Medium Similarity Renormalization Group: A novel ab initio method for nuclei”. In: *Physics Reports* 621 (Mar. 2016), pp. 165–222. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2015.12.007. URL: <http://dx.doi.org/10.1016/j.physrep.2015.12.007>.
- [4] Morten Hjorth-Jensen, Maria-Paola Lombardo, and Ubirajara van Kolck. *An advanced course in computational nuclear physics: bridging the scales from quarks to neutron stars*. Springer, 2017.
- [5] Fredrik Kjolstad et al. “The Tensor Algebra Compiler”. In: *Proc. ACM Program. Lang.* 1.OOPSLA (Oct. 2017), 77:1–77:29. ISSN: 2475-1421. DOI: 10.1145/3133901. URL: <http://doi.acm.org/10.1145/3133901>.
- [6] V. Murg et al. “Tree Tensor Network State with Variable Tensor Order: An Efficient Multireference Method for Strongly Correlated Systems”. In: *Journal of Chemical Theory and Computation* 11.3 (2015). PMID: 25844072, pp. 1027–1036. DOI: 10.1021/ct501187j. eprint: <https://doi.org/10.1021/ct501187j>. URL: <https://doi.org/10.1021/ct501187j>.
- [7] Román Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”. In: *Annals of Physics* 349 (Oct. 2014), pp. 117–158. ISSN: 0003-4916. DOI: 10.1016/j.aop.2014.06.013. URL: <http://dx.doi.org/10.1016/j.aop.2014.06.013>.
- [8] I. V. Oseledets. “Tensor-Train Decomposition”. In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317. DOI: 10.1137/090752286. eprint: <https://doi.org/10.1137/090752286>. URL: <https://doi.org/10.1137/090752286>.
- [9] Chase Roberts et al. *TensorNetwork: A Library for Physics and Machine Learning*. 2019. arXiv: 1905.01330 [physics.comp-ph].
- [10] Anton Rodomanov. *Introduction to the Tensor Train Decomposition and Its Applications in Machine Learning*. HSE Seminar on Applied Linear Algebra. Mar. 2016. URL: https://bayesgroup.github.io/team/arodomanov/tt_hse16_slides.pdf.

- [11] I. Shavitt and R.J. Bartlett. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge Molecular Science. Cambridge University Press, 2009. ISBN: 9780521818322. URL: <https://books.google.com/books?id=SWw6ac1NHZYC>.
- [12] Szilárd Szalay et al. “Tensor product methods and entanglement optimization for ab initio quantum chemistry”. In: *International Journal of Quantum Chemistry* 115.19 (2015), pp. 1342–1391. DOI: 10.1002/qua.24898. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.24898>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.24898>.
- [13] A. Tichai, J. Ripoché, and T. Duguet. “Pre-processing the nuclear many-body problem”. In: *The European Physical Journal A* 55.6 (June 2019). ISSN: 1434-601X. DOI: 10.1140/epja/i2019-12758-6. URL: <http://dx.doi.org/10.1140/epja/i2019-12758-6>.
- [14] A. Tichai et al. “Tensor-decomposition techniques for ab initio nuclear structure calculations: From chiral nuclear potentials to ground-state energies”. In: *Physical Review C* 99.3 (Mar. 2019). ISSN: 2469-9993. DOI: 10.1103/physrevc.99.034320. URL: <http://dx.doi.org/10.1103/PhysRevC.99.034320>.
- [15] B. Zhu, R. Wirth, and H. Hergert. “Singular value decomposition and similarity renormalization group evolution of nuclear interactions”. In: *Physical Review C* 104.4 (Oct. 2021). DOI: 10.1103/physrevc.104.044002. URL: <https://doi.org/10.1103/5C%2Fphysrevc.104.044002>.

CHAPTER 6

OPTIMAL REFERENCE STATE ENSEMBLES FOR THE IN-MEDIUM SIMILARITY RENORMALIZATION GROUP

6.1 Introduction

In this chapter, we focus on the main methodological improvement to the IMSRG framework that is developed in this thesis. In general, the implementation of any RG method begins with the choice of an operator basis. If we can work in a complete basis, we can capture the RG flow exactly (for example, SRG for finite-size matrices). In most applications, and in particular the IMSRG(2), we only work with truncated bases due to the complexity of tracking induced operators. For example, in the IMSRG(2) we work with a finite set of normal ordered, second-quantized operators.

The RG flow comes with built in diagnostics—it can tell us which operators are relevant or irrelevant, and stabilities or divergences can be a sign of truncations that are too severe, or truncations that violate symmetries we would like or need to preserve. The response of the RG flow to the choice of the operator basis impacts the accuracy with which we can describe the observables of interest.

Here, we will explore the use of ensembles of reference states to enhance the robustness of the IMSRG operator basis, and to control truncation errors. We show that the use of an ensemble serves to “inform” the IMSRG flow about the excited states of the system, so that the RG improvement of the Hamiltonian is not merely targeted to the description of the ground state alone. Similar ideas have been successfully explored in the context of quantum chemistry [4, 7].

6.2 Truncation Error in the IMSRG

In the IMSRG, the unitarity of the transformation provides a key diagnostic, which is that observables from exact many-body calculations should be invariant. We have shown in Equation 2.30 that the product of two normal-ordered M -body and N -body operators produces a sum of up to $M + N$ -body operators. The commutator in the IMSRG flow equation, Equation 3.4, produces a

flowing operator of total rank $\text{rank}(\eta) + \text{rank}(H) - 1$. In the IMSRG(2) truncation, we disregard operators for three or more particles which are induced by the commutator. The error in this truncation scheme manifests as a violation of unitarity in the IMSRG transformation, which is evident from the non-preservation of the flowing eigenvalue spectrum that we would obtain in an exact diagonalization. In cases where this loss of unitarity is especially egregious, the assumption is that the terms we truncated from the operator basis are important for maintaining a stable IMSRG flow for that system configuration.

Increasing the truncation level to keep three and higher A -body operators is not feasible in realistic applications of the IMSRG, because the memory requirements ($O(N^4)$ in the IMSRG(2) versus $O(N^6)$ in the IMSRG(3)) and computational costs ($O(N^6)$ in the IMSRG(2) versus $O(N^9)$ in the IMSRG(3)) become prohibitive. The solution we propose in this Chapter is to recover physical information lost due to truncation by normal-ordering the IMSRG Hamiltonian with respect to an ensemble of Slater determinant reference states, thereby tuning the chosen operator basis in which we implement the IMSRG(2) flow. We will show that the impact on the cost of the IMSRG(2) flow is minimal. There is a non-negligible cost to the “mixing” optimization procedure, but we will show that improvements to the IMSRG result can still be obtained using less-than-optimal mixing parameters.

6.3 Formulation of the Reference State Ensemble

In the reference state ensemble scheme, the many-body Hamiltonian is normal-ordered with respect to an *ensemble* of orthonormal Slater determinant configurations, $\{\Phi_1, \dots, \Phi_P\}$. We choose a reference state ψ , which is a mixture of the Slater determinant configurations such that

$$|\psi\rangle = \sum_{i=1}^P c_i |\Phi_i\rangle. \quad (6.1)$$

The density matrix $\hat{\rho}$ (dropping explicit indices for brevity) is

$$\hat{\rho} = |\psi\rangle \langle\psi| = \sum_i c_i^* c_i |\Phi_i\rangle \langle\Phi_i| = \sum_i p_i |\Phi_i\rangle \langle\Phi_i|. \quad (6.2)$$

The trace $\text{tr } \hat{\rho}$ is simply the sum of the diagonal terms, i.e.

$$\text{tr } \hat{\rho} = \sum_i p_i \langle \Phi_i | \Phi_i \rangle = \sum_i p_i \quad (6.3)$$

Expectation values of operators O can be computed as traces in the density matrix, $\text{tr } \hat{\rho} O$. Consider the expectation value of operator pair $a_p^\dagger a_q$, which generates the one-body density matrix, in ψ :

$$\begin{aligned} \langle \psi | a_p^\dagger a_q | \psi \rangle &= \text{tr } \hat{\rho} a_p^\dagger a_q = \text{tr } \left(\sum_i p_i |\Phi_i\rangle \langle \Phi_i| a_p^\dagger a_q \right) \\ &= \sum_i p_i \langle \Phi_i | a_p^\dagger a_q | \Phi_i \rangle \\ &= \sum_i p_i \rho_i, \end{aligned} \quad (6.4)$$

where the factor ρ_i is the one-body density matrix corresponding to reference state configuration $|\Phi_i\rangle$. Thus, we have shown that the *reference state ensemble* is a simple sum of one-body density matrices, in a basis of Slater determinants states, with terms weighted by the probability in which they might accurately represent the state targeted for decoupling by the IMSRG (e.g., the ground state). Note that for a “pure”, correlated state $|\Psi\rangle$, the density operator $\hat{\rho} = |\Psi\rangle \langle \Psi|$ would generate additional terms in the expectation value that are missing in the “mixed” state ψ .

In this formulation, the occupation of a particular state in the reference ensemble becomes fractional (according to p_i) instead of an integer in the single Slater determinant case as in Equation 2.27. This implies that the two-body and three-body density matrices will not factor completely into one-body density matrices, which generates nonzero IDMs in Equations 2.34 and 2.35.

In our reference ensemble scheme, the irreducible terms do not encode pure correlations in a quantum mechanical sense, because they do not result from the “quantum interference” terms that are present within a pure wave function density operator. Since the reference ensemble consists of a simple mixture of Slater determinants, we may interpret the irreducible terms as providing “statistical” correlation information, in a effort to simulate a truly correlated reference state.

Concretely, the one-body density matrix corresponding to the reference state ensemble reads

$$\boldsymbol{\rho}^{(E)} = \sum_i^P p_i \boldsymbol{\rho}_i, \quad (6.5)$$

where P is the total number of probabilities in which we expand the reference, and each ρ_i corresponds to the irreducible one-body density matrix for a particular Slater determinant configuration, which is fixed by the number of hole and particle states in the single particle basis. The total number of Slater determinant configurations available to the ensemble is given by the binomial coefficient $P = C(N, N_h)$, where N is the number of single particle states and N_h is the number of hole states. States which are “excluded” from the ensemble correspond to weights $p_i = 0$. In order for particle number to be conserved, we assert that the probabilities are normalized such that

$$\sum_i^P p_i = 1. \quad (6.6)$$

The reference state ensemble adds two additional degrees of freedom to the IMSRG flow, namely the size of the ensemble and the distribution of probabilities across the ensemble. Thus, we may optimize the ensemble size and weighting scheme in order to minimize the truncation error of the IMSRG(2).

Truncation error is difficult to measure without an exact method to compare with the IMSRG(2) result. Fortunately, the exact spectrum is accessible for the P3H Hamiltonian (see Chapter 4.1), which we employ to study the efficacy of the reference state ensemble.

6.4 Ensemble Optimization Scheme

As a proof-of-concept, we demonstrate one technique which can be used to optimize the probability weights that parameterize the reference state ensemble when an exact method is tractable for comparison. In Algorithm 6.1, we define a residual as the difference between the exact eigenspectrum and the IMSRG(2) eigenspectrum, computed from a particular reference state ensemble characterized by the probability weight vector $\mathbf{p}$. We perform a least-squares minimization of this residual under variations of $\mathbf{p}$. Of course, this optimization scheme is not practical in a realistic application, where the IMSRG(2) may be expensive to complete (IMSRG emulator techniques may assist in reducing the expense, see Chapter 7.3) or the exact eigenspectrum may not be accessible due to the exponential cost of determining it.

Algorithm 6.1 Pseudocode implementation which demonstrates the minimization process, for which the residual involves a complete IMSRG(2) flow as well as a full configuration interaction (FCI) computation to obtain the eigenvalues of the vacuum Hamiltonian.

```

function RESIDUAL(p, y, M, m)
    p  $\leftarrow$  p/sum(p)                                 $\triangleright$  enforce constraint  $\sum_i p_i = 1$ 
    z = Mp                                              $\triangleright$  compute reference state
    x =  $\mathcal{R}(\mathbf{z}, \delta, g, b)$                         $\triangleright$  compute eigenvalues from  $H(\infty)$ 
    return x[1 : m] – y[1 : m]
end function
initialize reference ensemble matrix M
initialize p
y =  $\mathcal{F}(\delta, g, b)$                                  $\triangleright$  compute eigenvalues from FCI
repeat p  $\leftarrow$  lsq (residual(p, y, m))
until residual(p, y, M, m) < tolerance

```

6.5 Regularization of Generator Divergences

Let us consider the IMSRG(2) flow of a P3H Hamiltonian (see Chapter 4.1) with a strong pairing strength, $g = 2.0$. Let $b = 0.0$. Figure 6.1 illustrates an IMSRG(2) flow of this Hamiltonian. The diagonal matrix elements, which should converge to the true eigenspectrum as the IMSRG(2) flows, diverge from this destination and cross each other unpredictably. This is an issue of the White generator for states which are close to the Fermi surface; the generator, which depends on denominators of energy differences, blows up uncontrollably when the energy differences are small (see Chapter 3.4.2). We note that although this behavior is most obvious with the White generator because of the explicit energy denominators, this issue may persist in other generator forms due to the truncation of induced terms. In this example, the pairing strength squeezes the energy states close together such that the IMSRG(2) flow cannot sufficiently decouple the target ground state from excited states, and the flow fails.

In the proceeding results, we attempt to correct this failure by inserting information about the excitations of the system into the IMSRG(2) through the use of an ensemble of Slater determinants. While the use of fractional occupation factors implies that two-body and higher irreducible density matrices (IDMs) introduced in Chapter 2.6 are nonzero, as discussed above, we truncate terms containing the IDMs for now, and refer to this ensemble evaluation scheme as fractional occupation only, or FOO.

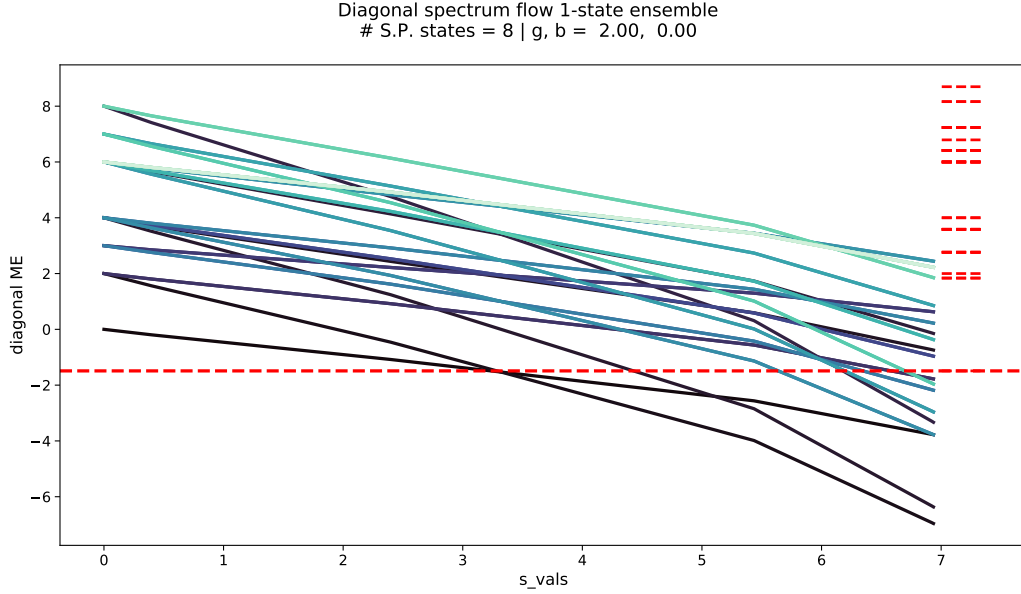


Figure 6.1 Diagonal matrix elements of a flowing P3H Hamiltonian with 8 single particle states. The dashed horizontal lines to the right represent the true eigenspectrum for this Hamiltonian, obtained through exact diagonalization (see Chapter 2.10). The longest dashed line represents the true ground state.

Figure 6.2 demonstrates the effect of the FOO ensemble scheme: We are able to cure the divergent behavior in the IMSRG(2) flow for strong pairing $g = 2.0$, and the ground state energy converges to the desired result; in addition, many of the unpredictable crossings in excited states have been mitigated. One disadvantage we can immediately note from this process is that the dynamical range has doubled compared to Figure 6.1, meaning that more computational effort is required to achieve the same result (without mentioning the computational effort required for optimization).

The spectrum of the flowing Hamiltonian in the FOO scheme is not necessarily invariant under changes of s (and unitarity is still violated). However, the extent of the unitarity violation is improved compared to the single reference case. Figure 6.3 compares the spectrum evolution according to the single reference flow versus the FOO scheme flow.

6.6 Improving the Eigenspectrum

In this section, we demonstrate the complexity in improving all eigenvalues and eigenstates of the IMSRG(2) evolved Hamiltonian simultaneously, using the FOO ensemble scheme from

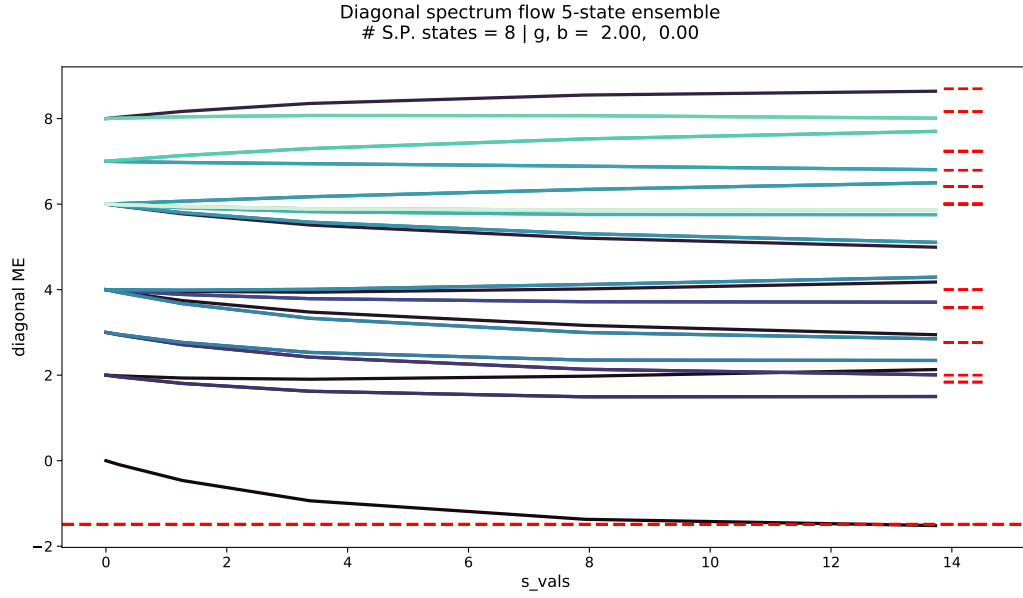


Figure 6.2 Diagonal matrix elements of a flowing P3H Hamiltonian with 8 single particle states. In contrast to Figure 6.1, the Hamiltonian has been normal-ordered with respect to a 5-state optimized reference ensemble.

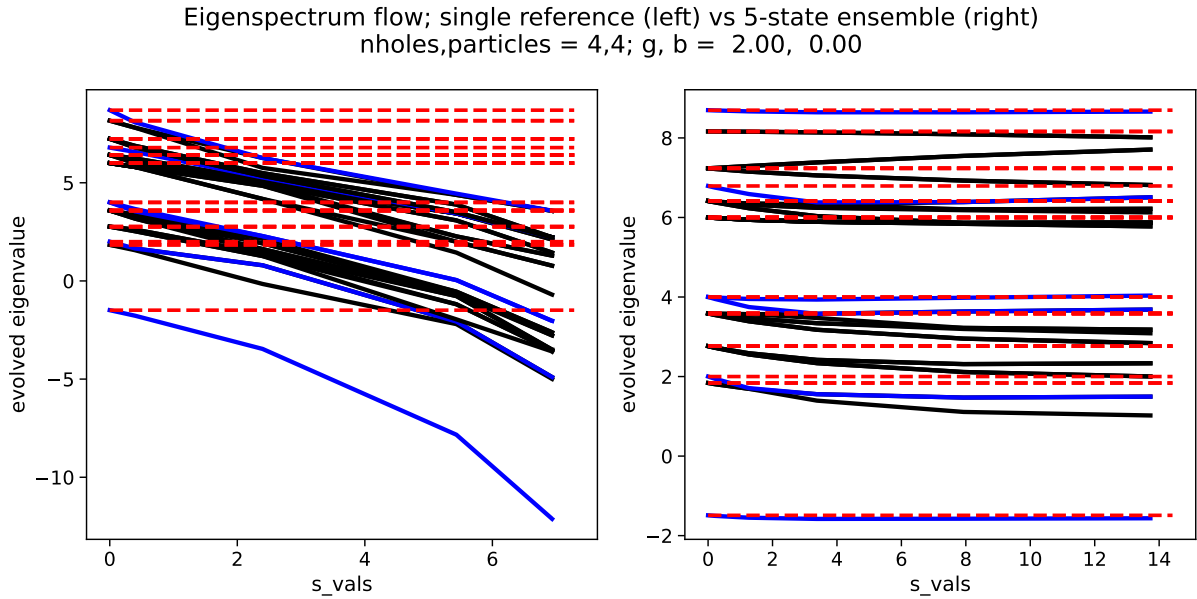


Figure 6.3 Comparison of the flowing spectrum for a single reference flow to a 5-state FOO scheme flow. The red dashed lines represent the true spectrum for this particular P3H Hamiltonian. Blue lines correspond to paired eigenstates. The spectrum is not invariant under IMSRG(2) evolution in either case, but the FOO scheme flow preserves more of the original spectrum.

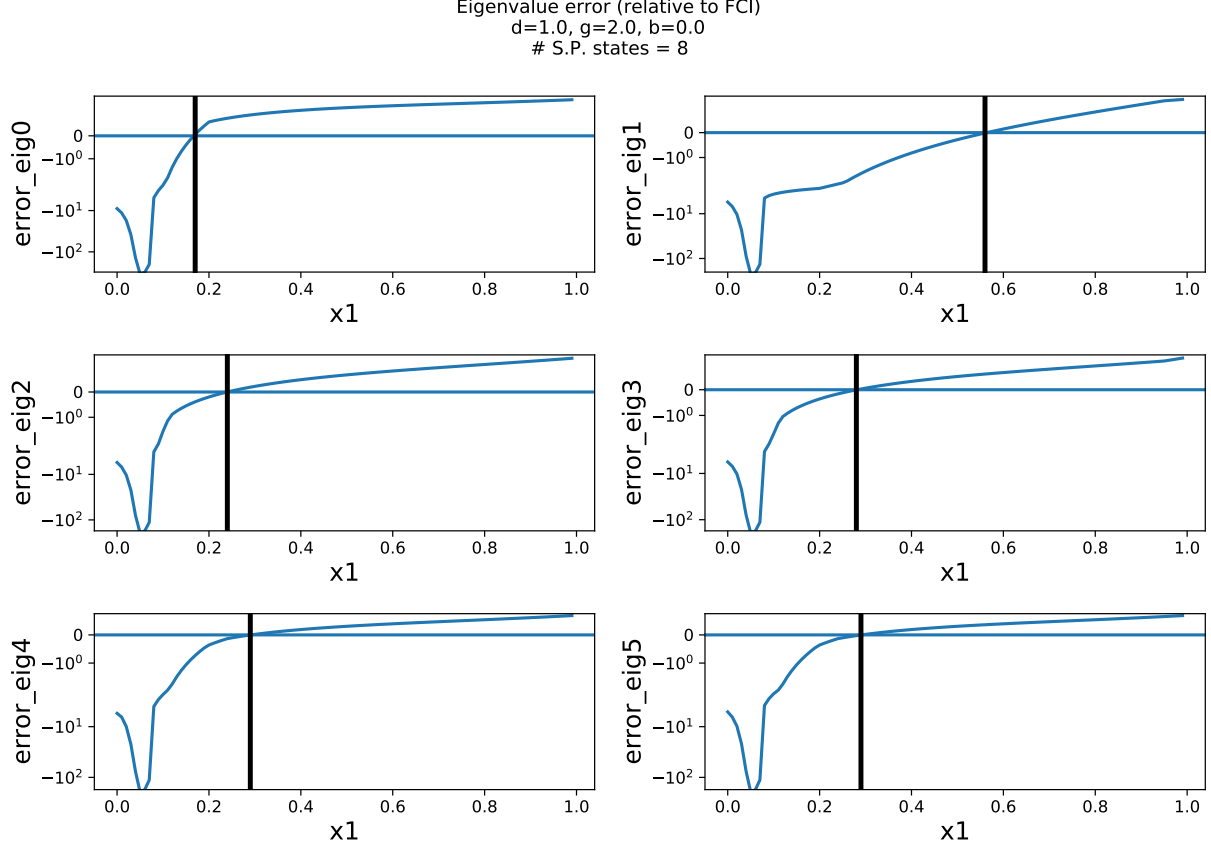


Figure 6.4 The vertical axis is the difference between the IMSRG(2) energy and the FCI energy for each eigenstate. The horizontal axis x_1 is the mixing strength, which controls the 2-state ensemble dependence on the first excited fully paired SD configuration.

the previous section. “Improvement” in the spectrum is interpreted as the preservation of the eigenvalues throughout the flow, which assesses the unitarity of the IMSRG transformation.

We have shown in Figure 6.2 that the reference state ensemble scheme “regularizes” the flow of matrix elements; compared to the extreme case in Figure 6.1, this accounts for an improvement in preservation of the eigenspectrum (especially, the ground state) which we observed in Figure 6.3.

Let us systematically study how a 2-state FOO ensemble changes the eigenspectrum preservation for 6 fully paired¹ eigenstates of the IMSRG(2) evolved Hamiltonian. The 2-state FOO ensemble, in this case, reads

$$\rho^{(E-2)} = (1 - x_1)\rho_0 + x_1\rho_1, \quad (6.7)$$

¹For the P3H Hamiltonian in question, the pair-breaking contribution is zero. Thus, we worry only about the accuracy of the eigenstates accessed by the pairing contribution.

where x_1 is a fraction which controls the proportion of the ground state SD configuration compared to the first excited state SD configuration. Note that when $x_1 = 0$, we recover the usual single SD reference state. In the FOO ensemble scheme, the weights $(1 - x_1), x_1$ directly affect the fractional values in the occupation factors, and nothing else. Figure 6.4 shows the error curves for each fully paired eigenstate, for the same Hamiltonian as in Figures 6.1 and 6.2 using the 2-state reference ensemble. The solid horizontal line in each plot denotes the zero crossing, where the error in the energy is zero.

These plots demonstrate that, for a single mixing parameter, there is no mixing for which the total error in the IMSRG(2) eigenspectrum is zero. This result suggests that we must make concessions based on the goal of the reference state ensemble scheme; tuning the ensemble to improve a single eigenstate, e.g. the ground state, does not necessarily guarantee all eigenstates will be improved directly proportionally. In the 2-state ensemble, there is maybe no good reason to expect improvements for high-lying excited states if we only use ground-state and first-excited-state Slater determinant configurations. Seeking improvement in high-lying excitations motivates the use of more basis configurations in the ensemble, at the cost of a more complicated multi-parameter optimization problem.

Figure 6.7 also displays a sweeping range of control over error in every IMSRG(2) eigenstate for only a single mixing parameter in a 2-state ensemble. The 2-state ensemble begins to see improvement over the single reference beyond around $x_1 = 0.1$, before which we observe a steep increase in error around $x_1 = 0.08$. In some way, this singularity reveals the topology of the optimization problem. The steep increase shows how much we must weight the addition of the excited state in order to decouple the ensemble result from the single reference result, such that the spectrum no longer diverges. These results suggest that we may be able to alleviate particular crossings with a simple 2-state ensemble.

6.7 Ensemble Consistency with Non-Targeted Observables

In the previous sections, we have shown that reference ensembles are capable of improving the IMSRG(2) flow of the Hamiltonian—in other words, the reference ensemble can reduce the error

in energy eigenvalues, or even improve the entire spectrum robustly. However, most of the time, we wish to also compute the expectation values of observables other than the energies, hence we must ensure that the use of the reference state ensemble does not spoil their description. We may evolve any additional observable O alongside the Hamiltonian according to Equation 3.4 [2],

$$\frac{dO(s)}{ds} = [\eta(s), O(s)], \quad (6.8)$$

where $\eta(s)$ is the IMSRG generator constructed from the Hamiltonian. Since the space for storing the full flow history of η is prohibitive in realistic applications, we must concatenate the operator flow equations in the same normal-ordered operator basis as the Hamiltonian to the Hamiltonian ODE system. Concatenation of each additional operator implies the addition of another full-size version of the ODE system, which restricts the simultaneous evolution to only a couple of observables in practice. One solution for this issue is the Magnus [5, 1] formulation of the IMSRG demonstrated by Morris, Parzuchowski, and Bogner [6] (also see Hergert et al. [2] and Chapter 10 of Hjorth-Jensen, Lombardo, and Kolck [3]). The Magnus method casts the unitary transformation to a true exponential, $U(s) = e^{\Omega(s)}$, and then reformulates the flow equations to evolve the Magnus operator $\Omega(s)$ instead of the Hamiltonian itself. Future work will implement in the reference state ensemble in the Magnus-IMSRG, but has not been implemented for the results in this study.

One disadvantage of the rather abstract P3H toy model is the lack of compelling observables with which to gather relevant data. The most interesting candidate beside H itself is the total spin-squared operator $\hat{S}^2$. We will explore how the use of a reference state ensemble that is optimized for H impacts the expectation value $\langle S^2 \rangle$; in this way, we show that the reference ensemble flow is “safe” in that important physical information in simultaneous observables is not destroyed.

The second-quantized spin-squared operator $\hat{S}^2$ reads

$$\begin{aligned} \hat{S}^2 = & \sum_{pq} \langle p | \hat{S}^2 | q \rangle a_p^\dagger a_q \\ & + \frac{1}{4} \sum_{pqrs} \langle pq | \hat{S}^{(1)} \cdot \hat{S}^{(2)} | rs \rangle_{\mathcal{A}} a_p^\dagger a_q^\dagger a_s a_r. \end{aligned} \quad (6.9)$$

The $\hat{S}^2$ matrix elements are constructed in the single particle basis states of the P3H model, with quantum numbers p , which denotes the energy level, and σ_p , which denotes the spin direction of

the particle at level p . The one-body matrix elements of $\hat{S}^2$, in units of $\hbar^2$, read

$$\langle (p, \sigma_p) | \hat{S}^2 | (q, \sigma_q) \rangle = \frac{3}{4} \delta_{pq} \delta_{\sigma_p \sigma_q}. \quad (6.10)$$

Using ladder operators $\hat{s}_-$ and $\hat{s}_+$, the two-body part of the operator can be expressed as

$$\hat{s}^{(1)} \cdot \hat{s}^{(2)} = \frac{1}{2} \left(\hat{s}_+^{(1)} \hat{s}_-^{(2)} + \hat{s}_-^{(1)} \hat{s}_+^{(2)} \right) + \hat{s}_z^{(1)} \hat{s}_z^{(2)}. \quad (6.11)$$

The ladder operator terms in Equation 6.11 correspond to a mutual “spin-flipping” interaction, whereas the spin projection $\hat{s}_z^{(1)} \hat{s}_z^{(2)}$ term is structurally similar to the one-body matrix element.

Introducing the two-body antisymmetrized product state

$$|(p, \sigma_p)(q, \sigma_q)\rangle_{\mathcal{A}} = \frac{1}{\sqrt{2}} (|(p, \sigma_p)(q, \sigma_q)\rangle - |(q, \sigma_q)(p, \sigma_p)\rangle), \quad (6.12)$$

we can compute the two-body matrix element corresponding to the $\hat{s}^{(1)} \cdot \hat{s}^{(2)}$ term in Equation 6.9.

In Figure 6.5 we present the evolution of the expectation value $\langle S^2 \rangle$ in the ground eigenstate of the Hamiltonian, compared to evolution of the ground state energy. The vertical axis measures the observables’ difference from the starting point, so that energy and $\langle S^2 \rangle$ can be plotted to the same scale. The horizontal axis represents the s evolution scale. In panels *A* and *B*, we have normal-ordered both operators with respect to the single, standard SD ground state configuration. Note that in these cases, the $\hat{S}^2$ matrix maps to zero because the total spin is zero for the SD ground state configuration. In this way, $\langle S^2 \rangle$ is trivially conserved. The energy behaves as expected, in that the strongly interacting Hamiltonian in panel *B* results in a divergent flow.

In panel *C*, we have normal-ordered both operators with respect to an optimized ensemble. Now that there are $S \neq 0$ SD configurations mixed into the ensemble, we see that $\langle S^2 \rangle$ is no longer conserved. However, we see that the evolution of $\langle S^2 \rangle$ closely resembles that of the energy, meaning that the character of the $\hat{S}^2$ has been preserved under the reference ensemble flow. This study suggests that the reference state ensemble is safely preserves simultaneous observables in the IMSRG flow.

6.8 Using Exact Eigenstates as References

We conclude the discussion in this chapter by considering the impact of using an exact eigenstate of H for setting up the normal ordered operator basis for the IMSRG. In this case, we expect the

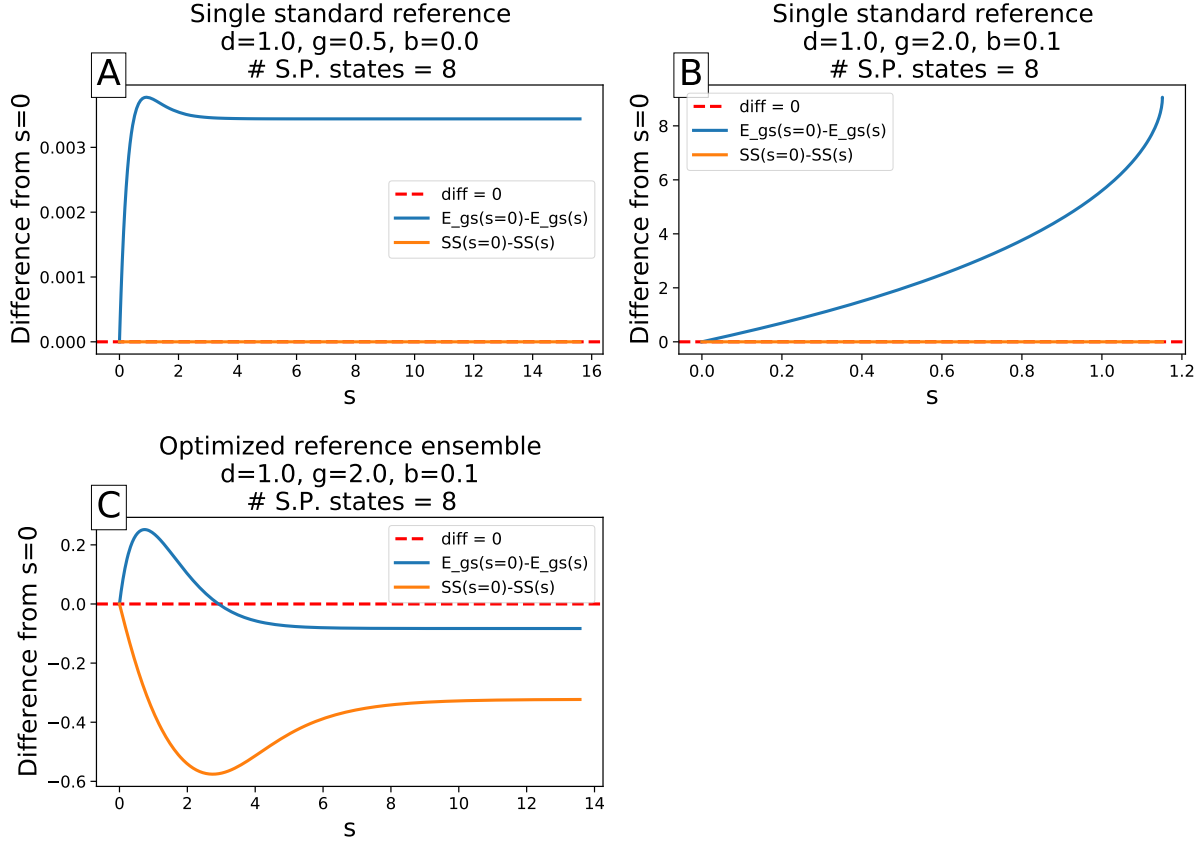


Figure 6.5 Evolution of $\langle S^2 \rangle$ and the ground state energy $E(s)$ for two P3H model Hamiltonians and two reference schemes. The vertical axis measures the change in the zero-body part of the evolving operator from the starting point $s = 0$, while the horizontal axis plots the s evolution range. Panel A features a weakly interacting P3H Hamiltonian where energy converges, according to a single standard ground state SD configuration reference. Panel B features a strongly interacting P3H Hamiltonian where energy diverges, according to a single standard ground state SD configuration reference. Panel C features a strongly interacting P3H Hamiltonian where energy converges, according to an optimized ensemble reference.

zero-body part of the operator to be identical to the eigenvalue associated with the reference state, and it should be completely insensitive to the IMSRG truncation.

Since the eigenstate is correlated, the IDMs will be non-zero, and we will perform the evolution in the MR-IMSRG(2) scheme (see Appendix B). We will use generalized form of the White-ArcTan generator (Chapter 3.4.2), as well as the Brillouin generator (Chapter 3.4.3), to generate the flow.

In Figure 6.6, we show the eigenvalue flow corresponding to the lowest four eigenstates of a weak-pairing P3H Hamiltonian and strong-pairing/weak-pair-breaking P3H Hamiltonian. Note that in both cases, the first and second excited states are degenerate in energy. The color coding indicates which exact eigenstate was used to normal order H . In the weak-pairing case, we find that all eigenstates remain invariant under evolution if we use the associated occupations and IDMs for constructing the normal ordered operator basis, and we find the expected lack of sensitivity to the MR-IMSRG truncation. Interestingly, the (fully paired) ground state remains invariant *regardless* of the exact eigenstate we use for the normal ordering. The third excited state, which is fully paired as well, also remains invariant if we use any of the excited states to construct the normal ordered operator basis, while a basis built from the ground state is not able to capture the evolution of this state completely. For the first and second excited states, evolving with the fully paired references only leads to minor deviations from the exact eigenvalue.

In the strong-pairing/weak-pair-breaking case, we find that the flow breaks down for all cases except in the case where Hamiltonian is normal-ordered with respect to the ground state. This suggests that for strongly interacting systems, a single exact eigenstate does not provide enough information to account for these induced terms which have been truncated. Note especially the “closeness” of the second and third energies in the strongly interacting P3H Hamiltonian—the eigenvalues in these state seem to be switching places in a level crossing.

In Figure 6.7, we systematically study how truncations of terms depending on the IDMs affect the unitarity of the ground state eigenvalue’s evolution. Here, we examine the flow generated by the Brillouin generator and normal order with respect to the exact ground state. In both cases, the inclusion of the fractional occupation numbers alone yields a converged flow, but the deviation

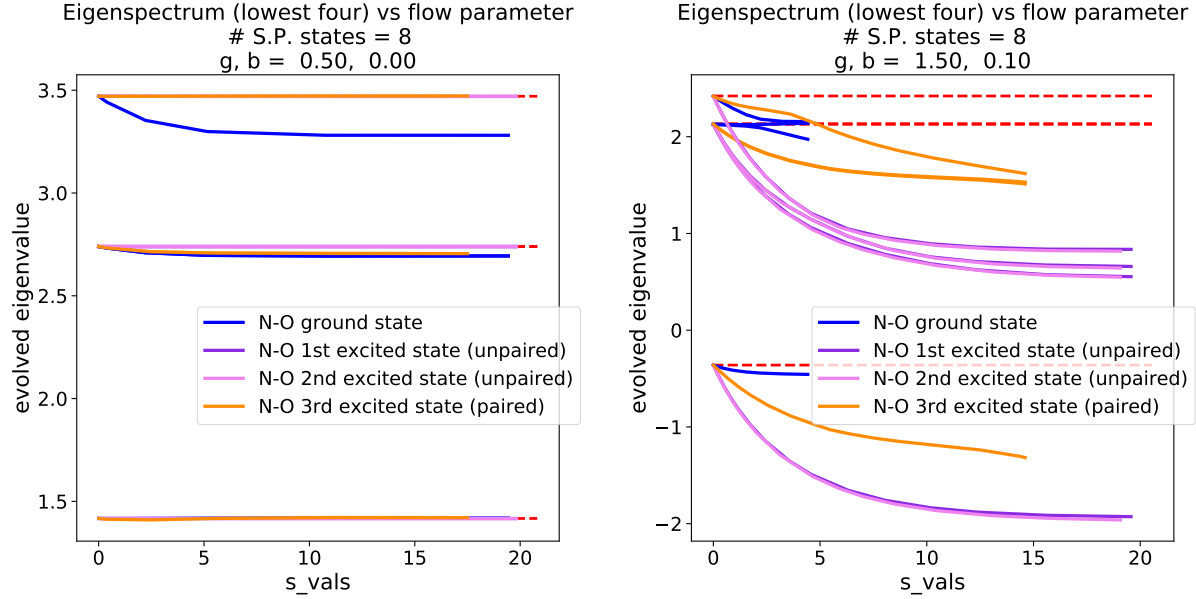


Figure 6.6 Evolution of the lowest four eigenvalues in two P3H models, varying the exact eigenstate against which the Hamiltonian has been normal-ordered. The flow was evaluated using a White-ArcTan generator. In the left pane, we evolve a weak pairing Hamiltonian with no pair-breaking. In the right pane, we evolve a strong pairing Hamiltonian with weak pair-breaking.

from the exact eigenvalue can be substantial, and in the strongly interacting case it is difficult to tell whether the evolution is completely leveled off. In the weakly-interacting case, inclusion of the two-body IDM in the Brillouin generator greatly reduces the violation of unitarity, and inclusion of the three-body IDM guarantees the invariance of the eigenvalue.

In the strongly-interacting case, the inclusion of the two-body IDM terms on top of the fractional occupation numbers actually causes the flow to “diverge”—since the Brillouin generator is effectively the gradient of the energy under unitary variations, a likely explanation is that we overshoot the stationary point because the gradient is unbalanced. The inclusion of the three-body IDM fixes this issue, hence the takeaway is that IDM truncations in the generator need to be carefully considered. Since the IDMs encode correlations in the reference state that are coupling to the correlations treated by the MR-IMSRG during the evolution, it is not surprising that the balance of the two- and three-body IDMs is more delicate in the strongly interacting case.

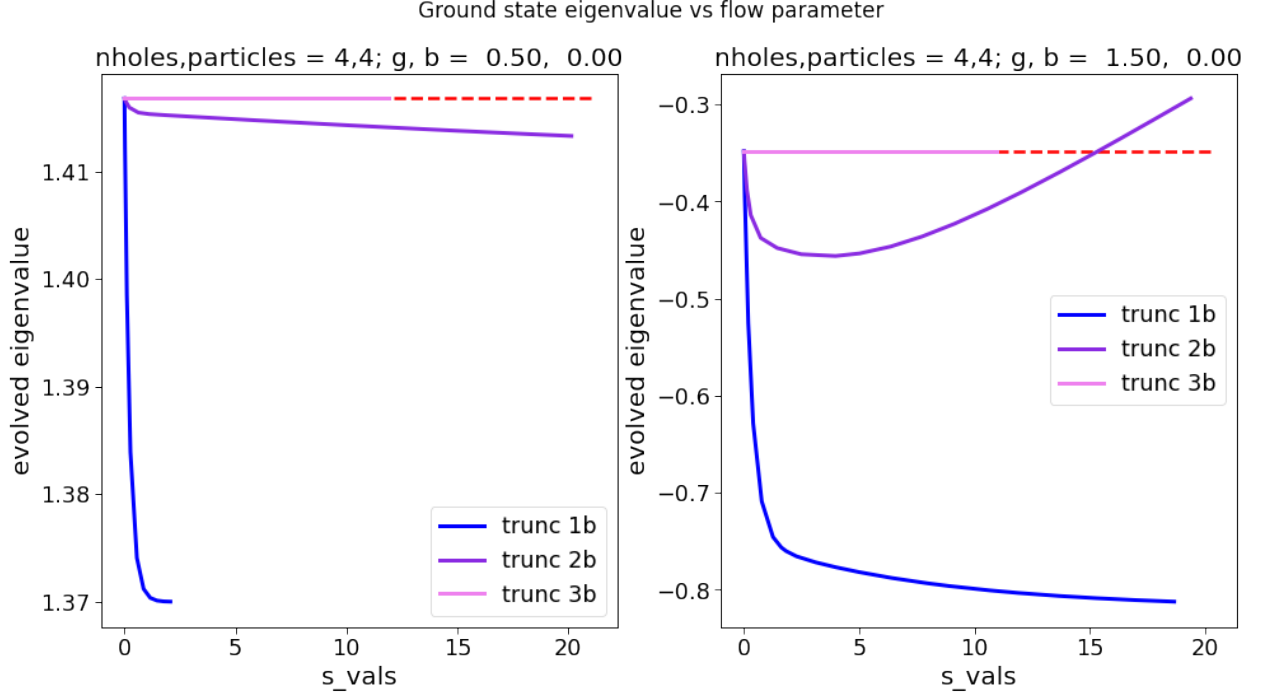


Figure 6.7 Flow of the ground state eigenvalue when normal ordering with respect to the ground state for two P3H Hamiltonians, with no-pairing breaking. The left pane is a weak pairing case, where the right pane is the strong pairing case. The truncation level of the IDMs in the generator and the MR-IMSRG(2) flow has been varied.

6.9 Outlook

In this chapter, we have presented the reference state ensemble scheme for reducing truncation error in the IMSRG(2) flow. We have shown that by introducing information about potential excitations of the system into the IMSRG(2) flow via the reference state ensemble, we can reduce the violation of the transformation's unitarity due to truncations. This is evident from the improvement in the preservation of the spectrum of evolved P3H Hamiltonians which have been normal-ordered with respect to reference ensembles. We have also shown that the use of reference ensembles that are optimized for the Hamiltonian does not seem to significantly distort observables which are evolved alongside the Hamiltonian, although this matter will have to be revisited in more realistic applications.

In the future, we hope to improve the optimization process in Algorithm 6.1 in several ways. For one, the exact spectrum will be inaccessible in applications to realistic systems, so we will need to conceive of different characteristics for the loss of unitarity in observables, e.g., partial

traces. Or, perhaps, we could implement a neural network optimization strategy to “learn” the best ensemble mixing from a large set of training points. Of course, this method is also constrained by the computational effort for performing the full (MR-)IMSRG evolutions, as well as the richness of viable evolutions in the parameter surface spanned by the mixing parameters. We may be able to leverage the emulator techniques discussed in Chapter 7 for generating comprehensive training sets for reference ensemble optimization.

BIBLIOGRAPHY

- [1] S. Blanes et al. “The Magnus expansion and some of its applications”. In: *Physics Reports* 470.5 (2009), pp. 151–238. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2008.11.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0370157308004092>.
- [2] H. Hergert et al. “The In-Medium Similarity Renormalization Group: A novel ab initio method for nuclei”. In: *Physics Reports* 621 (Mar. 2016), pp. 165–222. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2015.12.007. URL: <http://dx.doi.org/10.1016/j.physrep.2015.12.007>.
- [3] Morten Hjorth-Jensen, Maria-Paola Lombardo, and Ubirajara van Kolck. *An advanced course in computational nuclear physics: bridging the scales from quarks to neutron stars*. Springer, 2017.
- [4] Chenyang Li and Francesco A. Evangelista. “Driven similarity renormalization group for excited states: A state-averaged perturbation theory”. In: *The Journal of Chemical Physics* 148.12 (Mar. 2018), p. 124106. ISSN: 0021-9606. DOI: 10.1063/1.5019793. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.5019793/16710196/124106_1_online.pdf. URL: <https://doi.org/10.1063/1.5019793>.
- [5] Wilhelm Magnus. “On the exponential solution of differential equations for a linear operator”. In: *Communications on Pure and Applied Mathematics* 7.4 (1954), pp. 649–673. DOI: <https://doi.org/10.1002/cpa.3160070404>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160070404>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160070404>.
- [6] T. D. Morris, N. M. Parzuchowski, and S. K. Bogner. “Magnus expansion and in-medium similarity renormalization group”. In: *Phys. Rev. C* 92 (3 Sept. 2015), p. 034331. DOI: 10.1103/PhysRevC.92.034331. URL: <https://link.aps.org/doi/10.1103/PhysRevC.92.034331>.
- [7] Thomas J. Jr. Watson and Garnet Kin-Lic Chan. “Correct Quantum Chemistry in a Minimal Basis from Effective Hamiltonians”. In: *Journal of Chemical Theory and Computation* 12.2 (2016). PMID: 26756223, pp. 512–522. DOI: 10.1021/acs.jctc.5b00138. eprint: <https://doi.org/10.1021/acs.jctc.5b00138>. URL: <https://doi.org/10.1021/acs.jctc.5b00138>.

CHAPTER 7

KOOPMAN OPERATOR FOR THE IN-MEDIUM SIMILARITY RENORMALIZATION GROUP

7.1 Introduction

In this chapter, we present the results of several algorithms developed from Koopman operator approximation techniques, namely the Dynamic Mode Decomposition (DMD), to emulate the solution to the IMSRG(2) flow (Equation 3.18). There are two algorithms which work in tandem to generate a complete emulator to the parametric IMSRG problem (i.e. emulate IMSRG solution varying couplings): 1) a standard DMD formula, which we show can emulate the solution to a particular IMSRG(2) flow [4, 3] and 2) an algorithm which interpolates DMD operators across a parametric coupling surface which characterizes the model Hamiltonian we wish to evolve via IMSRG(2) [6].

7.2 The Koopman Operator

The *Koopman operator* [7] is an infinite-dimensional linear operator which acts on a measurement subspace of a nonlinear dynamical system. The system itself may be finite or infinite-dimensional. The spectral decomposition of this operator characterizes the behavior of the nonlinear dynamical system [3, 4]. In principle, the measurement subspace is continuous in time; however, for our applications we concern ourselves with the *discrete-time* Koopman operator which acts on a *discrete* measurement subspace.

Consider a measurement function g which evaluates a system x , where x is interpreted as a vector of state variables, at some point in time x_k . Forward time-steps in the system are governed by a flow map $\mathbf{F}$, which propagates the *system* forward in time such that,

$$\mathbf{F}(x_k) = x_{k+1}. \quad (7.1)$$

The Koopman operator is defined by the composition of g and $\mathbf{F}$ [4],

$$\mathcal{K}g = g \circ \mathbf{F}, \quad (7.2)$$

which shows that the Koopman operator propagates the *measurement function* forward in time, so that [4],

$$\mathcal{K}g(x_k) = g(F(x_k)) = g(x_{k+1}). \quad (7.3)$$

In other words, the Koopman operator characterizes the flow map $\mathbf{F}$, under which x evolves, via measurements of the system by g . In the eigenbasis of $\mathcal{K}$ [4],

$$\mathcal{K}\phi(x_k) = \lambda\phi(x_k) = \phi(x_{k+1}), \quad (7.4)$$

which shows that evaluation of the Koopman eigenfunctions characterize the dynamics of the system. In other words, the dynamics of the nonlinear system are fully captured by an infinite-dimensional linear operator (the Koopman operator) that acts on the space of measurements of the system.

Since the Koopman operator is infinite-dimensional, we must find techniques which can approximate the Koopman operator in a finite dimension. The technique which we use in this chapter is called Dynamic Mode Decomposition, or DMD.

7.3 Dynamic Mode Decomposition for IMSRG

The DMD is a finite-dimensional, finite-measurement, approximation to the discrete Koopman operator for a nonlinear dynamical system. It was originally developed by Schmid [12] in the context of fluid dynamics. The DMD is formulated as the best-fit linear operator which connects successive, time-ordered measurements of the target system [12, 11]; as a result, Singular Value Decomposition (SVD) is most often the technique used to solve for the DMD operator, due to its computationally efficient scaling and flexibility in low-rank truncated spaces.

The DMD begins with N successive observations from the evolving dynamical system, organized into offset data matrices X and X' , where X' is one step forward relative to X . In matrix form,

$$X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_{N-1} \\ | & & | \end{bmatrix}; \quad X' = \begin{bmatrix} | & & | \\ x_2 & \cdots & x_N \\ | & & | \end{bmatrix}. \quad (7.5)$$

Algorithm 7.1 Pseudocode implementation which demonstrates the standard algorithm for DMD.

```

collect  $N$  observations from evolving dynamical system, taking steps of fixed width  $dt$ 
organize offset matrices  $X$  and  $X'$ , with shapes  $M \times N - 1$ 
solve  $X^\dagger$  (via SVD)
 $A = X'X^\dagger$ 
compute eigendecomposition  $A\Phi = \Lambda\Phi$ 
compute DMD mode amplitudes  $b = \Phi^\dagger x_0$ 
expand dynamical state  $x_k = \Phi\Lambda^{k-1}b$  ▷ for  $k > 0$ ,

```

For simplicity, we assert that each observation be taken at a regular intervals dt in the dynamical variable. The DMD operator A shifts X one step forward in time (or whatever the dynamical variable may be), i.e.

$$AX = X'. \quad (7.6)$$

We may proceed to solve for the “best-fit” A in a least-squares sense, obtaining

$$A = X'X^\dagger, \quad (7.7)$$

where † denotes the Moore-Penrose inverse (pseudoinverse) of X . The Moore-Penrose inverse minimizes the L2 loss function $\|X' - AX\|^2$ such that

$$X^\dagger = (X^T X)^{-1} X^T, \quad (7.8)$$

where we must have that $(X^T X)$ is invertible.

Standard Dynamic Mode Decomposition The *standard* DMD algorithm is written, in pseudocode, in Algorithm 7.1. In the final line of Algorithm 7.1, the state vector x_k represents the state of the dynamical system at step k , where $k = t_k/dt$. The DMD eigenfunctions, or *modes*, are denoted Φ and the DMD eigenvalues are denoted Λ . The vector x_0 represents the initial state of the dynamical system; the product $\Phi^\dagger x_0$, usually denoted b , is the DMD amplitudes which represent the DMD eigenfunctions evaluated in the initial conditions.

Algorithm 7.1 explains how the dynamical system is expanded in DMD modes, according to the *DMD* expansion [8, 3, 4, 12], written

$$x_k = \Phi\Lambda^{k-1}b, \quad k > 0. \quad (7.9)$$

We may translate this expansion to continuous time by converting the diagonal DMD eigenvalue matrix Λ to its continuous analog Ω , with diagonal elements $\omega_i = \ln(\lambda_i)/dt$, so that the continuous expansion reads [4],

$$x(t) = \Phi \exp(\Omega t) b. \quad (7.10)$$

In order to fully characterize the dynamical system, the necessary information that must be extracted from DMD are the modes Φ and the eigenvalues Λ . Adjacent objects, such as the amplitudes b , are inferred or directly computed from Φ and Λ . Thus, we refer to a *DMD decomposition* of a dynamical system as the pair (Φ, Λ) .

Projected Dynamic Mode Decomposition One prohibiting issue with the standard DMD algorithm is that the memory and computational cost scales heavily with the number of dynamical variables M . In our case, these would be the coefficients of the IMSRG(2) Hamiltonian. The DMD operator in Algorithm 7.1 is a dense matrix with shape $M \times M$; for a typical nuclear many-body calculation using IMSRG(2), M might be on the order 10^6 or greater. The viability of the standard DMD algorithm is quickly challenged for practical uses of the IMSRG(2).

Instead, we leverage the flexibility of the SVD to solve the *projected* DMD, which solves for the DMD operator in a reduced measurement subspace. Beginning with an SVD on the data matrix X ,

$$X \approx U \Sigma V^*, \quad (7.11)$$

we can truncate the number of singular values used in the reconstruction at r while minimizing the reconstruction error, assuming that the singular values are exponentially decaying [8]. The truncated SVD expression then reads

$$X_{\text{trunc}} \approx U_r \Sigma_r V_r^*. \quad (7.12)$$

Note that the shapes of X and X_{trunc} are equivalent, so we assume that $X \approx X_{\text{trunc}}$ up to some negligible truncation error (due to the exponentially decaying singular values), which is controllable by choosing an appropriate r . Then we may interpret the matrix U_r as a transformation from the full measurement space to the reduced measurement subspace.

The *reduced* DMD operator is the projection of the full DMD operator onto the reduced SVD modes [8],

$$A_r = U_r^* A U_r, \quad (7.13)$$

where A_r is the $r \times r$ projected DMD operator onto the reduced space. From here, we may substitute the expression for the full DMD operator A , Equation 7.7, as well as the SVD on X , to arrive at

$$\begin{aligned} A_r &= U_r^* (X' X^\dagger) U_r \\ &= U_r^* X' (U_r \Sigma_r V_r^*)^\dagger U_r \\ &= U_r^* X' V_r \Sigma_r^{-1}. \end{aligned} \quad (7.14)$$

The reduced DMD operator A_r and the full DMD operator A necessarily share up to r eigenvalues. Thus, the eigendecomposition of A_r reads

$$A_r W = W \Lambda. \quad (7.15)$$

Tu et al. [13] showed that the full DMD eigenvectors Φ can be recovered from the reduced DMD eigenvectors W via

$$\Phi = X' V_r \Sigma_r^{-1} W. \quad (7.16)$$

At this point, the full DMD decomposition (Φ, Λ) has been recovered from the reduced measurement subspace. The dynamical system can then be expanded in the DMD modes according to Equations 7.9 or 7.10.

Algorithm 7.2 summarizes the procedure for computing a reduced DMD decomposition. Note that the computational effort for computing the eigendecomposition of the DMD operator has been significantly reduced by solving for the reduced operator A_r ; the projection softens the scaling from $O(M^3)$ in the full measurement space, to $O(r^3)$ in the reduced measurement subspace, where $r \ll M$.

7.4 Emulating the IMSRG Flow

In this section, we justify how the DMD can be used to “learn” the unitary transformation which propagates the IMSRG flow. We have studied only IMSRG up to 2-body truncations, i.e. IMSRG(2), although we emphasize that the emulation procedure is consistent for IMSRG(N).

Algorithm 7.2 Pseudocode implementation which demonstrates the algorithm for projected DMD.

collect N observations from evolving dynamical system, at fixed width dt
organize offset matrices X and X' , with shapes $M \times N - 1$
compute truncated SVD on $X \approx U_r \Sigma_r V_r^*$
compute $A_r = U_r^* X' V_r \Sigma_r^{-1}$
compute eigendecomposition $A_r W = W \Lambda$
project back to full space $\Phi = X' V_r \Sigma_r^{-1} W$ or $\Phi = U_r W$
compute DMD mode amplitudes $b = \Phi^\dagger x_0$
expand dynamical state $x_k = \Phi \Lambda^{k-1} b$ ▷ for $k > 0$,

In Chapter 3.2, Equation 3.1, we expressed the continuous, unitary, s -dependent transformation $U(s)$ as the transformation which drives the target Hamiltonian toward a desired form (e.g., to the minimal decoupling of an eigenvalue), and in the standard IMSRG approach, we “solve” for this transformation implicitly by propagating the system of flow equations for the Hamiltonian. We can also think of the transformation in the form

$$H(s) = \mathcal{U}(s)H(0), \quad (7.17)$$

where $\mathcal{U}(s)$ is an unknown “super operator” that acts on the algebra of second-quantized operators, and encodes the desired continuous unitary transformation. The expression for the DMD expansion in Equation 7.10 provides a description of exactly this form. Replacing $x \rightarrow H$ so that the coefficients of $H(s)$ becomes the state variables tracked in x , we can express $\mathcal{U}(s)$ in terms of the Koopman modes and frequencies as

$$H(s) = \mathcal{U}(s)H(0) = \{\Phi \exp(\Omega t) \Phi^\dagger\} H(0). \quad (7.18)$$

Note that the product $\{\Phi \exp(\Omega t) \Phi^\dagger\}$ is unitary, provided that the columns of Φ are sufficiently linearly independent. We may take for granted that this transformation provides the desired evolution behavior, simply because the transformation is informed by measurement data of the IMSRG flow. Of course, this assumption is predicated on *good* measurement data, meaning the IMSRG flow being measured is well-behaved to begin with. In this way, we consider the DMD a *data-driven, non-intrusive emulator* for the IMSRG—“data-driven” in the sense that the DMD is built from, or informed by, data measured from the evolving system, and “non-intrusive” in the sense that DMD does not modify the IMSRG method itself.

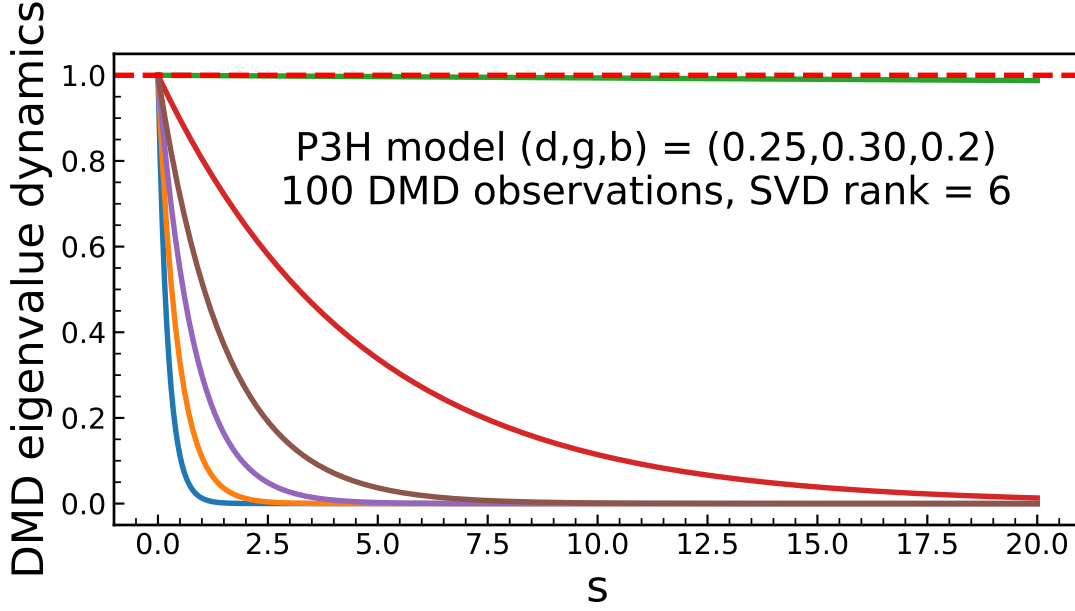


Figure 7.1 DMD mode dynamics in s , for a rank-6 DMD expansion of a strongly interacting P3H model. Each curve correspond to exponentials $e^{\omega_1 s}$ through $e^{\omega_6 s}$ in Equation 7.19, ordered from slowest to fastest frequency. Note that one mode in particular remains approximately constant in s .

Equation 7.18 reveals an interesting feature of DMD emulation in the IMSRG context. Practical implementation of DMD emulation will treat $H(s)$ as a one-dimensional vector which contains all three pieces of the normal-ordered operator in the IMSRG(2) truncation, $H(s) = [E(s), f(s), \Gamma(s)]^T$. In this way, the rows of Φ correspond to the elements of the vector $H(s)$. For example, we need only the first row of Φ to reconstruct the flow of $E(s)$,

$$E(s) = \phi_{11}e^{\omega_1 s}b_1 + \phi_{12}e^{\omega_2 s}b_2 + \cdots + \phi_{1r}e^{\omega_r s}b_r. \quad (7.19)$$

For the energy to be *convergent*, the DMD modes must be convergent. This is readily achieved with *decaying* mode frequencies, ω_k —and, in fact, this is what we observe from from DMD decomposition of IMSRG flow data. Figure 7.1 displays the exponentials which drive the IMSRG dynamics in the DMD reconstruction of a strongly interacting P3H model. If we take s to infinity, DMD expansion will take the Hamiltonian to zero. In some cases, it may be beneficial or necessary to define a termination constraint in the expansion. Figure 7.1 is typical in IMSRG data in the sense that the slowest decaying mode, where the ω 's are ordered from slowest to fastest, tends to “look constant” in the range of relevant dynamics. In this case, we may approximate $\omega_1 \approx 0$ so

that $e^{\omega_1} \approx 1$. In this way, the DMD expansion will converge to a nonzero result as s increases. Formally, we might view this process as multiplying through the expansion a common factor $e^{-\omega_1 s}$, to arrive at

$$E(s) = \phi_{11}e^0 b_1 + \phi_{12}e^{(\omega_2 - \omega_1)s} b_2 + \dots + \phi_{1r}e^{(\omega_r - \omega_1)s} b_r. \quad (7.20)$$

Note that if ω_1 is truly small, then for ω_k where $k > 0$, $\omega_k - \omega_1 \approx \omega_k$.

In this special case that we wish to calculate the theoretically converged result as $s \rightarrow \infty$, note that

$$E(s \rightarrow \infty) \approx \phi_{11} b_1, \quad (7.21)$$

where the converged energy is simply the first element of Φ multiplied with the first DMD mode amplitude. We refer to this as the no-decay approximation or ND approximation. The approximation may be extended to the full Hamiltonian, evolved to $s \rightarrow \infty$, by multiplying with the entire DMD eigenvector corresponding to ω_1 .

Now we may characterize the efficacy of forecasting the IMSRG flow using DMD. There are two primary parameters that can be varied to tune emulation performance, namely the number of observations of the evolving system, N , and the SVD truncation rank r . Increasing N provides the DMD emulation with more context, in the sense that the dynamics are more constrained to the target problem. Increasing truncation rank r improves the fidelity of the reduced DMD operator in use, in the sense that more SVD directions are available to the dynamical expansion. However, increasing r too much may introduce noise to the expansion in the form of singular values which are close to zero, causing instability in the dynamical expansion. One must tune these parameters for the optimal performance in the target problem.

In the IMSRG context, we want to minimize the number of observations (i.e. polynomial-cost ODE solver steps) while also minimizing the error of the forecasted evolved energy. The DMD forecast is most useful if we can stop observing the flow *before the convergence elbow in the energy* (see e.g. the top row of Figure 7.4, which plots the IMSRG-evolving energy). Beyond this elbow, the IMSRG is essentially solved and forecasting is no longer useful. The s value where the convergence elbow begins and ends depends on the chosen flow generator; in general, however, we

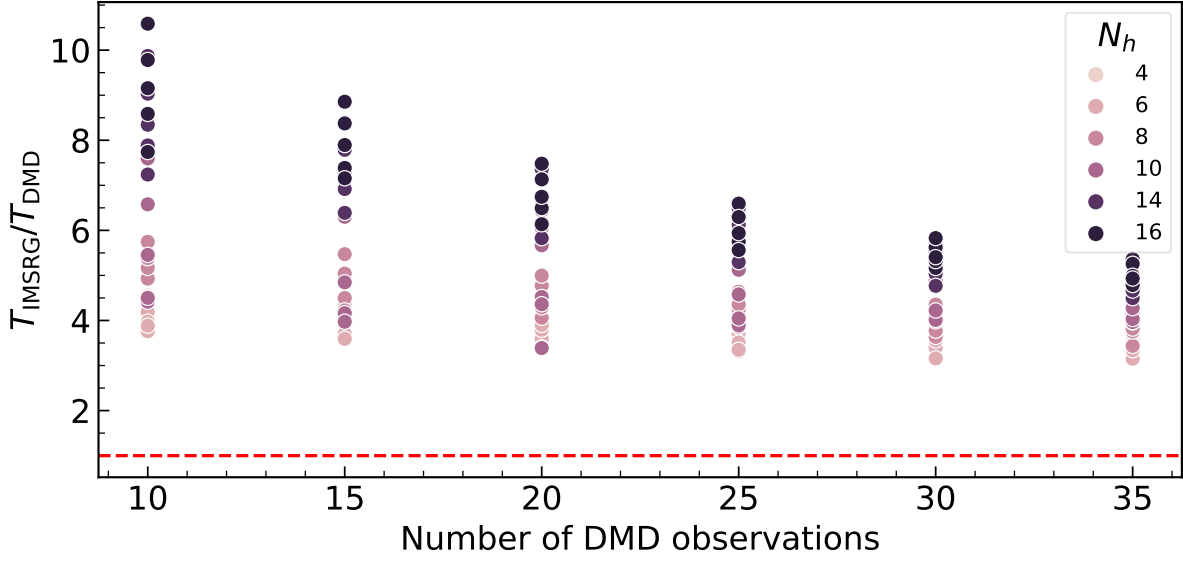


Figure 7.2 The ratio of TACO IMSRG(2) wall time to DMD emulation time. The total wall time for DMD includes the time cost for calculating x number of DMD observations. The hue represents the single particle basis size, with darker color corresponding to larger bases. The red dotted line is where the ratio equals 1.

find that taking 20-30 observations, at a step-width characteristic to the generator, provides enough observations to safely converge to a reasonably accurate energy. This number of observations also tends to fall close to the onset of convergence.

As a first example, we return to the P3H model in Chapter 4.1, calculated using the `tcimsrg` code from Chapter 5.6. In Figure 7.2, we show the speedup of DMD emulation over the TACO implementation of the IMSRG(2). The hue represents the size of the single particle basis. The horizontal axis reflects the number of observations used to build the DMD operator; note that a larger measurement matrix means a larger SVD system to compute, which is why speedups decrease for increasing observations. These results show sizeable performance gains using DMD emulation. Figure 7.3 shows that these performance gains come at minimal cost to accuracy.

Motivated by the cost-savings from DMD emulation in the P3H model, we continue with an example involving an IMSRG(2) calculation of ^{16}O , computed with a SRG-softened $\text{NN}^3\text{LO}(500)$ interaction, and study the accuracy of the DMD forecast on the number of observations. Our results are shown in Figure 7.4. The observation steps were taken in steps of $ds = 0.05$, such that the

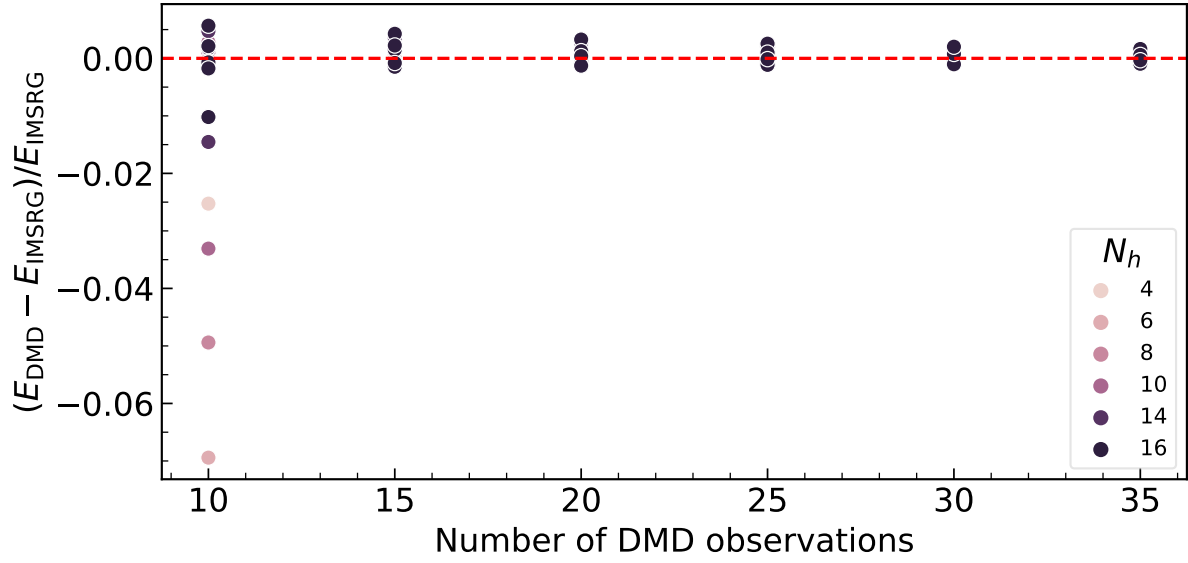


Figure 7.3 The relative error in the evolved energy versus number of DMD observations. We use the same hue scheme as Figure 7.2. The error is controllable by the number of DMD observations, but generally falls on the order of less than a percent no matter the size of the measurement matrix.

max observed s at 1.0 corresponds to 20 observation points. We see from the bottom row of plots that the relative error in the evolved energy improves quickly as the maximum observation point approaches the elbow of convergence; beyond this region, the error improvement saturates. In this way, observations beyond the elbow are not technically relevant for obtaining a good DMD decomposition.

Figure 7.5 provides more detail on how error in the predicted energy interacts with the number of observations and reduced DMD truncation rank. For fewer observations, the SVD is more sensitive to noise—in other words, directions which correspond to near-zero singular values appear at lower truncation levels, which causes instability in the DMD result. This is evident from the way relative error increases sharply beyond $r = 10$ in the row corresponding to 20 observations, for example. Figure 7.5 suggests that the most robust, and effective, way to reduce DMD error is to increase the number of observations provided to the operator. In order for DMD emulation of the IMSRG flow to be effective, we must balance the amount of observations with the error that can be tolerated for the target problem.

Figure 7.6 plots the Frobenius error relative to the IMSRG result for DMD emulation of the

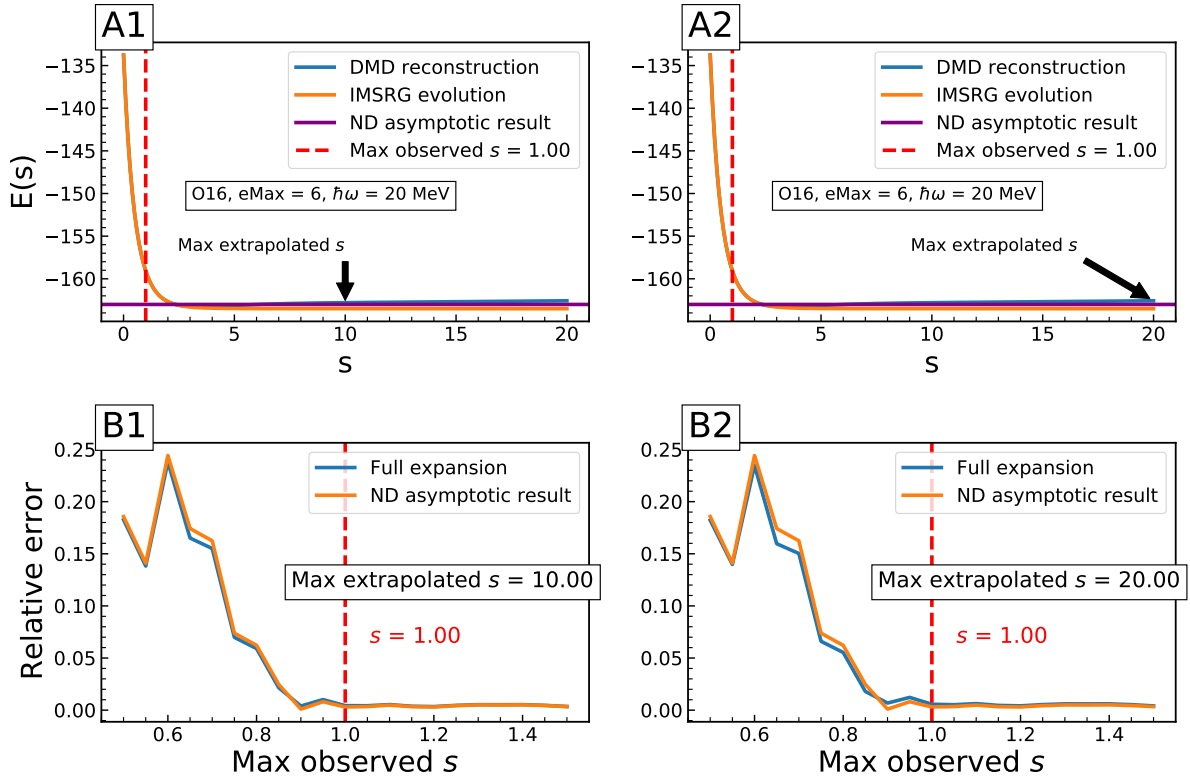


Figure 7.4 DMD forecast results for ^{16}O , in an $e\text{Max} = 6$ harmonic oscillator basis with oscillator width $\hbar\omega = 20$ MeV, with SRG-softened $\text{NN}^3\text{LO}(500)$ interaction. The top row plots evolution of the ground state energy, according to the IMSRG as well as DMD reconstruction of the IMSRG. The vertical dashed line indicates where direct IMSRG observations stop. The bottom row plots the relative error in the energy at $s = 10$ (panel B1) and $s = 20$ (panel B2). The vertical dashed line indicates the relative error result for the flow in the panel above.

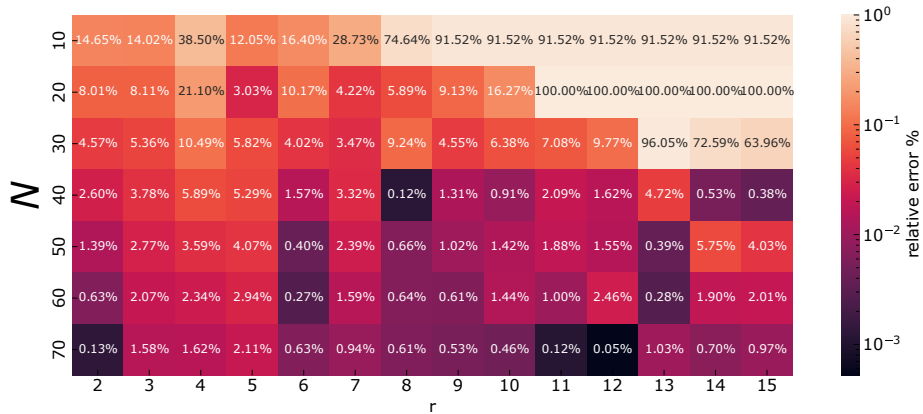


Figure 7.5 DMD emulation of ^{16}O , details as in Figure 7.4. Relative error in the DMD-predicted evolved energy as a function of number of observations and truncation rank r .

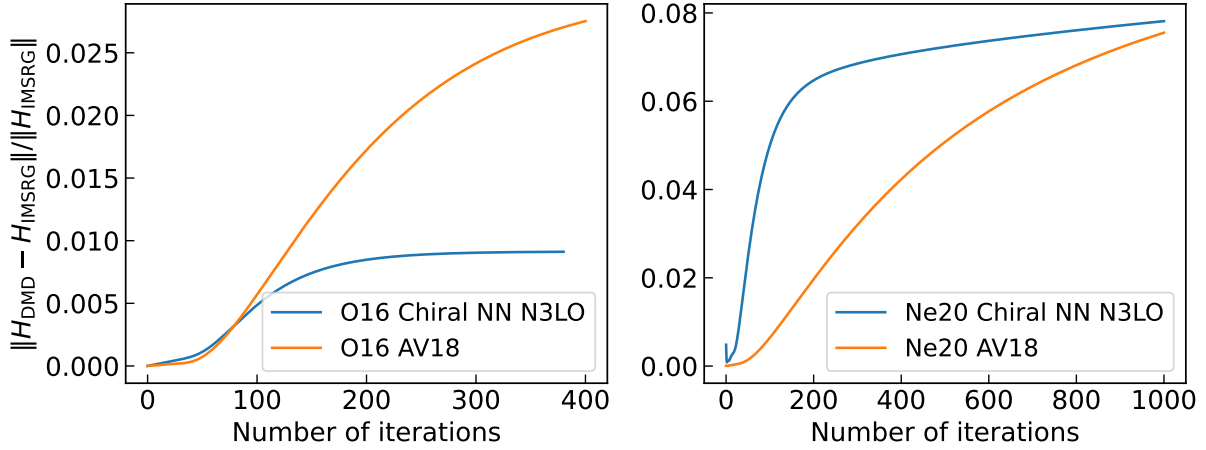


Figure 7.6 DMD emulation of the IMSRG(2) flow for ^{16}O ^{20}Ne in two different interactions AV18 and a chiral NN interaction. The right pane is eMax = 6 basis size, while the right pane is eMax = 10. The vertical axis plots the Frobenius error relative to the IMSRG(2) result.

flow in different interactions and different nuclei, ^{16}O and ^{20}Ne . Note that ^{20}Ne open-shell, which involves stiffer IMSRG(2) flow equations because the reference state is harder to decouple. Also note that AV18 [14] is a very “hard” phenomenological interaction, compared to the chiral interactions in Chapter 4.2. We show the DMD is able to reproduce the full flowing Hamiltonian with high accuracy in every case, meaning the DMD is able to fully capture the dynamics of the problem regardless of the interaction. The DMD emulates the results within the finite subset of the operator algebra that characterizes the IMSRG flow.

7.5 Emulating the Parametric IMSRG Flow

The “parametric” IMSRG is an extension to the IMSRG problem, which includes the coupling constants that parameterize the evolving the Hamiltonian. For the purposes of uncertainty quantification and sensitivity analyses, we need to explore the dependence of the IMSRG results on the parameters of the chiral interactions (Chapter 4.2). The parameter space is very large at 15+ dimensions and we require millions, or even billions, of samples to characterize the space. Collecting this volume of samples is not feasible with direct IMSRG calculations. Thus, we want to leverage DMD reduced-order models of the IMSRG flow to emulate the results. The super operator

we wish to emulate becomes

$$H(c_1, \dots, c_{N_c}; s) = \mathcal{U}(c_1, \dots, c_{N_c}; s)H(c_1, \dots, c_{N_c}; 0) \quad (7.22)$$

where c_i represents a particular coupling constant (e.g. the parameters of the P3H Hamiltonian, or the LECs of the chiral interactions). The coupling constants which characterize the evolving Hamiltonian also characterize the unitary transformation. In this emulation scheme, we apply the DMD to emulate the unitary transformation as a trajectory s along the *coupling manifold* characterized by $c_1, \dots, c_{N_c}$. The primary algorithm we employ to tackle the parametric problem is Reduced Koopman Operator Interpolation (rKOI), introduced by Huhn et al. [6]. In the rKOI scheme, we directly interpolate the DMD operator by sampling a training set of DMD operators computed from varying IMSRG trajectories along the Hamiltonian coupling manifold.

Algorithm 7.3 provides an overview of the rKOI algorithm. The key feature of this strategy is that the reduced DMD operator is interpolated directly. The DMD expansion computed from

Algorithm 7.3 Pseudocode implementation which demonstrates the algorithm for rKOI DMD.

```

vary coupling constants  $c_1, \dots, c_{N_c}$   $N_{\text{train}}$  times and compute IMSRG trajectory  $D_i$  for each realization
for each  $D_i$  in  $N_{\text{train}}$  size set do
    compute projected DMD in Algorithm 7.2
    record  $A_r^{(i)}, U_r^{(i)}, b^{(i)}$ 
end for
train interpolators to predict  $\mathcal{I}_{A_r}, \mathcal{I}_{U_r}, \mathcal{I}_b$  on training data  $\{A_r^{(i)}\}, \{U_r^{(i)}\}, \{b^{(i)}\}$ 
compute DMD expansion in Algorithm 7.2 from interpolated operators

```

interpolated operators reads

$$x_k^I = U_r^I W^I (\Lambda^I)^{k-1} b^I \quad (7.23)$$

where we explicitly include how the interpolated operators enter into the expansion. Note that W^I, Λ^I are computed by the eigendecomposition of A_r^I .

We refer to Algorithm 7.3, or the process of ingesting training data to compute the interpolated operators, as the *interpolation engine* (IE). The computed operators from which we may construct the DMD expansion, and therefore predict the evolution of the system given a target set of coupling

constants, we refer to as the interpolation engine model, or IEM. The trained IEM fully captures the transformation to reduced space U_r , the dynamical trajectory through the coupling constant manifold encoded by A_r , and the space of initial conditions—this means that a prediction using this interpolation scheme is *data-free*, in the sense that the only input necessary to predict every feature of the dynamical system is the target set of coupling constants. Of course, the accuracy of the prediction depends on the success of the interpolation method, which itself depends on the well-behaved nature of the manifold we wish to interpolate.

7.6 Interpolating DMD using RBFs

We have chosen to interpolate the required DMD operators using a radial basis function (RBF) interpolator¹, in linear and thin plate spline kernels. The *mesh-free* nature of RBF interpolation was the deciding factor for its implementation in DMD interpolation—the RBF-IEM can freely provide predictions not constrained to a mesh. Instead of a mesh, we employ Latin hypercube sampling for optimal coverage of the parameter space, and the nature of the RBF interpolation allows us to generalize the model inside and outside of the sampled region. In the following, we demonstrate the success of the RBF-IEM in one and two dimensions, corresponding to the interaction parameters g , b in the P3H model (Chapter 4.1), as well as the success of interpolation in a realistic nuclear system.

Interpolating in 1D Toy Model Figure 7.7 illustrates how the RBF-IEM reproduces the evolved energy in the P3H model across a wide range of pairing strengths g , with pair-breaking strength b fixed at zero. In the 1D case, the training points have been sampled uniformly except at $g = 0$ (we exclude $g = 0$ from the training space because the energy is trivially zero). The predicted energy matches the IMSRG energy with reasonable accuracy, at least on the order 10^{-3} , with a notable loss in accuracy beyond the training point boundaries—this is a disadvantage of the *mesh-free* interpolation.

¹SciPy documentation: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.RBFInterpolator.html#scipy.interpolate.RBFInterpolator>.

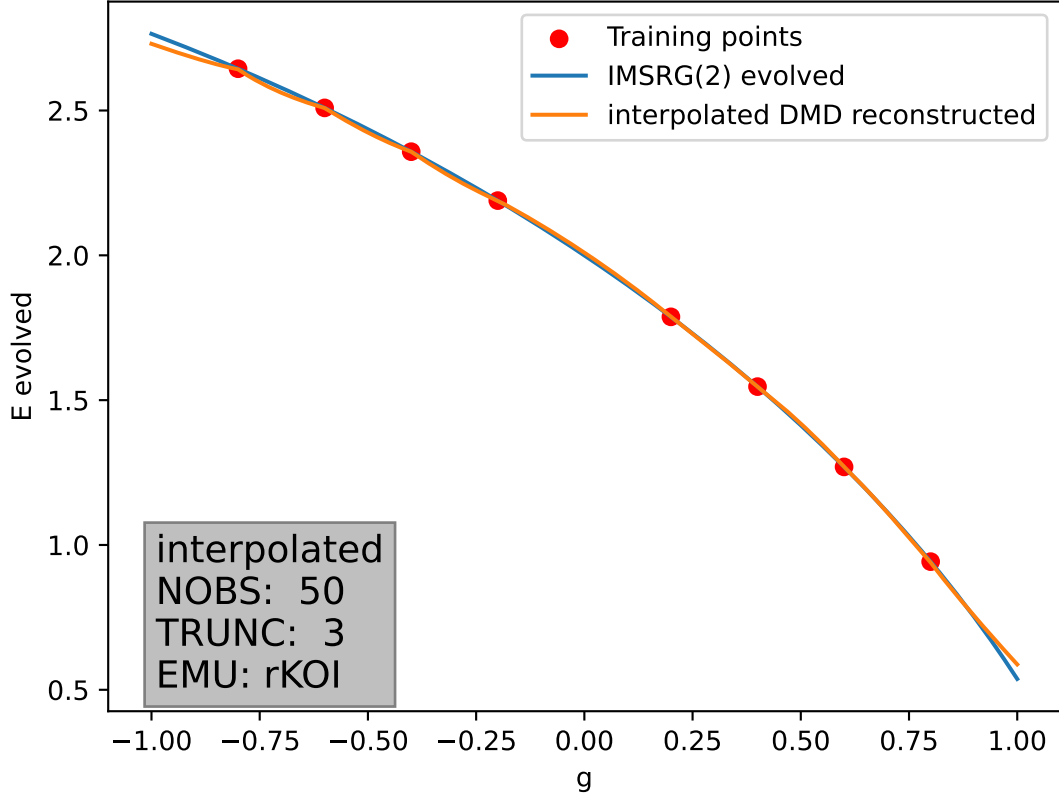


Figure 7.7 IMSRG(2) ground-state energy as a function of the pairing strength g , comparing the interpolated DMD prediction against the IMSRG(2)-evolved result. The red dots indicate IMSRG(2) evolution training points for the interpolation engine. Each training point represents $N = 50$ IMSRG(2) iterations which are used to construct a training DMD operator at SVD truncation level $r = 3$.

Interpolating in 2D Toy Model In Figure 7.8, we now vary g and b in the P3H model. The training points have been collected using a Latin hypercube sampling strategy. The Latin hypercube sampling promotes optimal coverage by dividing the parameter space into equal partitions, and then sampling one point from each partition. Note that the error range is fairly wide, with most predictions received at an error 10^{-1} to 10^{-4} . The distribution of training points seems to affect the locality of error; that is, regions which are less dense with training points impute a larger error in those regions. At the same time, there are particular regions, such as the lower left quadrant, where increasing the number of training points does *not* seem to affect the error significantly. This is likely improvement saturation constrained by the reconstruction error of the individual DMD

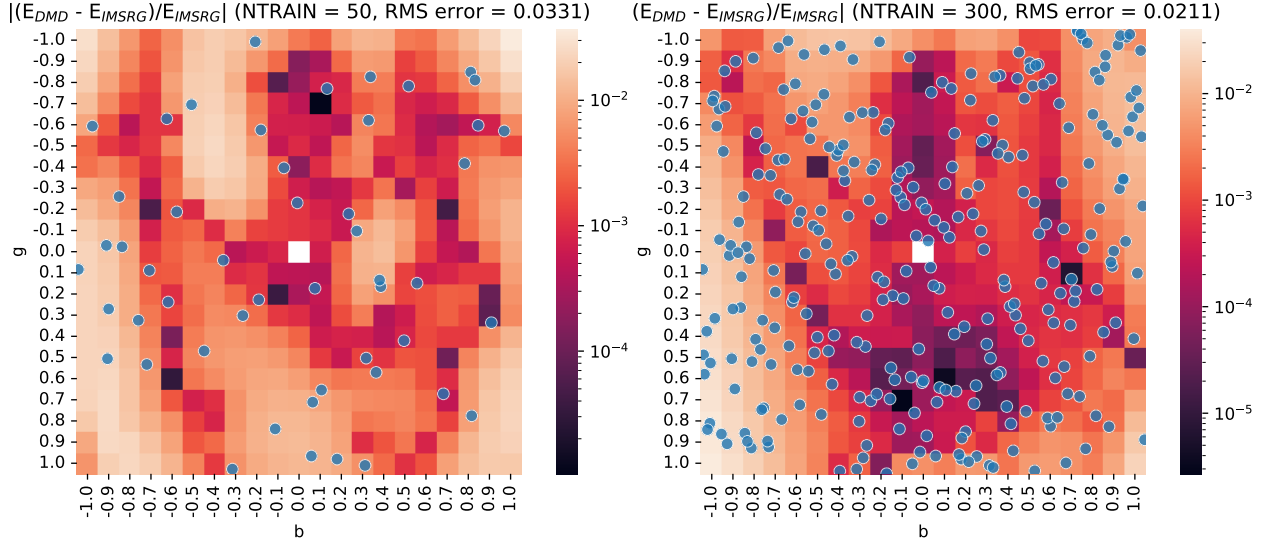


Figure 7.8 Absolute relative error in the evolved energy, comparing DMD emulation to IMSRG(2) evolution. Each training point represents a rank 3 DMD operator computed from 50 observations. (*LEFT*) 50 training points. (*RIGHT*) 300 training points.

training points. In other words, the parametric emulator can only predict as well as the DMD operators on which it has been trained.

In Figure 7.9, we plot the absolute relative error of direct DMD forecasts on the same grid points as Figure 7.8. The results of DMD forecasting reveals that the RBF-IEM error is constrained by the accuracy of the DMD itself. We might improve the error of the DMD forecast by increasing the truncation rank. However, a larger DMD operator results in a more complicated parameter space to interpolate.

Interpolating a Realistic Nucleus Ultimately, our goal is to construct a practical IMSRG(2) emulator for realistic (nuclear) many-body calculations. There is an array of additional challenges associated with the IE applied in this case, including memory requirements (compare the P3H Hamiltonian of 10^3 IMSRG(2) coefficients to a realistic nucleus which requires $10^6 - 10^9$ IMSRG(2) coefficients), but we will demonstrate that they can be overcome. As an example, we will construct an RBF-IEM to model realistic IMSRG(2) calculations of the closed-shell ^{16}O nucleus, varying 24 low energy constants (LECs) in the underlying chiral N^3LO NN interaction by Entem and Machleidt

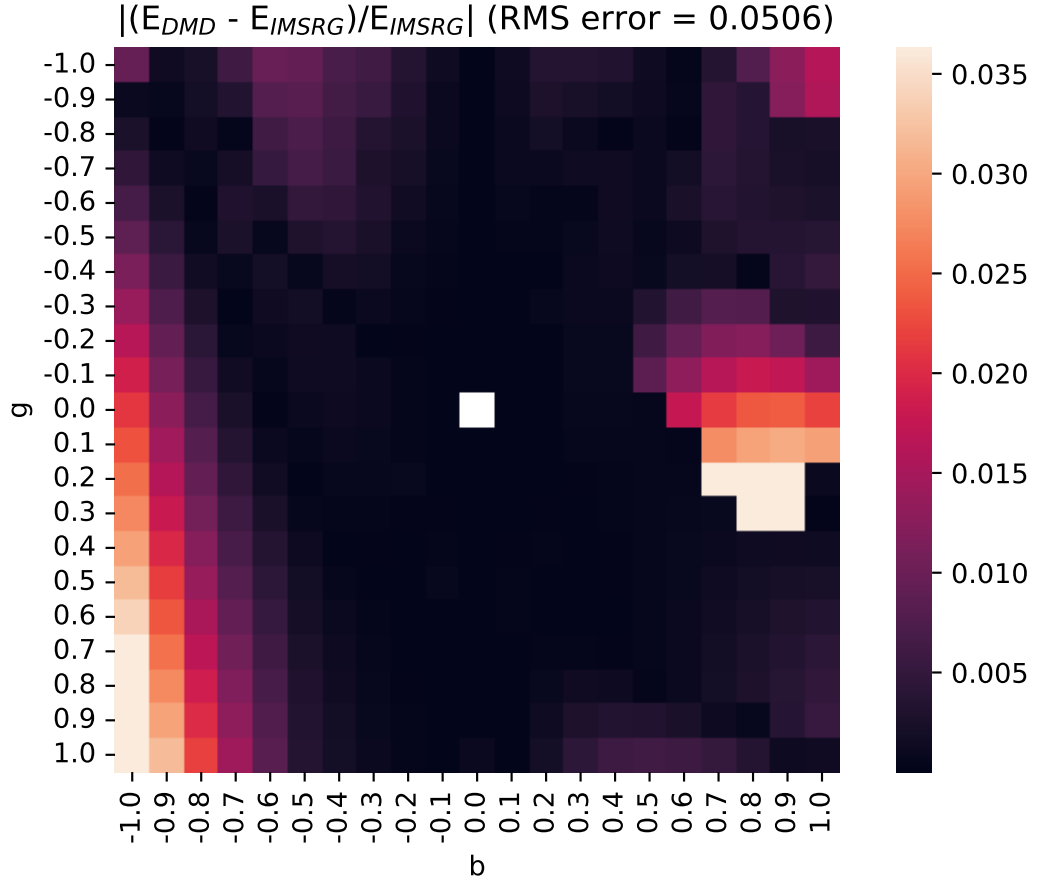


Figure 7.9 Absolute relative error in the evolved energy for DMD emulation, measured by IMSRG(2). Each grid point is the evolved energy predicted by a rank 3 truncated DMD operator directly computed from the flow, for the corresponding grid point.

(Chapter 4.2).

Figure 7.10 summarizes the interpolation error on ^{16}O . The RBF-IEM was trained on 300 points, where each training point is a rank 8 truncated DMD operator computed from 50 observations of the evolving ^{16}O Hamiltonian. We plot two error measurements, relative error in the evolved energy and flow reconstruction error (mean-squared error of discrete flow matrix reproduced by DMD). The relative error in the energy measures how well the RBF-IEM predicts the target evolved energy, whereas the reconstruction error measures how well the RBF-IEM models the full dynamics of the associated IMSRG(2) flow. Note the variability is rather high in both error measurements. We measure the error according to direct IMSRG(2) calculations using parameters which were not in the training set. Validation points with particularly high error likely are “far” from the training

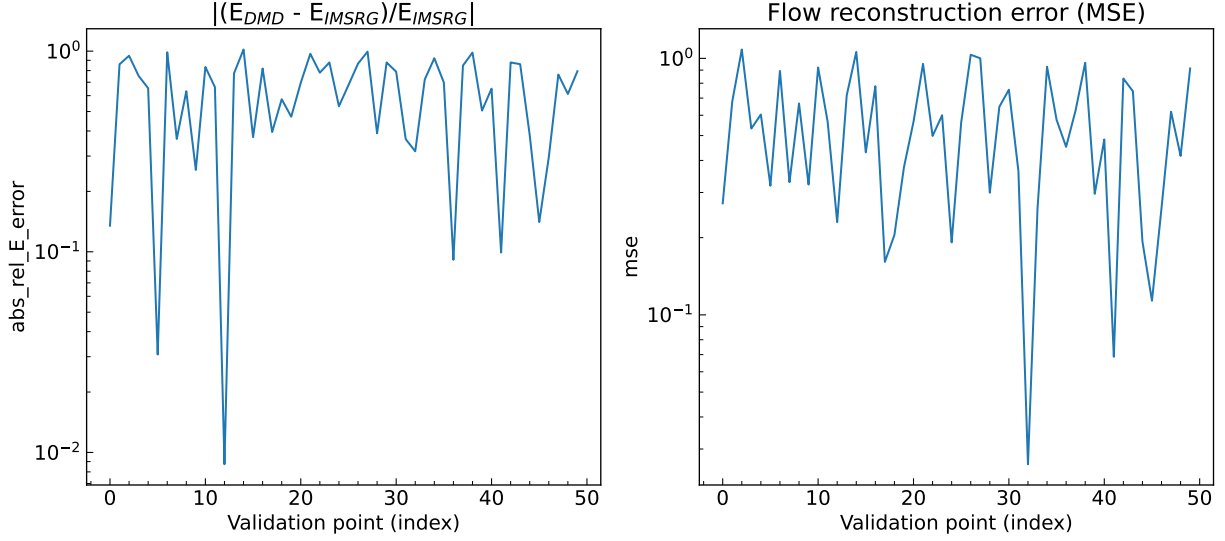


Figure 7.10 Error measurements from interpolation engine trained on 300 training points. In both panels, the error measurement is distributed across 50 random validation points. The left pane plots absolute relative error in the evolved energy. The right pane plots mean-squared error in the reconstructed flow from the interpolated DMD operators.

points, meaning that the model does not seem to generalize well, and that more training points could tighten the error spread. Figure 7.11 provides some evidence for this claim. Note that variations in the reconstruction error seem to correlate, although not perfectly, with variations in the average distance from each validation point to the training set. Additionally, interpolating the parameter space of the operator U_r , which transforms the DMD operator to reduced space, is difficult since the size of the operator is equal to the number of state variables in the dynamical system. This is in contrast to the parametric space of the DMD operator itself, which is 8^2 parameters in this case.

To tackle this issue, we modify our approach by performing a reduction on the *training space* of IMSRG(2) flows, taking inspiration from the partitioned approach to parametric DMD introduced by Andreuzzi, Demo, and Rozza [1]. Instead of interpolating the full dimension of the training space, we reduce the feature dimension of the training space to the principal components with the help of an SVD.

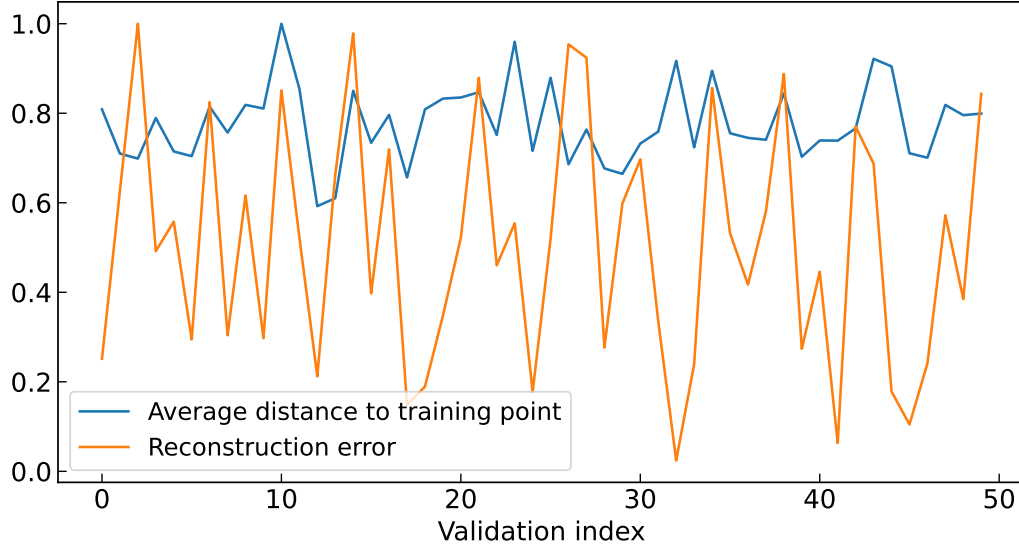


Figure 7.11 Illustrated in this plot is the reconstruction error (mean-squared error) in Figure 7.10, as well as the validation point average distance to the training points. The vertical axis for each measurement has been scaled so that the data appear between 0 and 1, so that variations can be compared.

Borrowing the notation of Andreuzzi, Demo, and Rozza [1], let

$$X^{c_i} := \begin{bmatrix} | & \dots & | \\ x_1^{c_i} & \dots & x_N^{c_i} \\ | & \dots & | \end{bmatrix} \in \mathbb{R}^{m \times N}, \mathcal{X}_1 := \begin{bmatrix} X^{c_1} & \dots & X^{c_p} \end{bmatrix} \in \mathbb{R}^{m \times Np}, \quad (7.24)$$

where X^{c_i} represents N observations of the IMSRG(2) flow according to parametric set c_i (e.g., a particular set of LECs). There is evidence of low rank structure in the IMSRG(2) flow (we thank Boyao Zhu and Heiko Hergert for their fruitful discussions in this regard; in addition, see [5]), hence we expect that the training matrix of IMSRG(2) flows given by $\mathcal{X}_1$ will also exhibit low rank structure. Indeed, Figure 7.12 shows that its singular value spectrum is decaying exponentially.

We proceed with SVD on $\mathcal{X}_1$, truncating at t singular values. We emphasize that t denotes the reduction of the *feature space* associated with the training data, while r denotes the truncation of individual reduced DMD operators. Note that in the IMSRG context t is likely to be much greater than the truncation rank r of the reduced DMD operator. The truncated SVD of $\mathcal{X}_1$ reads

$$\mathcal{X}_1 \approx \mathbf{L}_t \mathbf{\Sigma}_t \mathbf{R}_t^*. \quad (7.25)$$

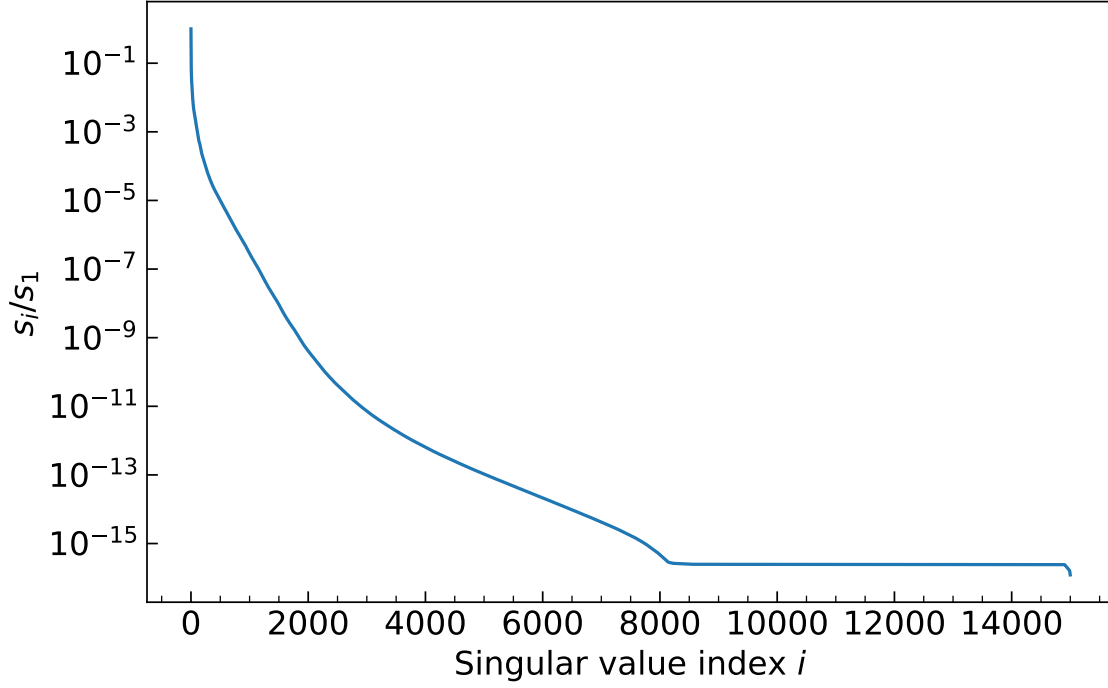


Figure 7.12 Singular value spectrum of the training matrix $\mathcal{X}_1$, scaled by the largest singular value.

Finally, the state variable information in $\mathcal{X}_1$ is compressed using $\mathbf{L}_t$,

$$\tilde{\mathcal{X}}_1 = \mathbf{L}_t^* \mathcal{X}_1 \in \mathbb{R}^{t \times Np}, \quad (7.26)$$

and parametric training may proceed with Algorithm 7.3 using the reduced training matrix. We refer to the model produced by the reduction on the training space as reduced IEM, or rIEM. RBF-rIEM is the RBF-interpolated version of the model. We might notate the truncation rank of the reduced training space directly in the acronym for clarity, e.g. RBF-r(t)IEM

In Figure 7.13, we have trained the RBF-r(1000)IEM on 300 rank-2 reduced DMD operators in the reduced training space corresponding to truncation $t = 1000$. Comparing with Figure 7.10, we observe smaller variations in both measurements, with fluctuations staying between the orders 10^{-2} and 10^{-1} .

We have show that the modified parametric DMD algorithm for training space reduction increased prediction accuracy across a wide range of validation points. As a result of the training space reduction, the time to compute the DMD operator training points, as well as the time to train

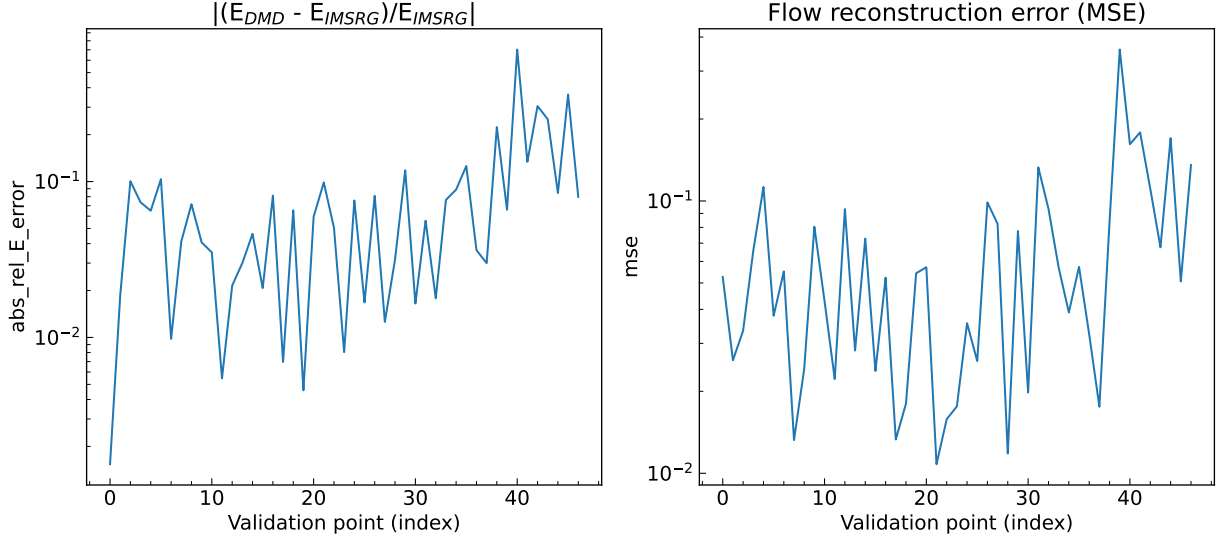


Figure 7.13 Error measurements from interpolation engine trained on 300 *reduced* training points. In both panels, the error measurement is distributed across 50 random validation points. (*LEFT*) Absolute relative error in the evolved energy. (*RIGHT*) Mean-squared error in the reconstructed flow from the interpolated DMD operators.

the IEM, was also reduced.

7.7 Optimized Training for Large Bases using Streaming SVD

The reduced IE method introduces the new cost of computing the SVD on the training space in Equation 7.25. More specifically, we need the left singular vectors $\mathbf{L}_r$ in order to perform the reduction on the training space. Thus, the training algorithm performance hinges on the computation of a very “tall and skinny” matrix $\mathcal{X}_1$, which likely will not completely fit into memory.

To address this problem, we employ the *streaming SVD*, or Sequential Karhunen-Loeve (SKL) algorithm introduced by Levey and Lindenbaum [9], for computing the left singular vectors U of the full SVD via looping over data rectangles, stored on disk, for which each iteration R-SVD is performed to extract the major singular vectors from a QR-decomposition of the rectangle. We show the method’s pseudocode in Algorithm 7.4 and provide a schematic diagram in Figure 7.14. The forget factor $\mathbf{ff}$ is a fraction between 0 and 1 that controls the contribution of each previous set of singular vectors to the next. Algorithm 7.4 partitions calculation of U such that each partition of columns of the full matrix can be introduced into memory sequentially. The complexity of computing the left singular values this way is comparable to the complexity of the full SVD, except

Algorithm 7.4 Pseudocode implementation which demonstrates the algorithm for streaming SVD, originally conceived by Levey and Lindenbaum [9]. Algorithm adapted from Maulik and Mengaldo [10].

Initialization (rectangle $k = 0$): Compute QR-decomposition $A_0 = Q_0 R_0$
 Compute SVD $R_0 = U'_0 D^{(k=0)} V_0^T$ (truncating at desired rank)
 Compute $U^{(k=0)} = Q_0 U'_0$ (where U^k corresponds to the full SVD modes at update k)
for each rectangle $k \geq 1$ and matrix A_k **do**
 Append new data A_k to previous basis (ff $\cdot U_{k-1} D_{k-1} | A_k = A'_k$)
 Compute QR-decomposition $A'_k = Q_k R_k$
 Compute SVD $R_k = U'_k D^{(k=0)} V_k^*$ (truncating at desired rank)
 Update singular vectors $U^{(k)} = Q_k U'_k$
end for

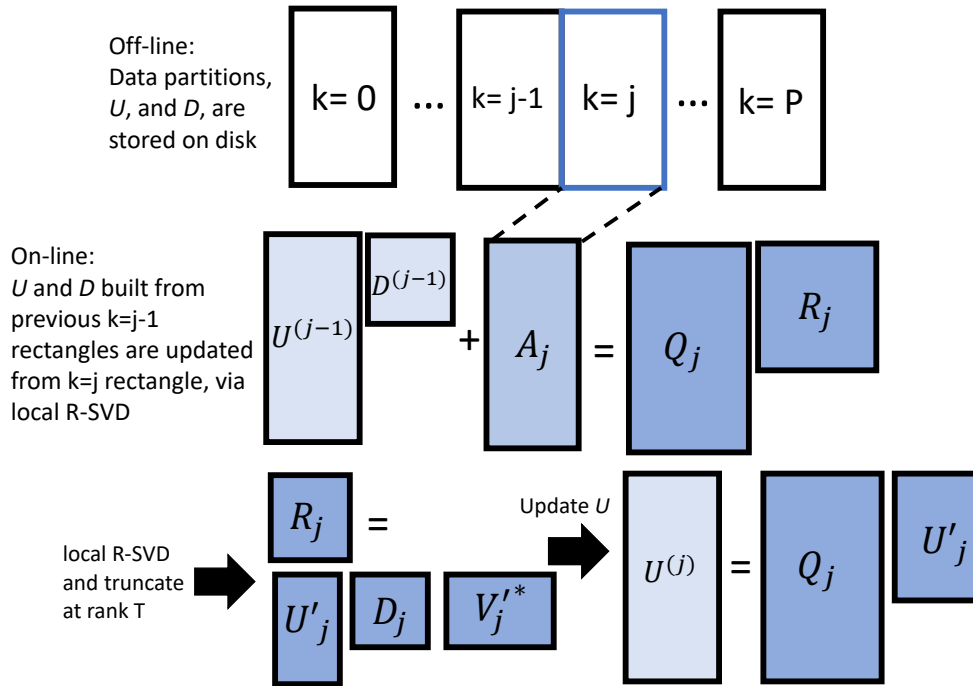


Figure 7.14 Schematic representation of the SKL algorithm [9]. In the on-line stage, the SVD modes and singular values are updated according to the local data at $k = j$. The updated modes and singular values can be written to disk and unloaded from memory until the next update.

that the memory constraint has been relaxed. Note that the modes produced by the SKL algorithm are robust to noise introduced by concatenation of new data [9]. In general, one does not worry about significant singular vectors being lost in the concatenation. This fact has been tested and proven by Levey and Lindenbaum [9]. An intuitive interpretation of this fact may be that for a “good” random initialization point of the stream, stochastic introduction of new data tends to

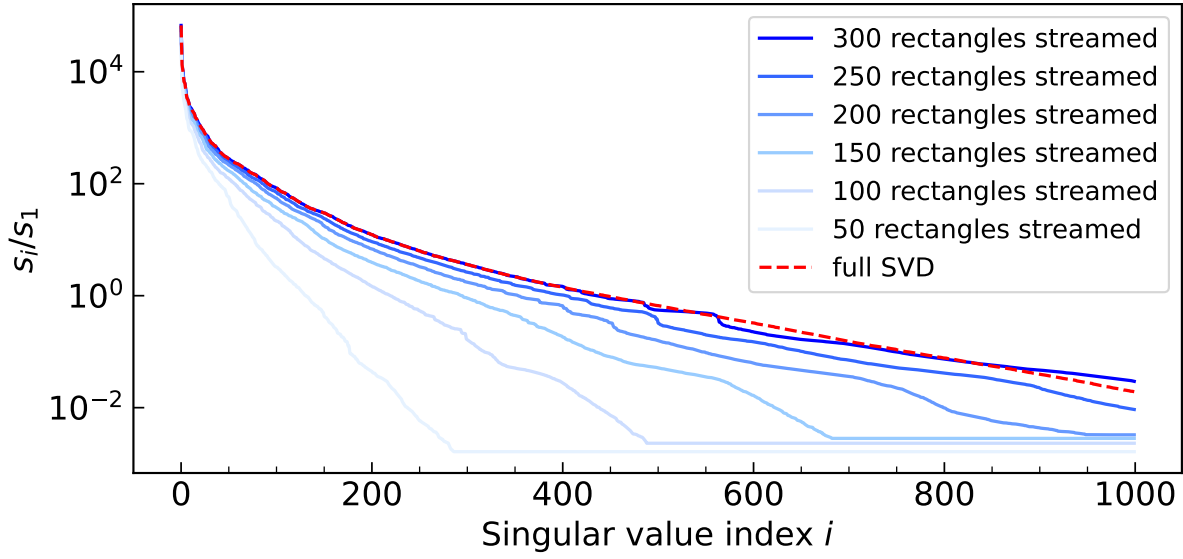


Figure 7.15 The singular value spectrum of 300 training points corresponding to ^{16}O , in an $\text{eMax} = 6$ harmonic oscillator basis size and oscillator width $\hbar\omega = 20$.

suppress modes which are significantly different from previous modes, in an orthogonality sense. Additionally, the rate of convergence of the streaming modes depends on the underlying density of the low rank structure in the streaming data; that is, streaming data which exhibits a quickly decaying singular value spectrum will converge faster under the SKL algorithm than streaming data for which the spectrum decays more slowly. Figure 7.15 displays the singular value spectrum convergence of the streaming SVD as new rectangles are incorporated. Note the dashed line which displays the spectrum of the equivalent full SVD, i.e. performing SVD on the full dataset at once. The largest 25-50 singular values converge the fastest in this truncation scheme. This means that, depending on the desired truncation rank in the streaming SVD, we may not need to stream every available data point to arrive at a satisfactory low rank approximation of the dataset.

7.7.1 A Parallel Implementation of the Streaming SVD

While current parallel implementations of streaming SVD focus on parallelization of the QR algorithm itself [2, 10], we present a different approach that focuses on increasing the throughput of the SKL algorithm. We simply call this approach parallel-SKL, or PSKL. We start with the fact that the SKL algorithm makes no assumptions on the ordering of the data partitions stored on disk,

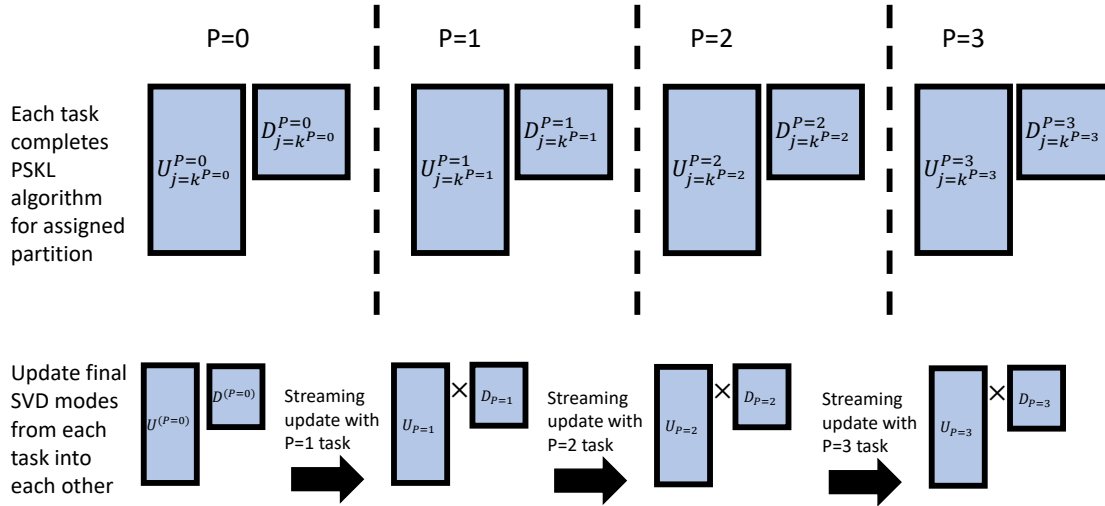


Figure 7.16 Diagram of a 4-task parallel-SKL algorithm. Each task P is assigned a partition of data indices, which together span the total range of the target data. Then, each task P computes its own streaming SVD of its data partition. Finally, the modes computed from each task are updated into each other, sequentially.

nor does the on-line update of SVD modes in Figure 7.14 account for which data partitions are used, or when they are ingested. However, the SVD modes at update step k do contain information about all data partitions *before* step k (to the extent controlled by the forget-factor ff), in the form of iterative projections of old information onto new information. Therefore, the parallel structure we present is to simply partition the ingestion of off-line data over several, parallel tasks, each of which perform their own streaming SVD of the data they are assigned. Once each task is completed, these separate, parallel SVD modes may be sequentially updated into each other, generating a single set of SVD modes which contain information from all tasks. Figure 7.16 displays a diagram which illustrates the PSKL algorithm. Note that the information required for sequential updating after each task is completed is stored in the matrix multiplication of $\mathbf{U}_P \mathbf{D}_P$.

Figure 7.17 confirms that information from each separate task is preserved during the final combination step. The PSKL singular value spectrum, computed using 2 tasks in this example, is consistent with the spectrum of a full SVD calculation.

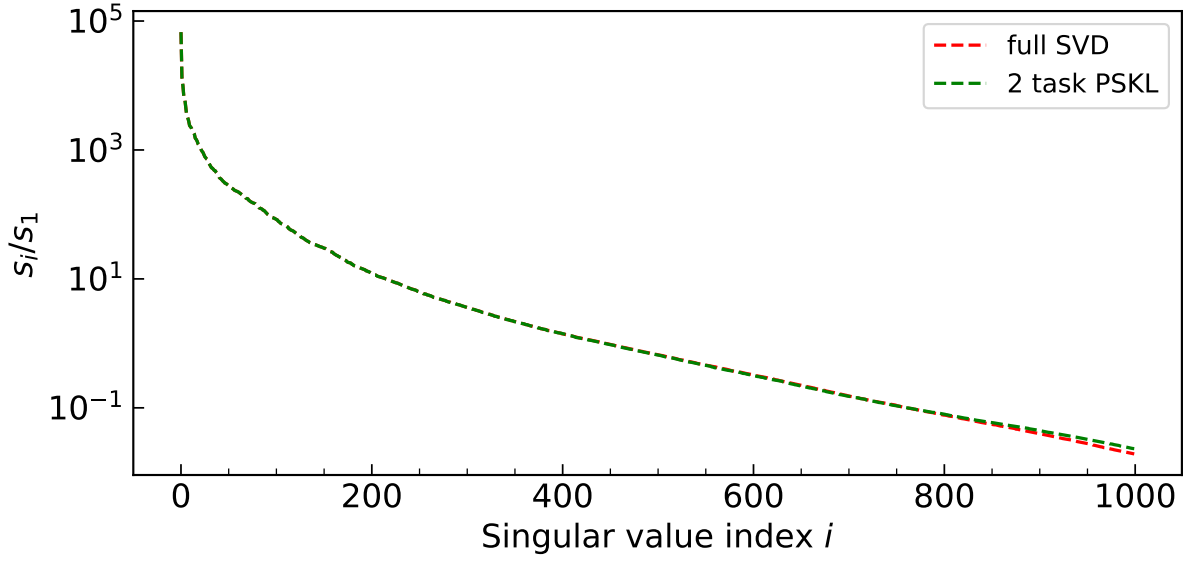


Figure 7.17 Singular value spectrum comparison between the full SVD and a 2-task PSKL run, for a training matrix of 300 points corresponding to ^{16}O , in an $\text{eMax} = 6$ harmonic oscillator basis size and oscillator width $\hbar\omega = 20$.

Assuming each task has access to resources identical to the sequential version, then the PSKL algorithm performance scales at least linear in the number of tasks. Of course, this resource duplication across tasks also implies linear scaling in the total memory consumption of the algorithm. However, the performance gains may be worth the memory costs on systems where large amounts of memory are available.

7.8 Outlook

In this chapter, we have presented a method for constructing an IMSRG(2) emulator, using DMD as the driving mechanism for predicting the IMSRG dynamics. We have shown that DMD may be used to forecast a single IMSRG(2) evolution (robust to different interactions and nuclei), reducing the number of polynomial-cost ODE solver iterations required for a converged result. We have also shown that DMD may be used to construct an interpolation engine for the purpose of training a model that emulates the flow for any set of parameters that characterize the Hamiltonian including resolution scales and cutoffs. In this case, we desire a rich set of IMSRG(2) evolutions in order to populate the IEM training set with high-fidelity DMD operators. The larger the training

set, the more difficult data ingestion becomes; we have presented a solution to this problem using PCA to reduce the training set, in tandem with the SKL algorithm for streaming SVD.

With all these improvements combined, we are able to construct a confident model of the IMSRG(2) evolutions of a target nucleus and the associated Hamiltonian, parameterized by the LECs in chiral EFT. In Chapter 8, we present a practical application of RBF-rIEM for IMSRG(2) applied to nuclear Hamiltonians: namely, global sensitivity analysis of energies to variations of the LECs. This analysis is important to determining the contribution of the variance of particular LECs to the total variance in the evolved energy. In future work, this sensitivity knowledge may let us generate more efficient training bases for higher fidelity IEMs, as well as construct efficient sampling distributions for a proper statistical uncertainty quantification of observables that are computed with IMRSG methods.

BIBLIOGRAPHY

- [1] Francesco Andreuzzi, Nicola Demo, and Gianluigi Rozza. *A dynamic mode decomposition extension for the forecasting of parametric dynamical systems*. 2021. DOI: 10.48550/ARXIV.2110.09155. URL: <https://arxiv.org/abs/2110.09155>.
- [2] Austin R. Benson, David F. Gleich, and James Demmel. “Direct QR factorizations for tall-and-skinny matrices in MapReduce architectures”. In: *2013 IEEE International Conference on Big Data*. 2013, pp. 264–272. DOI: 10.1109/BigData.2013.6691583.
- [3] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. DOI: 10.1017/9781108380690.
- [4] Steven L. Brunton et al. *Modern Koopman Theory for Dynamical Systems*. 2021. arXiv: 2102.12086 [math.DS].
- [5] J. Hoppe et al. “Importance truncation for the in-medium similarity renormalization group”. In: *Phys. Rev. C* 105 (3 Mar. 2022), p. 034324. DOI: 10.1103/PhysRevC.105.034324. URL: <https://link.aps.org/doi/10.1103/PhysRevC.105.034324>.
- [6] Quincy A. Huhn et al. *Parametric Dynamic Mode Decomposition for Reduced Order Modeling*. 2022. DOI: 10.48550/ARXIV.2204.12006. URL: <https://arxiv.org/abs/2204.12006>.
- [7] B. O. Koopman. “Hamiltonian Systems and Transformation in Hilbert Space”. In: *Proceedings of the National Academy of Sciences* 17.5 (1931), pp. 315–318. DOI: 10.1073/pnas.17.5.315. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.17.5.315>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.17.5.315>.
- [8] J. Nathan Kutz et al. *Dynamic Mode Decomposition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2016. DOI: 10.1137/1.9781611974508. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611974508>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611974508>.
- [9] A. Levey and M. Lindenbaum. “Sequential Karhunen-Loeve basis extraction and its application to images”. In: *IEEE Transactions on Image Processing* 9.8 (2000), pp. 1371–1374. DOI: 10.1109/83.855432.
- [10] Romit Maulik and Gianmarco Mengaldo. *PyParSVD: A streaming, distributed and randomized singular-value-decomposition library*. 2021. arXiv: 2108.08845 [cs.MS].
- [11] Peter Schmid and Joern Sesterhenn. “Dynamic Mode Decomposition of numerical and experimental data”. In: *APS Division of Fluid Dynamics Meeting Abstracts*. Vol. 61. APS Meeting Abstracts. Nov. 2008, MR.007, MR.007.

- [12] Peter J. Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28. DOI: 10.1017/S0022112010001217.
- [13] Jonathan H. Tu et al. “On dynamic mode decomposition: Theory and applications”. In: *Journal of Computational Dynamics* 1.2 (2014), pp. 391–421. ISSN: 2158-2491. DOI: 10.3934/jcd.2014.1.391. URL: /article/id/1dfebc20-876d-4da7-8034-7cd3c7ae1161.
- [14] R. B. Wiringa, V. G. J. Stoks, and R. Schiavilla. “Accurate nucleon-nucleon potential with charge-independence breaking”. In: *Phys. Rev. C* 51 (1 Jan. 1995), pp. 38–51. DOI: 10.1103/PhysRevC.51.38. URL: <https://link.aps.org/doi/10.1103/PhysRevC.51.38>.

CHAPTER 8

GLOBAL SENSITIVITY ANALYSIS USING EMULATION OF THE IMSRG

8.1 Introduction

In this chapter, we present a practical application of the parametric IMSRG emulator: A global sensitivity analysis (GSA) of energies to variations in the low-energy constants (LECs) which define the nuclear interactions constructed from Chiral EFT, as explained in Chapter 4.2. We present several GSAs for selected nuclei at interaction chiral orders $N^2\text{LO}$ (including NN and 3N interactions) and $N^3\text{LO}$ (using NN forces only). Our strategy for GSA mirrors the strategy of Ekström and Hagen [3], so that our results for ^{16}O with the NNLOsat interaction should be directly comparable. Note that this GSA would not be feasible without the improvements achieved by the reduced-basis interpolation strategy of Chapter 7.6.

The sensitivity analysis was performed with aid from the SALib¹ Python package [8, 6]. We follow the Sobol’ approach to global sensitivity analysis [10, 9]. The sensitivity measures the variance in the model output captured by a change in the model parameters. To begin, the total variance D of the model output Y , with respect to the model parameters, is decomposed into a sum of partial variances relative to simultaneous changes in an increasing number of parameter c_i . Denoting the variance for varying a single parameter while keeping all others fixed D_i , for two parameters D_{ij} , etc., the decomposition reads

$$D = \sum_i D_i + \sum_{i,j,i \neq j} D_{i,j} + \cdots . \quad (8.1)$$

The first-order *sensitivity* to parameter c_i is given by

$$S_i = \frac{D_i}{D}, \quad (8.2)$$

and the total sensitivity to the parameter c_i is given by a sum over all orders of sensitivity,

$$S_{T,i} = S_i + S_{ij} + \cdots S_{i\dots N}. \quad (8.3)$$

¹Documentation can be found at <https://salib.readthedocs.io/en/latest/index.html>.

The first-order sensitivity integral with respect to parameter c_i is [9]

$$S_i = \frac{\text{Var}_i [E_{\sim \mathbf{i}} [Y | c_i]]}{D}, \quad (8.4)$$

where the expectation value E index $\sim \mathbf{i}$ represents the *set* of all parameters excluding c_i . The total sensitivity integral with respect to the parameter c_i is [9]

$$S_{T,i} = 1 - \frac{\text{Var}_{\sim \mathbf{i}} [E_i [Y | c_{\sim \mathbf{i}}]]}{D}. \quad (8.5)$$

S_i describes the sensitivity of the model output Y to c_i while fixing all other parameters, i.e. $c_{\sim \mathbf{i}}$. $S_{T,i}$ describes the sensitivity of Y to c_i including the first-order effect *plus all higher orders*. Thus, the total sensitivity encodes correlations between parameters in the model output.

In the Sobol' approach to global sensitivity analysis, the integrals in Equations 8.4 and 8.5 are estimated via Sobol' sequence sampling of the model output, which is used for quasi-MC methods. The Sobol' sequence successively divides the parameter space in half, for an optimal number of partitions, such that random samples in each partition optimally cover the full parameter space.

Note that sampling the RBF-rIEM can be very fast, provided the *reduced* model is being sampled, i.e. the model *before* transformation back to the full IMSRG space via $\mathbf{L}_r$ (see Equation 7.26). Once all relevant samples have been collected, the *reduced* samples may be transformed back to the full space in a single batch by applying the learned transformation.

8.2 GSA with a Chiral N3LO Nucleon-Nucleon Interaction

In Figure 8.1, we present a Sobol' sensitivity analysis of the ^{16}O ground-state energy from IMSRG(2) calculations with chiral N3LO NN interactions with cutoff $\Lambda_b = 500$ MeV. Our baseline is the $\text{NN}^3\text{LO}(500)$ interaction by Entem and Machleidt [4], for nucleus ^{16}O . We trained a RBF-r(50)IEM of ^{16}O on 300 DMD operator training points built from 50 observations of corresponding IMSRG(2) evolutions. We used an $e\text{Max} = 12$ harmonic oscillator basis size with oscillator width $\hbar\omega = 20$ Mev. The corresponding LEC training set has been sampled from a Latin hypercube with $\pm 20\%$ relative variations around the standard N3LO LEC set in Table 4.2. The particular contact LECs which were varied appear on the horizontal axis of 8.1. The analysis was performed with

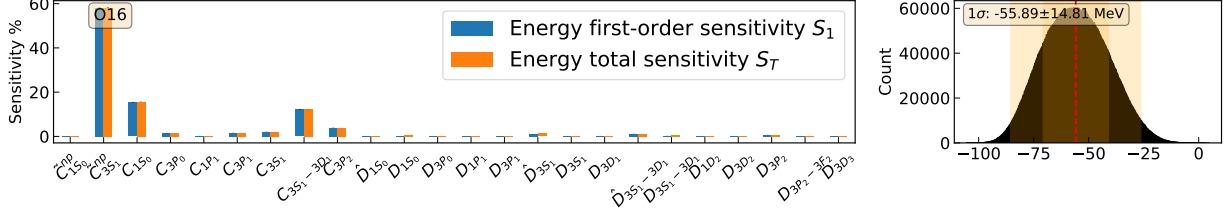


Figure 8.1 First-order and total sensitivity of the IMSRG(2) ground-state energy for ^{16}O energy to variation of the LECs around the standard $\text{N}^3\text{LO}(500)$ interaction of Entem and Machleidt [4]. The left panel contains the sensitivity information per LEC. The right panel plots the total variance in the energy. The shaded region in the variance plot represents the 1σ and 2σ range.

13,631,488 Sobol’ sequence samples of the RBF-r(50)IEM, totaling about 10 minutes of wall-time on a high performance computer, in Figure 7.13. The same number of complete IMSRG(2) calculations, on the same high performance computer, would require approximately 7700 years of compute time. We sample within a $\pm 10\%$ relative variation of the standard N^3LO LECs. The two most sensitive parameters in $\tilde{C}_{3S_1}$ and C_{1S_0} are consistent with the N^2LO sensitivities presented by Ekström and Hagen [3]. The first-order sensitivity matches almost exactly to the total sensitivity in all LECs, meaning that variations in single LECs do not affect the others, at least according to the fitted RBF-r(50)IEM.

The top four most sensitive parameters, in this model, are $\tilde{C}_{3S_1}$, C_{1S_0} , $C_{3S_1-3D_1}$, and C_{3P_2} . Note that we neglect the isospin breaking in the leading 1S_0 contact $\tilde{C}_{1S_0}^{np}$. C_{1S_0} is the subleading contact in the 1S_0 partial wave, so it probably renormalizes the core. $\tilde{C}_{3S_1}$ and $C_{3S_1-3D_1}$ are in the deuteron partial waves — the mixed waves is impacted by the strength of the tensor interaction. The $S=1$ P-waves (C_{3P_x}), and C_{3P_2} in particular, probably renormalize short-range spin-orbit physics.

8.3 GSA with Chiral NNLO Two- Plus Three-Nucleon Interactions

In this section, we present a Sobol’ sensitivity analysis of the Chiral NNLO LECs by Ekström et al. [2, 1], up to the 3B forces. Here we have performed the analysis using interactions constructed from “delta-full” and “delta-less” chiral EFT. In this “delta-full” $\Delta\text{NNLO}_{\text{GO}}(394)$ interaction we explicitly add fixed resonance saturation values to the c_3 and c_4 LECs in the Fujita-Miyazawa force, because the low cutoff allows us to absorb into the adjust of c_3 and c_4 , without the need to

implement the long-range term explicitly. The LECs c_3, c_4 are adjusted by an amount c_3^Δ, c_4^Δ [1],

$$c_3^\Delta = -2c_4^\Delta = \frac{4h_A^2}{9\delta} = -2.972246 \text{ GeV}^{-1}, \quad (8.6)$$

for final values that read,

$$\begin{aligned} c_3^{3N} &= c_3 + c_3^\Delta = -3.61 \\ c_4^{3N} &= c_4 + c_4^\Delta = +2.44 \end{aligned} \quad (8.7)$$

Additionally, we must include the c_2 πN LEC in the variation, for a total of 19 variation parameters (see Table 4.1, Chapter 4.2). The c_3 and c_4 LECs in the “delta-less” $\text{NNLO}_{\text{sat}}(450)$ interaction, which read [2]

$$\begin{aligned} c_3 &= -3.93 \\ c_4 &= 3.77, \end{aligned} \quad (8.8)$$

take on similar values to the resonance-saturation-adjusted c_3^{3N} and c_4^{3N} in the $\Delta\text{NNLO}_{\text{GO}}(394)$ interaction after introduction of the Fujita-Miyazawa term. The $\text{NNLO}_{\text{sat}}(450)$ does not include the c_2 LEC, for a total of 18 variation parameters.

For both interactions, we have trained RBF-r(50)IEMs for each nucleus that was studied. The models were trained on 130 DMD training operators, which were constructed from 50 observations of each corresponding IMSRG(2) evolution. We used an $\text{eMax} = 12$ harmonic oscillator basis size with oscillator width $\hbar\omega = 20 \text{ MeV}$. Every LEC training set has been sampled from a Latin hypercube with $\pm 20\%$ relative around the corresponding standard LEC set for the chosen interaction.

8.3.1 Delta-full NNLO

The trained RBF-r(50)IEM has been Sobol’-sampled 11,010,048 times within a $\pm 10\%$ relative variation around the corresponding standard LEC in Table 4.1. The total wall-time for each Sobol’ sampling routine is on the order of a few minutes, on a high performance computer. A full IMSRG(2) calculation for the same number of Sobol’ samples, on the same high performance computer, would require approximately 6000 years of compute time.

The sensitivity analyses in Figure 8.3 show near consistent parameter sensitivity across the studied neutron-rich Calcium isotopes. The most sensitive parameter is the $\tilde{C}_{3S_1}^{(np)}$. There seems

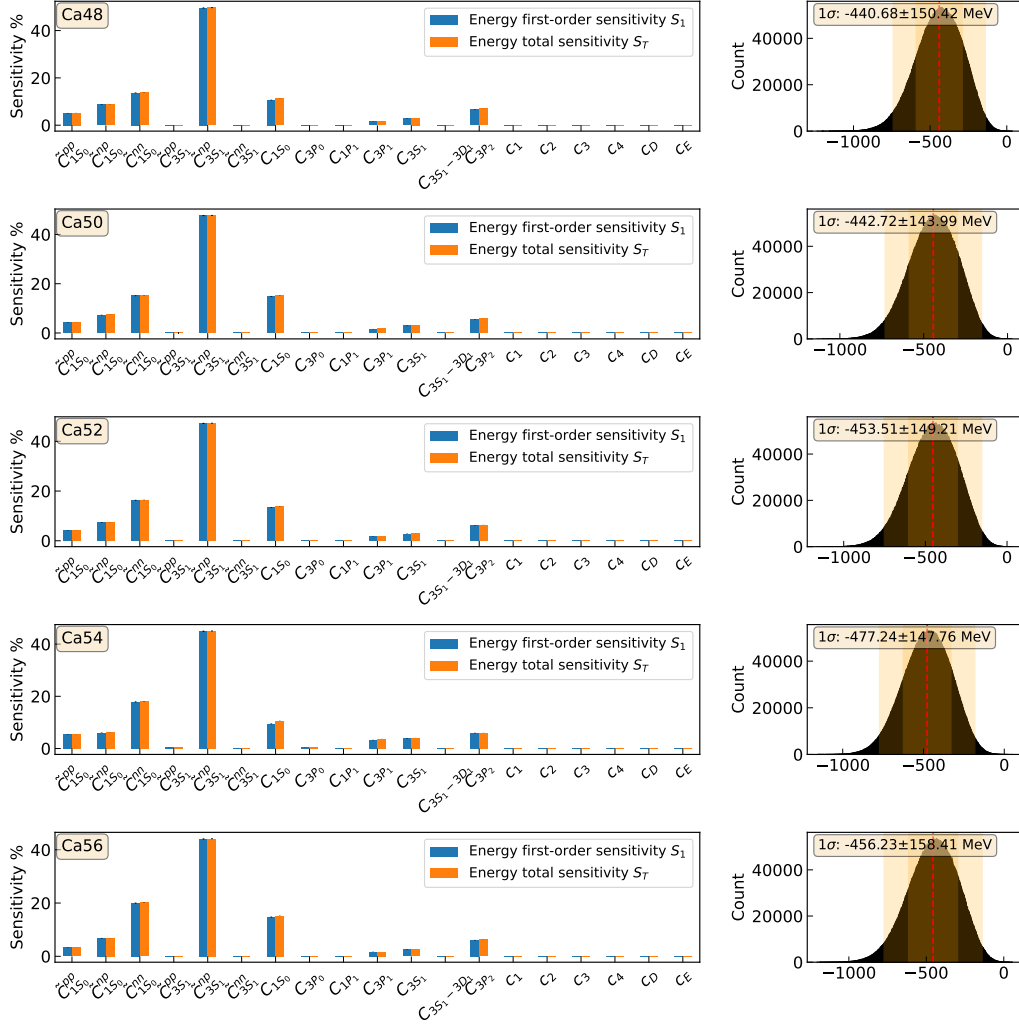


Figure 8.2 Summary of $\Delta\text{NNLO}_{\text{GO}}(394)$ LEC sensitivity as sampled by the RBF-r(50)IEMs corresponding to each nucleus in neutron-rich Calcium isotopes, based on samples of interactions around the $\Delta\text{NNLO}_{\text{GO}}(394)$ baseline. The left column plots of the sensitivities of each nucleus per LEC. The right column plots the total variance in the energy as measured by the model for that nucleus. The shaded region in the variance plot represents the 1σ and 2σ range.

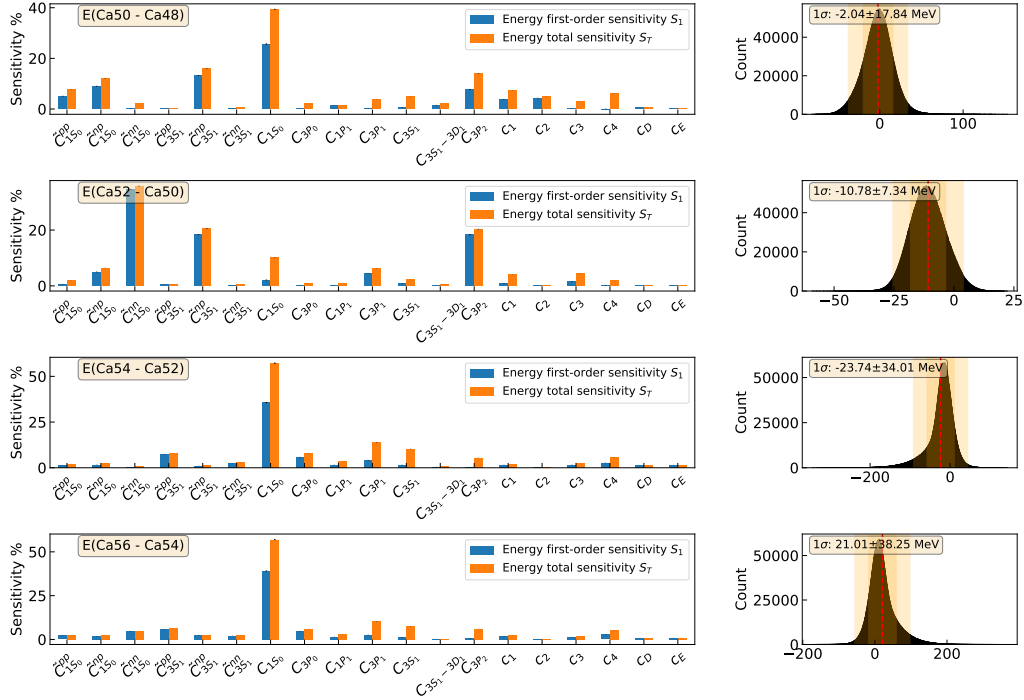


Figure 8.3 Summary of two-neutron separation energies (S_{2n}) as sampled by the RBF-r(50)IEMs corresponding to each nucleus in neutron-rich Calcium isotopes, based on samples of interactions around the $\Delta\text{NNLO}_{\text{GO}}(394)$ baseline. The left column plots of the sensitivities of each S2N per LEC. The right column plots the total variance in the energy as measured by the model for that S2N. The shaded region in the variance plot represents the 1σ and 2σ range.

to be very little dependence on the c_i 's. As in the $\text{NN}^3\text{LO}(500)$ interaction, the LEC sensitivities are fully first-order in the RBF-r(50)IEM model output. In Figure 8.3 we plot the two-neutron separation S_{2n} energies corresponding to the Calcium isotopes presented. We calculate the S_{2n} energy by simply subtracting between the sampled model outputs of two nuclei (carefully ensuring that each subtraction is between two outputs on the same input LEC set), e.g. $E(^{50}\text{Ca} - ^{48}\text{Ca}) = E(^{50}\text{Ca}) - E(^{48}\text{Ca})$, $E(^{52}\text{Ca} - ^{50}\text{Ca}) = E(^{52}\text{Ca}) - E(^{50}\text{Ca})$, and so on. The sensitivity patterns change here, because small differences in the separate nuclear model outputs are enhanced by division of the total variance in the subsequent sensitivity analysis. A major differing feature is that there appears to be second- and higher-order sensitivity effects in play, since the first-order sensitivity does not fully account for the total sensitivity. This is expected, because the S_{2n} energy

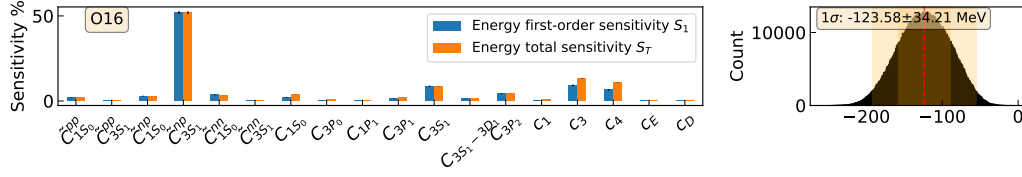


Figure 8.4 Summary of $\text{NNLO}_{\text{sat}}(450)$ LEC sensitivity in ^{16}O as sampled by the RBF-r(50)IEMs around the $\text{NNLO}_{\text{sat}}(450)$ baseline. The left panel plots of the sensitivities per LEC. The right panel plots the total variance in the energy as measured by the RBF-IEM. The shaded region in the variance plot represents the 1σ and 2σ range.

output depends on the outputs of two independent nuclear models.

8.3.2 Delta-less NNLO

Each trained RBF-r(50)IEM has been Sobol'-sampled 1,310,720 times within a $\pm 10\%$ relative variation around the corresponding standard $\text{NNLO}_{\text{sat}}(450)$ LEC set in Table 4.1. The total wall-time for each Sobol' sampling routine is on the order of a few minutes, on a high performance computer. A full IMSRG(2) calculation for the same number of Sobol' samples, on the same high performance computer, would require approximately 750 years of compute time.

The sensitivity analysis for ^{16}O in Figure 8.4 was performed using the same interaction as in Ekström and Hagen [3] (with slightly different variation parameters). We note slight differences between our results and Ekström and Hagen [3], especially in the C_{1S_0} contact, which may be due to differences between our learned models, as well as parametric model implementations. In this RBF-r(50)IEM model, there appears to be some higher-order effects in the c_3 and c_4 contacts which are important. There does not appear to be much dependence on the c_E, c_D contacts relevant to the 3N force.

As in the previous section for $\Delta\text{NNLO}_{\text{GO}}(394)$, in Figure 8.5 we show the sensitivity analyses for several neutron-rich Calcium isotopes as sampled by learned RBF-r(50)IEMs working in the $\text{NNLO}_{\text{sat}}(450)$ interaction. In contrast to the previous section, we find increased importance, as well as higher-order sensitivities, in the c_3, c_4 contacts. These effects carry into the S_{2n} energy

8.4 Discussion and Outlook

The right panel total variances reveal important information about the fitted RBF-rIEM for each nucleus. First of all, since in every case the LEC variation samples at a fixed width around standard LEC sets (Tables 4.1 and 4.2), we would expect the most probable value, e.g. the peak of the distribution, to reflect the IMSRG(2) outcome for the interactions associated with the standard LEC sets. However, there seems to be systematic error in the expected result from each emulator; in particular, the expected energies are consistently overbound relative the IMSRG(2) result (see Table 8.1). The reason seems to be an overbinding bias in the training sets generated to train the RBF-rIEM for each nucleus.

We measured over/underbinding bias by checking the proportion of IMSRG(2) NNLO training points which resulted in an evolved energy outside an arbitrary fixed energy width of $\pm 20\%$ relative to the IMSRG(2) evolved energy using the standard LEC set. Then, an overbound data point is $E_{\text{EMU}} < E_{\text{IMSRG(2)}} - 20\%$, whereas an underbound data point $E_{\text{EMU}} > E_{\text{IMSRG(2)}} + 20\%$. A “safe” data point is inside the 20% tolerance. We found that the training sets for the calcium isotopes in Figures 8.2 and 8.5 typically presented more overbound data points relative to safe and underbound data points. We did not check the bias for the N3LO training points for ^{16}O explicitly (see Figure 8.1) although the safe assumption is that this training set presented an underbinding bias, because the mode of the distribution is above the baseline result from Hergert et al. [5], who report a converged energy of $E = -126.01$ MeV. The takeaway message is that the emulator is successfully capturing the sensitivity in the chosen model space, as expected.

By exploring the robustness of the emulator in the number of training points, varying observables and training space reduction rank, we have shown that these biases are likely not caused by the emulation machinery, but the crude uniform prior from which we sampled the training points. This motivates better choices in the sampling distribution. In the future, we intend to implement a Bayesian history matching approach akin to Vernon et al. [11], which has been used in recent nuclear physics applications [7]. In this approach, a training set is constructed from so-called *non-implausible* interactions by validating the corresponding randomly-sampled LEC sets according to

(A)Ca	$E_{\text{IMSRG}(2)} \text{ (MeV)}$	$E_{\text{exp}} \text{ (Mev)}$
48	-403.351178	-416.001197 (19)
50	-411.5000324	-427.508 (2)
52	-419.1716484	-438.32786 (68)
54	-421.911536	-445.37 (5)
56	-423.3308106	-449.85 (2)
(A)O	$\Delta\text{NNLO}_{\text{GO}}(394) E_{\text{IMSRG}(2)} \text{ (MeV)}$	$E_{\text{NNDC}} \text{ (Mev)}$
16	-108.8349651	127.6193152 (32)

Table 8.1 IMSRG(2) evolved energies according to the $\Delta\text{NNLO}_{\text{GO}}(394)$ interaction. The experimental values from the NNDC are tabulated here for reference, but are not necessarily relevant to the emulator results.

some implausibility criteria. Ideally, we use data for selected observables and expert knowledge to define the implausibility measure in appropriate fashion.

Naively, we could validate a non-implausible LEC set based on the converged energy of IMSRG(2) evolutions for selected nuclei and families of interaction; however, full IMSRG(2) calculations are expensive. Instead, we might choose to determine non-implausibility by evaluating NN scattering phase-shifts which result from each LEC sample. Once the training data is “clean” in the implausibility sense, we may employ importance re-sampling to dynamically update the LEC sampling distributions to more realistic ranges, based on the results from the GSAs in the previous sections. The goal of combining these methods is to progress toward a quantifiable theoretical error on a prediction provided by the emulator.

BIBLIOGRAPHY

- [1] A. Ekström et al. “ Δ isobars and nuclear saturation”. In: *Phys. Rev. C* 97 (2 Feb. 2018), p. 024332. DOI: 10.1103/PhysRevC.97.024332. URL: <https://link.aps.org/doi/10.1103/PhysRevC.97.024332>.
- [2] A. Ekström et al. “Accurate nuclear radii and binding energies from a chiral interaction”. In: *Phys. Rev. C* 91 (5 May 2015), p. 051301. DOI: 10.1103/PhysRevC.91.051301. URL: <https://link.aps.org/doi/10.1103/PhysRevC.91.051301>.
- [3] Andreas Ekström and Gaute Hagen. “Global Sensitivity Analysis of Bulk Properties of an Atomic Nucleus”. In: *Physical Review Letters* 123.25 (Dec. 2019). DOI: 10.1103/physrevlett.123.252501. URL: <https://doi.org/10.1103%5C%2Fphysrevlett.123.252501>.
- [4] D. R. Entem and R. Machleidt. “Accurate charge-dependent nucleon-nucleon potential at fourth order of chiral perturbation theory”. In: *Physical Review C* 68.4 (Oct. 2003). DOI: 10.1103/physrevc.68.041001. URL: <https://doi.org/10.1103%5C%2Fphysrevc.68.041001>.
- [5] H. Hergert et al. “The In-Medium Similarity Renormalization Group: A novel ab initio method for nuclei”. In: *Physics Reports* 621 (Mar. 2016), pp. 165–222. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2015.12.007. URL: <http://dx.doi.org/10.1016/j.physrep.2015.12.007>.
- [6] Jon Herman and Will Usher. “SALib: An open-source Python library for Sensitivity Analysis”. In: *The Journal of Open Source Software* 2.9 (Jan. 2017). DOI: 10.21105/joss.00097. URL: <https://doi.org/10.21105/joss.00097>.
- [7] Baishan Hu et al. “Ab initio predictions link the neutron skin of ^{208}Pb to nuclear forces”. In: *Nature Physics* 18.10 (Aug. 2022), pp. 1196–1200. DOI: 10.1038/s41567-022-01715-8. URL: <https://doi.org/10.1038%5C%2Fs41567-022-01715-8>.
- [8] Takuya Iwanaga, William Usher, and Jonathan Herman. “Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses”. In: *Socio-Environmental Systems Modelling* 4 (May 2022), p. 18155. DOI: 10.18174/sesmo.18155. URL: <https://sesmo.org/article/view/18155>.
- [9] Andrea Saltelli et al. “Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index”. In: *Computer Physics Communications* 181.2 (2010), pp. 259–270. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2009.09.018>. URL: <https://www.sciencedirect.com/science/article/pii/S0010465509003087>.
- [10] I.M. Sobol’. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. In: *Mathematics and Computers in Simulation* 55.1 (2001). The Second IMACS Seminar on Monte Carlo Methods, pp. 271–280. ISSN: 0378-4754. DOI: [https://doi.org/10.1016/S0167-9119\(01\)00060-6](https://doi.org/10.1016/S0167-9119(01)00060-6).

//doi.org/10.1016/S0378-4754(00)00270-6. URL: <https://www.sciencedirect.com/science/article/pii/S0378475400002706>.

- [11] Ian Vernon et al. “Bayesian Emulation and History Matching of JUNE”. in: *medRxiv* (2022). DOI: 10.1101/2022.02.21.22271249. eprint: <https://www.medrxiv.org/content/early/2022/02/22/2022.02.21.22271249.full.pdf>. URL: <https://www.medrxiv.org/content/early/2022/02/22/2022.02.21.22271249>.

CHAPTER 9

CONCLUSIONS AND OUTLOOK

In this work, we have made significant progress toward the theoretical and computational improvement of the IMSRG, as outlined in Chapter 1. Here, we summarize the main scientific contributions presented in this work and outline the future directions that naturally arise from the studies conducted therein.

Chapter 5 presented a tensor network architecture scheme for factorizing data structures relevant to the numerical solution of the IMSRG flow. We showed that the tensor-train decomposition can be used to factorize the occupation factors, which are constant in the flow, thereby reducing the memory cost (e.g. N^2 to N in some cases) of storing on disk. For large single particle bases, this can result in several orders of magnitude of memory savings. We recast the IMSRG flow equations as tensor contractions to be performed by a tensor library, which when combined with factorization of the occupation factors has favorable scaling over naive loop implementations in a toy model. Success here motivates the continued exploration of reduced representations of the IMSRG itself. In the future, we wish to leverage the performance improvements provided by these techniques in the IMSRG production codes. Tensor library integration will also guarantee that the IMSRG codes take advantage of improvements developed by the maintainers of the library, so that the IMSRG calculation is always as efficient as possible. Improvements to the speed and efficiency of the IMSRG production codes would carry into all other advances presented in this work, since they rely on full-scale IMSRG calculations, e.g., for the construction of training data.

Chapter 6 presented our main methodological improvement to the IMSRG, called the *reference state ensemble*. We formulated the reference state ensemble as a mixture of independent one-body density matrices for selected Slater determinant configurations, which are weighted according to tunable probabilities. We showed that we can reduce error due to truncation of induced terms by informing the standard IMSRG operator basis with excited state information using the ensemble—the evidence being how the reference ensemble stabilizes the spectrum of an evolving Hamiltonian,

where a corresponding single reference IMSRG flow would diverge. Ultimately, we use the reference ensemble to attempt to maintain the unitarity of the IMSRG flow as measured by the invariance of the Hamiltonian’s spectrum. Future work in this area will involve a new formulation of the optimization problem using a proxy observable (i.e. not the spectrum) whose evaluation does not require costly exact diagonalizations of the flowing Hamiltonian. Following this breakthrough, we may extend the reference ensemble to realistic nuclear calculations, as well as integrate it into our emulator techniques presented in Chapter 7.

Chapter 7 presented our main computational improvement to the IMSRG, which is a method for emulating the evolution with the help of the so-called Dynamic Mode Decomposition (DMD). We showed that the DMD expansion, informed by direct measurements of the IMSRG-evolved Hamiltonian, may be interpreted as approximating the unitary transformation generated by integration of the IMSRG flow equations. Using this emulator technique, we presented a method for emulating the parametric IMSRG based on an “interpolation engine” powered by DMD. We improved the interpolation method further by effectively performing a Principal Component Analysis (PCA) to reduce the feature space of the IMSRG training points—and we showed, critically, how to implement the PCA calculation via a streaming Singular Value Decomposition in order to mitigate the memory constraints that come with larger realistic IMSRG model spaces. The IE generalized well enough to the full parameter space to perform sensitivity analyses on the parameters of the chiral interactions, the LECs, which we presented in Chapter 8. We showed that a statistically significant number of calculations for GSA could be performed in just a few minutes using our parametric emulator, whereas the same number of full IMSRG(2) calculations would take thousands of years. Future work will attempt to apply the DMD method to the Magnus formulation of the IMSRG, which has the critical advantage that the unitary transformation driving the IMSRG flow is measured directly, instead of implicitly through measurements of the flowing Hamiltonian (as is the case in the standard IMSRG). Emulating the Magnus operator has the advantage that we may easily emulate the flow of observables besides the Hamiltonian, without ever taking direct measurements of their evolution. Additionally, we wish to improve the parametric emulator by informing the

baseline, uniform prior with nonimplausibility criteria, e.g., through history-matching, in order to curtail the training set biases and optimize the useful physical information present in the training space. In this way, we may draw better, physically-motivated conclusions from the sensitivity analyses. Improvements in all these areas will take us closer to proper theoretical error bars on the IMSRG results, which will allow us to use the method to address scientific challenges in nuclear physics, nuclear astrophysics, and the exploration of the fundamental symmetries of the Standard Model of Particle Physics.

APPENDIX A

TENSOR TRAIN DECOMPOSITION OF THE OCCUPATION FACTORS

A.1 Tensor Train Decompositions for Occupation Factors in the IMSRG(2)

$$A(a, b) = n_a - n_b = \begin{bmatrix} n_a & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -n_b \end{bmatrix} \quad (\text{A.1})$$

$$B(a, b) = 1 - n_a - n_b = \begin{bmatrix} 1 - n_a & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -n_b \end{bmatrix} \quad (\text{A.2})$$

$$\begin{aligned} C^{(1)}(a, b, c) &= n_a n_b \bar{n}_c + \bar{n}_a \bar{n}_b n_c = n_a n_b + n_c - n_b n_c - n_a n_c \\ &= \begin{bmatrix} n_a & 1 & 1 & n_a \end{bmatrix} \begin{bmatrix} n_b & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -n_b & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ n_c \\ n_c \\ n_c \end{bmatrix} \end{aligned} \quad (\text{A.3})$$

$$\begin{aligned} C^{(2)}(a, b, c) &= n_a \bar{n}_b \bar{n}_c + \bar{n}_a n_b n_c = n_b n_c + n_a - n_a n_c - n_a n_b \\ &= \begin{bmatrix} n_b & 1 & 1 & n_b \end{bmatrix} \begin{bmatrix} n_c & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -n_c & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ n_a \\ n_a \\ n_a \end{bmatrix} \end{aligned} \quad (\text{A.4})$$

$$D(a, b, c, d) = n_a n_b \bar{n}_c \bar{n}_d = \begin{bmatrix} n_a & 0 \end{bmatrix} \begin{bmatrix} n_b \\ 0 \end{bmatrix} \begin{bmatrix} 1 - n_c & 0 \end{bmatrix} \begin{bmatrix} 1 - n_d \\ 0 \end{bmatrix} \quad (\text{A.5})$$

$$D^{(2)}(a, b, c, d) = \bar{n}_a \bar{n}_b n_c n_d - n_a n_b \bar{n}_c \bar{n}_d =$$

$$= \begin{bmatrix} 1 & -n_a & 1 & -n_a & n_a & n_a \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -n_b & 0 & 0 & 0 \\ 0 & 0 & 0 & n_b & 0 & 0 \\ 0 & 0 & 0 & 0 & n_b & 0 \\ 0 & 0 & 0 & 0 & 0 & n_b \end{bmatrix} \times$$

$$\begin{bmatrix} n_c & 0 & 0 & 0 & 0 & 0 \\ 0 & n_c & 0 & 0 & 0 & 0 \\ 0 & 0 & n_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & n_c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_d \\ n_d \\ n_d \\ 1 \\ 1 \\ n_d \end{bmatrix} \quad (\text{A.6})$$

$$E(a, b, c, d, e, f) = n_a n_b n_c \bar{n}_d \bar{n}_e \bar{n}_f = \begin{bmatrix} n_a & 0 \end{bmatrix} \begin{bmatrix} n_b \\ 0 \end{bmatrix} \begin{bmatrix} n_c & 0 \end{bmatrix} \begin{bmatrix} 1 - n_d \\ 0 \end{bmatrix} \begin{bmatrix} 1 - n_e & 0 \end{bmatrix} \begin{bmatrix} 1 - n_f \\ 0 \end{bmatrix} \quad (\text{A.7})$$

APPENDIX B

MULTI-REFERENCE IMSRG(2) FLOW EQUATIONS

B.1 Multi-Reference IMSRG(2) Flow Equations

The multi-reference IMSRG(2) flow equations include irreducible density matrices (IDMs), introduced in Chapter 2.6, which encode correlations in the reference state that couple to the correlations described by the IMSRG flow. Here, we record the MR-IMSRG(2) flow equations as they are introduced by Hergert [1]:

$$\begin{aligned} \frac{dE}{ds} = & \sum_{ab} (n_a - n_b) \eta_{ab} f_{ba} + \frac{1}{4} \sum_{abcd} (\eta_{abcd} \Gamma_{cdab} - \Gamma_{abcd} \eta_{cdab}) n_a n_b \bar{n}_c \bar{n}_d \\ & + \frac{1}{4} \sum_{abcd} \left(\frac{d}{ds} \Gamma_{abcd} \right) \lambda_{abcd} + \frac{1}{4} \sum_{abcdklm} (\eta_{abcd} \Gamma_{klam} - \Gamma_{abcd} \eta_{klam}) \lambda_{bklcdm}, \end{aligned} \quad (\text{B.1})$$

$$\begin{aligned} \frac{df_{ij}}{ds} = & \sum_a (\eta_{ia} f_{aj} - f_{ia} \eta_{aj}) + \sum_{ab} (\eta_{ab} \Gamma_{biaj} - f_{ab} \eta_{biaj}) (n_a - n_b) \\ & + \frac{1}{2} \sum_{abc} (\eta_{iabc} \Gamma_{bcja} - \Gamma_{iabc} \eta_{bcja}) (n_a \bar{n}_b \bar{n}_c + \bar{n}_a n_b n_c) \\ & + \frac{1}{4} \sum_{abcde} (\eta_{iabc} \Gamma_{deja} - \Gamma_{iabc} \eta_{deja}) \lambda_{debc} + \sum_{abcde} (\eta_{iabc} \Gamma_{bejd} - \Gamma_{iabc} \eta_{bejd}) \lambda_{aecd} \\ & - \frac{1}{2} \sum_{abcde} (\eta_{iajb} \Gamma_{cdae} - \Gamma_{iajb} \eta_{cdae}) \lambda_{cdbe} + \frac{1}{2} \sum_{abcde} (\eta_{iajb} \Gamma_{bcde} - \Gamma_{iajb} \eta_{bcde}) \lambda_{acde}, \end{aligned} \quad (\text{B.2})$$

$$\begin{aligned} \frac{d\Gamma_{ijkl}}{ds} = & \sum_a (\eta_{ia} \Gamma_{ajkl} + \eta_{ja} \Gamma_{iakl} - \eta_{ak} \Gamma_{ijal} - \eta_{al} \Gamma_{ijka} - f_{ia} \eta_{ajkl} - f_{ja} \eta_{iakl} + f_{ak} \eta_{ijal} + f_{al} \eta_{ijka}) \\ & + \frac{1}{2} \sum_{ab} (\eta_{ijab} \Gamma_{abkl} - \Gamma_{ijab} \eta_{abkl}) (1 - n_a - n_b) \\ & + \sum_{ab} (n_a - n_b) ((\eta_{iakb} \Gamma_{jbla} - \Gamma_{iakb} \eta_{jbla}) - (\eta_{jakb} \Gamma_{ibla} - \Gamma_{jakb} \eta_{ibla})). \end{aligned} \quad (\text{B.3})$$

Note that these equations reduce to the single reference flow equations, in Equation 3.18, when we set the IDMs equal to zero. In contrast to the regular IMSRG(2), the MR-IMSRG(2) naively

scales as $O(N^7)$ when the term proportional to the three-body IDM is included in the zero-body flow equation B.1. For common types of reference states, we can often exploit the specific structure of the IDMs to reduce the scaling of the associated terms in the flow equations, and thereby recover the same $O(N^6)$ computational cost as in the conventional IMSRG(2). Evolving the MR-IMSRG(2) flow equations usually requires one to two orders of magnitude more integration steps [1].

BIBLIOGRAPHY

- [1] H Hergert. “In-medium similarity renormalization group for closed and open-shell nuclei”. In: *Physica Scripta* 92.2 (Dec. 2016), p. 023002. ISSN: 1402-4896. DOI: 10.1088/1402-4896/92/2/023002. URL: <http://dx.doi.org/10.1088/1402-4896/92/2/023002>.

APPENDIX C

CONTRIBUTED CODE PROJECTS

In this appendix, we record the code projects that were created, developed, or improved upon as contribution to this thesis. These code projects were also used in the studies presented within the main body of this work.

C.1 IMSRG Codes

The codes in Table C.1 were developed to implement the IMSRG(2) flow of schematic Hamiltonians like the P3H model. One project was written in Python, the other was written in C++. Although originally developed for the P3H model, both codes have been designed with modularity in mind; in principle, one can apply either code to a different model Hamiltonian or Observable with the addition of a new Hamiltonian or Observable class that implements the required features. In chapter 6, this functionality was used in the Python code to evolve and evaluate the total spin-squared operator $\hat{S}^2$ alongside the P3H Hamiltonian.

The Python project Tensor-factor IMSRG (TFIMSRG) outsources tensor contraction at the backend to the library TensorNetwork [2]. The C++ project Tensorized C++ IMSRG (TCIMSRG) employs the Tensor Algebra Compiler (TACO) [1] as a backend for tensor contraction computation. Both codes implement the tensor-train decomposition of occupation factors described in Chapter 5.3 and presented explicitly in Appendix A. The summations for the IMSRG flow equations have been converted to tensor contractions and computed with these libraries. Future work may include the extension of TCIMSRG with a new, distributed version of TACO called Distributed Tensor Algebra Compiler (DISTAL) [3], as well as the deployment of the tensor network framework in a new IMSRG production code.

The P3H model IMSRG codes in Table C.1 are available at <https://github.com/davis9ja/tfimsrg> and <https://github.com/davis9ja/tcimsrg>. The IMSRG production code, which implements the IMSRG for realistic nuclear Hamiltonians, is available on request to the author, H. Hergert.

Project Name	Language	Dependencies	Description
tfimsrg	Python	PyCI, TensorNetwork ¹	Implements the IMSRG(2) flow for the P3H model. Object oriented approach. Machinery for IMSRG(3) implemented, but not ready for experiment.
tcimsrg	C++	dmcpp, TACO ²	Implements the IMSRG(2) flow for the P3H model. Object oriented approach.
IMSRG	C	MEQ, ME3BX, HFB, GSL ³ , SUNDIALS ⁴	Feature-complete production code project which was updated with emulation functionality as part of this dissertation.

Table C.1 Table of IMSRG codes developed for this dissertation. The listed dependencies are non-standard libraries or code projects which must be installed and are listed in Tables C.2 and C.3.

C.2 Full Configuration Interaction Code for Schematic Hamiltonians

This code implements the Full Configuration Interaction approach, i.e., exact diagonalization, for schematic Hamiltonians, as described in Chapter 2.10. The Hamiltonian is fully constructed and diagonalized in the many-body basis. This is often possible for a schematic model like the P3H model.

For realistic interactions, the model space dimensions required for the FCI approach quickly become prohibitive, even if the sparsity of the Hamiltonian is exploited and computations are limited to a few low-lying states, e.g., through the use of Lanczos or Arnoldi methods.

In the present work, we used this code for diagnostic purposes in IMSRG calculations with reference ensembles, as discussed in Chapter 2.10. The code was also used to study the structure of exact eigenstate of the P3H Hamiltonian as a function of its parameters (e.g., for the determination of phase diagrams of this model).

We developed two FCI implementations for this work: one in Python and one in C++. The Python code, located at https://gitlab.msu.edu/imsrg/pyci/-/tree/jacob-dev/imsrg_ci is forked from

¹Tensor contraction library that wraps TensorFlow, among other backends: <https://github.com/google/TensorNetwork/> [2]

²Tensor Algebra Compiler (TACO) which performs efficient tensor contraction in C++: <https://github.com/tensor-compiler/taco> [1]

³GNU Scientific Library, <https://www.gnu.org/software/gsl/>

⁴SUite of Nonlinear and Differential/ALgebraic equation Solvers, <https://github.com/LLNL/sundials>

a rudimentary version originally developed by A. Zitzelberger, another student in the working group. The C++ code is located at <https://github.com/davis9ja/cci>.

Project Name	Language	Dependencies	Description
PyCI	Python	None	Constructs the P3H Hamiltonian in a Slater determinant basis via FCI.
CCI	C++	DMCPP	Constructs the P3H Hamiltonian in a Slater determinant basis via FCI.

Table C.2 Table of FCI codes developed for this dissertation. Refer to Chapter 2.10 for information on FCI.

C.3 Density Matrix Codes

Both P3H IMSRG codes in Table C.1 implement reference state ensemble functionality, as well as the normal ordering with respect to exact FCI eigenstates. This requires the computation of density matrices and their use in normal ordering schemes. The following codes may be used to generate one-, two-, and three-body density matrices, according to a user-specified Slater determinant basis, for use with reference state ensemble studies. The DMCPP code is located at <https://github.com/davis9ja/dmcpp>.

Project Name	Language	Dependencies	Description
PyCI	Python	None	The <code>density_matrix</code> module may be exported to compute density matrices independent of FCI use.
DMCPP	C++	None	Compute density matrices for specified SD basis. Compiled as a static library and intended for import to IMSRG C++ code.

Table C.3 Table of density matrix codes developed for this dissertation. These codes are necessary for generating the density matrices required for generalized normal-ordering.

C.4 IMSRG Emulation using Dynamic Mode Decomposition

The functionality for emulating an IMSRG flow using Dynamic Mode Decomposition (DMD) is wrapped into the IMSRG production code as well as TCIMSRG. For independent implementations

and parametric emulation, the package IMSRG-EMU was developed as part of this dissertation. The package is written in Python and provides a simple command-line interface for running emulations on archived data; however, the classes which perform the emulations can be exported to user Python codes for additional functionality and independent experimentation. The IMSRG-EMU code is located at https://github.com/davis9ja/imsrg_emu.

Project Name	Language	Dependencies	Description
IMSRG-EMU	Python	None	Provides interface for computing a standard, projected DMD emulation on an input dataset (Chapter 7.2) as well as computing a parametric emulation with training on an appropriate training set (Chapter 7.3).

Table C.4 Table of IMSRG emulation codes, using DMD, developed as contribution to this dissertation work.

C.5 Full Table of Contributed Codes

For reference, we have compiled all contributed codes outlined in Appendix sections C.1 through C.4 into a single table. Refer to each code’s user documentation for a complete description of their syntax and usage.

Project Name	Language	Dependencies	Location
TFIMSRG	Python	PyCI, TensorNetwork	https://github.com/davis9ja/tfimsrg
TCIMSRG	C++	dmcpp, TACO	https://github.com/davis9ja/tcimsrg
IMSRG	C	MEQ, ME3BX, HFB, GSL, SUNDIALS	Available upon request to the author, H. Hergert
PyCI	Python	None	https://gitlab.msu.edu/imsrg/pyci/-/tree/jacob-dev/imsrg_ci
CCI	C++	DMCPP	https://github.com/davis9ja/cci
DMCPP	C++	None	https://github.com/davis9ja/dmcpp
IMSRG-EMU	Python	None	https://github.com/davis9ja/imsrg_emu

Table C.5 Complete list of code projects and contributed extensions developed as part of this dissertation.

BIBLIOGRAPHY

- [1] Fredrik Kjolstad et al. “The Tensor Algebra Compiler”. In: *Proc. ACM Program. Lang.* 1.OOPSLA (Oct. 2017), 77:1–77:29. ISSN: 2475-1421. DOI: 10.1145/3133901. URL: <http://doi.acm.org/10.1145/3133901>.
- [2] Chase Roberts et al. *TensorNetwork: A Library for Physics and Machine Learning*. 2019. arXiv: 1905.01330 [physics.comp-ph].
- [3] Rohan Yadav, Alex Aiken, and Fredrik Kjolstad. “DISTAL: The Distributed Tensor Algebra Compiler”. In: *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation*. PLDI 2022. San Diego, CA, USA: Association for Computing Machinery, 2022, pp. 286–300. ISBN: 9781450392655. DOI: 10.1145/3519939.3523437. URL: <https://doi.org/10.1145/3519939.3523437>.