Syracuse University
SURFACE

Dissertations - ALL SURFACE

August 2020

Development and Efficient Implementation of Electron-Hole Interaction Kernel for Investigation of Electronic Excitations in Semiconductor and Metallic Nanoparticles

Peter Francis McLaughlin
Syracuse University

Follow this and additional works at: https://surface.syr.edu/etd

Part of the Physical Sciences and Mathematics Commons

Recommended Citation

McLaughlin, Peter Francis, "Development and Efficient Implementation of Electron-Hole Interaction Kernel for Investigation of Electronic Excitations in Semiconductor and Metallic Nanoparticles" (2020). *Dissertations - ALL*. 1249.
https://surface.syr.edu/etd/1249

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Abstract

The focus of this work is to construct theoretical methods or approaches to tackle computationally challenging problems in quantum chemistry. The primary topic of the work will focus on efficient theories that allow for capturing electron-correlation in semiconductor or metallic nanoparticles. These systems are computationally challenging due to the size of the systems, the number of electrons per atom, and the degeneracy of the particle-hole states. Three methods presented here are aimed at providing novel theoretical methods towards computationally inexpensive electronic excited state calculations. The frequency-dependent geminal-screened electron-hole interaction kernel (FD-GSIK) method provides a real-space approach towards eliminating the virtual space reducing the computational effort present in most excited state methods. Dressed molecular orbital basis is presented to compress the configuration space in metallic nanoparticles, resulting in a cheap multi-reference approach to obtaining electron-correlation. Finally, the moment generated molecular orbital basis is derived to provide an alternative approach that utilizes moments of the reference function to decouple particle and hole states. A secondary focus within this work is to provide efficient Monte Carlo sampling techniques for modeling and integration. Monte Carlo provides an efficient pathway to sampling experimental models and evaluation of complicated integrals which are normally computationally costly. The theoretical developments presented in this work allow for understanding theoretical properties in semiconductor or metallic nanoparticles which normally would be computationally prohibited.

Development and Efficient Implementation of Electron-Hole Interaction Kernel for Investigation of Electronic Excitations in Semiconductor and Metallic Nanoparticles

by

Peter F. McLaughlin

B.S. Ball State University, 2012

M.Phil Syracuse University, 2015

Dissertation

Submitted in partial fulfillment of the requirements for

the degree of Doctor of Philosophy in Chemistry.

Syracuse University

August 2020

Copyright © Peter F. McLaughlin 2020

All rights reserved.

0.1 Acknowledgements

I want to begin by thanking my advisor Ari for his support and guidance through my Ph.D. career. His willingness to educate me in quantum mechanics, mathematics and computer science in my pursuit towards achieving this goal was phenomenal. Ari is a truly remarkable educator who ultimately wants the best for his students, setting them up to be prepared for the rigors of this field. As I approach the end of my education at Syracuse University I feel confident as a scientist and researcher thanks to Ari.

I next want to thank my parents and siblings who have always been there encouraging me on this path. They have been supporting me from afar, curious of my well-being to support in anyway they could. I appreciate the weekly calls from my dad as that gave me time to relax and contemplate my week and future. It is a ritual that I look forward too. Additionally, my mom was always there to accept a phone call whenever I need to decompress or discuss any of my troubles. My parents influence on my life and time here is appreciated beyond words.

I would like to thank my fiancé, Molly for her endless support during this journey. I want to thank her for kind and selfless nature, as my best friend. I am excited to continue our life together as we both venture out into the world beyond our academic careers.

I want to thank my labmates: Dr. Michael Bayne, Dr. Jeremy Scher and Nicole Spanedda that have helped me in my Ph.D. career with the invigorating and thorough discussions that made the days enjoyable. In particular, I want to thank Elyse Kleist who has been a friend through thick and thin since we both started at Syracuse. As well as Dr. Shawn Kowal, for our friendship that provided joy and laughter during our time here. Also, I want to provide special thanks to all the people in the chemistry main office and graduate school at Syracuse University for all their help throughout the years.

Lastly, I would like to thank my committee members, Dr. Pranav Soman, Dr. Weiwei Zheng, Dr. Joseph Chaiken, Dr. Timothy Korter, and Dr. John Franck for taking their time to be a part of my committee. All have you shown interest in my work and have provided me with guidance throughout my Ph.D. career. Either through research or teaching, each of you have impacted my

overall education and Ph.D. career.

There are numerous people who have further contributed to supporting me throughout my Ph.D. career. But I wanted to personally acknowledge all the people mentioned here for their efforts in assisting me to reach this accomplishment.

Contents

0.1 Acknowledgements iv

List of Tables **x**

List of Figures **xi**

1 Introduction **1**

1.1 Scope 1

2 Quantum Chemistry Background **5**

2.1 Hartree-Fock approximation 5

2.2 Electron Correlation 7

3 Second quantization **9**

3.1 Creation and Annihilation operators 9

3.2 Anticommutation relations 11

3.3 Second quantized operators 14

3.4 Particle-hole formulation 15

4 Electronic Excited State Methods **19**

4.1 Configuration Interaction Singles 19

4.2 Linear Response Matrices 22

4.2.1 Casida Equations 22

4.2.2 Limitations 23

4.3 Time-Dependent Density Funtional Theory 24
4.4 GW Bethe-Salpeter equation 25
4.5 Coupled-Cluster methods 26
4.6 Multi-Reference 28
4.7 Summary 29

5 Compact real-space representation of excited states using frequency-dependent explicitly-correlated electron-hole interaction kernel 30

5.1 Introduction 30
5.2 Theory 31
  5.2.1 Construction of frequency-dependent electron-hole-interaction kernel . . . 31
  5.2.2 Determination of the $|Q\rangle$ state 37
  5.2.3 Reducing the number of particle-hole states 38
  5.2.4 Eliminating the AO-to-MO integral transformation 39
  5.2.5 Summary of steps in the overall calculation 40
5.3 Results 42
  5.3.1 Chemical system and system setup 42
  5.3.2 Scaling of FD-GSIK cost 42
  5.3.3 Comparison of FD-GSIK with correlated methods 43
  5.3.4 Electron-hole binding energy 44
  5.3.5 Error analysis of the FD-GSIK method 44
  5.3.6 Summary of results from the FD-GSIK calculations 45
5.4 Discussion 48
  5.4.1 Low-scaling excited state method 48
  5.4.2 Frequency dependent kernel 48
  5.4.3 Absence of adjustable parameters 49
  5.4.4 Future directions and extensions 49
5.5 Conclusions 50

6 Investigation of Dense Manifold of Particle-Hole Excitations in Metallic Nanowires Using r12-Correlated Frequency Dependent Electron-Hole Interaction Kernel 51

6.1 Introduction . . . 51

6.2 Theory . . . 53

6.2.1 Dressed Molecular Orbitals . . . 53

6.2.2 Frequency-dependent geminal-screened electron-hole interaction kernel . . 56

6.2.3 Optimization of $|\tilde{Q}\rangle$ Dressing Coefficients . . . 58

6.3 Results and Discussion . . . 60

6.3.1 Computational Details . . . 60

6.3.2 HOMO-LUMO Gap for Ag Nanowires . . . 60

6.3.3 Collection of single-particle excitations for Ag Nanowires . . . 62

6.3.4 Error analysis and timing data . . . 63

6.4 Conclusions . . . 64

7 Derivation of Moment Generated Molecular Orbital Basis 66

7.1 Introduction . . . 66

7.2 Theory . . . 67

7.3 Conclusion . . . 70

8 Implementation of Permutation Sampling Monte Carlo Integration Method 71

8.1 Introduction . . . 71

8.2 Theory . . . 71

8.3 Object-Oriented Programming . . . 75

8.3.1 Polymorphism . . . 75

8.3.2 Composition and Inheritance . . . 75

8.3.3 Abstraction . . . 76

8.3.4 Encapsulation . . . 77

8.4 C++ Language Features . . . 77

8.5 Conclusions . . . 79

9 Collaborative Projects **81**

9.1 Introduction 81

9.2 Investigation of Optical Properties of Silver Nanoclusters 81

9.2.1 Introduction 81

9.2.2 Results 82

9.2.3 Conclusion 85

9.3 Understanding Mn^{2+} Dopant Migration in CdS/ZnS Quantum Dots 86

9.3.1 Introduction 86

9.3.2 Theory 86

9.3.3 Computational Details 87

9.3.4 Results and Discussion 88

9.3.5 Conclusion 91

10 Conclusions and Future Work **92**

A Monte Carlo Variance Reduction **95**

A.0.1 Antithetic Sampling 95

A.0.2 Stratification 96

A.0.3 Control Variate 97

A.0.4 Importance Sampling 98

A Manual for Permutation Sampling Monte Carlo **99**

Bibliography **185**

Curriculum Vitae 203

List of Tables

4.1 Size of excited state calculation based on basis function and active electrons 24

4.2 Non-linear scaling of various excited state methods. 29

5.1 Comparison of the first excitation energy (eV) obtained from CIS and the FD-GSIK method for PbS nanoparticles. 43

5.2 Comparison of electron-hole binding energies (eV) obtained from EOM-CCSD, ADC(3), and FD-GSIK method for PbS nanoparticles. 43

5.3 Comparison of excitation energy (eV) obtained from FD-GSIK method and GW-BSE[63] for small CdSe nanoparticles. 44

5.4 Maximum and minimum standard deviation (eV) of FD-GSIK excitation energy for each nanoparticle series. 45

5.5 FD-GSIK results for PbS nanoparticles. 46

5.6 FD-GSIK results for PbSe nanoparticles. 46

5.7 FD-GSIK results for CdS nanoparticles. 46

5.8 FD-GSIK results for CdSe nanoparticles. 47

5.9 Largest clusters investigated using LANL2DZ basis. 48

6.1 The standard deviation of $\omega_P^{FD-GSIK}$ from 1+10 runs of the Monte Carlo integration with total single-core CPU computational time. 64

6.2 The standard deviation of $\omega_{\tilde{P}}^{FD-GSIK}$ from 1+10 runs of the Monte Carlo integration with total single-core CPU computational time. 64

9.1 **The number of basis functions for 2nm Ag cluster** 83
9.2 **Computational impact of Ag clusters.** 83

List of Figures

3.1 Diagram of the particle-hole representation. The left side is the many-particle picture for the ground state and excited state. While the right shows the particle-hole picture for the same ground and excited state of the many-particle picture. 16

5.1 Interacting and non-interacting Hamiltonian 34
5.2 The single-core timing comparison of CIS and FD-GSIK calculations for the PbS series ($Pb_4S_4 - Pb_{80}S_{80}$) using LANL2DZ basis. 43
5.3 Electron-hole binding energy obtained using FD-GSIK method for Pb_4S_4-$Pb_{140}S_{140}$, Pb_4Se_4-$Pb_{140}Se_{140}$, Cd_3S_3-$Cd_{87}S_{87}$, and Cd_3Se_3-$Cd_{144}Se_{144}$ as a function of ω_p^0. . . . 45

6.1 Excitation energy of Ag nanowires using FD-GSIK for the HOMO-LUMO gap in comparison to reported TD-DFT results. †[144] 61
6.2 Histogram of molecular orbital energies for Ag_{100} near the HOMO-LUMO gap. 62
6.3 Comparison of the uncorrelated gap to the first-order corrected and FD-GSIK correlated excitation obtained for $|\tilde{P}\rangle$ in Ag nanowires. 63

8.1 Generic C++ class that demonstrates OOP principles. 78
8.2 Main source code for permutation sampling Monte Carlo 80
8.3 Graphical Hierarchy of the class structure from the main source code 80

9.1 (a) Fluorescence intensity of Ag polymer matrix during irradiation. The inset demonstrates the effect of fluorescent intensity with various irradiation intensities. (b) Fluorescence microscopy images at time points 1-8 from (a). (Figure 1)[146] 82

9.2 A plot of the total SCF energy for Ag_{80} at various geometries that were systematically searched for by a scalar radial approach. 84

9.3 A plot which shows a histogram of oscillator strengths for the 10 highest occupied and 10 lowest unoccupied orbitals of silver nanoclusters. 85

9.4 Density function theory calculation of Mn:CdS core QDs dispersed in hexane. (a) The average relative potential the Mn(II) ion experiences at various radial positions in the CdS core and solution and (b) The average relative free energy of the Mn(II) ion, CdS core, and solvent system as the Mn(II) ion moves radial through the system at the lowest (393 K) and highest temperature (573 K) from the experimental study. 89

9.5 (a) The free energy profile along the reaction coordinate for Mn(II) ion migration in the CdS/ZnS and CdS host lattice at 573 K. The green, yellow, and purple regions indicate the core, shell, and solvent regions, respectively for the CdS/ZnS QDs. Two distinct barriers are observed in the CdS/ZnS QDs, first from the core to core/shell interface and second from the core/shell interface to solvent. In contrast, only one barrier from core to solvent, is observed for CdS QDs. (b) Rate constants of reactions for Mn(II) ion in both CdS/ZnS and CdS host lattice systems as a function of temperature. 90

9.6 Density function theory calculation of Mn:CdS/ZnS core/shell QDs dispersed in hexane. (a) The average relative potential the Mn(II) ion experiences at various radial positions in the quantum dot and solution and (b) The average relative free energy of the Mn(II) ion, quantum dot, and solvent system as the Mn(II) ion moves radial through the system at the lowest and highest temperature from the experimental study. 90

A.1 A graph of a normalized Gaussian function. 96

Chapter 1

Introduction

1.1 Scope

Quantum chemistry is an area within chemistry that focuses on obtaining theoretical insight into chemical systems with the use of quantum mechanics. In general, quantum chemistry research lies in obtaining answers accurately and efficiently. Accuracy in quantum chemistry begins by trying to minimize the energy within the system to match the ground-state energy or lowest energy state. But in addition, efficiency is of high importance within quantum chemistry to obtain results as quickly as possible. Where efficiency is measured by the computational effort needed for the theoretical investigation for the particular system. Typically, computational effort refers to the memory requirements, floating point operations, or speed of the calculation, method or algorithm. Throughout this presented work, path to efficient algorithms or computational methods is of key importance towards advancing the field of quantum chemistry.

A foundation within quantum chemistry is the Hartree-Fock method which provides a framework for calculating the energy of the chemical system from a wave-function and the Schrödinger equation. Hartree-Fock method captures about 99% of the total electronic energy of the system, and 95% of the wavefunction thereby contributing as a primary starting point for more advance quantum chemistry methods. [1] The goal of advance methods are to collect the remaining 1% of energy to complete the theoretical understanding of the system, and accuracy. In general, the energy differences in chemical and spectroscopic properties of interest are smaller than the missing 1% of electronic energy therefore it is crucial to develop methods that correct for the remaining energy. The energy that is not obtained in the Hartree-Fock method is the correlation energy and is defined as the energy missing from the Hartree-Fock model to matching the exact energy. The Hartree-Fock method is an introductory method and key baseline for much of quantum chemistry and provides the starting point for much of the presented work and discussed in Chapter 2 along with the correlation energy.

A key aspect to appreciating and understanding the presented work comes from comprehension of the mathematical notation within the theory. I dedicate all of Chapter 3 to second quantization which is a useful representation in quantum mechanics. It provides a simplistic notation towards more complicated equations and is used throughout the presented work. Once I have provided second quantization I will immediately use it to provide brief insight into many popular and commonly used electronic excited state methods, in Chapter 4. The chapter will not be in-depth derivations of each method, but provide a primer to key mathematical expressions that define the methods. This chapter sets up the main motivation of the presented work, along with the problems faced towards obtaining correlation energy in modern quantum chemistry. Background material provided in Chapter 2, Chapter 3, and Chapter 4 will allow for understanding of the presented work in Chapter 5, Chapter 6, Chapter 7, and Chapter 9.

In Chapter 5, the frequency-dependent geminal-screened electron-hole interaction kernel (FD-GSIK) method for describing electron-hole correlation in electronically excited many-electron systems is presented. The FD-GSIK is a parameter-free, first-principle method derived from excited state wavefunction that was both frequency-dependent and r12-explicitly correlated. The FD-GSIK avoids using unoccupied orbitals for kernel construction by performing an infinite-order summation of particle-hole excitation and representing it as a compact real-space operator. It bypasses the computationally demanding steps of evaluation, storage, and transformation of atomic-orbital integrals by directly evaluating molecular orbital integrals in real space using the stratified Monte Carlo method. This chapter will demonstrate and discuss the advantages of this method by presenting excitation and electron-hole binding energies of large nanoparticles including $Pb_{140}S_{140}$, $Pb_{140}Se_{140}$, $Cd_{144}Se_{144}$, and $Cd_{72}S_{72}$. This work was recently accepted for publication in *Journal of Chemical Theory and Computation* DOI:10.1021/acs.jctc.9b01238.

In Chapter 6, a dressed molecular orbital basis approach will be presented. I present a dressed particle-hole basis that allows for compression of the configuration space, allowing for a computationally efficient representation for investigation of multi-reference characteristics in electronic excitations. Multi-reference wavefunctions are approximations to the full configuration interaction (FCI) wavefunction, which is desirable since it provides an exact solution to the many-electron problem. The FCI is in practice computationally prohibitive for many-electron systems due to the number of excited determinants. In general, truncation of the configuration space is limited to the singly or doubly excited determinants for construction of the multi-reference wavefunction. Or by limiting the active space of the occupied or unoccupied orbitals thereby limiting the number of possible excited determinants. The objective of the presented work is to construct a molecular orbital basis that includes information for all particle-hole orbitals as well as compressing the configuration space without the use of truncation. The derived basis will be used in conjunction with the developed frequency-dependent geminal-screened electron-hole kernel (FD-GSIK) method to obtain electron-correlated excitation energies in silver nanowires, Ag_n where $n = 2, 4, 6, 8, 10, 12, 20, 40, 60, 80, 100$. These metallic like systems are of great interest due to there electronic properties but are computationally challenging due to the number of electrons and their highly degenerate molecular orbitals. The presented work aims to provide an approach that can efficiently understand mutli-reference excitation characteristics within this system. This work is in preparation for submission to *The Journal of Chemical Physics*.

In Chapter 7, a derivation for a moment generated molecular orbital basis for obtaining electron-correlation is presented. This approach serves as an alternative to the frequency-dependent geminal-screened electron-hole interaction kernel reported in previous chapters. The operator in the method are the moments of the ground state particle or hole state, enabling a simplistic operator for decoupling the single particle-hole reference state. The benefit of the derived method would be that the operator is not limited by the choice of reference states.

In Chapter 8, I will briefly present the permutation sampling Monte Carlo integration method and focus on the computational implementation of the method. The primary goal of the method is to decouple the dimensionality in integration to provide an efficient and accurate evaluation of numerical integrals. Within quantum chemistry numerical integration is ever-present and a key computational bottleneck within calculations. The main focus within this chapter will be the theory and computer science in efficient implementation of the permutation sampling Monte Carlo method. Providing insight into design decisions towards programming efficient software that is general purpose for future development but versatile in functionality. The use of permutation sampling Monte Carlo towards understanding a 12+1 dimensional Green's function integral is in preparation for journal submission.

In Chapter 9, various theoretical investigations within collaborative projects will be presented. These studies are centered around nanoparticles, which are of huge interest amongst many groups of science, including chemistry, biology and engineering. These nanoparticles that contain a large number of electrons are computationally limited with commercial electronic structure software packages. Therefore theoretical properties are a challenge to obtain and often avoided. The presented work will focus on the development of quantum chemical calculations for these large chemical systems. The first investigation will present the preliminary results for the collaborative investigation of Ag nanoclusters. The Ag nanoclusters were formed in a polymer film by direct-laser writing. It was found that these nanoclusters exhibited unique optical properties. This investigation will present theoretical insight on the HOMO-LUMO gap energies for large Ag nanoclusters to support the experimental observations. The next work will provide an insight into the theoretical works provided for understanding Mn^{2+} dopant migration in CdS/ZnS quantum dots. The goal of the theoretical investigation is to provide insight into dopant migration versus dopant ejection within these systems. The presented work will focus on the theoretical method developed towards understanding the thermodynamics of this molecular system in a computationally efficient manner. The work with Mn^{2+} dopant migration in CdS/ZnS quantum dots was recently accepted into *The Journal of Physical Chemistry Letters*.

Finally, Chapter 10 will cover an overall conclusion to the presented work within this thesis and the future investigations of this work.

Chapter 2

Quantum Chemistry Background

2.1 Hartree-Fock approximation

I begin by briefly introducing the Hatree-Fock method as this is the foundation for most of the work that I have developed. The common nomenclature and conventions for describing many-electron wave functions for quantum chemists is a single Slater determinant, Φ,

$$\Phi = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_N(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(N) & \chi_2(N) & \cdots & \chi_N(N) \end{vmatrix} \tag{2.1}$$

$$= A|\chi_1(1)\chi_2(2)\cdots\chi_N(N)|. \tag{2.2}$$

In Equation 2.1, there is a normalization factor, $\frac{1}{\sqrt{N!}}$, the antisymmetrizer, A, and a spin orbital, $\chi_i(\mu)$. The spin orbital describes both the spatial distribution and spin of the μth electron. Electrons can have either be spin up (α) or spin down (β) within a spin orbital. The Slater determinant is the initial convention of choice for a physical wave function for several crucial reasons. First, the antisymmetry rule is followed by construction of the determinant. The antisymmetry rule is simply that interchange of both space and spin of two electrons must be opposite wave functions,

$$\begin{vmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_N(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(N) & \chi_N(N) & \cdots & \chi_N(N) \end{vmatrix} = - \begin{vmatrix} \chi_1(2) & \chi_2(2) & \cdots & \chi_N(2) \\ \chi_1(1) & \chi_2(1) & \cdots & \chi_N(1) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(N) & \chi_N(N) & \cdots & \chi_N(N) \end{vmatrix} \tag{2.3}$$

Where by interchanging the first two rows in Equation 2.1 yields a negative sign. The next reason is that a Slater determinant follows the Pauli exclusion principle, which is that two electrons can not occupy the same spin orbital. This occurs in a Slater determinant when two columns are exactly equal, which results in the determinant being zero.

The Hatree-Fock equation, in atomic units allows for optimization of the spin orbitals through minimization of the spin orbital energies,

$$\hat{f}\chi_i = \varepsilon_i \chi_i. \tag{2.4}$$

With the Fock operator $\hat{f}$,

$$\hat{f} = \hat{h} + \hat{v}^{HF} \tag{2.5}$$

$$\hat{f} = \hat{h} + \sum_{j=1}^{N_{elec.}} (\hat{J}_j - \hat{K}_j) \tag{2.6}$$

containing the one-electron Hamiltonian,

$$\hat{h} = [\hat{t} + \hat{v}^{ne}] \tag{2.7}$$

$$\hat{h} = -\frac{1}{2}\nabla_i^2 - \sum_{A=1}^{M} \frac{Z_A}{r_{iA}} \tag{2.8}$$

and the two-electron Coulomb ($\hat{J}$) and exchange ($\hat{K}$) operators,

$$\hat{J}_j(1)\chi_i(1) = \langle \chi_j(2) | \frac{1}{r_{12}} | \chi_j(2) \rangle \chi_i(1) \tag{2.9}$$

$$\hat{K}_j(1)\chi_i(1) = \langle \chi_j(2)|\frac{1}{r_{12}}|\chi_i(2)\rangle \chi_j(1). \tag{2.10}$$

The Coulomb and exchange operators are an effective one-electron potential operator called the Hartree-Fock potential,

$$v^{HF} = \sum_{j=1}^{N_{elec.}} (\hat{J}_j - \hat{K}_j) \tag{2.11}$$

The v^{HF} is simply an average potential felt by the μth electron from the other electrons. By minimizing the energy of the spin orbitals a Slater determinant is obtained that is a good first approximation to electronic wave function. [2, 1]

The truth is that the exact wave function, χ_{exact}, for a many-electron interacting system is not a single Slater determinant but a combination of Slater determinants as there are many combinations of Slater determinants which can be formed. The use of one single Slater determinant is a good approximation to the ground state of a many-electron system for which the N electrons occupy the lowest N spin orbitals but neglects correlation affects.

2.2 Electron Correlation

Electron correlation is the correction to the Hartree-Fock wave function that is needed to obtain the exact wave function,

$$\Psi_{\text{exact}} = \Phi_{\text{HF}} + \chi_{\text{Corr}} \tag{2.12}$$

In general, interest lies with trying to obtain the correlation energy, which is the energy required to correct the Hartree-Fock energy, E_{HF} to the exact energy E_{exact},

$$\Delta E_{\text{corr}} = E_{\text{exact}} - E_{\text{HF}} \tag{2.13}$$

Since the instantaneous correlation of the electrons due to their repulsion is missing within the Hartree-Fock model, where the motion of the electron is described by an average field with respect to the other electrons. This limitation of the Hartree-Fock model is easily seen in bond dissociation energies, for which it fails. [2, 1, 3]

Chapter 3

Second quantization

A formulation that will be used throughout the presented work is second quantization. This formulation allows for a compact and convenient notation of representing wave-functions (*i.e.* Slater determinants) and operators, while providing a simplified way of manipulating the functions and operators.

3.1 Creation and Annihilation operators

Let's begin with the a normalized Slater determinant,

$$\Phi = \Phi_{ijk\ldots N} \equiv A|\chi_i\chi_j\chi_k\cdots\chi_N| \equiv |\chi_i\chi_j\chi_k\cdots\chi_N\rangle \equiv |ijk\ldots N\rangle \tag{3.1}$$

with the antisymmetrizer A, and a spin orbital χ, for each particle. The above notations (Equation 3.1) are all equivalent but the last notation will be of primary use throughout this work when utilizing second quantization formulation. This notation is just the simplest notation by dropping χ symbol for every spin orbital. It is important to note that occupancy within second quantization is denoted by the presence of a basis spin orbital in the determinant, for example the spin orbitals $ijk\ldots N$ are occupied in Equation 3.1 and all other spin orbitals are unoccupied.

The creation or annihilation of a particle represent the addition or removal of a spin orbital through the creation and annihilation operators. The creation operator is designated by the dagger symbol, (†) and creates a particle in a specific spin orbital at the beginning of the Slater determinant. The removal of a particle from the determinant is done through the annihilation operator on a specific spin orbital. The notations for creation and annihilation:

$$\text{creation operator for spin orbital } \chi_i, \quad i^\dagger \tag{3.2}$$

$$\text{annihilation operator for spin orbital } \chi_i, \quad i \tag{3.3}$$

These operators can be used on the Slater determinant, $|jk\ldots N\rangle$, as follows:

$$i^\dagger|jk\ldots N\rangle = |ijk\ldots N\rangle \tag{3.4}$$

$$i|ijk\ldots N\rangle = |jk\ldots N\rangle \tag{3.5}$$

thereby creating a particle in i (Equation 3.4), then removing that particle from i (Equation 3.5). It is relevant to point out that creating a particle in an already occupied spin orbital or annihilating a particle that does not exist within the Slater determinant will result in a invalid determinant.

$$i^\dagger|ijk\ldots N\rangle = 0 \tag{3.6}$$

$$i|jk\ldots N\rangle = 0 \tag{3.7}$$

In general a lexical order of the spin orbitals is convenient,

$$|ijk\ldots N\rangle, \quad \text{where} \quad i < j < k < \cdots < N \tag{3.8}$$

but we must discuss the creation and annihilation of a particle that is out of lexical order, for example creation/annihilation of p. The consequences of this is demonstrated below,

$$p^\dagger|ijk\ldots N\rangle = |pijk\ldots N\rangle = (-1)^{\sigma(p)}|ijk\ldots p\ldots N\rangle \tag{3.9}$$

$$p|ijk\ldots p\ldots N\rangle = (-1)^{\sigma(p)}|pijk\ldots N\rangle = (-1)^{\sigma(p)}|ijk\ldots N\rangle \tag{3.10}$$

where $(-1)^{\sigma(p)}$ is needed to maintain the antisymmetry principle in Slater determinants for which an interchange of the spin orbital p with the other spin orbitals $ijk\ldots$ results in a negative sign occurring $\sigma(p)$ times. With all this points we can now define a *vacuum* Slater determinant, denoted as $|\rangle$, which is a determinant with no spin orbitals. This can be constructed through successive operations on a Slater determinant,

$$ijk\ldots N|ijk\ldots N\rangle = |\rangle \tag{3.11}$$

We can also create a Slater determinant from the *vacuum* Slater determinant through successive creation operations. [2, 1]

$$i^\dagger j^\dagger k^\dagger \ldots N^\dagger|\rangle = |ijk\ldots N\rangle \tag{3.12}$$

3.2 Anticommutation relations

With the definition of creation and annihilation operators let us now look at anticommutation relations of these operators. Let us first consider the creation operators $p^\dagger$ and $q^\dagger$. The order of which these two operators can performed are as follows:

$$p^\dagger q^\dagger|ijk\ldots\rangle = |pqijk\ldots\rangle \tag{3.13}$$

$$q^\dagger p^\dagger|ijk\ldots\rangle = |qpijk\ldots\rangle = -|pqijk\ldots\rangle \tag{3.14}$$

This is valid for any Slater determinant for which χ_p or χ_q does not exist in $|ijk\ldots\rangle$, as the product would be zero. We then get the anticommutation relation for the two creation operators as,

$$p^\dagger q^\dagger = -q^\dagger p^\dagger \tag{3.15}$$

$$[p^\dagger, q^\dagger]_+ \equiv p^\dagger q^\dagger + q^\dagger p^\dagger = 0 \tag{3.16}$$

which can be generically written as:

$$[\hat{A},\hat{B}]_+ \equiv \hat{A}\hat{B}+\hat{B}\hat{A} \tag{3.17}$$

is the anticommutator of $\hat{A}$ and $\hat{B}$ and commonly written as $\{\hat{A},\hat{B}\}$. It is important to point out the case of $p=q$ this relation still holds true because $p^\dagger p^\dagger = 0$ (Equation 3.6). The creation anticommutation relation is unique as $[\hat{A},\hat{B}]_+ = [\hat{B},\hat{A}]_+$ which for a normal commutator $[\hat{A},\hat{B}]_+ = -[\hat{B},\hat{A}]_+$.

Now we can move onto the anticommutation relations for the annihilation operator. Let us now consider the annihilation operators p and q:

$$pq|qpijk\ldots\rangle = p|pijk\ldots\rangle = |ijk\ldots\rangle \tag{3.18}$$

$$qp|qpijk\ldots\rangle = -qp|qpijk\ldots\rangle = -p|pijk\ldots\rangle = -|ijk\ldots\rangle \tag{3.19}$$

Note that if χ_p or χ_q are in the interior of the Slater determinant then a sign change will occur as described in Equation 3.10, for both cases of above equations (Equation 3.18,Equation 3.19). If χ_p and/or χ_q are not present within the Slater determinant the resulting determinant is zero. (Equation 3.7) Therefore we arrive at the anticommutator relation for the annihilation operator,

$$pq = -qp \tag{3.20}$$

$$[p,q]_+ = 0 \tag{3.21}$$

The last case to consider is a creation and annihilation operator pair, $p^\dagger q$ when $p \neq q$:

$$p^\dagger q|qijkl\ldots\rangle = p^\dagger|ijkl\ldots\rangle = |pijkl\ldots\rangle \tag{3.22}$$

$$qp^\dagger|qijkl\ldots\rangle = q|pqijkl\ldots\rangle = -q|qpijkl\ldots\rangle = -|pijkl\ldots\rangle \tag{3.23}$$

This operation removes the particle from spin orbital q to put it into spin orbital p but there is difference of a negative sign between the two expressions. Therefore we have the relation:

$$[p^\dagger, q]_+ = 0 \quad (p \neq q) \tag{3.24}$$

Let us now look at the case when $p = q$ in which p does or doesn't exist within the initial Slater determinant,

$$p^\dagger p|pijkl\ldots\rangle = p^\dagger|ijkl\ldots\rangle = |pijkl\ldots\rangle \tag{3.25}$$

$$pp^\dagger|pijkl\ldots\rangle = 0 \tag{3.26}$$

$$p^\dagger p|ijkl\ldots\rangle = 0 \tag{3.27}$$

$$pp^\dagger|ijkl\ldots\rangle = p^\dagger|pijkl\ldots\rangle = |ijkl\ldots\rangle \tag{3.28}$$

Which generalized shows that,

$$\left(p^\dagger p + pp^\dagger\right)|\ldots\rangle = |\ldots\rangle \tag{3.29}$$

For which we obtain the relation:

$$[p^\dagger, p]_+ = [p, p^\dagger]_+ = 1 \tag{3.30}$$

In the case of a creation and annihilation operator pair we then have the final anticommutation relation,

$$[p^\dagger, q]_+ = [p, q^\dagger]_+ = \delta_{pq} \tag{3.31}$$

that contains the Kronecker-delta operator, δ_{pq}. The Kronecker-delta operator is a commonly used operator that differentiates the expression for the instances when $p \neq q$ or $p = q$ resulting in zero or 1, respectively.[2, 1, 4]

In summary there are three anticommutation relations for creation and annihilation operators,

$$[p^\dagger, q^\dagger]_+ = 0 \tag{3.32}$$

$$[p, q]_+ = 0 \tag{3.33}$$

$$[p^\dagger, q]_+ = [p, q^\dagger]_+ = \delta_{pq} \tag{3.34}$$

3.3 Second quantized operators

The benefit with second quantization as presented so far allows is that we have presented a notation for a many-electron wave function without the need of Slater determinants. We can now represent a many-electron wave function simply through creation and annihilation operators, anticommutation relations, and the vacuum state. Our focus now shifts to the second quantization representation of many-particle operators, $\mathscr{O}_1$ and $\mathscr{O}_2$ for one-particle and two-particle operators, respectively. This will allow for representation of theory of many-electron systems without the use of Slater determinants by obtaining the second quantization equivalent of the matrix elements of $\langle K|\mathscr{O}|L\rangle$.

The second quantization expression for $\mathscr{O}_1$, which will be the sum of one-electron operators (Equation 2.5) and for $\mathscr{O}_2$ the total coulomb repulsion between electrons are:

$$\mathscr{O}_1 = \sum_i^N \hat{h}(i) \implies \sum_{ij} \langle i|\hat{h}|j\rangle i^\dagger j \tag{3.35}$$

$$\mathscr{O}_2 = \frac{1}{2}\sum_{i\neq j}^{N^2} \hat{K}(i,j) \implies \frac{1}{2}\sum_{ijkl} \langle ij|kl\rangle i^\dagger j^\dagger lk \tag{3.36}$$

The left expression in Equation 3.35 and Equation 3.36 are the wave function representation of the operators, where the right hand side are the second quantization representation. The eloquence and benefit of second quantization is easily expressed from these expressions in terms of the summation of over the spin orbitals. For the wave function representation operators depend on the summation of each particle, where for second quantization the number of particles is independent of number of particles. This is one of the main benefits of second quantization as the expressions are not limited by number of particles, on the other hand with wave function representation expressions depend on the number of particles. Therefore when dealing with large many-electron systems, the expressions in second quantization are simple while becoming burdensome in wave function representation.[1]

3.4 Particle-hole formulation

We are now able to fully describe theory of many-electron wave functions and operators in terms of second quantization. The reference state that we have constructed up to this point is a vacuum Slater determinant, Equation 3.11 and Equation 3.12. It will be more convenient if we had a fixed reference state that contained occupied spin orbitals, where all other Slater determinants can be constructed from this reference. The reference state that we will use is the Fermi vacuum defined as,

$$|0\rangle \equiv |\Phi_0\rangle = |ijk\dots N\rangle \tag{3.37}$$

The Fermi vacuum is typically the ground state determinant obtained from Hatree-Fock. Within the particle-hole formulation the spin orbitals $i, j, k, \dots N$, in Equation 3.37 are the hole states(occupied orbitals). Particle states (unoccupied orbitals) are defined as $a, b, c, \dots$, the indices $p, q, r, s, \dots$ are used to describe states that can be either hole or particle states.

The construction of the Fermi vacuum can easily be demonstrated through Figure 3.1. Starting with the many-particle picture there is a ground state which contains particles filling spin orbitals χ_i, according to the Pauli exclusion principle. The highest filled orbital is then designated the Fermi level, as ε_F. There is an example of an excited state for the many-particle picture, where the particle below the Fermi level, χ_3 has been excited above the Fermi level, χ_9. Applying the Fermi vacuum as the reference, the ground state in the particle-hole picture appears empty. But in the particle-hole formulation we are only interested in changes from the ground state reference. Therefore the depiction of the excited state in the particle-hole picture contains a particle in χ_9 and a hole (the empty circle) in χ_3 as these are the only changes from the Fermi vacuum.

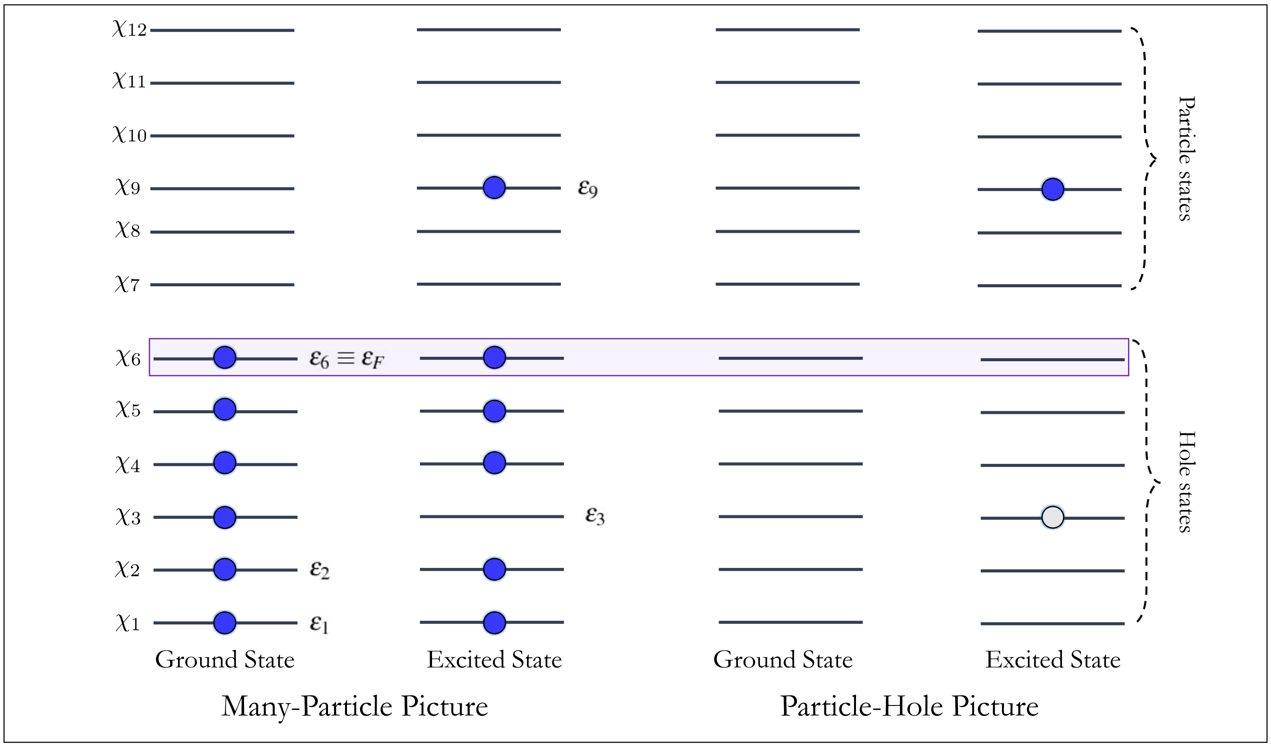


Figure 3.1: Diagram of the particle-hole representation. The left side is the many-particle picture for the ground state and excited state. While the right shows the particle-hole picture for the same ground and excited state of the many-particle picture.

This now allows us to describe excited state determinants with respect to the Fermi vacuum,

$$\text{(single excitation)} \qquad |\Phi_i^a\rangle \equiv a^\dagger i|0\rangle = |ajk\ldots N\rangle \tag{3.38}$$

$$\text{(double excitation)} \qquad |\Phi_{ij}^{ab}\rangle \equiv a^\dagger b^\dagger ji|0\rangle = |abk\ldots N\rangle \tag{3.39}$$

$$\text{(electron removal)} \qquad |\Phi_i\rangle \equiv i|0\rangle = |jk\ldots N\rangle \tag{3.40}$$

$$\text{(electron attachment)} \qquad |\Phi^a\rangle \equiv a^\dagger|0\rangle = |aijk\ldots N\rangle \tag{3.41}$$

with particle-hole representation operators that are hole or particle creators and annihilators.

$$i^\dagger - \text{hole annihilator} \tag{3.42}$$

$$i - \text{hole creator} \tag{3.43}$$

$$a^\dagger - \text{particle creator} \tag{3.44}$$

$$a - \text{particle annihilator} \tag{3.45}$$

It is important to note that a hole can not be annihilated if the hole doesn't exist within the reference state, that is,

$$i^{\dagger}|ijk\ldots N\rangle = 0 \tag{3.46}$$

also a particle can not be destroyed if it doesn't exist,

$$a|ijk\ldots N\rangle = 0 \tag{3.47}$$

Therefore, when describing a single excitation in particle-hole formalism like in Equation 3.38, first a hole is created in i by removing the particle from spin orbital i, then that particle is created in spin orbital a. This has allowed for creation of single excited state determinant that only contains a hole annihilator, particle creator and the Fermi vacuum reference state. Likewise in describing a double excitation (Equation 3.39) the hole states are creating first by removing i then j, and creating the particles in b then a.

The order which these operators occur is through normal ordering of the pseudo-creation operators to the left of all pseudo-annihilation operators. The pseudo-creation operators are the operators in the particle-hole formalism that create, i and $a^{\dagger}$. These are referenced as pseudo-creation operators because they both create but both do not contain the † symbol, which referees to creation in section 3.1. The opposite is true for the pseudo-annihilation operators, $i^{\dagger}$ and a as these operators annihilate. The process of moving all the pseudo-creation operators to left of the pseudo-annihilation operators is done through the anticommutation relations described in Equation 3.32-Equation 3.34,

$$[p^{\dagger}, q^{\dagger}]_{+} = 0 \tag{3.48}$$

$$[p, q]_{+} = 0 \tag{3.49}$$

$$[p^\dagger, q]_+ = [p, q^\dagger]_+ = \delta_{pq} \tag{3.50}$$

which were also written as,

$$p^\dagger q^\dagger = -q^\dagger p^\dagger \tag{3.51}$$

$$pq = -qp \tag{3.52}$$

$$p^\dagger q = \delta_{pq} - qp^\dagger \tag{3.53}$$

as these will relations keep track of the sign changes that occur during the reordering of the operators. Let us start with this simple example,

$$p^\dagger q \implies p^\dagger q \tag{3.54}$$

in this case the operators are already in normal order, therefore no sign change is required. Now consider the follow case,

$$qp^\dagger \implies -p^\dagger q \tag{3.55}$$

following the anticommutation relation in Equation 3.53 a negative sign is obtained during the normal ordering. Normal ordering particle-hole operators is crucial when evaluating matrix elements in second quantization.[1]

Chapter 4

Electronic Excited State Methods

A key component of the present work is to develop excited state methods to obtain electron-correlation energy to correct for the Hartree-Fock approximation to the exact Schrödinger wave-function. Within this chapter I will be presenting the underlying theory of many common excited state methods to serve as a introduction to the challenges addressed within the presented work after I have derived the linear response matrices. The derivation of each of these formulations nor their ranking in terms of accuracy are within the nature of this work, but their overall computational performance for excited state calculations is our primary interest and will be summarized towards the end.

4.1 Configuration Interaction Singles

Configuration interaction singles (CIS) is the most simplistic wave-function based *ab initio* method for electronic excited states, both conceptually and computationally. We will first start with our best single Slater determinant,

$$|\Phi_0\rangle = |\chi_1 \chi_2 \cdots \chi_N\rangle \tag{4.1}$$

where χ_i corresponds to a spin orbital with N number of electrons. $|\Phi_0\rangle$ is obtained by solving the time-independent HF equation, Equation 2.5,

$$\hat{F}|\Phi_0\rangle = E_0|\Phi_0\rangle \tag{4.2}$$

where,

$$\hat{F} = \sum_{i}^{N_{elec.}} \hat{f}_i \tag{4.3}$$

In configuration interaction singles, the electronic wave-function is a combination of the singly excited determinants generated through all possible single particle creator and single hole creator operators.

$$|\Psi_{\mathrm{CIS}}\rangle = \sum_{ia} c_i^a a^\dagger i|\Phi_0\rangle \tag{4.4}$$

$$|\Psi_{\mathrm{CIS}}\rangle = \sum_{ia} c_i^a |\Phi_i^a\rangle \tag{4.5}$$

The summation in Equation 4.4 is over all *ia* pairs which have a size of number of occupied ($N_{occ.}$) times number of virtual orbitals ($N_{vir.}$), commonly refereed to as the configuration space. This many-body wave function is then substituted into the exact time-independent electronic Schrödinger equation,

$$\hat{H}|\Psi\rangle = \left[\hat{T} + \hat{V}_{\mathrm{elec-nucl}} + \hat{V}_{\mathrm{elec-elec}}\right]|\Psi\rangle = E|\Psi\rangle \tag{4.6}$$

where the electronic Hamiltonian contains the kinetic energy operator, $(\hat{T})$

$$\hat{T} = -\sum_{i}^{N_{elec.}} \frac{1}{2}\nabla_i^2 \tag{4.7}$$

the electron-nuclei attraction, ($\hat{V}_{\text{elec}-\text{nucl}}$)

$$\hat{V}_{\text{elec}-\text{nucl}} = -\sum_{i}^{N_{elec.}}\sum_{K}^{N_{nuc.}} \frac{Z_K}{|\mathbf{r}_i - \mathbf{R}_K|} \tag{4.8}$$

where i is iterating through all electrons and K runs over all nuclei, that have a nuclear charge, Z_K. Finally the electron-electron interaction term, ($\hat{V}_{\text{elec}-\text{elec}}$)

$$\hat{V}_{\text{elec}-\text{elec}} = \sum_{i}^{N_{elec.}}\sum_{j>i}^{N_{elec.}} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{4.9}$$

Then by projecting onto Equation 4.6 or multiplication of $\langle\Phi_j^b|$ on the right hand side,

$$\sum_{i,a}^{N_{occ.},N_{vir.}} \langle\Phi_j^b|\hat{H}|\Phi_i^a\rangle c_i^a = E_{\text{CIS}} \sum_{i,a}^{N_{occ.},N_{vir.}} c_i^a \delta_{ij}\delta_{ab} \tag{4.10}$$

where a single term within the left summation is

$$\langle\Phi_j^b|\hat{H}|\Phi_i^a\rangle = (E_0 + \varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\text{A}} \tag{4.11}$$

where $f_H(\mathbf{r}_1,\mathbf{r}_2) = 1/|\mathbf{r}_1 - \mathbf{r}_2|$ is the Hartree kernel. The full expression is then defined as,

$$\sum_{i,a}^{N_{occ.},N_{vir.}} \left\{(E_0 + \varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\text{A}}\right\} c_i^a = E_{\text{CIS}} \sum_{ia}^{N_{occ.},N_{vir.}} c_i^a \delta_{ij}\delta_{ab} \tag{4.12}$$

which can be simplified with the equation for the excitation energy $\omega_{\text{CIS}} = E_{\text{CIS}} - E_0$,

$$\sum_{i,a}^{N_{occ.},N_{vir.}} \left\{(\varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\text{A}}\right\} c_i^a = \omega_{\text{CIS}} \sum_{ia} c_i^a \delta_{ij}\delta_{ab} \tag{4.13}$$

in the above expression ε is the orbital energy of the χ_a or χ_i orbitals, and $\langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\text{A}}$ is the antisymmetrized two-electron integrals defined as,

$$\langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\text{A}} = \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle - \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ba\rangle \tag{4.14}$$

$$\langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle = \int\int d\mathbf{r}_1 d\mathbf{r}_2 \chi_i^*(\mathbf{r}_1)\chi_j^*(\mathbf{r}_2) f_H(\mathbf{r}_1,\mathbf{r}_2)\chi_a(\mathbf{r}_1)\chi_b(\mathbf{r}_2) \tag{4.15}$$

$$\langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ba\rangle = \int\int d\mathbf{r}_1 d\mathbf{r}_2 \chi_i^*(\mathbf{r}_1)\chi_j^*(\mathbf{r}_2) f_H(\mathbf{r}_1,\mathbf{r}_2)\chi_b(\mathbf{r}_1)\chi_a(\mathbf{r}_2) \tag{4.16}$$

The expression Equation 4.13 can be more conveniently written in matrix notation,

$$\mathbf{AX} = \omega\mathbf{X} \tag{4.17}$$

as an eigenvalue equation that uses **A** for the matrix representation of Equation 4.11. ω is the diagonal matrix of the excitation energies, and **X** is the matrix of the CIS expansion coefficients. The matrix elements of **A** are,

$$A_{ia,jb} = (\varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\mathrm{A}} \tag{4.18}$$

The excitation energies can be obtained by,

$$(\mathbf{A} - \omega)\,\mathbf{X} = 0 \tag{4.19}$$

or diagonalization of **A**.[3]

4.2 Linear Response Matrices

4.2.1 Casida Equations

Within section 4.1 the derivation of **A** matrix was the primary focus. This matrix is of key importance for linear response theory for excited state calculations. The **A** matrix is only one component of what is commonly refereed to as the Casida equations or linear response matrices,[5, 3]

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{A}^* \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \omega \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \tag{4.20}$$

From the perspective of Time-dependent Hatree-Fock (TD-HF) the matrix elements are defined as follows,

$$A_{ia,jb} = (\varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle_{\mathrm{A}} \tag{4.21}$$

$$B_{ia,jb} = \langle ib|f_H(\mathbf{r}_1,\mathbf{r}_2)|aj\rangle_{\mathrm{A}} \tag{4.22}$$

or more generically written and known as,

$$A_{ia,jb} = (\varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + K_{ia,jb} \tag{4.23}$$

$$B_{ia,jb} = K_{ia,bj} \tag{4.24}$$

with the electron-hole interaction kernel (K_{eh}) or coupling matrix,

$$K_{pq,rs} = \langle pr|\hat{K}_{eh}|qs\rangle - \langle pr|\hat{K}_{eh}|sq\rangle \tag{4.25}$$

Where we have seen the terms for the **A** matrix which describes the resonant coupling between the excitations, but now have a non-resonant coupling matrix in the form of **B** between the excitations and de-excitations. It is very common to set $\mathbf{B} = 0$ in Equation 4.20, this is commonly refereed to as the Tamm-Dancoff aproximation (TDA)[4, 6]. When the TDA is applied to Equation 4.20 it reduces to the CIS formulation Equation 4.17. The last term in both the matrices is response of the nonlocal HF exchange potential, which yields a Coulomb-like term.

4.2.2 Limitations

It is important to note some key computational cost features of these response matrices as these are a primary focus of the present research. The first being the size of these matrices, they are $(N_{occ.})^2 \times (N_{vir})^2$ in size, where $N_{occ.}$ and $N_{vir.}$ is the number of occupied molecular orbitals and number of virtual orbitals, respectively. The size of these matrices are inherently linked to the square of the occupation space of the system, because *ia* iterates through all single particle excitation's in the system. The next computational cost arises in the construction of the coupling matrix, where the atomic-orbital to molecular-orbital (AO-to-MO) integral transformation is required. Where the integral transformation is,

$$\left[\chi_p(\mathbf{r}_1)\chi_q(\mathbf{r}_1)|\hat{K}_{\text{eh}}|\chi_r(\mathbf{r}_2)\chi_s(\mathbf{r}_2)\right] = \sum_{\mu=1}^{N_{AO}}\sum_{\nu=1}^{N_{AO}}\sum_{\lambda=1}^{N_{AO}}\sum_{\sigma=1}^{N_{AO}} C_{\mu p}C_{\nu q}C_{\lambda r}C_{\sigma s}\left[\phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)|\hat{K}_{\text{eh}}|\phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)\right] \tag{4.26}$$

For which the linear combination of atomic orbitals (LCAO) is a summation over the number of atomic orbitals (N_{AO}). This transformation has the scaling of $O(N_b^5)$, where N_b is the number of basis functions for the system. [2] To put these limitations into perspective, Table 4.1 shows some numerical data to better describe the computational effort for the response matrices. The data provided in Table 4.1 shows the unfavorable scaling of the singly excited configurations and two-electron integral transformation for a system of experimental interest, $Cd_{250}Se_{250}$ which is approximately a 2 nm quantum dot. A key limitation to also take note of is the computational effort in regards to storage of this information which can greatly exceed general use devices rapidly.

Table 4.1: Size of excited state calculation based on basis function and active electrons

Molecule	**Number of Active Electrons**	**Basis Functions**	**Singly Exited Configurations**	**Two-Electron Transformation**
H_2O	10	19 (6-31G*)	140	2,476,099
CdSe	18	26 (LANL2DZ)	153	11,881,376
$Cd_{12}Se_{12}$	216	312 (LANL2DZ)	22,032	2,956,466,552,832
$Cd_{250}Se_{250}$	4,500	6,750 (LANL2DZ)	10,125,000	14,012,604,492,187,500,000

In the following sections I will go over how various other excited state methods reduce the computational effort examined here in the Hartree-Fock linear response matrices. Each method can arrive at the Casida equations but I will only be presenting the form of the **A** and **B** matrices.

4.3 Time-Dependent Density Funtional Theory

Time-dependent density functional theory (TD-DFT) is a very common excited state method due to the widespread use of DFT for ground state calculations. Since the ground state reference is constructed through DFT, there requires a treatment of the exchange-correlation (xc) functional which has replaced the HF exchange potential. In LR-TD-DFT the **A** and **B** matrices are defined as,

$$A_{ia,jb} = (\varepsilon_a - \varepsilon_i)\,\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle + \langle ij|f_{xc}|ab\rangle \tag{4.27}$$

$$B_{ia,jb} = \langle ib|f_H(\mathbf{r}_1,\mathbf{r}_2)|aj\rangle + \langle ib|f_{xc}|aj\rangle \tag{4.28}$$

where the last term is the response of the chosen xc potential. The response to the xc potential is the second functional derivative of the exchange-correlation energy, known as the xc kernel,

$$\langle ij|f_{xc}|ab\rangle = \int \mathrm{d}^3\mathbf{r}_1\mathrm{d}^3\mathbf{r}_2\chi_i^*(\mathbf{r}_1)\chi_j^*(\mathbf{r}_2)\frac{\partial^2 E_{\mathrm{ex}}}{\partial\rho(\mathbf{r}_1)\partial\rho(\mathbf{r}_2)}\chi_a(\mathbf{r}_1)\chi_b(\mathbf{r}_2) \tag{4.29}$$

The limitations of LR-TD-DFT are still consistent with those described in subsection 4.2.2 regarding the size of the **A** and **B** matrices, along with the AO-to-MO transformation. But in addition there is now the computational effort in calculation and storage of the xc kernel. Also, since the exact xc potential is not know, but approximated in use the choice of density functional is an inherited limitation of using DFT based excited state methods.[3]

4.4 GW Bethe-Salpeter equation

The GW formulation is a correction to the ground state Kohn-Sham(KS) DFT orbitals utilizing the energy-dependent $\Sigma(E)$ self-energy as follows,[7, 8, 9]

$$\varepsilon_n^{GW} = \varepsilon_n^{KS} + \langle\phi_n^{KS}|\Sigma(\varepsilon_n^{GW}) - V^{\mathrm{xc}}|\phi_n^{KS}\rangle \tag{4.30}$$

where V^{xc} is the DFT exchange-correlation potential. Where the self energy is defined as follows,

$$\Sigma(\mathbf{r},\mathbf{r}';E) = \frac{i}{2\pi}\int d\omega e^{i\omega 0^+}G(\mathbf{r},\mathbf{r}';E+\omega)W(\mathbf{r},\mathbf{r}';\omega) \tag{4.31}$$

$$G(\mathbf{r},\mathbf{r}';E) = \sum_n \frac{\phi_n(\mathbf{r})\phi_n^*(\mathbf{r}')}{E-\varepsilon_n+0^+\times\mathrm{sgn}(\varepsilon_n-E_f)} \tag{4.32}$$

$$W(\mathbf{r},\mathbf{r}';\omega) = v(\mathbf{r},\mathbf{r}') + \int d\mathbf{r}_1 d\mathbf{r}_2 v(\mathbf{r},\mathbf{r}_1) P_0(\mathbf{r}_1,\mathbf{r}_2;\omega) W(\mathbf{r}_2,\mathbf{r}';\omega) \tag{4.33}$$

$$P_0(\mathbf{r},\mathbf{r}';\omega) = \sum_{i,j} (f_i - f_j) \frac{\phi_i^*(\mathbf{r})\phi_j(\mathbf{r})\phi_j^*(\mathbf{r}')\phi_i(\mathbf{r}')}{\varepsilon_i - \varepsilon_j - \omega - i0^+} \tag{4.34}$$

containing a screened Coulomb potential W, where $v(\mathbf{r},\mathbf{r}')$ is the bare Coulomb potential, G is the time-ordered one-body Green's function, P_0 the independent-electron susceptibility, 0^+ is an infinitesimally small positive number, and $\{f\}$ are occupation numbers.[10] The above equations obtain corrected ground state orbitals, through construction of the self-energy. But the best Σ will be the one in which the left and right ε_n^{GW} in Equation 4.30 are equal, therefore self-consistent. The computational effort needed to construct the ground state reference in GW is important in terms of understanding overall computational cost in calculation of excited state excitations as the self energy is a 6-dimensional integral. With the construction of the ground state reference it is at this point that the Bethe-Salpeter equations (BSE) are used to obtain the linear response matrices,

$$A_{ia,jb} = \left(\varepsilon_a^{GW} - \varepsilon_i^{GW}\right)\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1,\mathbf{r}_2)|ab\rangle + \langle ij|W|ab\rangle \tag{4.35}$$

$$B_{ia,jb} = \langle ib|f_H(\mathbf{r}_1,\mathbf{r}_2)|aj\rangle + \langle ib|W|aj\rangle \tag{4.36}$$

which look familiar to the LR-TD-DFT formulation except for the last term, which instead of the response the xc functional, there is now a response to the screened Coulomb potential, W. With GW-BSE the ground state reference is corrected but the excited state calculations have not gained much in terms of computational cost reduction. The size of **A** and B are still linked to number of singly excited states and the cost of the AO-to-MO integral transformation has not been addressed.

4.5 Coupled-Cluster methods

Couple cluster methods deal with the coupled-cluster (CC) wave-function,[1, 11]

$$\Psi = e^{\hat{T}}|0\rangle \tag{4.37}$$

where $|0\rangle$ is the reference wave-function and

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \ldots \tag{4.38}$$

Which are *m*-body cluster operators given as,

$$\hat{T}_m = \frac{1}{(m!)^2} \sum_{\substack{ij\ldots \\ ab\ldots}} t_{ij\ldots}^{ab\ldots} \left\{ a^\dagger i b^\dagger j \ldots \right\} \tag{4.39}$$

where m is the pairs of creation and annihilation operators, producing m-fold excitations. The benefit of this exponential *ansatz* is that if expanded in a Taylor series,

$$e^{\hat{T}} = 1 + \frac{1}{2}\hat{T}^2 + \frac{1}{3!}\hat{T}^3 + \ldots \tag{4.40}$$

and then by expanding $\hat{T}$,

$$\Psi = |0\rangle + \hat{T}_1|0\rangle + \hat{T}_2|0\rangle + \ldots \tag{4.41}$$

$$+ \frac{1}{2!}\hat{T}_1^2|0\rangle + \hat{T}_1\hat{T}_2|0\rangle + \frac{1}{2!}\hat{T}_2^2|0\rangle + \ldots \tag{4.42}$$

$$+ \frac{1}{3!}\hat{T}_1^3|0\rangle + \frac{1}{2!}\hat{T}_1^2\hat{T}_2|0\rangle + \frac{1}{2!}\hat{T}_1\hat{T}_2^2|0\rangle + \frac{1}{3!}\hat{T}_2^3|0\rangle + \ldots \tag{4.43}$$

$$+ \ldots \tag{4.44}$$

for which one can obtain a wave-function that has m-tuple excitations from a single ground state reference Slater determinant. This is the general description of a CC wave-function, in practice the excitation operator is limited to singles and/or doubles as the expansion in $\hat{T}$ can be costly, but provides a quality description of the ground state reference. With the ground state reference in CC described I present the response matrices,

$$A_{ai,bj} = \left(\varepsilon_a^{CC} - \varepsilon_i^{CC}\right)\delta_{ij}\delta_{ab} + \langle ij|f_H(\mathbf{r}_1, \mathbf{r}_2)|ab\rangle_{\mathrm{A}} \tag{4.45}$$

$$B_{ai,bj} = \langle ib|f_H(\mathbf{r}_1, \mathbf{r}_2)|aj\rangle_{\mathrm{A}} \tag{4.46}$$

for which look similar to that obtained in TDHF but the wave-functions are CC wave-functions, therefore contain more information on the m-tuple excitations in $\hat{T}$.

4.6 Multi-Reference

Multi-reference (MR) approaches deal with construction of a wave-function from a collection of Slater determinants mainly from the Full CI wave-function,

$$|\Psi\rangle = c_0|\Phi_0\rangle + \sum_{ia} c_i^a|\Phi_i^a\rangle + \left(\frac{1}{2!}\right)^2 \sum_{ijab} c_{ij}^{ab}|\Phi_{ij}^{ab}\rangle + \left(\frac{1}{3!}\right)^2 \sum_{ijkabc} c_{ijk}^{abc}|\Phi_{ijk}^{abc}\rangle + \ldots \tag{4.47}$$

which fully permutes the lowest Slater determinant to n-tuply excited determinants. The number of n-tuply excited determinants is determined by the number of spin orbitals ($2K$), number of occupied orbitals, $N_{occ.}$ with $N_{vir.} = 2K - N_{occ.}$, number of virtual orbitals. Since we can also choose from $N_{vir.}$ the number of n-tuply excited determinants is,

$$\binom{N_{occ.}}{N_{vir.}}\binom{2K - N_{occ.}}{N_{vir.}} \tag{4.48}$$

which for many-electron systems with a large number of one-electron basis sets result in poor computational scaling, [2] recall the scaling of the just the singly excited determinants in Table 4.1.

The MR approach that I wish to discuss briefly is the complete active space (CAS) approach, which constructs a CI like wave-function through restriction of the number of excitations in the expansion. Within CAS the molecular orbital space is separated into subspaces containing: inactive, active and secondary (virtual) orbitals. Where the inactive space are doubly occupied orbitals, typically core orbitals and the virtual orbitals are unoccupied in all possible configurations. Finally the active orbitals have variable occupancy of 0, 1 or 2. The CI space then contains all possible determinants that can be constructed from the active orbitals. Therefore when utilizing CAS there is an inherited choice in the active space for construction of the wave-function.

4.7 Summary

The limitations and computational cost of these methods reduce to several components: *a*) form of the reference wave function, *b*) size of configuration space, *c*) number of excitation permutations, *d*) limit of the virtual space, *e*) two-electron integral transformation, see Table 4.2

Table 4.2: Non-linear scaling of various excited state methods.

Method	**Exponential Scaling of basis functions**
CIS	$O(N^4)$ [3]
TD-DFT	$O(N^4)$ [3]
EOM-CCS(D)	$O(N^4)$ ($O(N^6)$) [12]
GW-BSE	$O(N^4)$ [7]
CASPT2	$O(N^4)$ [13]
LPNO	$O(N^5)$ [14]

The reference wave function can be as simple as the HF/DFT wave function or in the case of MR or CC methods a set of configurations. All configurations in a many-electron system consist of all possible permutations of electron-hole n-tuple excitations. But often these reference configurations are reduced to a set of single, double or triple electron-hole excitation's near the Fermi level.[11, 15] Which in practice leads to complications with user-experience in choosing appropriate active spaces.[16, 17] This reference configuration is then subject to n-tuple excitations up to the limit of the virtual space, which is restricted by the size of the basis set. When for the exact solution the virtual space is infinite.[2] Each of the configurations are formed through an atomic orbital to molecular orbital transformation, (AO-to-MO), which is limited by the two-electron integral transformation. The two-electron transformation is in general the slowest computational step with a $O(N^5)$ scaling, with N related to the number of basis functions.[2, 3, 18, 5, 19] In some methods there is a basis transformation from molecular orbitals basis to a different basis that can reduce computational scaling, like pair natural orbitals (PNOs).[20] While some methods utilize the resolution-of-the-identity (RI) approximation to reduce the scale of the transformation.[21, 3, 5]

Chapter 5

Compact real-space representation of excited states using frequency-dependent explicitly-correlated electron-hole interaction kernel

5.1 Introduction

Accurate treatment of electron-hole interaction and inclusion of electron-correlation effect is central for describing electronic excitation states. Development of efficient theoretical methods for excited states continues to be an active field of research. The construction of the electron-hole interaction kernel $K_{\mathrm{eh}}(\omega)$ is the main source of difference for various approaches including: configuration interaction (CI),[3, 22] coupled-cluster (CC),[12, 23] GW-Bethe-Salpeter equation (GW-BSE), [24, 25, 26, 27] time-dependent density functional theory (TDDFT), [28, 29, 30, 31] multireference (MRCI), [15] pair natural orbitals (PNO), [32, 33] and explicity-correlated Hartree-Fock methods.[34, 35, 36, 37]

The steep computational cost originates from various components including: *a*) treatment of electron-electron correlation, *b*) number of particle-hole excitations, *c*) form of the $K_{\mathrm{eh}}(\omega)$, and *d*) two-electron AO-to-MO integral transformation. The number of particle-hole states effects the size of the A and B response matrices, that arise in various theories including, TDDFT, GW-BSE, EOM-CCSD, and CIS methods.[5, 3] Various strategies exist to obtain a compact representation of the basis. Information from the polarization propagator,[38] 1-particle reduced density matrix, [32, 33] and Cholesky decomposition (CD)[39] have been used for calculation of excited states. A different strategy using stochastic basis functions has been demonstrated in GW-BSE,[40, 41] many-body perturbation theory (MBPT),[42] and TDDFT formulations.[43, 44] Another alternative approach is to move away from occupation-number representation and use real-space representation.[45, 37] Methods such as reduced-density matrix,[46] electron-hole explicitly-correlated Hartree-Fock,[34, 35, 36] geminal screened electron-hole interaction kernel,[45, 37] electron-correlator method, [47] and geminal-augmented MCSCF[48] have successfully demonstrated the efficacy of real-space representation.

The goal of this work is to present an efficient and practical method to account for electron-hole correlation in large semiconductor NPs without using a large number of particle-hole states. Using Löwdin's partitioning technique,[49] we have derived a real-space, frequency-dependent, explicitly-correlated, electron-hole interaction kernel (FD-GSIK method) that is capable of describing the dynamical screening of electrons and holes in a many-electron system. The method was implemented by performing real-space Monte Carlo (MC) integration for computation of all matrix elements. We demonstrate that the FD-GSIK method provides a compact representation of $K_{\mathrm{eh}}(\omega)$ while avoiding the steep N^5 computational cost of performing AO-to-MO integral transformation, and circumventing storing and evaluating frequency-dependent AO integrals. The FD-GSIK addresses the need for a first-principle excited-state method that does not exhibit the steep scaling of the computational cost with increasing system size.

5.2 Theory

5.2.1 Construction of frequency-dependent electron-hole-interaction kernel

The effective many-body electron-hole Hamiltonian has the general form [50, 51, 52, 35, 53, 54]

$$H_{\mathrm{eh}} = H_{\mathrm{e}}^{0} + H_{\mathrm{h}}^{0} + V_{\mathrm{ee}} + V_{\mathrm{hh}} + V_{\mathrm{eh}} \tag{5.1}$$

where

$$H_{\mathrm{e}}^0 = \sum_{e_1 e_2} \langle e_1 | f^e | e_2 \rangle e_1^\dagger e_2 \tag{5.2}$$

$$H_{\mathrm{h}}^0 = \sum_{h_1 h_2} \langle h_1 | f^h | h_2 \rangle h_1^\dagger h_2 \tag{5.3}$$

$$V_{\mathrm{ee}} = \sum_{e_1 e_2 e_3 e_4} \langle e_1 e_2 | w_{\mathrm{ee}} | e_4 e_3 \rangle e_1^\dagger e_2^\dagger e_3 e_4 \tag{5.4}$$

$$V_{\mathrm{hh}} = \sum_{h_1 h_2 h_3 h_4} \langle h_1 h_2 | w_{\mathrm{hh}} | h_4 h_3 \rangle h_1^\dagger h_2^\dagger h_3 h_4 \tag{5.5}$$

$$V_{\mathrm{eh}} = \sum_{h_1 h_2 e_1 e_2} \langle e_1 h_1 | v_{\mathrm{eh}} | e_2 h_2 \rangle e_1^\dagger h_1^\dagger h_2 e_2 \tag{5.6}$$

where $e^\dagger$ and $h^\dagger$ are particle and hole creation operators, and f_{e} and f_{h} are the Fock operators. The electron-hole Hamiltonian can be factored into a sum of non-interaction terms and interaction terms. For a 1-particle 1-hole state, further simplification can be achieved and the total Hamiltonian can be expressed as

$$H_{\mathrm{eh}} = H_{\mathrm{eh}}^0 + V_{\mathrm{eh}} \tag{5.7}$$

The non-interacting Hamiltonian H_{eh}^0 is obtained from the Fock operator whose eigenfunctions are (uncorrelated) particle-hole excited states

$$H_{\mathrm{eh}}^0 = \omega_P^0 |P\rangle = (\varepsilon_a - \varepsilon_i) |\chi_i^{\mathrm{h}} \chi_a^{\mathrm{e}}\rangle \tag{5.8}$$

We are interested in calculating the many-body correction to ω_P^0 due to electron-hole correlation in the system. We start by defining the zeroth-order approximation of the electron-hole wavefunction as,

$$|\Psi_X\rangle^{(0)} = |P\rangle \tag{5.9}$$

and the correlated electron-hole wavefunction Ψ_X is defined as

$$|\Psi_X\rangle = C_P|P\rangle + C_\perp|P_\perp\rangle \tag{5.10}$$

where C_P and $C_\perp$ are expansion coefficients and $P_\perp$ is the correlated wavefunction that exists in the orthogonal subspace of P. This is a very general form of the wavefunction and both MBPT and CI wavefunctions can be expressed in this form, where choice and construction of $P_\perp$ differentiates between treatment of electron-hole correlation within different methods. Here, we will define an electron-hole correlator operator Λ and express the $P_\perp$ as,

$$|\Psi_X\rangle = C_P|P\rangle + C_Q\Lambda(\omega, r_{\text{eh}})|Q\rangle \tag{5.11}$$

where C_P and C_Q are expansion coefficients, $|P\rangle$ and $|Q\rangle$ are two eigenvectors of H_{eh}^0 operators with eigenvalues ω_P^0 and ω_Q^0, respectively and $\langle P|Q\rangle = 0$. These two eigenvectors represent two different particle-hole excitation of the form $|P\rangle = |\chi_i^{\text{h}}\chi_a^{\text{e}}\rangle$ and $|Q\rangle = |\chi_j^{\text{h}}\chi_b^{\text{e}}\rangle$ which are obtained from applying the following particle-hole excitation operators $\{a^\dagger i\}$ and $\{b^\dagger j\}$, as shown in Figure 5.1. The central quantity of interest is the $\Lambda(\omega, r_{\text{eh}})$ operator, a frequency-dependent, r12-explicitly correlated operator which operates simultaneously on both particle and hole states and is responsible for capturing electron-hole correlation. Construction of Λ and determination of $|Q\rangle$ are presented in subsection 5.2.4 and subsection 5.2.2, respectively. In this section, we will focus on the construction of the $K_{\text{eh}}(\omega)$ and its derivation is independent of the specific form of Λ. The two conditions that Λ must satisfy are: (1) frequency-dependent, and (2) explicitly-correlated. Both the expansion coefficients (C_P, C_Q) and $\Lambda(\omega, r_{\text{eh}})$ are obtained from the solution of the resulting

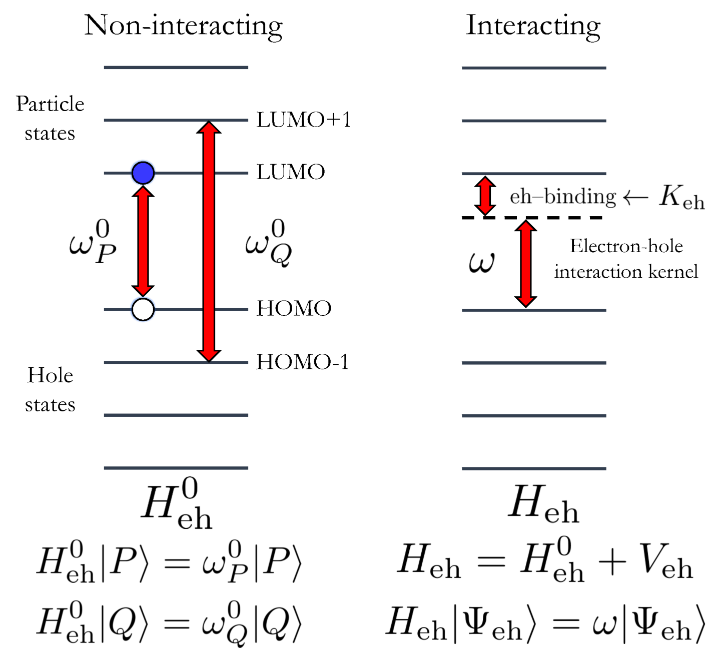

Non-interacting
Interacting
Particle states
LUMO+1
LUMO
ω_P^0
ω_Q^0
HOMO
HOMO-1
Hole states
eh–binding ← K_{eh}
ω
Electron-hole interaction kernel
H_{eh}^0
H_{eh}
$H_{\text{eh}}^0|P\rangle = \omega_P^0|P\rangle$
$H_{\text{eh}}^0|Q\rangle = \omega_Q^0|Q\rangle$
$H_{\text{eh}} = H_{\text{eh}}^0 + V_{\text{eh}}$
$H_{\text{eh}}|\Psi_{\text{eh}}\rangle = \omega|\Psi_{\text{eh}}\rangle$

Figure 5.1: Interacting and non-interacting Hamiltonian

electron-hole pseudo eigenvalue equation,

$$H_{\text{eh}}|\Psi_X(\omega)\rangle = \omega|\Psi_X(\omega)\rangle \tag{5.12}$$

where,

$$|\Psi_X\rangle = C_P|P\rangle + C_Q\Lambda(\omega)|Q\rangle \tag{5.13}$$

and $|P\rangle$ and $|Q\rangle$ are eigenkets of the non-interacting electron-hole Hamiltonian H_{eh}^0.

$$H_{\text{eh}}^0|P\rangle = \omega_P^0|P\rangle \tag{5.14}$$

$$H_{\text{eh}}^0|Q\rangle = \omega_Q^0|Q\rangle \tag{5.15}$$

Because $\Lambda(\omega, r_{\text{eh}})$ is frequency-dependent and depends on the eigenvalue ω, the solution is obtained iteratively. We have used the Löwdin's partitioning scheme[49] to obtain the analytical expression of $K_{\text{eh}}(\omega)$ in terms of the relevant matrix elements associated with V_{eh} and $\Lambda(\omega, r_{\text{eh}})$. The step-by-step derivation of the frequency-dependent electron-hole interaction kernel using Löwdin's partitioning technique is presented below. We start with projection on the subspace of $\langle P|$ and $\langle Q|$, and using orthonormalization of $\langle P|Q\rangle = \delta_{PQ}$ we get,

$$\begin{bmatrix} \langle P|H_{\text{eh}}|P\rangle & \langle P|H_{\text{eh}}\Lambda(\omega)|Q\rangle \\ \langle Q|H_{\text{eh}}|P\rangle & \langle Q|H_{\text{eh}}\Lambda(\omega)|Q\rangle \end{bmatrix} \begin{bmatrix} C_P \\ C_Q \end{bmatrix} = \omega \begin{bmatrix} 1 & \langle P|\Lambda(\omega)|Q\rangle \\ 0 & \langle Q|\Lambda(\omega)|Q\rangle \end{bmatrix} \begin{bmatrix} C_P \\ C_Q \end{bmatrix} \tag{5.16}$$

Performing Löwdin's partitioning,

$$H_{PP}C_P + \tilde{H}_{PQ}C_Q = \omega C_P + \omega \tilde{S}_{PQ}C_Q \tag{5.17}$$

$$H_{QP}C_P + \tilde{H}_{QQ}C_Q = \omega \tilde{S}_{QQ}C_Q \tag{5.18}$$

where the tilde terms are defined as:

$$\tilde{S}_{PQ} = \langle P|\Lambda(\omega)|Q\rangle \tag{5.19}$$

$$\tilde{S}_{QQ} = \langle Q|\Lambda(\omega)|Q\rangle \tag{5.20}$$

$$\tilde{H}_{PQ} = \langle P|H_{\text{eh}}\Lambda(\omega)|Q\rangle = \omega_P^0 \tilde{S}_{PQ} - \langle P|r_{12}^{-1}\Lambda(\omega)|Q\rangle \tag{5.21}$$

$$\tilde{H}_{QQ} = \langle Q|H_{\text{eh}}\Lambda(\omega)|Q\rangle = \omega_Q^0 \tilde{S}_{QQ} - \langle Q|r_{12}^{-1}\Lambda(\omega)|Q\rangle \tag{5.22}$$

Rearranging Equation 5.17 and Equation 5.18,

$$H_{PP}C_P + \left[\tilde{H}_{PQ} - \omega \tilde{S}_{PQ}\right] C_Q = \omega C_P \tag{5.23}$$

$$\left[\tilde{H}_{QQ} - \omega \tilde{S}_{QQ}\right] C_Q = -H_{QP}C_P \tag{5.24}$$

Therefore,

$$C_Q = -\frac{H_{QP}}{\tilde{H}_{QQ} - \omega\tilde{S}_{QQ}} C_P \tag{5.25}$$

We obtain the equation for the electron-hole interaction kernel,

$$[H_{PP} + K_{PP}(\omega)]\, C_P = \omega C_P \tag{5.26}$$

where $K_{PP}(\omega)$ is the electron-hole interaction kernel and is defined as

$$K_{PP}(\omega) = -\frac{\left[\tilde{H}_{PQ} - \omega\tilde{S}_{PQ}\right] H_{QP}}{\tilde{H}_{QQ} - \omega\tilde{S}_{QQ}} \tag{5.27}$$

Substituting the definitions of the tilde operators defined in Equation 5.19 - Equation 5.22, the closed-form analytical expression for $[K_{\text{eh}}(\omega)]_{PP}$ is given by

$$K_{PP}(\omega) = -\frac{\left[\langle P|r_{12}^{-1}\Lambda(\omega)|Q\rangle_A - (\omega_P^0 - \omega)\langle P|\Lambda(\omega)|Q\rangle\right] \langle Q|r_{12}^{-1}|P\rangle_A}{\langle Q|r_{12}^{-1}\Lambda(\omega)|Q\rangle_A + (\omega_Q^0 - \omega)\langle Q|\Lambda(\omega)|Q\rangle} \tag{5.28}$$

where $\langle\ldots\rangle_A$ represents anti-symmetrized matrix elements. The expression for $[K_{\text{eh}}(\omega)]_{PP}$ can be further simplified by performing Gram - Schmidt orthogonalization to make $\langle P|\Lambda|Q\rangle = 0$. This can be achieved by replacing

$$\Lambda|Q\rangle \rightarrow \Lambda|Q\rangle - \langle P|\Lambda|Q\rangle|P\rangle \tag{5.29}$$

in the expression for K_{PP} in Equation 6.27. The orthogonalization procedure imposes intermediate normalization condition for the electron-hole wave

$$\langle P|\Psi_X\rangle = 1 \tag{5.30}$$

The $[K_{\text{eh}}(\omega)]_{PP}$ terms has two sources of frequency-dependency. The first are the $(\omega_P^0 - \omega)$ and $(\omega_Q^0 - \omega)$ terms from the operator inversion step in the Löwdin's partitioning. The second is the frequency-dependency from $\Lambda(\omega, r_{\mathrm{eh}})$ and this feature is the main content of this article. Indeed, setting $\Lambda = 1$ results in a simple 2×2 configuration-interaction singles (CIS) expression.

5.2.2 Determination of the $|Q\rangle$ state

The zeroth-order electron-hole wavefunction is given by $|\Psi_X\rangle^{(0)} = |P\rangle$ and the choice of the $|Q\rangle$ plays an important role in accurate description of electron-hole correlation and calculation of electron-hole binding energies. In this work, we determine $|Q\rangle$ by minimizing the trace $T[Q]$ of the 2×2 uncorrelated Hamiltonian.

$$\mathbf{H}_{PQ} = \begin{bmatrix} \langle P|H_{\mathrm{eh}}|P\rangle & \langle P|H_{\mathrm{eh}}|Q\rangle \\ \langle Q|H_{\mathrm{eh}}|P\rangle & \langle Q|H_{\mathrm{eh}}|Q\rangle \end{bmatrix} \tag{5.31}$$

where, the trace is given by

$$T[Q] = \omega_P^0 + \omega_Q^0 + \langle P|v_{\mathrm{eh}}|P\rangle + \langle Q|v_{\mathrm{eh}}|Q\rangle \tag{5.32}$$

and the search for Q is only restricted to states $\omega_P^0 \leq \omega_Q^0$. Since, Q is a uncorrelated particle-hole state, the last two terms in the previous expression, can be written in terms of particle and hole densities,

$$T[\rho_Q^{\mathrm{h}}, \rho_Q^{\mathrm{e}}] = \omega_P^0 + \omega_Q^0 - \langle \rho_P^{\mathrm{h}} \rho_P^{\mathrm{e}} r_{\mathrm{eh}}^{-1} \rangle - \langle \rho_Q^{\mathrm{h}} \rho_Q^{\mathrm{e}} r_{\mathrm{eh}}^{-1} \rangle \tag{5.33}$$

The search for the optimal Q scales $N_{\mathrm{occ}} \times N_{\mathrm{vir}}$ and requires searching over all 1p-1h states $|\chi_j \chi_b\rangle$. To reduce the computational cost, we define an approximate trace operator given by $\tilde{T}$

$$\tilde{T}[\tilde{\rho}_Q^{\mathrm{h}}, \tilde{\rho}_Q^{\mathrm{e}}] = \omega_P^0 + \omega_Q^0 - \langle \rho_P^{\mathrm{h}} \rho_P^{\mathrm{e}} r_{\mathrm{eh}}^{-1} \rangle - \langle \tilde{\rho}_Q^{\mathrm{h}} \tilde{\rho}_Q^{\mathrm{e}} r_{\mathrm{eh}}^{-1} \rangle \tag{5.34}$$

which uses approximate densities $\tilde{\rho}_Q^{\mathrm{h}}\tilde{\rho}_Q^{\mathrm{e}}$. The densities $\tilde{\rho}_Q^{\mathrm{h}}\tilde{\rho}_Q^{\mathrm{e}}$ are the Gaussian approximation to the true densities $\rho_Q^{\mathrm{h}}\rho_Q^{\mathrm{e}}$

$$\tilde{\rho}_Q^{\mathrm{e,h}} = \frac{1}{\sigma_Q^{\mathrm{e,h}}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{r-\mu_Q^{\mathrm{e,h}}}{\sigma_Q^{\mathrm{e,h}}}\right)} \tag{5.35}$$

and are obtained by minimizing the Kullback–Leibler divergence[55] between the true densities $\rho_Q^{\mathrm{h}}\rho_Q^{\mathrm{e}}$ and the Gaussian probability distribution functions.

$$D_{\mathrm{Kullback-Leibler}} = \int_V d\mathbf{r}\rho_Q^{\mathrm{e,h}} \ln\left[\frac{\rho_Q^{\mathrm{e,h}}}{\tilde{\rho}_Q^{\mathrm{e,h}}}\right] \tag{5.36}$$

The computational effort for determination of approximate densities $\tilde{\rho}_Q^{\mathrm{h}}\tilde{\rho}_Q^{\mathrm{e}}$ scales linearly with $(N_{\mathrm{occ}}+N_{\mathrm{vir}})$ and the calculation of $\langle\tilde{\rho}_Q^{\mathrm{h}}\tilde{\rho}_Q^{\mathrm{e}}r_{\mathrm{eh}}^{-1}\rangle$ is performed analytically. Both of these steps allow us search for the optimal $|Q\rangle$ state at a reduced computational cost.

5.2.3 Reducing the number of particle-hole states

The $\Lambda(\omega, r_{\mathrm{eh}})$ operator introduces the electron-hole correlation and add correction beyond the zeroth-order approximation to the electron-hole wavefunction. Motivated by both CIS and MBPT theory for excited states, we represent Λ in project space of 1p-1h states as

$$\Lambda = \sum_{ij}^{\mathrm{Nocc}} \sum_{ab}^{\infty} \Lambda_{ijab}\{a^\dagger i\}|0\rangle\langle 0|\{j^\dagger b\} \tag{5.37}$$

The action of $\Lambda(\omega, r_{\mathrm{eh}})$ in the electron-hole wavefunction can be interpreted as a compact representation of an infinite-order particle-hole excitation operator in the occupation-number representation [37]

$$\Lambda(\omega, r_{\mathrm{eh}})|Q\rangle = \sum_{K=0}^{\infty} c_K^{\Lambda}(\omega)|K\rangle \tag{5.38}$$

where the expansion coefficient $c_K^{\Lambda}(\omega)$ is a functional of Λ and are calculated from the matrix element $\langle K|\Lambda|Q\rangle$. In the present work, we will use the explicitly-correlated Slater-geminal function for constructing the Λ operator in real-space representation. Both Slater-geminal and Guassian-geminal functions have been used extensively for many-electron,[56, 48, 47] electron-proton[57, 58], and electron-hole systems[34, 35, 36, 45, 37] and provide a compact representation for construction of correlated many-particle wavefunction. The correlation operator, $\Lambda(\omega, r_{\text{eh}})$ has the general form of,

$$\Lambda(\omega, r_{\text{eh}}) = \frac{(\omega_P^0 - \omega)}{\gamma} e^{-r_{\text{eh}}\gamma} \tag{5.39}$$

where $\gamma = \langle P|r_{\text{eh}}^{-1}|P\rangle$. *The most important feature of the above expression for Λ is the complete absence of any adjustable parameters.* All quantities needed to define the operator are calculated during the course of the calculations. The explicit correlation comes from the Slater-geminal function $e^{-\gamma r}/\gamma$. We have made γ proportional to the Coulomb interaction of the electron-hole pair in $|P\rangle$ by setting $\gamma = \langle P|r_{\text{eh}}^{-1}|P\rangle$. The Slater-geminal is scaled by the $\left(\omega_P^0 - \omega\right)$ and is the frequency dependence of Λ. Although ω_P^0 is known before start of the calculation, ω is the final result and therefore, determination of ω requires a self-consistent solution of the equation derived in Equation 5.12.

5.2.4 Eliminating the AO-to-MO integral transformation

One of the principle computational bottleneck in performing first-principle excited state calculation is the evaluation of matrix elements. Traditionally, for atom-centered basis functions, the integrals are evaluated in AO representation and are later transformed in MO basis after the SCF step. This is an expensive transformation that scales as N^5. In this work, such a procedure is impractical because of three reasons. First, the scaling of the AO-to-MO is computationally impractical for large NPs that have > 200 heavy atoms with > 1000 basis functions. Second, as shown in Equation 6.28, the matrix elements depends on Slater-geminal function which are more complicated than the conventional 2-electron integrals and further adds to the already high computational cost. Third, the matrix elements depend on the final solution ω and consequently needs to be evaluated multiple times during the course of the calculations. In this work, we have addressed all these challenges by calculating the necessary integrals directly in the MO basis using MC numerical integration procedure. Specifically,

$$\langle \psi_r(\mathbf{r})\psi_s(\mathbf{r})|\Lambda(r_{12},\omega)|\psi_u(\mathbf{r})\psi_v(\mathbf{r})\rangle = \int d\mathbf{r}_1 d\mathbf{r}_2\, \psi_r(\mathbf{r}_1)\psi_s(\mathbf{r}_2)\Lambda(r_{12},\omega)\psi_u(\mathbf{r}_1)\psi_v(\mathbf{r}_2) \tag{5.40}$$

$$= \left(\omega_P^0 - \omega\right)\int d\mathbf{r}_1 d\mathbf{r}_2\, F(\mathbf{r}) \tag{5.41}$$

$$= \frac{\left(\omega_P^0 - \omega\right)}{N_{\text{sample}}} \sum_{\mathbf{r}_i \in \rho_{\text{pdf}}} \left[\frac{F(\mathbf{r}_i)}{\rho_{\text{pdf}}(\mathbf{r}_i)}\right] \tag{5.42}$$

where the MOs at any point in $\mathbf{r}$ in 3D space is obtained by $\psi_r(\mathbf{r}) = \sum_{\mu=1}^{N_{\text{AO}}} C_{\mu r}\phi_\mu(\mathbf{r})$. The MC step can be performed either using importance or stratified sampling and the details of the MC procedure used in this work has been described earlier, Bayne et. al.[59] . Additionally, the knowledge of analytic solution of the AO integrals using the explicitly correlated function is not needed and the method is not restricted by the choice of explicitly correlated function used in Λ .

5.2.5 Summary of steps in the overall calculation

Step 1 - Performing Hartree-Fock calculation: The first step involved the calculation of single-particle states and orbital energies which provided the zeroth-order description of the electronic excitation. We demonstrate the overall procedure using the Hartree-Fock (HF) method, however, both Kohn-Sham and pseudo-potential methods can be used as well.

Step 2 - Selection of the zeroth-order state: The next step involved selection of the P and Q states. Here, we are interested in investigating the effect of electron-hole correlation on the lowest excitation energy and therefore the HOMO-LUMO excitation in the non-interaction system was selected. Specifically, $|P\rangle$ was set to $P = \chi^{\text{h}}_{\text{HOMO}}\chi^{\text{e}}_{\text{LUMO}}$. The orthogonal space $|Q\rangle$ was selected using the procedue described in subsection 5.2.2. The zeroth-order excitation energy ω_P^0 is given by the HOMO-LUMO gap. It is important to note that this is a conventional HF procedure using conventional AO integrals. No explicitly-correlated integrals were evaluated in this step.

Step 3 - Evaluation of spatial molecular orbitals at point $\mathbf{r}$: The MC integration required evaluation of the spatial MOs $\psi(\mathbf{r})$ at any point in 3D space and was achieved by pre-computing the values of the MOs on a spatial grid before the start of the MC step. This made the implementation potable and demonstrated universal applicability, as it did not require source-code modification of electronic structure packages.

Step 4 - Performing the *1*$^{\text{st}}$ Monte Carlo integration for construction of Λ: After the MOs were obtained on a 3D grid, MC integration was performed to calculate $\langle P|r_{\text{eh}}^{-1}|P\rangle$. This quantity is needed for construction of the explicitly-correlated function in Λ as shown in Equation 6.28. In this work, we advocate the stratified-sampling approach and used the combined control-variate stratified sampling method (CCVSS) for evaluating two-electron integrals. [59]

Step 5 - Performing the 2$^{\text{nd}}$ Monte Carlo integration for construction $H[\Lambda(\omega)]$ and $S[\Lambda(\omega)]$: After $\Lambda(\omega)$ was constructed from ω_P^0 and $\langle P|r_{\text{eh}}^{-1}|P\rangle$, the matrix elements in Equation 6.27 were calculated numerically using the MC integration method described above. Since both P and Q states are eigenvectors of H_{eh}^0, matrix elements involving the reference Hamiltonian were obtained analytically. The remaining matrix elements are of the general form $\langle *|r_{\text{eh}}^{-1}|*\rangle$, $\langle *|\Lambda|*\rangle$, $\langle *|r_{\text{eh}}^{-1}\Lambda|*\rangle$, and were evaluated simultaneously in a single MC integration step.

Step 6 - Calculation of the excitation energy from the iterative solution of the projected equation: In the final step, matrix elements calculated previously were used to construct the $K_{\text{eh}}(\omega)$ using Equation 6.27. The excitation energy ω was obtained iteratively by evaluation of the left-hand-side of Equation 5.26 by setting $\omega = \omega_P^0$ as the starting guess.

5.3 Results

5.3.1 Chemical system and system setup

The FD-GSIK method was used to calculate excitation energy on a series of common NPs: PbS, PbSe, CdS, and CdSe up to 2.5 nm in diameter. The chemical formulas for the NPs investigated range from Pb_4S_4-$Pb_{140}S_{140}$, Pb_4Se_4-$Pb_{140}Se_{140}$, Cd_3S_3-$Cd_{87}S_{87}$, Cd_3Se_3-$Cd_{144}Se_{144}$, totaling in 48 unique structures. Details of the NPs including (*1*) chemical formula, (*2*) ω_p^0 energies, (*3*) electron-hole binding energies, (*4*) $\omega_{\mathrm{FD-GSIK}}$ excitations, (*5*) standard deviation of $\omega_{\mathrm{FD-GSIK}}$, and (*6*) computational cost are presented in Appendix A. Geometries for the NPs were generated by cutting dots of the desired size from bulk solid structure: for PbS and PbSe using the rock-salt crystal structure, while for CdS and CdSe the wurtzite crystal structure. The ground state single-particle states were obtained using the Q-CHEM package[60] using LANL2DZ basis and ECP. The 3D spatial grid for the MOs was generated with 100 points per dimension to generate a 10^6 point grid to obtain a dense grid for the integral evaluation. The grid boundary was determined by imposing a cutoff of $|\psi(\mathbf{r})| \leq 10^{-12}$ at the grid edges. The quality of the 3D spatial grid was verified by evaluating the normalization integral $\langle\psi|\psi\rangle$ before start of the MC calculations.

5.3.2 Scaling of FD-GSIK cost

The computational effort of the FD-GSIK method was compared to CIS calculation (Figure 5.2), for a series of PbS quantum dots ($Pb_4S_4 - Pb_{80}S_{80}$). Unsurprisingly, we found evidence of $O(N^5)$-scaling of computational cost for the CIS calculations. In contrast, the FD-GSIK method exhibited linear-scaling of the computational cost with increasing system size. Comparison of the excitation energies between FD-GSIK and CIS (Table 5.1) shows that the FD-GSIK energies are always lower than the CIS values. This trend is consistent with both MBPT and coupled-cluster formulations where we expect inclusion of electron-hole correlation to decrease the excitation energies.

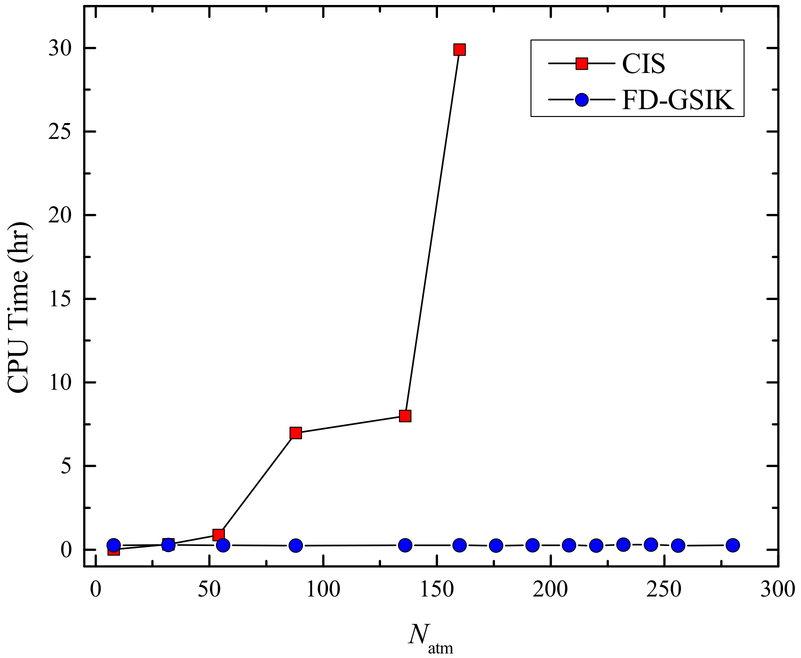


Figure 5.2: The single-core timing comparison of CIS and FD-GSIK calculations for the PbS series ($Pb_4S_4 - Pb_{80}S_{80}$) using LANL2DZ basis.

Table 5.1: Comparison of the first excitation energy (eV) obtained from CIS and the FD-GSIK method for PbS nanoparticles.

Chemical Formula	ω_{CIS}	$\omega_{\mathrm{FD-GSIK}}$
Pb_4S_4	3.7386	3.2915
$Pb_{16}S_{16}$	3.1944	2.4134
$Pb_{28}S_{28}$	3.1370	2.8188
$Pb_{44}S_{44}$	3.3432	3.0030
$Pb_{68}S_{68}$	2.4257	1.6547
$Pb_{80}S_{80}$	3.1582	3.1496

Table 5.2: Comparison of electron-hole binding energies (eV) obtained from EOM-CCSD, ADC(3), and FD-GSIK method for PbS nanoparticles.

	EOM-CCSD		**ADC(3)**		**FD-GSIK**	
Chemical Formula	Electron-hole binding eV	Time hour	Electron-hole binding eV	Time hour	Electron-hole binding eV	Time hour
Pb_4S_4	5.5922	0.0137	4.4832	0.008	5.0621	0.129
$Pb_{16}S_{16}$	4.5701	127.311	3.2128	1.372	3.9938	0.144
$Pb_{28}S_{28}$	–	–	3.3213	13.333	3.6390	0.132

5.3.3 Comparison of FD-GSIK with correlated methods

To further inspect the impact of electron-hole correlation operator in the FD-GSIK method, we performed calculations on a small subset of the PbS nanoparticles using EOM-CCSD and ADC(3) methods [61, 62] and the comparison of the electron-hole binding energies are presented in Table 5.2. These results showed two key insights; first, that FD-GSIK is successfully able to capture electron-hole correlation within the same magnitude as other correlated methods. Specifically, as compared to EOM-CCSD, the FD-GSIK method was found to capture 91% and 87% of the electron-hole binding energies. Second, the total time needed to perform the FD-GSIK calculations for these systems scales sub-linearly with respect to increasing system size. The FD-GSIK results were also compared with GW-BSE calculations[63] for CdSe clusters (Table 5.3) and were found to be in good agreement with the GW-BSE results. The results from Table 5.2 and Table 5.3 demonstrate that that the FD-GSIK is able to capture electron-hole correlation that is comparable to high-level many-body methods at a significantly low computational cost that has a favorable scaling with respect to system size.

Table 5.3: Comparison of excitation energy (eV) obtained from FD-GSIK method and GW-BSE[63] for small CdSe nanoparticles.

Chemical Formula	$\omega_{\mathrm{FD-GSIK}}$	$\omega_{\mathrm{GW-BSE}}$**[63]**
Cd_3Se_3	3.75	3.99
Cd_6Se_6	3.31	3.33

5.3.4 Electron-hole binding energy

The FD-GSIK method was used to calculate electron-hole binding energies for PbS, PbSe, CdS, and CdSe NPs of varying sizes and the results are presented in Figure 5.3 as a function of ω_p^0 energies. The complete list of excitation energies and electron-hole binding energies for these NPs is also available in Appendix A. We found that electron-hole binding energy exhibit a strong correlation with the non-interaction excitation energy (ω_p^0) and increased with increasing HOMO-LUMO gap.

5.3.5 Error analysis of the FD-GSIK method

The relationship between unbiased estimation of numerical error in the integral from the standard deviation of the integration kernel in MC method is well-known[59]. In the FD-GSIK method,

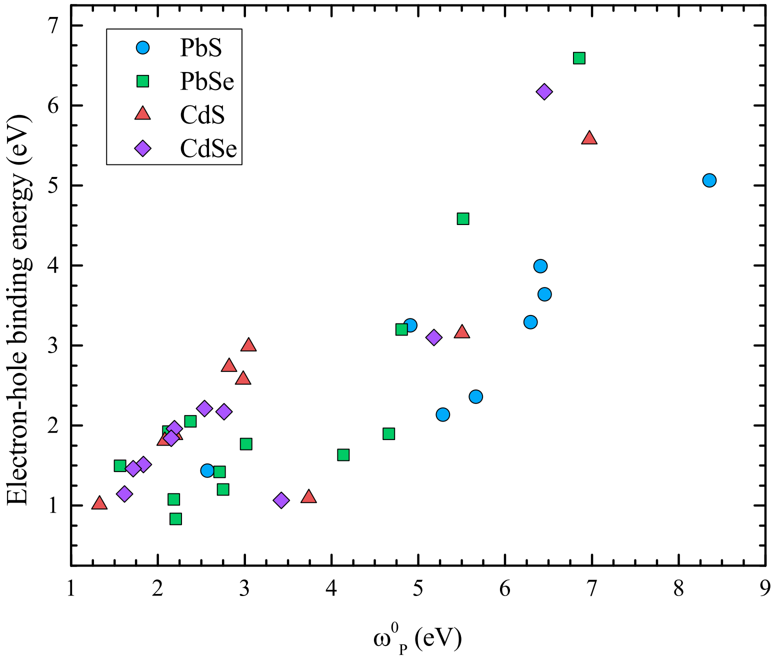


Figure 5.3: Electron-hole binding energy obtained using FD-GSIK method for Pb_4S_4-$Pb_{140}S_{140}$, Pb_4Se_4-$Pb_{140}Se_{140}$, Cd_3S_3-$Cd_{87}S_{87}$, and Cd_3Se_3-$Cd_{144}Se_{144}$ as a function of ω_P^0.

Table 5.4: Maximum and minimum standard deviation (eV) of FD-GSIK excitation energy for each nanoparticle series.

Standard Deviation	**PbS**	**PbSe**	**CdS**	**CdSe**
Maximum	0.018	0.042	0.064	0.059
Minimum	0.002	0.002	0.005	0.006

this relationship allows us to monitor and reduce the error in the calculated excitation energies. In Table 5.4, the maximum and minimum standard deviation is provided for each semiconducting NP series. The results from the standard deviation demonstrate the FD-GSIK method is able to achieve meV level of accuracy for all the NPs investigated in this work. The detailed description of errors for each NPs are presented in Appendix A.

5.3.6 Summary of results from the FD-GSIK calculations

Table 5.5: FD-GSIK results for PbS nanoparticles.

Chemical Formula	ω_P^0 eV	Electron-hole Binding eV	$\omega_{\mathrm{FD-GSIK}}$ eV	Std. Dev. of $\omega_{\mathrm{FD-GSIK}}$ eV	Time Hours
Pb_4S_4	8.35357	5.06212	3.29145	0.01781	0.12903
$Pb_{16}S_{16}$	6.40714	3.99376	2.41338	0.00590	0.14436
$Pb_{28}S_{28}$	6.45783	3.63899	2.81884	0.00889	0.13147
$Pb_{44}S_{44}$	6.29512	3.29209	3.00304	0.00440	0.12013
$Pb_{68}S_{68}$	4.90833	3.25367	1.65466	0.01082	0.13333
$Pb_{80}S_{80}$	5.28516	2.13558	3.14958	0.00194	0.12999
$Pb_{88}S_{88}$	2.88694	1.48855	1.39839	0.00360	0.11765
$Pb_{96}S_{96}$	2.56988	1.43828	1.13160	0.00801	0.13228
$Pb_{104}S_{104}$	4.57152	2.20100	2.37052	0.00720	0.12742
$Pb_{110}S_{110}$	3.95853	2.16843	1.79010	0.01080	0.12437
$Pb_{116}S_{116}$	3.11239	2.43603	0.67636	0.01776	0.14637
$Pb_{122}S_{122}$	5.73418	2.58570	3.14848	0.00688	0.15156
$Pb_{128}S_{128}$	2.89678	1.21088	1.68590	0.00434	0.11909
$Pb_{140}S_{140}$	5.66474	2.36135	3.30339	0.00396	0.12660

Table 5.6: FD-GSIK results for PbSe nanoparticles.

Chemical Formula	ω_P^0 eV	Electron-hole Binding eV	$\omega_{\mathrm{FD-GSIK}}$ eV	Std. Dev. of $\omega_{\mathrm{FD-GSIK}}$ eV	Time Hours
Pb_2Se_2	6.85533	6.58847	0.26686	0.04166	0.13833
Pb_4Se_4	4.80834	3.2009	1.60744	0.00526	0.14786
Pb_7Se_7	5.46048	4.31369	1.14679	0.03670	0.12674
Pb_8Se_8	2.1203	1.92466	0.19564	0.00368	0.11353
$Pb_{11}Se_{11}$	5.5160	4.58155	0.93445	0.01665	0.12359
$Pb_{17}Se_{17}$	2.46372	1.85218	0.61154	0.00695	0.14408
$Pb_{21}Se_{21}$	2.37579	2.05145	0.32434	0.02217	0.12609
$Pb_{25}Se_{25}$	4.65940	1.89550	2.76389	0.00643	0.11488
$Pb_{29}Se_{29}$	3.01619	1.76997	1.24622	0.00489	0.14735
$Pb_{39}Se_{39}$	4.13753	1.63465	2.50287	0.00221	0.14521
$Pb_{44}Se_{44}$	1.56409	1.49666	0.06742	0.00865	0.13585
$Pb_{52}Se_{52}$	2.18492	1.07936	1.10556	0.00197	0.12230
$Pb_{63}Se_{63}$	2.74999	1.20218	1.54780	0.00201	0.14599
$Pb_{82}Se_{82}$	2.71174	1.42091	1.29083	0.00321	0.15032
$Pb_{140}Se_{140}$	2.20412	0.83281	1.37131	0.00033	0.15515

Table 5.7: FD-GSIK results for CdS nanoparticles.

Chemical Formula	ω_P^0 eV	Electron-hole Binding eV	$\omega_{\mathrm{FD-GSIK}}$ eV	Std. Dev. of $\omega_{\mathrm{FD-GSIK}}$ eV	Time Hours
Cd_3S_3	6.97115	5.57372	1.39743	0.06397	0.14822
Cd_6S_6	2.81942	2.73266	0.08676	0.01682	0.13336
$Cd_{12}S_{12}$	3.04753	2.98927	0.05826	0.02259	0.13933
$Cd_{15}S_{15}$	5.50360	3.15073	2.35286	0.02525	0.13288
$Cd_{24}S_{24}$	2.98365	2.57143	0.41221	0.00626	0.14498
$Cd_{33}S_{33}$	2.08133	1.80746	0.27386	0.00539	0.13671
$Cd_{39}S_{39}$	2.19995	1.88188	0.31806	0.00566	0.10922
$Cd_{45}S_{45}$	3.73826	1.09407	2.64420	0.00084	0.12661
$Cd_{72}S_{72}$	1.32678	1.01447	0.31231	0.00052	0.13217

Table 5.8: FD-GSIK results for CdSe nanoparticles.

Chemical Formula	ω_P^0 eV	Electron-hole Binding eV	$\omega_{\mathrm{FD-GSIK}}$ eV	Std. Dev. of $\omega_{\mathrm{FD-GSIK}}$ eV	Time Hours
Cd_3Se_3	6.45328	6.17198	0.28129	0.05927	0.11632
Cd_6Se_6	2.53726	2.21264	0.32462	0.00402	0.14679
$Cd_{15}Se_{15}$	5.18003	3.10078	2.07925	0.01406	0.11319
$Cd_{24}Se_{24}$	2.76076	2.17122	0.58954	0.00348	0.16356
$Cd_{33}Se_{33}$	2.18953	1.95920	0.23033	0.01078	0.16583
$Cd_{42}Se_{42}$	1.83354	1.51175	0.32179	0.00310	0.11716
$Cd_{45}Se_{45}$	3.42302	1.06525	2.35778	0.00141	0.12704
$Cd_{54}Se_{54}$	2.15215	1.84267	0.30948	0.01002	0.11883
$Cd_{66}Se_{66}$	1.61582	1.14503	0.47079	0.00056	0.12920
$Cd_{144}Se_{144}$	1.71432	1.46049	0.25383	0.01236	0.12726

5.4 Discussion

5.4.1 Low-scaling excited state method

In this work we present a frequency-dependent electron-hole interaction kernel that does not exhibit steep scaling of computational cost with increasing system size. Traditional occupation-number approaches that use particle-hole excitation basis for excitation energy calculation suffer from high computational cost because the number of basis functions increase as $(N_{\text{occ}} \times N_{\text{vir}})$. As an example, this point is illustrated in Table 5.9 for the largest NPs investigated in this work. The last column of Table 5.9 shows that for all systems investigated in this work, use of the conventional method would require a particle-hole basis that is at least 10^6 or higher in size. In addition to that, the number of 2-electron integrals needed to construct the $A-$matrix increases as $(N_{\text{occ}} + N_{\text{vir}})^2$ which further contributes to the increasing the cost of the calculation. Finally, the AO-to-MO integral transformation which scales as $(N_{\text{occ}} + N_{\text{vir}})^5$ is a major contributor to the computational cost. All these factors contribute to the steep increase in the overall computational cost and is captured in Figure 5.2, where CIS line shows the cost of the conventional CIS calculation. The FD-GSIK line in Figure 5.2 showcases computation cost of the method developed in this work.

Table 5.9: Largest clusters investigated using LANL2DZ basis.

Cluster	Diameter (nm)	Number of AOs	Number of 1p-1h states
$Cd_{144}Se_{144}$	2.5	3744	3.17×10^6
$Cd_{87}S_{87}$	2.0	1392	1.16×10^6
$Pb_{140}Se_{140}$	2.5	2240	1.08×10^6
$Pb_{140}S_{140}$	2.5	2240	1.08×10^6

5.4.2 Frequency dependent kernel

The importance of frequency dependence for describing chemical systems is well-understood and have been studies extensively both in the context of MBPT[38, 40, 41] and TDDFT[64, 65, 66, 67]. In the FD-GSIK method, the frequency dependence in the interaction kernel allows the electron-hole interaction to change dynamically for different excitation and for different chemical systems.

The expression for the K_{eh} derived in Equation 6.27 is general and is applicable to a wide variety of choices for Λ by users.

5.4.3 Absence of adjustable parameters

Another significant feature of this work is that the kernel does not have adjustable parameters. The functional form presented here allows only two parameters, both of which are calculated during the course of the calculation. The inclusion on $(\omega_P^0 - \omega)$ makes the kernel dependent on the difference between the reference and the interacting system. The quantity $\langle P|r_{\mathrm{eh}}^{-1}|P\rangle$ is equal to the first-order correction in Rayleigh-Schrödinger perturbation theory (RSPT). Both of these terms modify the kernel dynamically with respect to the strength of the electron-hole interaction.

5.4.4 Future directions and extensions

In terms of directions for future research, the FD-GSIK kernel presents opportunities for extensions in other excited state phenomenons. Specifically, it is well-known in linear-response treatment of excited states that frequency-dependence in the electron-hole interaction kernel is crucial for describing two-electron excitations[68] and the present method can be extended to study biexcitonic interactions in semiconductor nanoparticles. Application to biexcitonic systems will involve defining $|P\rangle$ as the non-interacting 2p-2h reference state. The state $|Q\rangle$ will be obtained from the search in orthogonal 2p-2h space similar to procedure described here for 1p-1h systems. The FD-GSIK method can also be combined with electron-hole multicomponent coupled-cluster theory (eh-mcCC)[53, 54] for studying biexcitonic and multiexcitonic states. The FD-GSIK in its present form is not size-consistent and combining it with eh-mcCC will require a size-consistent formulation of the method.

The FD-GSIK method can also be extended to study systems that exhibit high density of states near the HOMO-LUMO gap such as gold and silver metallic clusters and large quantum dots. The issue of performing excited-state calculations in the presence of high density of particle-hole states is challenging and would require modification of the method presented here. One promising strategy is to use an orbital-optimized version of the present method, where dressed particle and hole state are obtained from a linear-combination of the bare particle-hole states.

$$|\tilde{h}\rangle = \sum_i c_i |i\rangle \tag{5.43}$$

$$|\tilde{e}\rangle = \sum_a c_a |a\rangle \tag{5.44}$$

where the $\mathbf{c}$ are obtained by minimizing $\langle \tilde{h}\tilde{e}|H_0 + V_{\mathrm{eh}}|\tilde{h}\tilde{e}\rangle$. Preliminary studies using this approach on gold nanoclusters were found to be promising and is currently being pursued.

5.5 Conclusions

In summary, the FD-GSIK method presented here implements a series of theoretical strategies to reduce the cost of an excited state calculation while maintaining the accuracy of the method. First, we have reduced the number of particle-hole states needed to describe the electron-hole interaction kernel by using an explicitly-correlated ansatz. Second, we have avoided the storage, evaluation, and transformation of atomic orbital integrals by performing evaluating the molecular orbitals integrals directly using the Monte Carlo method. Third, we have developed a frequency-dependent term in the electron-hole interaction kernel to account for dynamically screening. Finally, the FD-GSIK method is a first-principle method that is free of external parametrization and all required information is evaluated during the course of the calculation. The FD-GSIK method was applied to a series of quantum dots and the results from the calculations demonstrate the efficacy of the FD-GSIK method for reducing the scaling of computational effort with respect to system size.

Chapter 6

Investigation of Dense Manifold of Particle-Hole Excitations in Metallic Nanowires Using r12-Correlated Frequency Dependent Electron-Hole Interaction Kernel

6.1 Introduction

Noble metal nanoparticles, in particular silver and gold are of great interest for their applications in biomedicine,[69, 70, 71, 72, 73] catalysis,[74] energy conversion,[75, 76, 77, 78] and sensing.[79, 80, 81, 82] The potential in the above applications stems from strong absorption peaks in the visible to near IR region called surface plasmon resonance (SPR). [83, 84] A SPR is collective oscillations of electronic transitions due to the density of states within these metallic-like systems. Noble metal nanoparticles have been extensively studied both experimentally[85, 86, 87, 88] and theoretically [89, 90] to better understand the plasmonic nature.

The effect of composition, [91, 92] environment, [93] size,[94, 95, 96] and shape [97, 98, 95, 96] of noble metal nanoparticles is of particular interest as these cause tunability in the absorption peaks. In particular, nanorods and nanowires are of great interest due to the impact on their optical properties as a function of increased length. [99, 100, 101] Where a longitudinal mode along the main axis and a transverse mode perpendicular to the main axis are known to have varying affects due to increasing chain length. For which the longitudinal peak red shifts as length is increased and there is minimal effect of energy on the transverse peak.[102, 103]

A proper treatment of electron-correlation is then imperative towards better understanding these systems. Configuration interaction (CI) provides a simplistic method towards capturing electron-correlation. It is well known that with the limit of infinite basis, full configuration interaction (FCI) will exactly solve electron-correlation in the Schrödinger equation. The challenges are also well known with CI in the dramatically increasing computational cost for CI space. Tackling the challenge of size configuration space is a vast field of interest with various methods: Quantum Monte Carlo (QMC),[104] complete active space configuration interaction (CASCI) method, [105] geminal-projected configuration interaction (GP-CI) method,[45] many-body expanded full configuration interaction (MBE-FCI) method,[106, 107] multireference configuration interaction (MRCI), [108, 109, 110] multireference self consistent field (MR-SCF) theory,[111, 112] multireference second-order perturbation (MRMP2) theory, [113, 114] multi-state complete active space perturbation (MS-CASPT2) theory,[115] and rank-reduced full configuration interaction (RR-FCI).[116] The general goal within these methods and theories is to either restrict the the number of configurations through limiting the active space, or selecting only important configurations. The use of CI for nanoparticles is also limited by the cost of CI space, due to the number of single-particle states.[117, 118, 119] Time-dependent density-functional theory (TD-DFT) has demonstrated impactful insight into understanding the plasmon resonance within many noble metal nanowires and nanoparticles in obtaining electron-correlation.[120, 121, 122, 123, 124, 125, 126, 127, 128, 103] Though these TD-DFT studies have been useful, there still exist limitation for theoretical investigations for complex noble metal. There exists a class of ”magic clusters” that are theoretically investigated simply due to the use of high-symmetry within the structure to eliminate computational effort.[129, 130, 96] In general, noble metal systems can contain many electrons which require the use of many basis functions, thereby increasing the number of single-particle excitations increasing the computational effort to obtain electron-correlation. The limitations with CI and TD-DFT to investigate plasmon resonance within truly unique noble metal nanoparticles requires the development of new theoretical methods.[131]

In this work we will present an orbital-optimized dressed particle-hole basis, that allows us to compress the configuration space in a many-electron system but still capture the contribution to electron-hole correlation from all the single-particle excitations. The ideal of orbital-optimized excited states methods is not new, and is an area of interest with orbital-optimized Møller-Plesset perturbation (OO-MP) theory,[132, 133, 134] orbital-optimized coupled-cluster (OO-CC) methods,[135, 136] and various others theories.[137, 138, 139, 140, 141, 142] The goal of this work is to provide a formulation that captures the electron-hole correlation for plasmonic systems, or systems with highly degenerate orbitals with computational efficiency in mind by compression of the CI space. We will utilize our recently developed first-principle method, the frequency-dependent geminal-screened electron-hole interaction kernel method (FD-GSIK) to capture electron correlation which provides us a computational cost that is independent of system size. This work presented will not focus on the derivation of FD-GSIK but the formulation of our dressed molecular orbital basis and its application to silver nanowires to obtain correlated excitation energy for the longitudinal peak, the highest occupied molecular orbital (HOMO) to lowest unoccupied molecular orbital (LUMO) gap.

6.2 Theory

6.2.1 Dressed Molecular Orbitals

The effective many-body electron-hole Hamiltonian has the general form [50, 51, 52, 35, 53, 54]

$$H_{\mathrm{eh}} = H_{\mathrm{e}}^0 + H_{\mathrm{h}}^0 + V_{\mathrm{ee}} + V_{\mathrm{hh}} + V_{\mathrm{eh}} \tag{6.1}$$

where

$$H_{\mathrm{e}}^0 = \sum_{e_1 e_2} \langle e_1 | f^e | e_2 \rangle e_1^\dagger e_2 \tag{6.2}$$

$$H_{\mathrm{h}}^{0} = \sum_{h_1 h_2} \langle h_1 | f^h | h_2 \rangle h_1^\dagger h_2 \tag{6.3}$$

$$V_{\mathrm{ee}} = \sum_{e_1 e_2 e_3 e_4} \langle e_1 e_2 | w_{\mathrm{ee}} | e_4 e_3 \rangle e_1^\dagger e_2^\dagger e_3 e_4 \tag{6.4}$$

$$V_{\mathrm{hh}} = \sum_{h_1 h_2 h_3 h_4} \langle h_1 h_2 | w_{\mathrm{hh}} | h_4 h_3 \rangle h_1^\dagger h_2^\dagger h_3 h_4 \tag{6.5}$$

$$V_{\mathrm{eh}} = \sum_{h_1 h_2 e_1 e_2} \langle e_1 h_1 | v_{\mathrm{eh}} | e_2 h_2 \rangle e_1^\dagger h_1^\dagger h_2 e_2 \tag{6.6}$$

where $e^\dagger$ and $h^\dagger$ are particle and hole creation operators, and f_{e} and f_{h} are the Fock operators. The electron-hole Hamiltonian can be factored into a sum of non-interaction terms and interaction terms. For a 1-particle 1-hole state, further simplification can be achieved and the total Hamiltonian can be expressed as

$$H_{\mathrm{eh}} = H_{\mathrm{eh}}^{0} + V_{\mathrm{eh}} \tag{6.7}$$

The non-interacting Hamiltonian H_{eh}^{0} is obtained from the Fock operator whose eigenfunctions are (uncorrelated) particle-hole excited states

$$H_{\mathrm{eh}}^{0} | P \rangle = \omega_P^0 | P \rangle = (\varepsilon_a - \varepsilon_i) | \chi_i^{\mathrm{h}} \chi_a^{\mathrm{e}} \rangle \tag{6.8}$$

We are interested in understanding the many-body correction to a collection of single-particle excitations $\{\omega_P^0\}$, which are made up from all possible single-particle excitations from a set of particle states and a set of hole states as described below.

$$\{I\} = \{i, j, \ldots, N^h\} \tag{6.9}$$

$$\{A\} = \{a, b, \ldots, N^e\} \tag{6.10}$$

$$\{\omega_P^0\} = \{\omega_{P_i}^0, \ldots, \omega_{P_N}^0\} \tag{6.11}$$

The set of particle states ($\{I\}$) and set of hole states ($\{A\}$) do not have to contain all the particle or hole states, respectively, but in general consist of a subset of all particle/hole states. We then define the set of particle/hole states that do not exist in $\{I\}$ and $\{A\}$ as,

$$\{J\} = \{i', j', \ldots, N^{h'}\} \tag{6.12}$$

$$\{B\} = \{a', b', \ldots, N^{e'}\} \tag{6.13}$$

Note, that $\langle\{I\}|\{J\}\rangle = 0$ and that $\{I\}$ and$\{J\}$ contain all hole states for the system, these relations are the case for for the particle states as well. The wave-function for this collection of single-particle excitations is defined as,

$$|\tilde{P}\rangle = |he\rangle \tag{6.14}$$

$$|h\rangle = \sum_{i\in\{I\}} c_i |i\rangle \tag{6.15}$$

$$|e\rangle = \sum_{a\in\{A\}} c_a |a\rangle \tag{6.16}$$

in which we introduce a dressed uncorrelated molecular orbital , $|\tilde{P}\rangle$. Where the summations run over $N^h_{\tilde{P}}$ and $N^e_{\tilde{P}}$ number of hole orbitals and particle orbitals in set $\{I\}$ and $\{A\}$, respectively that contribute to the collective excitation for which we want to correct, along with coefficients c_i and c_a. Since we are interested in the collective excitation, the coefficients used in $|\tilde{P}\rangle$ are weighted so each particle/hole state contributes uniformly. From this point on in this presented work, a tilde will refer to dressed states as described in Equation 6.14.

We start by defining the zeroth-order approximation of the electron-hole wavefunction as,

$$|\Psi_X\rangle^{(0)} = |\tilde{P}\rangle \tag{6.17}$$

We now define the correlated electron-hole wavefunction Ψ_X as

$$|\Psi_X\rangle = C_{\tilde{P}}|\tilde{P}\rangle + C_{\perp}|\tilde{P}_{\perp}\rangle \tag{6.18}$$

where $C_{\tilde{P}}$ and $C_{\perp}$ are expansion coefficients and $\tilde{P}_{\perp}$ is the correlated wavefunction that exists in the orthogonal subspace of $\tilde{P}$. This is a very general form of the wavefunction and both MBPT and CI wavefunctions can be expressed in this form, where choice and construction of $\tilde{P}_{\perp}$ differentiates between treatment of electron-hole correlation within different methods. Here, we will define an electron-hole correlator operator Λ and express the $\tilde{P}_{\perp}$ as,

$$|\Psi_X\rangle = C_{\tilde{P}}|\tilde{P}\rangle + C_{\tilde{Q}}\Lambda(\omega, r_{\mathrm{eh}})|\tilde{Q}\rangle \tag{6.19}$$

where we define $|\tilde{Q}\rangle$ as,

$$|\tilde{Q}\rangle = |h'e'\rangle \tag{6.20}$$

$$|h'\rangle = \sum_{j\in\{J\}} c_j|j\rangle \tag{6.21}$$

$$|e'\rangle = \sum_{b\in\{B\}} c_b|b\rangle \tag{6.22}$$

where $|\tilde{Q}\rangle$ contains all particle-hole states not used in construction of $|\tilde{P}\rangle$ and $\langle\tilde{P}|\tilde{Q}\rangle = 0$.

6.2.2 Frequency-dependent geminal-screened electron-hole interaction kernel

At this point we have derived dressed molecular orbital basis and how we construct our correlated electron-hole wavefunction, Ψ_X as

$$|\Psi_X\rangle = C_{\tilde{P}}|\tilde{P}\rangle + C_{\tilde{Q}}\Lambda(\omega, r_{\mathrm{eh}})|\tilde{Q}\rangle \tag{6.23}$$

and are interested in solving,

$$H_{\text{eh}}|\Psi_X\rangle = \omega|\Psi_X\rangle \tag{6.24}$$

In this next section we will discuss briefly the frequency-dependent geminal-screened electron-hole interaction (FD-GSIK) method in terms of our correlated wavefunction. Full details of the derivation are in our previous work. FD-GSIK starts by projecting of $\langle\tilde{P}|$ and $\langle\tilde{Q}|$ on the subspace and utilizing orthonormalization, $\langle\tilde{P}|\tilde{Q}\rangle = 0$ to obtain,

$$\begin{bmatrix} \langle\tilde{P}|H_{\text{eh}}|\tilde{P}\rangle & \langle\tilde{P}|H_{\text{eh}}\Lambda(\omega)|\tilde{Q}\rangle \\ \langle\tilde{Q}|H_{\text{eh}}|\tilde{P}\rangle & \langle\tilde{Q}|H_{\text{eh}}\Lambda(\omega)|\tilde{Q}\rangle \end{bmatrix} \begin{bmatrix} C_{\tilde{P}} \\ C_{\tilde{Q}} \end{bmatrix} = \omega \begin{bmatrix} 1 & \langle\tilde{P}|\Lambda(\omega)|\tilde{Q}\rangle \\ 0 & \langle\tilde{Q}|\Lambda(\omega)|\tilde{Q}\rangle \end{bmatrix} \begin{bmatrix} C_{\tilde{P}} \\ C_{\tilde{Q}} \end{bmatrix} \tag{6.25}$$

After performing Löwdin's partitioning, we are able to obtain a equation for the electron-hole interaction kernel,

$$[H_{\tilde{P}\tilde{P}} + K_{\tilde{P}\tilde{P}}(\omega)]\, C_{\tilde{P}} = \omega C_{\tilde{P}} \tag{6.26}$$

where we obtain a closed-form analytical expression for our electron-hole interaction kernel, $K_{\tilde{P}\tilde{P}}(\omega)$ is defined as,

$$K_{\tilde{P}\tilde{P}}(\omega) = -\frac{\left[\langle\tilde{P}|r_{12}^{-1}\Lambda(\omega)|\tilde{Q}\rangle_A - (\omega_{\tilde{P}}^0 - \omega)\langle\tilde{P}|\Lambda(\omega)|\tilde{Q}\rangle\right]\langle\tilde{Q}|r_{12}^{-1}|\tilde{P}\rangle_A}{\langle\tilde{Q}|r_{12}^{-1}\Lambda(\omega)|\tilde{Q}\rangle_A + (\omega_{\tilde{Q}}^0 - \omega)\langle\tilde{Q}|\Lambda(\omega)|\tilde{Q}\rangle} \tag{6.27}$$

where $\langle\ldots\rangle_A$ represents anti-symmetrized matrix elements and $\Lambda(\omega, r_{\text{eh}})$ is,

$$\Lambda(\omega, r_{\text{eh}}) = \frac{(\omega_{\tilde{P}}^0 - \omega)}{\gamma} e^{-r_{\text{eh}}\gamma} \tag{6.28}$$

where $\gamma = \langle\tilde{P}|r_{\text{eh}}^{-1}|\tilde{P}\rangle$. Some key features to point out in FD-GSIK, first is the form of $\Lambda(\omega, r_{\text{eh}})$, which is absent of any adjustable parameters and is an infinite-order particle-particle excitation operator in real space representation.[37] Next is that our electron-hole interaction kernel has two forms of frequency dependency, first were obtained through the Löwdin's partitioning and the second is in the $\Lambda(\omega, r_{\mathrm{eh}})$ operator. Therefore obtaining ω is performed through the iterative solution of Equation 6.26 by utilizing a starting guess of $\omega = \omega^0_{\tilde{P}}$ in $K_{\tilde{P}\tilde{P}}(\omega)$.

6.2.3 Optimization of $|\tilde{Q}\rangle$ Dressing Coefficients

The zeroth-order electron-hole wavefunction is given by $|\Psi_X\rangle^{(0)} = |\tilde{P}\rangle$ and the choice of the $|\tilde{Q}\rangle$ plays an important role in accurate description of electron-hole correlation. In this work, we construct $|\tilde{Q}\rangle$ as,

$$|\tilde{Q}\rangle = |h'e'\rangle \tag{6.29}$$

$$|h'\rangle = \sum_{j\in\{J\}} c_j|j\rangle \tag{6.30}$$

$$|e'\rangle = \sum_{b\in\{B\}} c_b|b\rangle \tag{6.31}$$

where a coefficient is calculated as,

$$c_r = \frac{e^{-\alpha\Delta E_r^2}}{\sum_r e^{-\alpha\Delta E_r^2}} \tag{6.32}$$

with $\Delta E_r^2 = (\varepsilon_s^0 - \varepsilon_r)^2$ where ε_s^0 is the particle/hole energies from the lowest single-particle excitation (ω^0_Q) in $|Q\rangle$.

Determination of $\vec{c}$ is done by minimizing the first-order corrected excitation energy of a given dressed molecular orbital,

$$\min_{\alpha}\left[\langle\tilde{Q}|H_{eh}|\tilde{Q}\rangle\right] = \omega^0_{\tilde{Q}} + \langle\tilde{Q}|v_{\mathrm{eh}}|\tilde{Q}\rangle \tag{6.33}$$

since $\tilde{Q}$ is an uncorrelated particle-hole basis the last term can be expressed in terms of density,

$$\langle\tilde{Q}|v_{\mathrm{eh}}|\tilde{Q}\rangle = \langle\rho_{\tilde{Q}} r_{\mathrm{eh}}^{-1}\rangle \tag{6.34}$$

where the density, $\rho_{\tilde{Q}}$ is defined as,

$$\begin{aligned}
\rho_{\tilde{Q}}(1,2) &= \tilde{Q}(1,2)\tilde{Q}(1,2) \qquad (6.35)\\
&= h'(1)e'(2)h'(1)e'(2) \qquad (6.36)\\
&= h'(1)h'(1)e'(2)e'(2) \qquad (6.37)\\
\rho_{\tilde{Q}}(1,2) &= \rho_{h'}(1)\rho_{e'}(2) \qquad (6.38)
\end{aligned}$$

which provides particle and hole densities of the dressed states. These particle and hole densities are obtained through,

$$\rho_{h'}(1) = \sum_{j\in\{J\}} c_j^2 \rho_j^{h'}(1) \tag{6.39}$$

$$\rho_{e'}(2) = \sum_{b\in\{B\}} c_b^2 \rho_b^{e'}(2) \tag{6.40}$$

using the particle and hole densities that eigenfunctions of the Fock operator. The search for the coefficients scales as $N_\alpha \times (N_{\tilde{Q}}^{h'} \times N_{\tilde{Q}}^{e'})$ requiring search over all states. We reduce the computational cost by approximating the true densities $\rho_j^{h'} \rho_b^{e'}$, through a Gaussian approximation such that,

$$\rho_r^{e,h} \approx \rho_r^{G;e,h} = \frac{1}{\sigma_r^{\mathrm{e,h}}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{r-\mu_r^{\mathrm{e,h}}}{\sigma_r^{\mathrm{e,h}}}\right)} \tag{6.41}$$

and are obtained by minimizing the Kullback–Leibler divergence[55] between the true densities $\rho_j^{h'} \rho_b^{e'}$ and the Gaussian probability distribution functions.

$$D_{\text{Kullback−Leibler}} = \int_V d\mathbf{r} \rho_j^{h'} \rho_b^{e'} \ln\left[\frac{\rho_j^{h'} \rho_b^{e'}}{\rho_j^{G;h'} \rho_b^{G;e'}}\right] \tag{6.42}$$

The computational effort for determination of approximate densities $\rho_r^{G;h'}, \rho_r^{G;e'}$ scales linearly with $N_\alpha \times (N_{\tilde{Q}}^{h'} + N_{\tilde{Q}}^{e'})$ and the calculation of $\langle \rho_{\tilde{Q}} r_{\mathrm{eh}}^{-1} \rangle$ is performed analytically. Where $N_{\tilde{Q}}^{h'}$ and $N_{\tilde{Q}}^{e'}$ are the number of particle/hole orbitals in set $\{J\}$ and $\{B\}$. Both of these steps allow us to search for the optimal α in construction of the $\vec{c}$ coefficients at a reduced computational cost. Once the coefficients are obtained construction of $|\tilde{Q}\rangle$ is performed numerically.

6.3 Results and Discussion

6.3.1 Computational Details

The current work utilizes the above derivation of orbital-optimized dressed molecular orbitals and our FD-GSIK method to obtain correlated excitation energies of Ag_n nanowires where $n = 2, 4, 6, 8, 10, 12, 20, 40, 60, 80, 100$ using FD-GSIK method. The nanowires were constructed using a bond distance of 3.51 Å which was obtained by using the central Ag-Ag bond distance obtained from a Hartree-Fock (HF) optimized Ag_{100} calculation with LANL2DZ basis and effective core potential (ECP). The uncorrelated ground state single-particle states were calculated using HF method and LANL2DZ basis and ECP for the nanowires with the TERACHEM package. [143] The spatial MOs, $\psi(\mathbf{r})$ were generated with 100 points per dimension to generate a 10^6 point grid to obtain a dense grid for construction of the dressed molecular orbitals, $\tilde{\psi}(\mathbf{r})$. The grid boundary was set by imposing a cutoff tolerance of $|\psi(\mathbf{r})| \leq 10^{-12}$. The Monte Carlo integration was performed in decuplicate for obtaining maximum and minimum errors on the order of meV. Within this work we performed two sets of calculations utilizing different $|\tilde{P}\rangle$ states. In the first set of calculations we were interested in only the HOMO-LUMO gap, therefore $|\tilde{P}\rangle = |P\rangle$. As this gap was either the highest or second highest oscillator strength for all Ag nanowires. The next set of calculations looked into $|\tilde{P}\rangle$ that included all the single-particle excitations that were within 1 eV of the HOMO-LUMO gap (ω_P^0). This therefore gave us a collection of single-particle excitations near the band edge of the system. The dressed molecular orbitals were calculated as described in subsection 6.2.3 to obtain the $|\tilde{Q}\rangle$ state for FD-GSIK.

6.3.2 HOMO-LUMO Gap for Ag Nanowires

The first investigation utilizing dressed molecular orbitals was to obtain electron-correlation for a specific excitation, the HOMO-LUMO gap. Which is classified as the longitudinal peak in various reported works,[102, 103] and known to red shift as length is increased. We are interesting in understanding this trend for longer Ag nanowires and comparing to results from the previously reported TD-DFT work of Guidez and Aikens.[144] Utilizing FD-GSIK allows for calculation of excitation energies where the computational cost is independent of system size, therefore if the ground state wavefunction can be obtained, FD-GSIK can provide correlated excitation energies. This allowed for an investigation up to Ag_{100}, Figure 6.1. The reported HOMO-LUMO gap (ω_P^{HF})

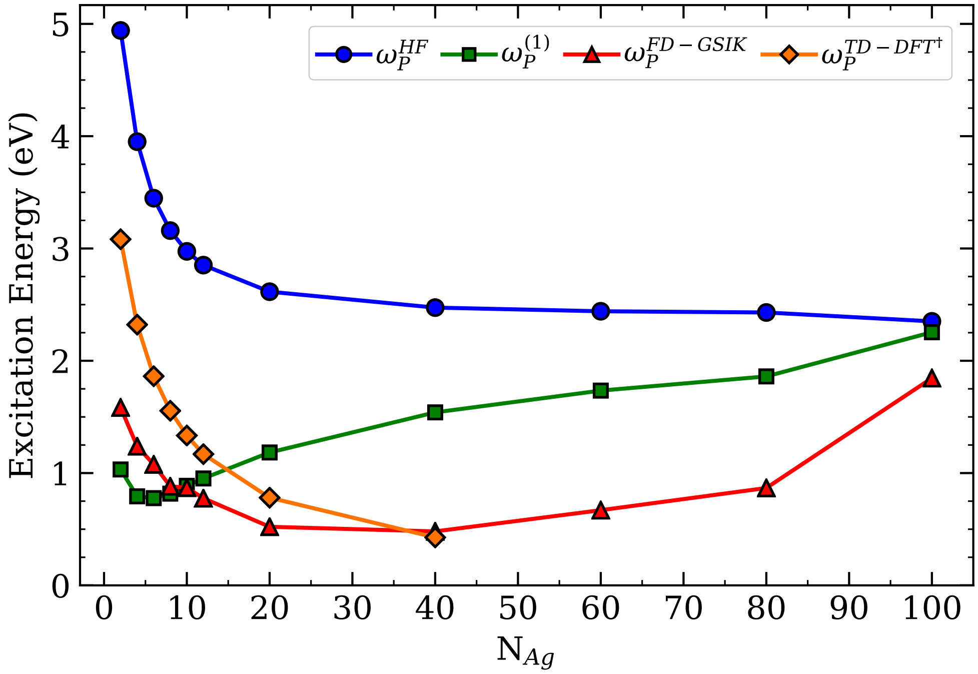


Figure 6.1: Excitation energy of Ag nanowires using FD-GSIK for the HOMO-LUMO gap in comparison to reported TD-DFT results. †[144]

shows a red shift in the energy as system size increases. In addition, the first order corrected excitation defined by $\omega_P^{(1)} = \omega_P^{HF} - \langle P|r_{12}^{-1}|P\rangle$, is reported. Also observed is that $\omega_P^{FD-GSIK}$ and ω_P^{TD-DFT} are in great agreement for the nanowires up to Ag_{40}, replicating the trend that is commonly reported for this small nanowire series. Interestingly as the number of Ag atoms increases the trend is non-convergent and there is an increase in energy, or a blue shift, in the excitation as the system approaches Ag_{100}. The use of dressed molecular orbitals and FD-GSIK has allowed for investigation of larger nanowires and demonstrates a non-obvious trend in the excitation energy with respect to system length. This was accomplished by the computationally efficient FD-GSIK method which was developed for computationally challenging systems through the use of a compressed molecular orbital basis.

6.3.3 Collection of single-particle excitations for Ag Nanowires

In addition we were interested in the set of single-particle excitations near the HOMO-LUMO gap. As this set will give us insight into the effect of the multi-reference nature of high density of states near the HOMO-LUMO gap, this is easily observed for Ag_{100} in, Figure 6.2. Which shows a clear HOMO-LUMO gap but that there exist numerous orbitals that lie near each other around the HOMO-LUMO gap. Clear breaks in the continuum of energy is not observed until -10 eV and 15 eV for the occupied and virtual space, respectively. In this work we are mainly interested in the single-particle excitations that exist within 1 eV of the HOMO-LUMO gap, as these are the most contributing to the degeneracy near the HOMO-LUMO gap. This defines our $|\tilde{P}\rangle$ state, in which each unique hole/particle state is evenly weighted and $|\tilde{Q}\rangle$ consists of every particle-hole state not in $|\tilde{P}\rangle$. The results are shown in Figure 6.3. Observed is a red shift in the collective excitation up to Ag_{40} and then an increase in energy for Ag_{60} and Ag_{70} with a slight decrease in energy for Ag_{100}.

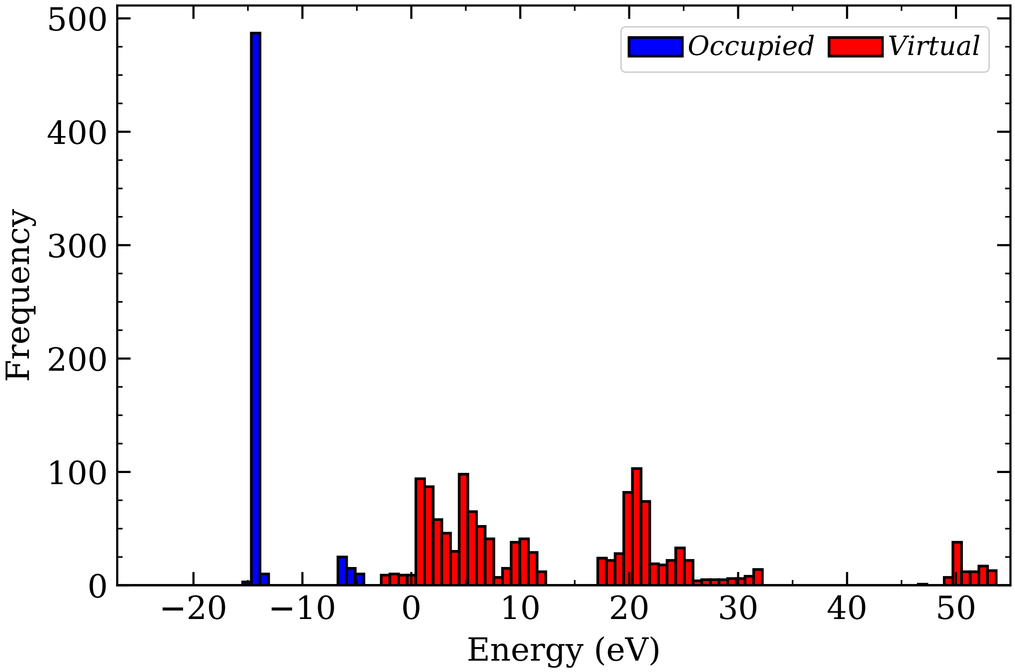


Figure 6.2: Histogram of molecular orbital energies for Ag_{100} near the HOMO-LUMO gap.

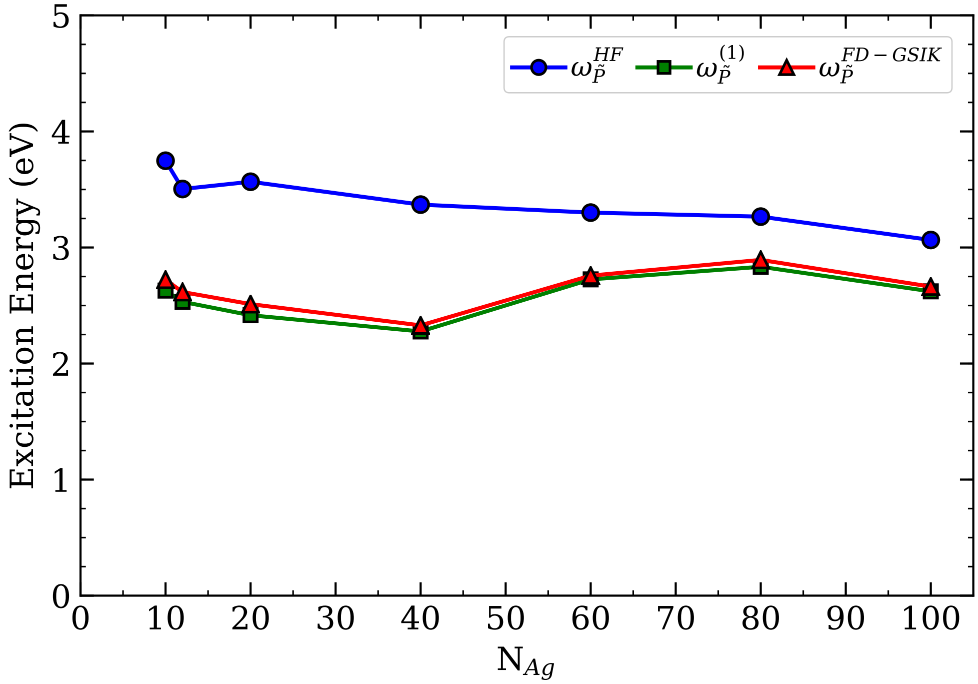


Figure 6.3: Comparison of the uncorrelated gap to the first-order corrected and FD-GSIK correlated excitation obtained for $|\tilde{P}\rangle$ in Ag nanowires.

The uncorrelated ground state energies are in general higher due to the multi-reference excitation energy used in construction of the dressed molecular orbital basis.

6.3.4 Error analysis and timing data

With FD-GSIK the use of Monte Carlo integration allows us to reduce the error in the excitation energies in a computationally efficient approach. Presented in Table 6.1 and Table 6.2 are the standard deviations of the FD-GSIK excitation energies for all the systems presented. In addition, the single-core total cumulative CPU time is reported for each system for the 1+10 MC integration's, one run to evaluate $\langle\tilde{P}|r_{12}^{-1}|\tilde{P}\rangle$ and then ten FD-GSIK integrations. These standard deviations result in meV errors, providing confidence in the reported energies. The error within a FD-GSIK calculation are systematically improvable through additional MC integrations, which can be performed in at low computational cost due to the construction of the dressed molecular orbital and FD-GSIK method.

Table 6.1: The standard deviation of $\omega_P^{FD-GSIK}$ from 1+10 runs of the Monte Carlo integration with total single-core CPU computational time.

Chemical Formula	$\omega_P^{FD-GSIK}$ (eV)	Standard Deviation (meV)	Total CPU Time (Hours)
Ag_2	1.5835	3.68E-01	1.31
Ag_4	1.2387	1.57E-01	1.35
Ag_6	1.0766	1.97E-01	1.36
Ag_8	0.8822	3.52E-01	1.35
Ag_{10}	0.8690	2.62E-01	1.32
Ag_{12}	0.7757	4.75E-01	1.33
Ag_{20}	0.5217	3.70E-01	1.34
Ag_{40}	0.4805	6.96E-01	1.32
Ag_{60}	0.6695	5.50E-01	1.33
Ag_{80}	0.8671	4.44E-01	1.35
Ag_{100}	1.8456	2.59E-01	1.33

Table 6.2: The standard deviation of $\omega_{\tilde{P}}^{FD-GSIK}$ from 1+10 runs of the Monte Carlo integration with total single-core CPU computational time.

Chemical Formula	$\omega_{\tilde{P}}^{FD-GSIK}$ (eV)	Standard Deviation (meV)	Total CPU Time (Hours)
Ag_{10}	2.7217	5.32E-02	1.34
Ag_{20}	2.5120	9.68E-02	1.36
Ag_{40}	2.3287	3.38E-01	1.34
Ag_{60}	2.7574	1.30E-01	1.35
Ag_{80}	2.8929	1.56E-01	1.34
Ag_{100}	2.6626	1.84E-01	1.34

6.4 Conclusions

The dressed molecular orbital basis was presented to compress the configuration space for understanding multi-reference single-particle excitations. Allowing for a computationally efficient multi-reference molecular orbital basis that can tackle large many-electron systems which have many single-particle excitations. The use of dressed molecular orbitals allowed us to investigate a highly degenerate metallic like system, silver nanowires. But in general, this formulation can be used for other chemical systems in which multi-reference characteristics are of interest. In conjunction with the frequency-depend geminal-screened electron-hole interaction kernel (FD-GSIK) method we were able to obtain electron-correlated excitation energies for the HOMO-LUMO gap and a collection of excitations near the HOMO-LUMO gap up to Ag_{100}. Observed within the presented work is a non-obvious trend in the energy of the HOMO-LUMO gap which blue shifts at longer nanowire lengths.

Chapter 7

Derivation of Moment Generated Molecular Orbital Basis

7.1 Introduction

Presented in this chapter is a derivation for the moment generated molecular orbital basis. The purpose of this derived work serves as an alternative to the frequency-dependent geminal-screened electron-hole interaction kernel presented in Chapter 5 and Chapter 6. The main limitation within FD-GSIK is that the $\Lambda(\omega, r_{eh})$ projection operator will not project to particle-hole states lower than the specified $|Q\rangle$ state. Therefore the selection of $|Q\rangle$ is limited to a particle-hole state that is in general the next particle-hole state after $|P\rangle$. The goal of the moment generate molecular orbital basis was to provide an electron-correlated approach that was free of the $\Lambda(\omega, r_{eh})$ projection operator and utilized the moments of the particle/hole states to capture electron-correlation. The body of this chapter is entirely the derivation of the method and will contain no results to validate the method.

7.2 Theory

The effective many-body electron-hole Hamiltonian has the general form

$$H_{\mathrm{eh}} = H_{\mathrm{e}}^{0} + H_{\mathrm{h}}^{0} + V_{\mathrm{ee}} + V_{\mathrm{hh}} + V_{\mathrm{eh}} \tag{7.1}$$

where

$$H_{\mathrm{e}}^{0} = \sum_{e_1 e_2} \langle e_1 | f^e | e_2 \rangle e_1^\dagger e_2 \tag{7.2}$$

$$H_{\mathrm{h}}^{0} = \sum_{h_1 h_2} \langle h_1 | f^h | h_2 \rangle h_1^\dagger h_2 \tag{7.3}$$

$$V_{\mathrm{ee}} = \sum_{e_1 e_2 e_3 e_4} \langle e_1 e_2 | w_{\mathrm{ee}} | e_4 e_3 \rangle e_1^\dagger e_2^\dagger e_3 e_4 \tag{7.4}$$

$$V_{\mathrm{hh}} = \sum_{h_1 h_2 h_3 h_4} \langle h_1 h_2 | w_{\mathrm{hh}} | h_4 h_3 \rangle h_1^\dagger h_2^\dagger h_3 h_4 \tag{7.5}$$

$$V_{\mathrm{eh}} = \sum_{h_1 h_2 e_1 e_2} \langle e_1 h_1 | v_{\mathrm{eh}} | e_2 h_2 \rangle e_1^\dagger h_1^\dagger h_2 e_2 \tag{7.6}$$

where $e^\dagger$ and $h^\dagger$ are particle and hole creation operators, and f_{e} and f_{h} are the Fock operators. The electron-hole Hamiltonian can be factored into a sum of non-interaction terms and interaction terms. For a 1-particle 1-hole state, further simplification can be achieved and the total Hamiltonian can be expressed as

$$H_{\mathrm{eh}} = H_{\mathrm{eh}}^{0} + V_{\mathrm{eh}} \tag{7.7}$$

The non-interacting Hamiltonian H^0_{eh} is obtained from the Fock operator whose eigenfunctions are (uncorrelated) particle-hole excited states

$$H^0_{\mathrm{eh}} = \omega^0_P |P\rangle = (\varepsilon^e_a - \varepsilon^h_i)|\Phi^h_0 \Phi^e_0\rangle \tag{7.8}$$

We are interested in calculating the many-body correction to ω^0_P due to electron-hole correlation in the system.

$$H_{\mathrm{eh}}|\Psi_X\rangle = \omega|\Psi_X\rangle \tag{7.9}$$

We start by defining the zeroth-order approximation of the electron-hole wavefunction as,

$$|\Psi_X\rangle^{(0)} = |P\rangle \tag{7.10}$$

and the correlated electron-hole wavefunction Ψ_X is defined as

$$|\Psi_X\rangle = c_P|P\rangle + c_Q|Q\rangle \tag{7.11}$$

where,

$$|P\rangle = \Phi^h_0 \Phi^e_0 \tag{7.12}$$

$$|Q\rangle = \left(\Omega_h \Phi^h_0\right)\left(\Omega_e \Phi^e_0\right) \tag{7.13}$$

In this presented derivation $\Omega_{e,h}$ is a moment operator that is defined as $\Omega_{e,h} = \left(r^m_{e,h} - \langle r^m_{e,h}\rangle\right)$, where m is the moment. The moment operator has the following property,

$$\langle \Phi^{e,h}_0 | \Omega_{e,h} | \Phi^{e,h}_0 \rangle = 0 \tag{7.14}$$

$$\langle \Phi^{e,h}_0 | \left(r^m_{e,h} - \langle r^m_{e,h}\rangle\right) | \Phi^{e,h}_0 \rangle = \langle \Phi^{e,h}_0 | r^m_{e,h} | \Phi^{e,h}_0 \rangle - \langle r^m_{e,h}\rangle = 0 \tag{7.15}$$

therefore $\Omega_h\Omega_e$ in Equation 7.13 can be expanded as,

$$\Omega_h\Omega_e = (r_h^m - \langle r_h^m \rangle)\left(r_e^{m'} - \langle r_e^{m'} \rangle\right) \tag{7.16}$$

$$= \left(r_h^m r_e^{m'} + \langle r_h^m \rangle \langle r_e^{m'} \rangle - r_h^m \langle r_e^{m'} \rangle - \langle r_h^m \rangle r_e^{m'}\right) \tag{7.17}$$

where m and m' are different moments for the hole and particle state respectively. Projecting $\langle P|$ and $\langle Q|$ onto Equation 7.9 results in the following matrix expression,

$$\begin{bmatrix} \langle P|H_{\text{eh}}|P\rangle & \langle P|H_{\text{eh}}\Omega_h\Omega_e|Q\rangle \\ \langle Q|H_{\text{eh}}|P\rangle & \langle Q|H_{\text{eh}}\Omega_h\Omega_e|Q\rangle \end{bmatrix} \begin{bmatrix} C_P \\ C_Q \end{bmatrix} = \omega \begin{bmatrix} 1 & 0 \\ 0 & \langle Q|\Omega_h\Omega_e|Q\rangle \end{bmatrix} \begin{bmatrix} C_P \\ C_Q \end{bmatrix} \tag{7.18}$$

simplified as,

$$\begin{bmatrix} H_{PP} & H_{PQ} \\ H_{PQ} & H_{QQ} \end{bmatrix} \begin{bmatrix} C_P \\ C_Q \end{bmatrix} = \omega \begin{bmatrix} 1 & 0 \\ 0 & S_{QQ} \end{bmatrix} \begin{bmatrix} C_P \\ C_Q \end{bmatrix} \tag{7.19}$$

for which **H** can be expanded as,

$$\begin{bmatrix} H_{PP} & H_{PQ} \\ H_{PQ} & H_{QQ} \end{bmatrix} = \begin{bmatrix} H_{PP}^0 & H_{PQ}^0 \\ H_{PQ}^0 & H_{QQ}^0 \end{bmatrix} + \begin{bmatrix} V_{PP} & V_{PQ} \\ V_{PQ} & V_{QQ} \end{bmatrix} \tag{7.20}$$

for which the definitions of the matrix elements are as follows:

$$V_{PP} = \langle \Phi_0^h \Phi_0^e | r_{12}^{-1} | \Phi_0^h \Phi_0^e \rangle \tag{7.21}$$

$$V_{PQ} = \langle \Phi_0^h \Phi_0^e | r_{12}^{-1} \Omega_h \Omega_e | \Phi_0^h \Phi_0^e \rangle \tag{7.22}$$

$$V_{QQ} = \langle \Phi_0^h \Phi_0^e | \Omega_h \Omega_e r_{12}^{-1} \Omega_h \Omega_e | \Phi_0^h \Phi_0^e \rangle \tag{7.23}$$

$$H_{PP}^0 = \omega_P^0 \tag{7.24}$$

$$H_{PQ}^0 = \langle P|H_{eh}^0|Q\rangle = \omega_P^0 \langle P|Q\rangle = 0 \tag{7.25}$$

$$H_{QQ}^0 = \langle Q|H_{eh}^0|Q\rangle = \langle \Omega_h \Phi_0^h \Omega_e \Phi_0^e | H_h^0 + H_e^0 | \Omega_h \Phi_0^h \Omega_e \Phi_0^e \rangle \tag{7.26}$$

$$= \langle \Omega_h \Phi_0^h | H_h^0 | \Omega_h \Phi_0^h \rangle \langle \Phi_0^e | \Omega_e \Omega_e | \Phi_0^e \rangle + \langle \Omega_e \Phi_0^e | H_e^0 | \Omega_e \Phi_0^e \rangle \langle \Phi_0^h | \Omega_h \Omega_h | \Phi_0^h \rangle \tag{7.27}$$

$$H_{QQ}^0 = \sum_k c_k^2 E_k^{0,h} \sum_k c_k^2 + \sum_k c_k^2 E_k^{0,e} \sum_k c_k^2 \tag{7.28}$$

with $c_k = \langle k | r^m | 0 \rangle$.

7.3 Conclusion

That concludes the derivation of moment generated molecular orbital basis which allows for obtaining electron-correlation for any given $|P\rangle$ and $|Q\rangle$ state. The advantage of the presented derivation is that the choice of $|P\rangle$ and $|Q\rangle$ is not restricted by the correlated operator, unlike the work presented in Chapter 5 and Chapter 6. The implementation and validation of the presented derivation is still in order, but the derivation in its presented form is still a novel approach for efficient implementation of electron-hole interaction kernel.

Chapter 8

Implementation of Permutation Sampling Monte Carlo Integration Method

8.1 Introduction

The permutation sampling Monte Carlo method is a universal integration technique that was developed within the Chakraborty reserach group to tackle the challenges of multi-dimensional integrals. Within the research group we are heavily interested in avoiding the atomic-orbital to molecular-orbital integral transformation. While developing new excited state methods for the grid based approaches, integrals become burdensome in higher dimensions. This chapter will focus on the computer science of the recent code implementation in C++ for permutation sampling Monte Carlo method as key decisions were made to program a code that was computationally efficient, portable, and multi-purpose.

8.2 Theory

The theory presented here is to briefly introduce the concepts of the permutation sampling Monte Carlo integration method which is a part of a larger investigation in ionization energies using

Green's function within the group. To begin, let us evaluate a simple 1-D Gaussian function,

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \tag{8.1}$$

using Monte Carlo method of a million uniform random points. This is a simple enough task as we can easily evaluate these million points in a matter of seconds, the accuracy of the integral evaluation is very reasonable with error of 1x10^{-3}. Now lets move up to a 3-D Gaussian function,

$$\int_{-\infty}^{\infty} e^{-\mathbf{r}^2} d\mathbf{r} \tag{8.2}$$

where $\mathbf{r}^2 = x^2 + y^2 + z^2$, therefore the 3-D Gaussian integral is a product of three 1-D Gaussian functions.

$$\int_{-\infty}^{\infty} e^{-\mathbf{r}^2} d\mathbf{r} = \int_{-\infty}^{\infty} e^{-x^2} dx \times \int_{-\infty}^{\infty} e^{-y^2} dy \times \int_{-\infty}^{\infty} e^{-z^2} dz \tag{8.3}$$

$$= \sqrt{\pi} \times \sqrt{\pi} \times \sqrt{\pi} = \pi^{\frac{3}{2}} \tag{8.4}$$

Now a choice must be made in terms of evaluating this 3-D integral with respect to the number of Monte Carlo points. If we want to maintain the initial conditions from the 1-D integral that means the 3-D integral could be evaluated with 3.0E6 Monte Carlo points. This choice in evaluation will result is great accuracy of the integral value, but the cost of the calculation is high, ${N_{\text{M.C.}}}^3$, a couple of hours. The other choice to be made in terms of evaluating the integral would be to maintain a million points between all three dimensions. The computational cost is now $(\frac{N_{\text{M.C.}}}{3})^3$ which is now a few minutes. But the accuracy of the integral evaluation will be on the order of 1.0E0. As this simple example demonstrates dimensionality of the integral evaluation can be computationally burdensome, even for 3-D. But what if someone is interested in two particle or four particle complex integrals that are 6-D or 12-D as is the nature of the work discussed in Chapter 5

Let us consider an generic 12-dimensional integral,

$$I = \int_D d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \rho(\mathbf{r}_1, \mathbf{r}_2 d, \mathbf{r}_3 d, \mathbf{r}_4) \frac{\Lambda(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)}{\rho(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)} \tag{8.5}$$

where $\Lambda(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$ is a 4-particle function which is evaluated through sampling from $\rho(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$, where $\rho(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$ is a valid probability distribution function that is defined as,

$$\int_D d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \rho(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4) = 1 \tag{8.6}$$

$$\rho(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4) \geq 0 \tag{8.7}$$

Permutation sampling Monte Carlo begins by decoupling the dimensionality of the N-dimensional point for evaluation by choosing each dimension of the N dimensions based on the probability distribution within each N dimension. In essence, instead of sampling each Monte Carlo point from a N-dimensional probability distribution, the N-dimensional point is constructed from uncoupled N probability distributions. The decoupling starts with a coordinate transformation where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4$ corresponds to q values, such that $q \in [0, 1]$.

$$I = \int_{-\infty}^{\infty} d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \Lambda(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4) \tag{8.8}$$

$$= \int_1^0 dq_1 dq_2 dq_3 \dots dq_{12} \Lambda(q_1 q_2 q_3 \dots q_{12}) J(q_1 q_2 q_3 \dots q_{12}) \tag{8.9}$$

where J is the Jacobian associated with the coordinate transformation. This allows for the integration to occur in intervals of $[0, 1]$ instead of infinite bounds. This coordinate transformation also allows for the decoupling of the $\rho(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$ distribution that will be sampled from in Equation 8.5 by making ρ to be a product of 12 one-dimensional probability distribution functions in q-space. In essence, instead of sampling each Monte Carlo point from a N-dimensional probability distribution, the N-dimensional point is constructed from uncoupled N probability distributions,

$$\rho(q) = \rho_1(q_1) \times \rho_2(q_2) \times \rho_3(q_3) \times \dots \rho_{12}(q_{12}) \tag{8.10}$$

The next key component to permutation sampling Monte Carlo is how these individual distributions, generically labeled as $\rho_d(q_d)$ are evaluated to sample, $\rho(q)$ in Equation 8.8. Imagine that each distribution, $\rho_d(q_d)$ is a deck of cards, and a hand ($\rho(q)$) is dealt by drawing the top card of each unique deck. This can be demonstrated as,

$$\begin{array}{ccccc} \rho_1 & \rho_2 & \rho_3 & \cdots & \rho_{12} \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ q_1^A & q_2^A & q_3^A & \cdots & q_{12}^A \\ q_1^B & q_2^B & q_3^B & \cdots & q_{12}^B \\ \vdots & \vdots & \vdots & & \vdots \\ q_1^Z & q_2^Z & q_3^Z & \cdots & q_{12}^Z \end{array} \tag{8.11}$$

where the superscripts $A, B, \ldots, Z$ are the number of hands dealt. In this situation a hand would then be the $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4$ point to be evaluated during Monte Carlo integration. The next Monte Carlo point will then be the next hand dealt from the 12 one-dimensional probability distribution functions. To improve the accuracy of the integral value each deck could then be shuffled continuing to evaluate Monte Carlo points as described, thereby allowing for redistribution of the probability distribution function.

$$\begin{array}{ccccc} \rho_1 & \rho_2 & \rho_3 & \cdots & \rho_{12} \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ q_1^G & q_2^F & q_3^T & \cdots & q_{12}^W \\ q_1^D & q_2^I & q_3^M & \cdots & q_{12}^V \\ \vdots & \vdots & \vdots & & \vdots \\ q_1^P & q_2^L & q_3^X & \cdots & q_{12}^A \end{array} \tag{8.12}$$

Permutation sampling Monte Carlo then allows for evaluation of complex multi-dimensional integrals as a one-dimensional integral in q-space. The derivation of permutation sampling Monte Carlo was demonstrated for a 12-dimensional generic integral with an undefined Λ function. In practice permutation sampling Monte Carlo is N-dimensional and the choice of function is independent to the sampling. Permutation sampling Monte Carlo is a generic integration technique that can be applied to any choice of integral evaluation.

8.3 Object-Oriented Programming

As stated, the theory of permutation sampling Monte Carlo is generic in terms of the number of dimensions for integration and the function of choice. Therefore it was a great coding project to practice the programming paradigm, object-oriented programming (OOP). OOP is a programming paradigm based on creating ”objects” that contains both data and the functions for the data. [145] In general objects are instances of classes, which are definitions of the data format and available procedures within the class. Objects and classes are readily seen within the real world and not simply in programming. There are major principles to discuss with regards to OOP, they are polymorphism, composition and inheritance, abstraction, and encapsulation.

8.3.1 Polymorphism

Polymorphism is one of many forms. For example, think of a hydrogen, helium and sodium atom, each of these are objects. They all belong to a element class, which contains the information on their name, symbol, atomic number, atomic weight, etc. The individual response each atom has to electromagnetic radiation is an example of a procedure as the electromagnetic radiation is information going to the atom, the atom then has its own procedure on what to do with that information. Polymorphism allows for construction of multiple objects from the same class.

8.3.2 Composition and Inheritance

Let us now consider dimers of the atoms, creating structure of two atoms. The structure can be stored in a class molecule, which contains information on the number of atoms, the position of the atoms and which atoms are bonded to each other. In the example I have created a new class that that contains two instances of class atoms. But the way the molecule class interacts with these atom classes has two different forms, composition or inheritance.

First is composition, where the molecule class is composed of two simple atoms classes. Composition is a useful tool when developing in that the top level object can be constructed through a complex network of other objects. When information is asked to the molecule class about the properties contained in the atom class, molecule will ask the atom class for that information. Second is inheritance where the molecule class has inherited two atom classes and their separate functionality. The key difference comes with how the molecule class uses the information from each atom class. In inheritance the information and functionality that is unique of an atom class can be utilized by molecule class. Therefore molecule class can take responsibility for passing information of atomic number or weight without have to ask the atom class. The doctrine of composition over inheritance was implemented within permutation sampling Monte Carlo. This doctrine means objects were responsible for their own functionality and their information.

8.3.3 Abstraction

Abstraction is the idea that the complexity associated with a function or information within a object is obscured by the requester of the information. For example in the theory of permutation sampling Monte Carlo, there was a discussion between integrating a Gaussian or a complex quantum chemistry problem. The user defined function is the example of the top level object, as this is the information that is needed when sampling the integral, *i.e.* what is the value of the integral at this position. In the case of a Gaussian it is simply, $f(\mathbf{r}) = e^{-\mathbf{r}^2}$, but for a complex quantum chemistry problem the value at position r needs information on molecular orbitals at $\chi(r)$. This is beneficial because it allows classes to be dynamic in their use as long as they contain the matching procedures at the top-level. But in terms of the implementation of permutation sampling Monte Carlo, it has no preference to how the value is being calculated, it just needs to know what value to update within the calculation. Abstraction is hiding internal implementation details to other objects when information is requested.

8.3.4 Encapsulation

Encapsulation is how the data within the object is made private to other objects. The principle of encapsulation is about which objects can obtain or modify data that is owned by another class. The information within an object can be public or private data. Where public data can just be utilized by outside objects, but private data can not be easily accessed. That does not mean private data is never recoverable, but has to be requested and it is up to owner of the data if that information can be passed along. Encapsulation is in the same light as abstraction but is further restrictive towards the access to information.

8.4 C++ Language Features

With this programming paradigm framework set, the natural programming language of choice was C++, a language that was built by Bjarne Stroustrup as ”C with Classes”. As it easily allows many of the principles of OOP in quick and easily implementation as seen in Figure 8.1. In addition C++ allows much more creative and beneficial features such as references, encapsulation of copy constructor and portability to CUDA C. References are denoted with ”&” and used as an alias to an already existing variable. References are important within coding in that references reduce the memory footprint of a code. Notice in Line 7 (Figure 8.1) the declaration and assignment of memory of two double variables (*atm1_weight* and *atm2_weight*). These variables are then passed by reference (Line 23) to the function *get_weight*() , which allows the function to directly effect the data stored in memory for *get_weight*() . If these variables were passed by value, a copy of the values would be sent to *get_weight*() where any modifications would not be reflected in Line 7 as these values are copies of the original data, *atm1_weight* and *atm2_weight*. Utilizing references within permutation sampling Monte Carlo allows for dramatic memory reduction as duplication of information is not performed. The next benefit within C++ is the encapsulation of the copy constructor for classes. This is demonstrated in Line 26 were the copy constructor is put into private information contained within atom class. This therefore enforces that any instance of the atom class must be passed by reference. Therefore any large object that is created will not be

Figure 8.1: Generic C++ class that demonstrates OOP principles.

```
class Molecule {
  public: //Public access for information
    //Functions that provide private information
    void get_number_atoms( int &natm) { natm = natm_ ; } ;
    void get_molecular_weight(double &weight) =
    {
          double atm1_weight, atm2_weight ;
          // Request for information from Atom class [Composition and
              Abstraction]
          atm1_get_weight(atm1_weight) ;
          atm2_get_weight(atm2_weight) ;
          weight = atm1_weight + atm2_weight;
    } ;
    [...]
  private: //Private information denoted with '_' at end [Encapsulation]
    int natm_ ;
    //[Polymorphism]
    Atom atm1_ ;
    Atom atm2_ ;
    [..]
 };
 class Atom {
  public:
    void get_weight(double &weight) { weight = weight_ ;} ;
    [...]
  private:
    double weight_ ;
    Atom(const Atom& rhs) ; //Copy Constructor
    [...]
 };
```

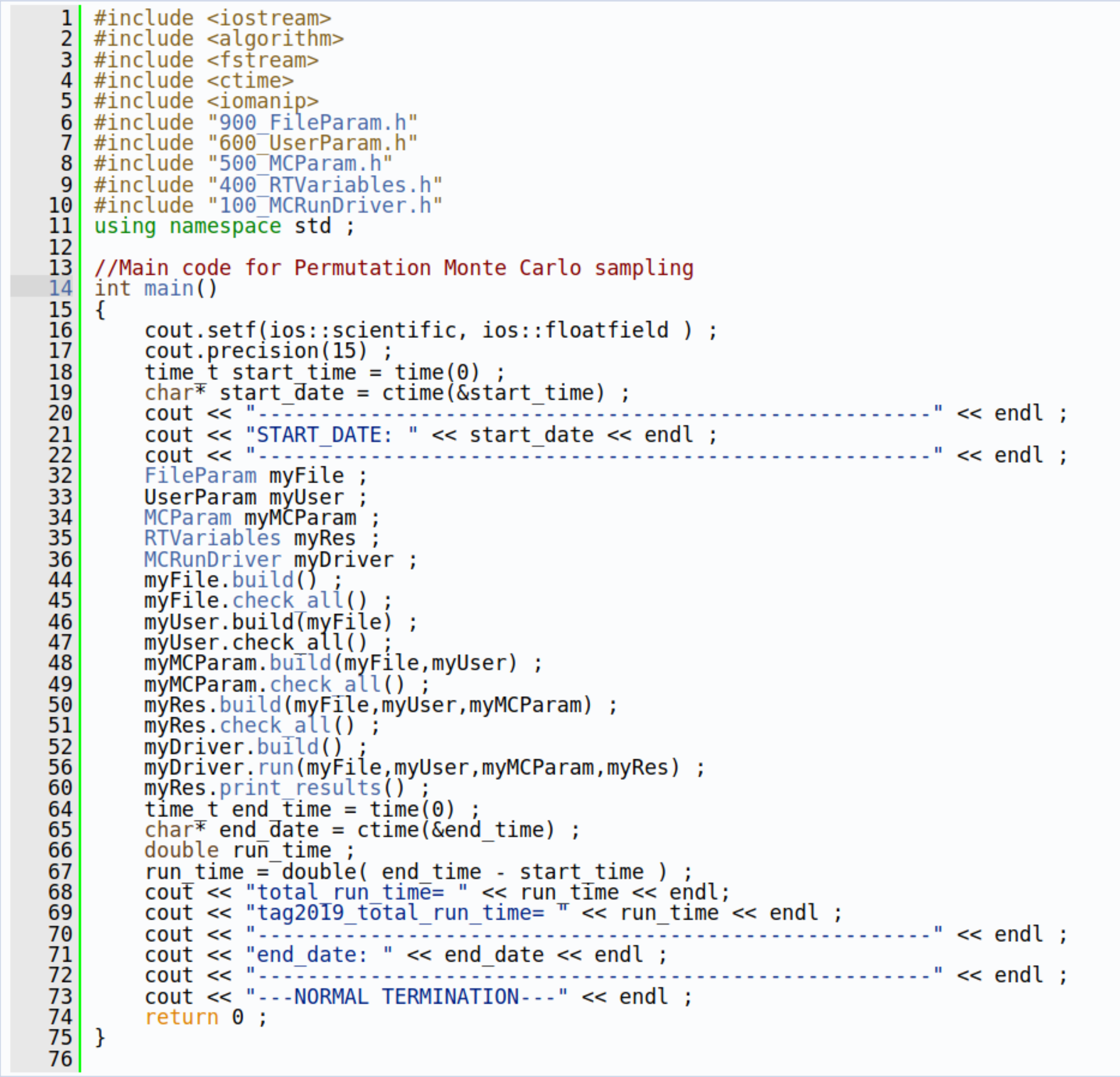
copied as it is passed to other objects and functions. The final reason for using C++ in development of permutation sampling Monte Carlo was for the portability to graphics processing units (GPU) using Compute Unified Device Architecture [CUDA] C. Which is a parallel computing platform in the C language created by NVIDIA as a tool for parallel programming on GPU's. The end goal of this project is to utilize GPU acceleration, allowing for massive parallization in a integral calculations resulting in fast sampling and compute time.

8.5 Conclusions

The coding project resulted in approximately 6,000 lines of code, the biggest coding project within the presented research, which is completely object-oriented for the groups future integral needs. Included in Figure 8.2 is the full source code for the main code for permutation sampling Monte Carlo. Which shows the techniques discussed in the previous sections within this chapter. It is important to deconstruct this source code to truly understand some of the complexity, first from Lines 16-22 is just preparation of how data will be printed and starting a timer to be used for calculation of run time. Next from Lines 32-36 there is declaration of all the classes that are needed for the code. The Lines 44-52 build each of these classes, then once building is complete a check is performed to ensure continuity within the code. [For example that number of dimensions is equal throughout each object] Next is Line 53 which is the run of the permutation sampling code. It isn't until this line that the calculation begins. Then there is the print results in Line 54, all the steps needed in run are abstracted, for once the run is complete results are ready to be printed. Finally are the lines 64-73 which calculate run time internally and proceed to normal termination of the code.

In addition I have included a graphical hierarchy of the class structure within the permutation sampling Monte Carlo code, Figure 8.3. This graphically depicts how classes are related within the code. The way information and data flows is designated by pointed arrows. As seen in the figure all the arrows flow downwards, this exemplifies composition over inheritance and encapsulation as no arrows are bi-directional.

Included in Appendix A is the manual for permutation sampling Monte Carlo.



```
#include <iostream>
#include <algorithm>
#include <fstream>
#include <ctime>
#include <iomanip>
#include "900_FileParam.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
#include "400_RTVariables.h"
#include "100_MCRunDriver.h"
using namespace std ;

//Main code for Permutation Monte Carlo sampling
int main()
{
    cout.setf(ios::scientific, ios::floatfield ) ;
    cout.precision(15) ;
    time_t start_time = time(0) ;
    char* start_date = ctime(&start_time) ;
    cout << "--------------------------------------------------------" << endl ;
    cout << "START_DATE: " << start_date << endl ;
    cout << "--------------------------------------------------------" << endl ;
    FileParam myFile ;
    UserParam myUser ;
    MCParam myMCParam ;
    RTVariables myRes ;
    MCRunDriver myDriver ;
    myFile.build() ;
    myFile.check_all() ;
    myUser.build(myFile) ;
    myUser.check_all() ;
    myMCParam.build(myFile,myUser) ;
    myMCParam.check_all() ;
    myRes.build(myFile,myUser,myMCParam) ;
    myRes.check_all() ;
    myDriver.build() ;
    myDriver.run(myFile,myUser,myMCParam,myRes) ;
    myRes.print_results() ;
    time_t end_time = time(0) ;
    char* end_date = ctime(&end_time) ;
    double run_time ;
    run_time = double( end_time - start_time ) ;
    cout << "total_run_time= " << run_time << endl;
    cout << "tag2019_total_run_time= " << run_time << endl ;
    cout << "--------------------------------------------------------" << endl ;
    cout << "end_date: " << end_date << endl ;
    cout << "--------------------------------------------------------" << endl ;
    cout << "---NORMAL TERMINATION---" << endl ;
    return 0 ;
}
```

Figure 8.2: Main source code for permutation sampling Monte Carlo

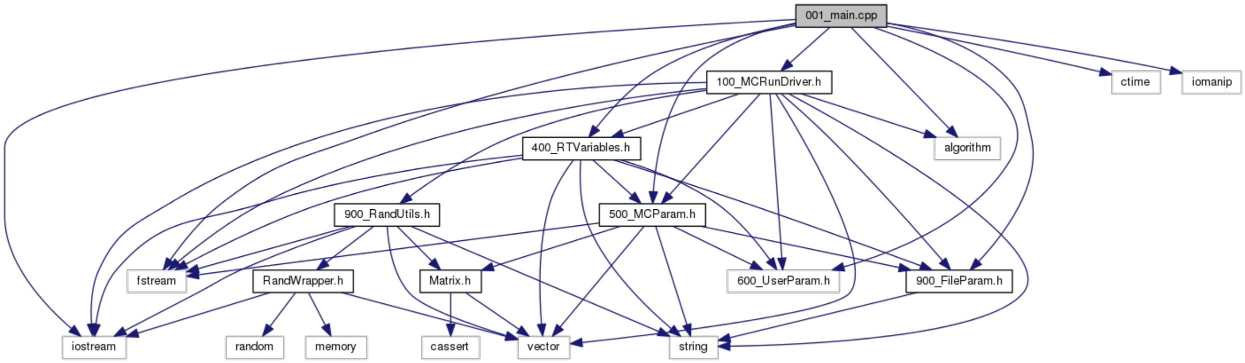


Figure 8.3: Graphical Hierarchy of the class structure from the main source code

Chapter 9

Collaborative Projects

9.1 Introduction

This chapter will introduce collaborative projects that I was apart of during my time at Syracuse University. I had the great opportunity to work on projects with the Dr. Pranav Soman group at the Syracuse Biomaterials Institute and the Dr. Weiwei Zhang group within the Department of Chemistry. I will provide brief introduction into the collaborative work and the results that I was able to contribute to the projects.

9.2 Investigation of Optical Properties of Silver Nanoclusters

9.2.1 Introduction

Recently a synthetic method was developed by Kunwar *et. al.*[146, 147] which demonstrated the creation of silver nanoclusters in a polymer film by direct-laser writing. Their method was able to create highly photostable Ag nanoclusters of various sizes entrapped in the polymer film. This synthetic procedure allows for size-controlled Ag nanocluster formation that is dependent on irradiation time. Since the Ag nanoclusters are trapped in a polymer matrix aggregation of the nanoclusters is prevented. Thereby allowing for the novel examination of the optical properties as a function of time and size, see Figure 9.1. They observe a sharp increase in fluorescent intensity to a maximum, followed by a gradual decay in intensity during Ag nanocluster formation. Currently, their experimental studies are limited in definitely providing information of the Ag nanocluster structure. Hence, they are unable to prove the transition of these Ag nanocluster to nanoparticles to bulk material.

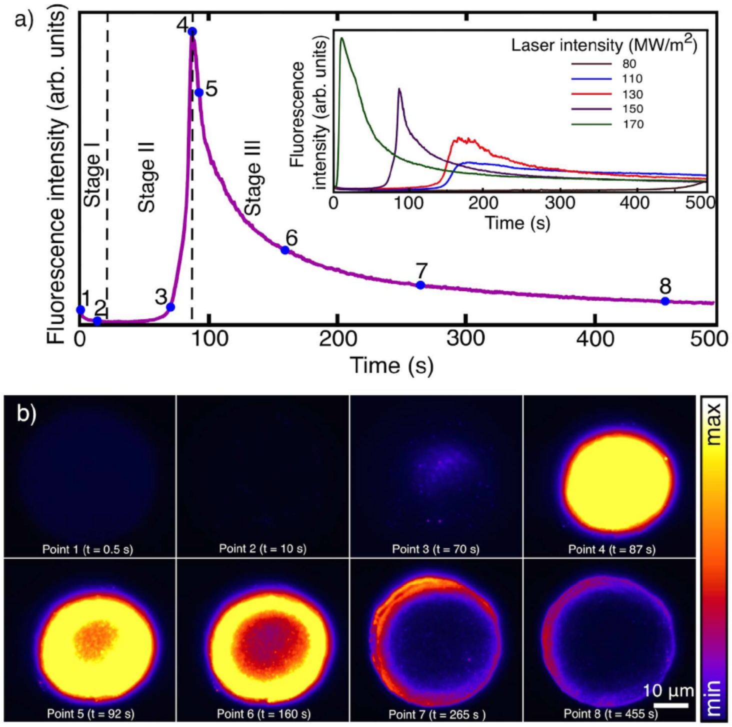


Figure 9.1: (a) Fluorescence intensity of Ag polymer matrix during irradiation. The inset demonstrates the effect of fluorescent intensity with various irradiation intensities. (b) Fluorescence microscopy images at time points 1-8 from (a). (Figure 1)[146]

9.2.2 Results

The intent of the presented work will be to provide collaborative theoretical support for their experimental claims. The theoretical investigation will primarily focus on understanding excited state properties like oscillator strength and excitation energy as a function of increasing number of atoms. This will hopefully gain insight into the experimental claims that the an increased florescent intensity can be observed within a series of silver clusters.

Since the theoretical investigation is trying to understand the fluorescent quenching from experiment, the Hartree-Fock calculations will need to account for Ag clusters ranging from <1 nm to 5 nm (16 ∼ 14000 atoms). This equates to a range of 752 to 658,000 total electrons, which is computationally expensive and impractical. The use of CRENBS basis and effective core potential (ECP) will assist with the theoretical choke by limiting the number of active electrons from 47 per atom to 11 per atom.[148] The use of ECP will allow for a significant reduction in the number of basis functions by approximating the core electrons as a psuedopotential as compared to a full electron basis like 3-21G, Table 9.1. Utilizing CRENBS basis and ECP also significantly reduces the number of basis functions to just the valence shell. Even though we are using a very minimal basis set, we are limited by the SCF procedure to (>2 nm) Ag clusters in size.

Table 9.1: **The number of basis functions for 2nm Ag cluster**

Basis set	N_b	N_{elec}
3-21G	17280	11280
CRENBS ECP	4320	2640

The presented work will focus on a range of Ag clusters under <2 nm in diameter, Table 9.2. The initial geometries for the clusters were generated by cutting dots of the desired size from metallic Ag structure.[149] The SCF procedure was performed using the Q-CHEM package[150] with CRENBS basis and ECP. Instead of performing geometry optimization calculations that would likely fail or be too computationally expensive, a systematic approach was taken to obtain the lowest energy structure.

Table 9.2: **Computational impact of Ag clusters.**

N_{atm}	N_b	N_{elec}
2	36	22
16	288	176
44	792	484
68	1224	748
80	1440	880
92	1656	1012
104	1872	1144
116	2088	1276
128	2304	1408

In order to keep the cost of the SCF procedure at a minimum it was decided that we would like to preserve as much symmetry of the cluster as possible. Since the geometry was cut from a pristine bulk structure it was expected that the average distance of the silver atoms should increased to relax the structure. Therefore it was decided that scaling the distance of the atoms from the origin would be the grounds for the geometry optimization. Where the inner shell is defined as the atoms that had a distance $\leq \frac{r}{2}$, where r is the radius of the cluster and the outer shell where distances $> \frac{r}{2}$. The inner shell will be scaled by α and the outer shell by β to allow the cluster to be dynamically scaled. The various permutations of α and β from the initial structure were then run to obtain their total SCF energy to find the lowest energy structure. This was then used to construct an energy surface for Ag_{80} in, Figure 9.2, where the blue dot was the lowest energy geometry from the systematic geometry search. This geometry search was performed for all the structures. This

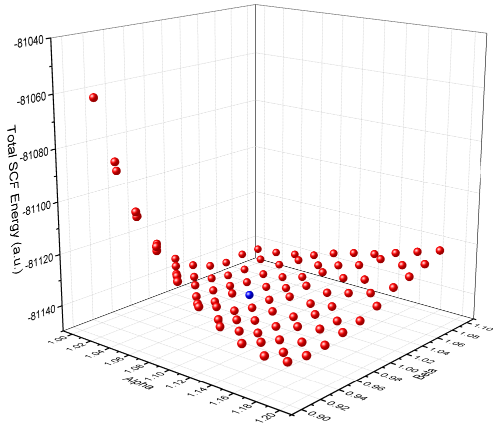


Figure 9.2: A plot of the total SCF energy for Ag_{80} at various geometries that were systematically searched for by a scalar radial approach.

radial scaling consisted of two components, which were separated by the inner shell and outer shell of the cluster.

A key excited state property that will aid in understanding the experimental observations is the oscillator strength. The oscillator strength is the probability of an excitation between an occupied and unoccupied orbital. If we can understand the oscillator strength as a function of increasing atom size, then the theoretical work can help understand the experimental observations. The oscillator strength was calculated for all single excitation permutations for the 10 highest energy occupied and 10 lowest energy unoccupied Hartree-Fock orbitals. These results were then histogrammed according the calculated oscillator strength to better interpret the results, Figure 9.3. The width of the bins are related to the energy of the excitation, therefore a narrow bin is the lowest excitation. These results show that as the number of silver atoms increase there is a point, Ag_{68}, for which there is a sharp increase in oscillator strength. Which is the exact observation seen from Kunwar *et. al.*[146, 147].

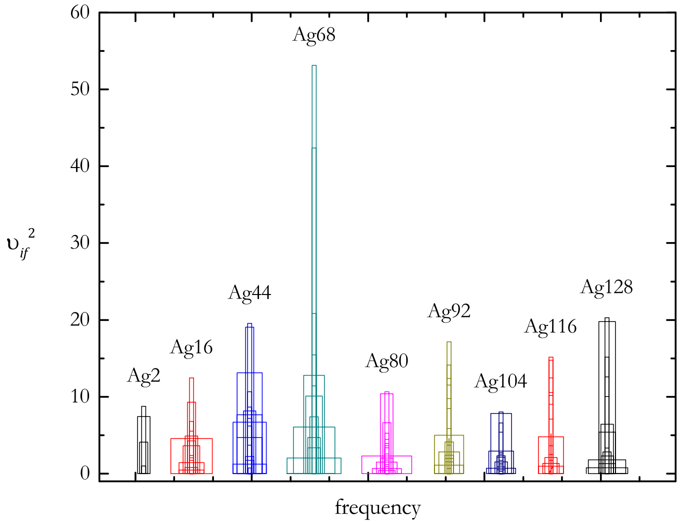


Figure 9.3: A plot which shows a histogram of oscillator strengths for the 10 highest occupied and 10 lowest unoccupied orbitals of silver nanoclusters.

9.2.3 Conclusion

The challenge within the presented Ag nanoclusters was in regards to the next interest in obtaining electron-correlated excitation energies. It was at this junction that development began on the previously reported work in Chapter 5 to assist within this collaborative project. The true advancement towards this project lies in the dressed molecular orbitals method presented in **??**. With these two methods developed the understanding of the excited states for the various Ag nanoclusters could continue.

9.3 Understanding Mn^{2+} Dopant Migration in CdS/ZnS Quantum Dots

9.3.1 Introduction

They were interested in gathering a computational understanding of dopant migration versus dopant ejection in incorporated dopants in core/shell quantum dots for a manuscript in preparation. This thermodynamic study expands on their previously reported work.[151] The goal of the computational work was to determine if Mn^{2+} dopant ion in a CdS/ZnS core/shell quantum dot was migrating or ejecting from the dot to provide further conclusions to the experimental work. The difference is subtle but is all dependent on the thermodynamics of the interaction of the dopant ion to the core/shell lattice. Dopant migration is an activated processes where energy is needed for the dopant to move over energy barriers. Dopant ejection is not an activated process and requires no energy for the dopant to traverse the quantum dot. Below I am providing the theoretical details and results for these studies that are being prepared for publication.

9.3.2 Theory

The thermodynamics of the dopant migration was investigated computationally by evaluating the relative effective radial Helmholtz free energy, $\Delta A_{\text{eff}}(r)$ experienced by the dopant ion. The $\Delta A_{\text{eff}}(r)$ provided key thermodynamic insight into effect of dopant ion at the core/shell interface and the experimentally observed temperature dependence of the migration. To calculate $\Delta A_{\text{eff}}(r)$, we define the radial configuration integral $Z(r,T)$ which is mathematically expressed as Equation 9.1

$$Z(r,T) = \int_{-\infty}^{+\infty} \partial \mathbf{r}' \exp\left[\frac{-U_{\text{dopant}-\text{lattice}}(\mathbf{r}')}{k_B T}\right] \partial(|\mathbf{r}'| - r) \tag{9.1}$$

where, T is temperature, k_B is Boltzmann constant. The relationship between the Helmholtz free energy and the configuration integral is well-known [152] and the free energy is proportional to the logarithm of the configuration integral

$$A(r,T) \propto -RT \ln\left[Z(r,T)\right] \tag{9.2}$$

Using $Z(r,T)$, we define the change in the radial Helmholtz free energy $\Delta A(r)$ as

$$\Delta A_{\text{eff}}(r) = A(r) - A(r_0) = -RT \ln\left[\frac{Z(r,T)}{Z(r_0,T)}\right] \tag{9.3}$$

where r_0 is some reference point. The free energy barrier associate with the migration of the dopant was used to calculate the rate constant of the transition using transistion state theory as shown below,

$$k(T) = \frac{k_B T}{h} e^{-\frac{\Delta A^\dagger}{k_B T}} \tag{9.4}$$

where $k(T)$ is the rate constant, $\Delta A^\dagger$ is the free enregy barrier along the transition path and h is the Planck's constant.

9.3.3 Computational Details

Computational investigation of the Mn^{2+} migration in the CdS/ZnS host lattice was performed at quantum mechanical level using density functional theory (DFT). The Q-CHEM electronic structure package [60] was used and the DFT calculation were performed using the B3LYP functional with LANL2DZ effective-core potential and basis set. The core/shell QD with the stoichiometric formula of $Cd_{288}S_{288}/Zn_{2488}S_{2488}$, having a core radius of 1.5 nm and a dot radius of 3.0 nm was constructed from the bulk crystal structures. Analysis of the quantum mechanics data revealed that the interaction of the Mn^{2+} with the host lattice can be predominantly described by a pairwise ion-dipole interaction potential as shown in Equation 9.5,

$$U_{\mathrm{dopant-lattice}} = \sum_i \frac{C}{|\mathbf{r}_{\mathrm{dopant}} - \mathbf{R}_i^{\mathrm{lattice}}|^2} \tag{9.5}$$

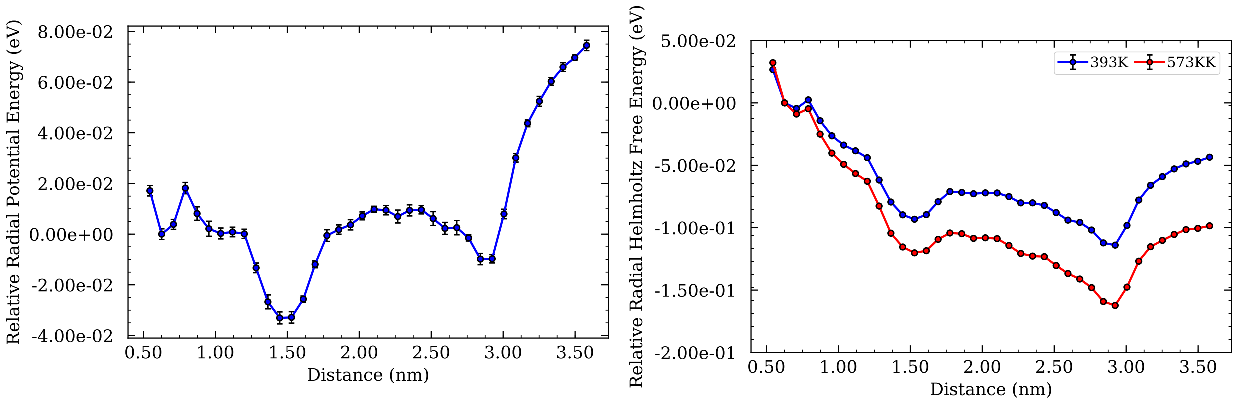
where $\mathbf{R}^{\mathrm{lattice}}$ is the center of the dipole, $\mathbf{r}$ is the position of the dopant, and C is an adjustable parameter optimized to fit the DFT data. The integrals needed for calculating $\Delta A_{\mathrm{eff}}(r)$ were evaluated numerically using Monte Carlo sampling. Specifically,

$$\left[\frac{Z(r,T)}{Z(r_0,T)}\right] \approx \frac{\sum_{i=1}^{N_{\mathrm{sample}}}\left[\frac{-U_{\mathrm{dopant-lattice}}(\mathbf{r}\hat{\eta}_i)}{k_B T}\right]}{\sum_{i=1}^{N_{\mathrm{sample}}}\left[\frac{-U_{\mathrm{dopant-lattice}}(\mathbf{r_0}\hat{\eta}_i)}{k_B T}\right]} \tag{9.6}$$

where, $\{\hat{\eta}_i\}$ is a set of unit vectors distributed randomly on a unit sphere. The vectors $\{r\hat{\eta}_i\}$ uniformly sampled points that are distance r from the center of the dot. For each value of the r, a million randomly generated points were used for performing Monte Carlo sampling, this procedure was repeated for 10 additional runs to provide data to obtain averages for each temperature. The for the Helmholtz free energy , was selected at the local minima in the core region of the QD. The calculated values of the relative Helmholtz free energy is presented in Figure S16 and S17 as a function of the dopant distance r from the center of the QD.

9.3.4 Results and Discussion

The calculated radial free energy for the movement of Mn dopants in CdS/ZnS QDs exhibits a local minimum at the core/shell interface of -0.111 eV at 573 K (r = 1.5 nm, Figure 9.4, Figure 9.5) Therefore, the dopant migration from the interior of the core towards the interface is due to decrease in the overall free energy. However, this decrease in the free energy is not monotonic and the minima at the core/shell interface is preceded by a free energy barrier within the CdS core, with a barrier height of 4.25E-3 eV at r = 0.8 nm at 573 K. The presence of this barrier indicates that dopant migration requires thermal activation to overcome the barrier within the core (Figure 9.5). The plot of the relative radial free energy also shows the presence of a high energy barrier with



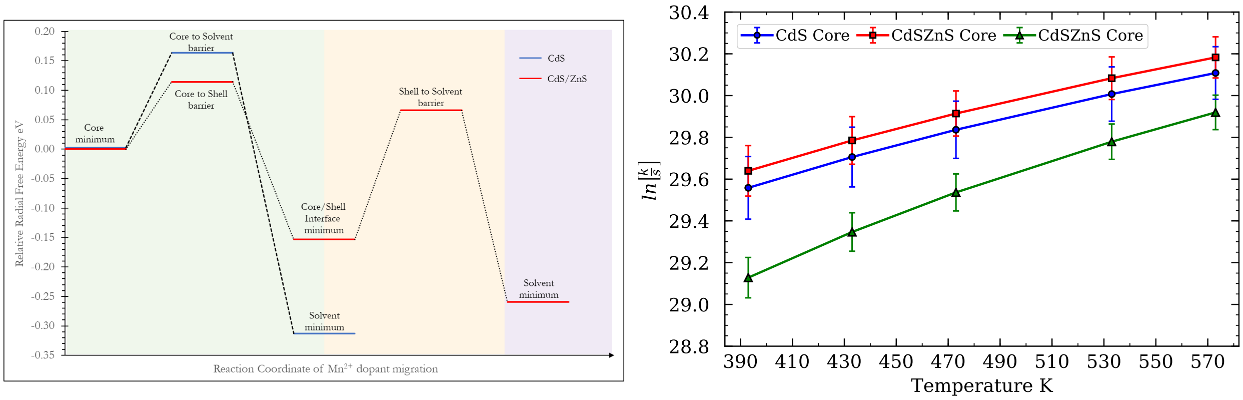
(a)

(b)

Figure 9.4: Density function theory calculation of Mn:CdS core QDs dispersed in hexane. (a) The average relative potential the Mn(II) ion experiences at various radial positions in the CdS core and solution and (b) The average relative free energy of the Mn(II) ion, CdS core, and solvent system as the Mn(II) ion moves radial through the system at the lowest (393 K) and highest temperature (573 K) from the experimental study.

the height of 1.59E-2 eV in the ZnS shell region (r = 1.5 - 3 nm, 573 K) for the ejection of the dopants from the core/shell minimum to solvent minimum. The barrier for dopant ejection was broader and higher than the migration to the core/shell interface, by 1.16E-2 eV, which is consistent with fact that higher temperature required for dopant ejection than dopant migration observed in our experiments. The rate constants of reactions for CdS/ZnS show that the rate limiting step is the interface to solvent reaction ($k_{Inteface}$) step, Figure 9.5. The ratio of the rate of reactions for the CdS/ZnS QD show that the rate for Mn dopant migration from k_{Core}, is 1.7 times higher than migration from interface to solvent ($k_{Interface}$) at lower temperatures, demonstrating that dopant migration to interface is faster than Mn dopant ejection to solvent.

The dopant migration was also computationally evaluated for core-only CdS QDs in solvent without ZnS shell and core/shell interface, Figure 9.6. The plot of the radial free energy (Figure S17b) shows a free energy barrier in the CdS QD with the highest free energy barrier of 2.69E-3 eV at r = 0.8 nm. This demonstrates that dopant ejection in the simple doped CdS QDs still require high thermal activation. The difference between CdS and CdS/ZnS is the presence of the core/shell minimum and shell to solvent barrier in the CdS/ZnS QDs (Figure 6a). The rate constant of for the migration of the dopant (k_{Core}) in the CdS QDs from the core region was found to be slower

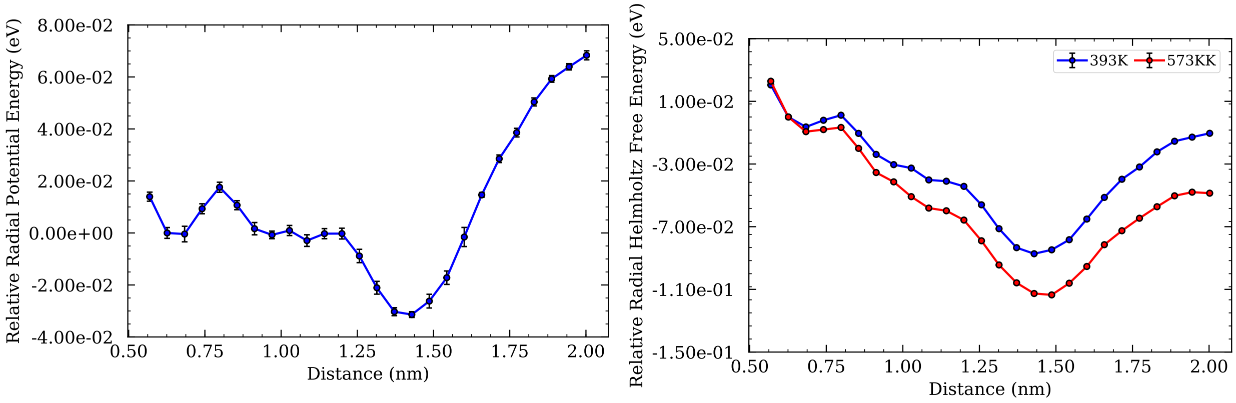


(a)

(b)

Figure 9.5: (a) The free energy profile along the reaction coordinate for Mn(II) ion migration in the CdS/ZnS and CdS host lattice at 573 K. The green, yellow, and purple regions indicate the core, shell, and solvent regions, respectively for the CdS/ZnS QDs. Two distinct barriers are observed in the CdS/ZnS QDs, first from the core to core/shell interface and second from the core/shell interface to solvent. In contrast, only one barrier from core to solvent, is observed for CdS QDs. (b) Rate constants of reactions for Mn(II) ion in both CdS/ZnS and CdS host lattice systems as a function of temperature.

than the corresponding (k_{Core}) in CdS/ZnS QDs (Figure 9.5). This observation indicates that the presence of the core/shell interface has a positive impact of increasing the rate constant of the migration of the dopant from the core region.



(a)

(b)

Figure 9.6: Density function theory calculation of Mn:CdS/ZnS core/shell QDs dispersed in hexane. (a) The average relative potential the Mn(II) ion experiences at various radial positions in the quantum dot and solution and (b) The average relative free energy of the Mn(II) ion, quantum dot, and solvent system as the Mn(II) ion moves radial through the system at the lowest and highest temperature from the experimental study.

9.3.5 Conclusion

In summary, the presented work shows that dopant migration is the primary reaction within these quantum dot systems. That dopant ejection is not present since thermal energy barriers exist within the Helmholtz free energy of the system. The presented theoretical work on Mn^{2+} dopant migration was added as part of the larger experimental work from the Zheng group that was recently accepted for publication. The theoretical method that was introduced has its own merit for further theoretical investigations into other dopant migration models. It provides a simplistic Monte Carlo method that can be used for large many-atom systems for understanding thermodynamic properties.

Chapter 10

Conclusions and Future Work

In this work, 5 methods were presented. In Chapter 5, the frequency-dependent geminal-screened electron-hole interaction kernel (FD-GSIK) method was presented. Within this work the FD-GSIK method was able to demonstrate new development towards efficient implementation of electron-hole interaction kernel. The method provided a frequency-dependent kernel which greatly reduced the computational effort in excited state calculations, demonstrated by large many-electron semiconductor quantum dots. In Chapter 6, the dressed molecular orbital basis was introduced. This method was an advancement of the FD-GSIK method by providing a basis to be used to understand the multi-reference nature of highly degenerate systems. The dressed molecular orbital basis provided a compression of the configuration space to reduce computational effort in the excited state calculations. This method was demonstrated with silver nanowires, which demonstrate highly degenerate orbitals, particularly near the HOMO-LUMO gap. With this approach I was able to demonstrate a blue shift in excitation energy for long silver nanowires. In Chapter 7, the derivation of moment generated molecular orbital basis was presented. This approach provides a method for obtaining electron-correlation through moments of a reference particle/hole state without the use of the projector operator $\Lambda(\omega, r_{eh})$. In Chapter 8, the permutation sampling Monte Carlo integration method was presented. The permutation sampling Monte Carlo method provides a sampling technique that decouples dimensionality within integrals. The presented work focuses on the design and implementation of the algorithm towards developing a integration tool that is computationally efficient. This was demonstrated through a graphical hierarchy that depicts the use of object-oriented programming paradigm. The final method was presented within Chapter 9 which contained the theoretical investigations within collaborative efforts. The method presented was a Monte Carlo sampling technique that allowed for modeling the thermodynamic interactions between Mn^{2+} and CdS/ZnS quantum dot. This approach allowed for determining that free energy barriers exist within the system which demonstrated that dopant migration was the dominate process. Rather than the expected dopant ejection which requires no energy.

Efficient implementations, particularly in the excited state calculations are of key focus within the presented work. Understanding the challenges in modern excited state theories are essential to the motivation of all the presented work. Chapter 4 provided a brief introduction to the theory and implementations of popular methods. Second quantization in Chapter 3 provides an understanding of the notation used throughout the thesis. Finally Chapter 2 provided background information to the main challenge within quantum chemistry obtaining a ground state wave function and electron correlation energy.

The future of the presented methods include utilizing FD-GSIK to investigate chemically interesting systems, that are computationally prohibited by existing excited state methods. Much of the work presented with FD-GSIK was towards validation of the theory and implementation. An example of a future investigation with FD-GSIK is to use it towards understanding the effect of distance between asphaltene dimers. The FD-GSIK can also advance with trying to understand thermal broadening due to Boltzmann distribution of structures. Since FD-GSIK is a computationally cheap excited state method, investigations into understanding an ensemble of structure can provide insight into experimental comparisons. In addition, dressed molecular orbital basis can be used to understand the energy transfer between two quantum dots. The dressed molecular orbital basis provides an approach to deal with high degeneracy that can be present in separated quantum dots interacting with each other. With the implementation of dressed molecular orbitals and FD-GSIK investigation into silver nanoclusters could continue. These two presented methods provide essential theory that was not developed during the collaborative work but could prove useful for understanding the experimental observations in the optical properties. The future of moment generate basis is to implement the theory to validate the approach towards obtaining electron-correlated excitation energies. The approached used for understanding dopant migration in quantum dots can also be used for other systems. The method was not derived specifically for the collaborative work, and has its own merit towards understanding thermodynamic properties in molecular systems. The future of permutation sampling Monte Carlo method is to integrate to a GPU platform through CUDA. The development was focused on providing GPU acceleration towards providing a fast and efficient integration package. Development will also include providing more functionality to the sampling techniques used within permutation sampling Monte Carlo, to provide a more accurate integration but maintain computational efficiency.

Appendix A

Monte Carlo Variance Reduction

Monte Carlo integration was an key component of many methods presented. It is relevant to take note of several key variance reduction techniques used throughout the works that improve on the quality of the Monte Carlo method. Within Monte Carlo integration, error variance has the form σ^2/n, where σ^2 is the variance within a sample and n is the number of samples. The final error variance and thereby the quality of the method can be improved in two ways, first by a larger value of n which comes at the cost of computational time, or by reducing σ. The variance reduction techniques that will be presented are antithetic sampling, stratification, control variate and importance sampling.

A.0.1 Antithetic Sampling

Suppose that the goal of the Monte Carlo integration is to evaluate the following integral,

$$\int_{-\pi}^{\pi} sin(x)dx = 0 \tag{A.1}$$

The exact solution of 0 is due to the symmetry of the function at $x = 0$, which means that, $f(x_i) = -f(-x_i)$. If we were to evaluate this integral using Monte Carlo, cancellation of sample x_i of would not occur until $-x_i$ is sampled, which isn't guaranteed unless $n \to \infty$. Antithetic sampling is based on improving cancellation of error within the Monte Carlo method by increasing the number of sampling points. This done by enforcing that for every sample, the equally probably opposite sample is also evaluated. So from the example, for every sample of x_i, $-x_i$ will also be evaluated, thereby ensuring cancellation within the integral evaluation since $f(x_1)+f(x_1)=-f(-x_1)=0$. In antithetic sampling the overall number of sampling points is increased exponentially based on the dimensionality of the sampling point, $n*(2)^{N_{dim}}$. Therefore even though antithetic sampling is useful for variance reduction, the dimensionality of the Monte Carlo integration could prove burdensome and computationally costly for evaluating all antithetic points.

A.0.2 Stratification

Stratified sampling is the idea of splitting up the sampling region into separate regions to create a fair distribution. Imagine that a Gaussian function, $g(x)$ is going to be evaluated through Monte Carlo sampling of random points, Figure A.1. Notice that the region where $g(x)$ would provide any significant contribution to the calculation is from $-2 \leq x \leq 2$, since $g(x) \approx 0$ when $|x| > 2$. Stratification takes advantage of this by creating strata, or regions in which sampling would occur. In the case of Figure A.1, strata could be constructed that are from the regions of $x < -2$, $-2 \leq x \leq 2$, and $x > 2$. From these strata the Monte Carlo sampling will be restricted to these regions, whether the sampling will be uniform or weighted. Since the function is known, we could therefore sample the 2nd strata significantly more than the other two strata. This way our sampling is improving σ of the overall calculation by directly improving the evaluate within each strata since the σ of strata 1 and 2 would not significantly change since $g(x) \approx 0$. In the long run stratification

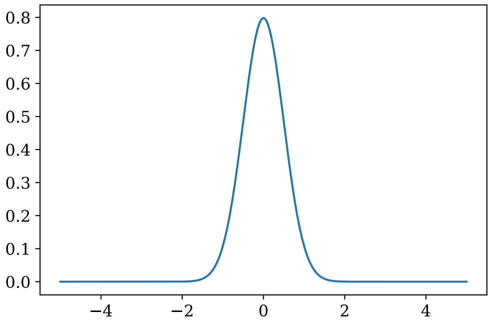


Figure A.1: A graph of a normalized Gaussian function.

will also improve the cost of the Monte Carlo sampling by ensuring that efficient sampling is taking place in strata that are significant.

A.0.3 Control Variate

The next variance reduction technique of interest is control variate. The purpose of a control variate is to improve the error in our desired calculation by understanding the error obtained from a known quantity. Say we want to know $\langle f(x)\rangle$ of function, $f(x)$ and we have a second function, $h(x)$ for which we know $\langle h(x)\rangle$, exactly. As we sample $f(x)$ we are going to subtract $h(x)$ and add $\langle h(x)\rangle$ to the end of the Monte Carlo integration,

$$\langle f(x)\rangle = \frac{1}{n}\sum_{i=1}^{n}[f(x_i)-h(x_i)]+\langle h(x)\rangle \tag{A.2}$$

What this does is that the error obtained at $h(x_i)$ is removed from $f(x_i)$, thereby leading to cancellation of errors from the whole integration because in the end of the evaluation we should have,

$$\langle f(x)\rangle = \frac{1}{n}\sum_{i=1}^{n}[f(x_i)-h(x_i)]+\langle h(x)\rangle \tag{A.3}$$

$$\langle f(x)\rangle = \langle f(x)\rangle - \langle h(x)\rangle_{MC} + \langle h(x)\rangle_{exact} \quad \text{if} \quad \langle h(x)\rangle_{MC} = \langle h(x)\rangle_{exact} \tag{A.4}$$

$$\langle f(x)\rangle = \langle f(x)\rangle \tag{A.5}$$

Note the distinction of $\langle h(x)\rangle_{MC} = \langle h(x)\rangle_{exact}$, for evaluation of $h(x)$ through Monte Carlo sampling and exactly. The use of control variate allows for cancellation in the error obtained in $\langle h(x)\rangle_{MC}$ from the Monte Carlo procedure by correcting with the exact result, $\langle h(x)\rangle_{exact}$. This in turn improves the error obtained for our desired calculation of $\langle f(x)\rangle$ by simply evaluating another function during each Monte Carlo step. The use of control variate also allows for easily evaluation of the quality of the Monte Carlo procedure as $\langle h(x)\rangle_{MC}$ can quantify if sampling was sufficient.

A.0.4 Importance Sampling

The final variance reduction technique that is of note is importance sampling. Importance sampling is simply taking a distribution, ρ with a known expectation value and creating a biased use of ρ that provides a different expectation value. Imagine that we have a ”fair” 6-sided die, which is a die that has a fair probability of $1/6$ for rolling any value and therefore an expectation value of 3.5. Now think of a probability, ρ' for which is a distribution of a ”biased” die, that has an expectation value lower than 3.5. This distribution is created when the rolls of the die are unfair and more heavily weighted toward lower values on the die, resulting in a lower expectation value. The key to importance sampling is creating an expectation value of the ”biased” die using ρ from the ”fair” die, by weighting the importance of a die roll or by sampling the ”important” values more frequently. If the important values are sampled more frequently then the variance in the expectation value will be lower.

Appendix A

Manual for Permutation Sampling Monte Carlo

Permutation_Sampling_Code

Generated by Doxygen 1.8.6

Tue Jun 23 2020 13:55:23

Contents

1 Class Index **1**
- 1.1 Class List 1

2 File Index **3**
- 2.1 File List 3

3 Class Documentation **5**
- 3.1 CoordParam Class Reference 5
  - 3.1.1 Detailed Description 7
  - 3.1.2 Constructor & Destructor Documentation 7
    - 3.1.2.1 CoordParam 7
    - 3.1.2.2 ∼CoordParam 7
    - 3.1.2.3 CoordParam 7
  - 3.1.3 Member Function Documentation 7
    - 3.1.3.1 build 7
    - 3.1.3.2 calc_volume 7
    - 3.1.3.3 check_all 7
    - 3.1.3.4 get_ndim 8
    - 3.1.3.5 get_vol 8
    - 3.1.3.6 get_xmax 8
    - 3.1.3.7 get_xmin 8
    - 3.1.3.8 operator= 8
    - 3.1.3.9 print_info 8
    - 3.1.3.10 set_xmax 8
    - 3.1.3.11 set_xmin 8
    - 3.1.3.12 transform_t_to_x 8
  - 3.1.4 Member Data Documentation 8
    - 3.1.4.1 length_ 8
    - 3.1.4.2 ndim_ 8
    - 3.1.4.3 vol_ 8
    - 3.1.4.4 xmax_ 9
    - 3.1.4.5 xmin_ 9
- 3.2 FileParam Class Reference 9
  - 3.2.1 Detailed Description 11
  - 3.2.2 Constructor & Destructor Documentation 11
    - 3.2.2.1 FileParam 11
    - 3.2.2.2 ∼FileParam 11
    - 3.2.2.3 FileParam 11
  - 3.2.3 Member Function Documentation 11
    - 3.2.3.1 build 11
    - 3.2.3.2 build 11
    - 3.2.3.3 check_all 12
    - 3.2.3.4 operator= 12
  - 3.2.4 Member Data Documentation 12
    - 3.2.4.1 binfile_ 12
    - 3.2.4.2 mcrun_file_ 12

    - 3.2.4.3 moefile_ 12
    - 3.2.4.4 moeinfo_ 12
    - 3.2.4.5 mohfile_ 12
    - 3.2.4.6 mohinfo_ 12
    - 3.2.4.7 usr_func_file_ 12
    - 3.2.4.8 usr_rangefile_ 13
- 3.3 Matrix< T > Class Template Reference 13
  - 3.3.1 Detailed Description 15
  - 3.3.2 Constructor & Destructor Documentation 15
    - 3.3.2.1 Matrix 15
    - 3.3.2.2 Matrix 15
    - 3.3.2.3 ∼Matrix 15
    - 3.3.2.4 Matrix 15
  - 3.3.3 Member Function Documentation 15
    - 3.3.3.1 get 15
    - 3.3.3.2 get_raw_ptr 15
    - 3.3.3.3 init 15
    - 3.3.3.4 ncol 16
    - 3.3.3.5 nrow 16
    - 3.3.3.6 operator= 16
    - 3.3.3.7 set 16
  - 3.3.4 Member Data Documentation 16
    - 3.3.4.1 A_ 16
    - 3.3.4.2 ncol_ 17
    - 3.3.4.3 nrow_ 17
- 3.4 MCParam Class Reference 17
  - 3.4.1 Detailed Description 19
  - 3.4.2 Constructor & Destructor Documentation 19
    - 3.4.2.1 MCParam 19
    - 3.4.2.2 ∼MCParam 19
    - 3.4.2.3 MCParam 20
  - 3.4.3 Member Function Documentation 20
    - 3.4.3.1 bin_init 20
    - 3.4.3.2 build 20
    - 3.4.3.3 calc_bin_index 20
    - 3.4.3.4 check_all 20
    - 3.4.3.5 check_bin_wt_read 21
    - 3.4.3.6 get_bin_start 21
    - 3.4.3.7 get_bin_wt 21
    - 3.4.3.8 get_ibin_start 21
    - 3.4.3.9 get_mc_base_pts 22
    - 3.4.3.10 get_mc_max_xnpt 22
    - 3.4.3.11 get_mc_nloop_randmat_update 22
    - 3.4.3.12 get_mc_nloop_wt_update 23
    - 3.4.3.13 get_mc_randmat_npt 23
    - 3.4.3.14 get_nbin 23
    - 3.4.3.15 get_ndim 24
    - 3.4.3.16 get_nshuffle 25
    - 3.4.3.17 get_tgt_npt 25
    - 3.4.3.18 mc_init 25
    - 3.4.3.19 normalize_binwts 25
    - 3.4.3.20 operator= 25
    - 3.4.3.21 read_bin_wt 25
    - 3.4.3.22 read_mc_info 25
  - 3.4.4 Member Data Documentation 26
    - 3.4.4.1 bin_start_ 26
    - 3.4.4.2 bin_wt_ 26
    - 3.4.4.3 mc_base_npt_ 26

3.4.4.4 mc_max_xnpt_ . . . 26
3.4.4.5 mc_nloop_randmat_update_ . . . 26
3.4.4.6 mc_nloop_wt_update_ . . . 26
3.4.4.7 mc_randmat_npt_ . . . 26
3.4.4.8 mc_wtdist_npt_ . . . 26
3.4.4.9 nbin_ . . . 26
3.4.4.10 ndim_ . . . 26
3.4.4.11 num_shuffle_ . . . 26
3.4.4.12 vol_ . . . 27
3.5 MCRunDriver Class Reference . . . 27
3.5.1 Detailed Description . . . 28
3.5.2 Constructor & Destructor Documentation . . . 28
3.5.2.1 MCRunDriver . . . 28
3.5.2.2 ∼MCRunDriver . . . 28
3.5.2.3 MCRunDriver . . . 28
3.5.3 Member Function Documentation . . . 28
3.5.3.1 build . . . 28
3.5.3.2 calc_npt_per_seg . . . 28
3.5.3.3 calc_npt_using_wt_1d . . . 29
3.5.3.4 calc_ref_density_using_binidx . . . 29
3.5.3.5 check_all . . . 30
3.5.3.6 check_for_underflow . . . 30
3.5.3.7 mc_engine . . . 30
3.5.3.8 mc_shuffle . . . 31
3.5.3.9 mc_with_fixed_dist . . . 32
3.5.3.10 operator= . . . 33
3.5.3.11 run . . . 33
3.6 MOGrid Class Reference . . . 34
3.6.1 Detailed Description . . . 36
3.6.2 Constructor & Destructor Documentation . . . 36
3.6.2.1 MOGrid . . . 36
3.6.2.2 ∼MOGrid . . . 37
3.6.2.3 MOGrid . . . 37
3.6.3 Member Function Documentation . . . 37
3.6.3.1 build . . . 37
3.6.3.2 calc_volume . . . 37
3.6.3.3 check_all . . . 37
3.6.3.4 check_read_grid . . . 37
3.6.3.5 get_grid_index_1d . . . 37
3.6.3.6 get_grid_index_3d . . . 37
3.6.3.7 get_ieng . . . 37
3.6.3.8 get_imoval_at_igrid_imo . . . 38
3.6.3.9 get_imoval_at_rvec . . . 38
3.6.3.10 get_ndim . . . 38
3.6.3.11 get_ngrid . . . 38
3.6.3.12 get_nmo . . . 38
3.6.3.13 get_npt_1d_at_idim . . . 38
3.6.3.14 get_rvec_from_xyzgrid . . . 38
3.6.3.15 get_volume . . . 38
3.6.3.16 get_xmax_at_idim . . . 38
3.6.3.17 get_xmin_at_idim . . . 38
3.6.3.18 init_grid_vectors . . . 38
3.6.3.19 init_info_vectors . . . 38
3.6.3.20 operator= . . . 39
3.6.3.21 read_gridfile . . . 39
3.6.3.22 read_gridinfo . . . 39
3.6.4 Member Data Documentation . . . 39
3.6.4.1 delta_1d_ . . . 39

3.6.4.2 eng_ 39
3.6.4.3 grid_x_ 39
3.6.4.4 grid_y_ 39
3.6.4.5 grid_z_ 39
3.6.4.6 length_ 39
3.6.4.7 mo_ 39
3.6.4.8 ndim_ 39
3.6.4.9 ngrid_ 39
3.6.4.10 nmo_ 40
3.6.4.11 npt_1d_ 40
3.6.4.12 npt_total_ 40
3.6.4.13 vol_ 40
3.6.4.14 xmax_ 40
3.6.4.15 xmin_ 40
3.7 RandUtils Class Reference 40
3.7.1 Detailed Description 42
3.7.2 Constructor & Destructor Documentation 42
3.7.2.1 RandUtils 42
3.7.2.2 ∼RandUtils 42
3.7.2.3 RandUtils 42
3.7.3 Member Function Documentation 42
3.7.3.1 build 42
3.7.3.2 calc_randmat 43
3.7.3.3 check_all 43
3.7.3.4 gen_rnd_seed 43
3.7.3.5 operator= 43
3.7.3.6 rand_num 43
3.7.3.7 rand_num 44
3.7.3.8 rand_num 44
3.7.3.9 shuffle_vec 44
3.7.4 Member Data Documentation 44
3.7.4.1 iseed_ 44
3.7.4.2 myRW_ 44
3.8 RandWrapper Class Reference 44
3.8.1 Detailed Description 46
3.8.2 Constructor & Destructor Documentation 46
3.8.2.1 RandWrapper 46
3.8.2.2 ∼RandWrapper 46
3.8.2.3 RandWrapper 46
3.8.3 Member Function Documentation 46
3.8.3.1 build 46
3.8.3.2 build 46
3.8.3.3 build 46
3.8.3.4 build 46
3.8.3.5 get_seed 47
3.8.3.6 operator= 47
3.8.3.7 rand_num_normal 47
3.8.3.8 rand_num_uniform 47
3.8.3.9 rand_num_uniform 47
3.8.3.10 rand_vec_antithetic_normal 47
3.8.3.11 rand_vec_antithetic_uniform 47
3.8.3.12 rand_vec_normal 47
3.8.3.13 rand_vec_uniform 47
3.8.3.14 shuffle_vec 47
3.8.3.15 shuffle_vec 47
3.8.4 Member Data Documentation 47
3.8.4.1 generator_ 47
3.8.4.2 iseed_ 48

3.8.4.3 mu_ . . . 48
3.8.4.4 sigma_ . . . 48
3.9 RTVariables Class Reference . . . 48
3.9.1 Detailed Description . . . 51
3.9.2 Constructor & Destructor Documentation . . . 51
3.9.2.1 RTVariables . . . 51
3.9.2.2 ∼RTVariables . . . 51
3.9.2.3 RTVariables . . . 51
3.9.3 Member Function Documentation . . . 51
3.9.3.1 build . . . 51
3.9.3.2 build . . . 51
3.9.3.3 calc_idim_xsum_tgt_npt . . . 52
3.9.3.4 calc_wt_for_rho . . . 52
3.9.3.5 check_all . . . 52
3.9.3.6 check_seg_tgt_npt . . . 52
3.9.3.7 check_seg_tgt_npt . . . 53
3.9.3.8 fill_seg_wt . . . 53
3.9.3.9 get_mc_xnpt . . . 53
3.9.3.10 get_nbin . . . 53
3.9.3.11 get_npt_per_dim_bin . . . 54
3.9.3.12 get_rho_wt . . . 54
3.9.3.13 get_xsum_npt_per_idim . . . 54
3.9.3.14 mc_calc_integral_and_error . . . 55
3.9.3.15 mc_with_fixed_dist . . . 55
3.9.3.16 operator= . . . 55
3.9.3.17 print_results . . . 55
3.9.3.18 print_seg_tgt_npt . . . 56
3.9.3.19 set_user_defined_seg_wt . . . 56
3.9.3.20 update_avg . . . 56
3.9.3.21 update_RTV_avg . . . 56
3.9.4 Member Data Documentation . . . 57
3.9.4.1 avg_f0_ . . . 57
3.9.4.2 avg_f0sq_ . . . 57
3.9.4.3 avg_ft_ . . . 57
3.9.4.4 avg_ftsq_ . . . 57
3.9.4.5 integral_error_f0_ . . . 57
3.9.4.6 integral_error_ft_ . . . 57
3.9.4.7 integral_value_f0_ . . . 57
3.9.4.8 integral_value_ft_ . . . 57
3.9.4.9 mc_xnpt_ . . . 57
3.9.4.10 nbin_ . . . 58
3.9.4.11 ndim_ . . . 58
3.9.4.12 rho_wt_ . . . 58
3.9.4.13 seg_avg_ft_ . . . 58
3.9.4.14 seg_avg_ftsq_ . . . 58
3.9.4.15 seg_est_wt_ . . . 58
3.9.4.16 seg_idim_xsum_tgt_npt_ . . . 58
3.9.4.17 seg_sigma_ . . . 58
3.9.4.18 seg_tgt_npt_ . . . 58
3.9.4.19 seg_wt_ . . . 58
3.9.4.20 seg_xnpt_ . . . 58
3.9.4.21 sigma_f0_ . . . 58
3.9.4.22 sigma_ft_ . . . 59
3.9.4.23 tot_sampl_npt_ . . . 59
3.9.4.24 variance_f0_ . . . 59
3.9.4.25 variance_ft_ . . . 59
3.9.4.26 vol_ . . . 59

4 File Documentation 61

4.1 001_main.cpp File Reference . . . 61
  4.1.1 Function Documentation . . . 61
    4.1.1.1 main . . . 61
4.2 100_MCRunDriver.cpp File Reference . . . 62
4.3 100_MCRunDriver.h File Reference . . . 63
4.4 400_RTVariables.cpp File Reference . . . 64
4.5 400_RTVariables.h File Reference . . . 64
4.6 500_MCParam.cpp File Reference . . . 65
4.7 500_MCParam.h File Reference . . . 66
4.8 800_CoordParam.cpp File Reference . . . 67
4.9 800_CoordParam.h File Reference . . . 67
4.10 800_MOGrid.cpp File Reference . . . 68
  4.10.1 Function Documentation . . . 69
    4.10.1.1 col_major_idx_3d . . . 69
    4.10.1.2 row_major_idx_3d . . . 69
4.11 800_MOGrid.h File Reference . . . 69
4.12 900_FileParam.cpp File Reference . . . 70
4.13 900_FileParam.h File Reference . . . 70
4.14 900_RandUtils.cpp File Reference . . . 71
  4.14.1 Function Documentation . . . 72
    4.14.1.1 col_major_idx_2d . . . 72
4.15 900_RandUtils.h File Reference . . . 72
4.16 999_col_major_idx.cpp File Reference . . . 73
  4.16.1 Function Documentation . . . 74
    4.16.1.1 col_major_idx_2d . . . 74
    4.16.1.2 col_major_idx_3d . . . 74
    4.16.1.3 row_major_idx_3d . . . 74
4.17 Matrix.h File Reference . . . 74
  4.17.1 Function Documentation . . . 75
    4.17.1.1 col_major_idx_2d . . . 75
4.18 RandWrapper.cpp File Reference . . . 76
4.19 RandWrapper.h File Reference . . . 76

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CoordParam 5
FileParam 9
Matrix< T > 13
MCParam 17
MCRunDriver 27
MOGrid 34
RandUtils 40
RandWrapper 44
RTVariables 48

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

001_main.cpp 61
100_MCRunDriver.cpp 62
100_MCRunDriver.h 63
400_RTVariables.cpp 64
400_RTVariables.h 64
500_MCParam.cpp 65
500_MCParam.h 66
800_CoordParam.cpp 67
800_CoordParam.h 67
800_MOGrid.cpp 68
800_MOGrid.h 69
900_FileParam.cpp 70
900_FileParam.h 70
900_RandUtils.cpp 71
900_RandUtils.h 72
999_col_major_idx.cpp 73
Matrix.h 74
RandWrapper.cpp 76
RandWrapper.h 76

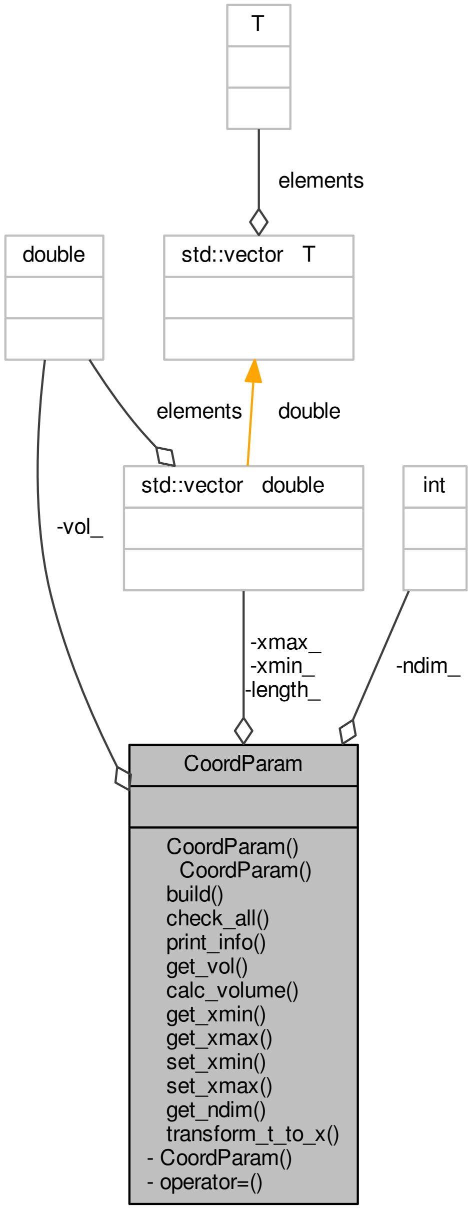
Chapter 3

Class Documentation

3.1 CoordParam Class Reference

`#include <800_CoordParam.h>`

Collaboration diagram for CoordParam:



Public Member Functions

- CoordParam ()
- ∼CoordParam ()
- void build (const string &infile)
- void check_all () const
- void print_info () const

- void get_vol (double &vol) const
- void calc_volume ()
- void get_xmin (vector< double > &xmin) const
- void get_xmax (vector< double > &xmax) const
- void set_xmin (vector< double > &xmin)
- void set_xmax (vector< double > &xmax)
- void get_ndim (int &ndim) const
- void transform_t_to_x (const vector< double > &tvec, vector< double > &xvec) const

Private Member Functions

- CoordParam (const CoordParam &rhs)
- CoordParam & operator= (const CoordParam &rhs)

Private Attributes

- int ndim_
- double vol_
- vector< double > xmin_
- vector< double > xmax_
- vector< double > length_

3.1.1 Detailed Description

Definition at line 12 of file 800_CoordParam.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 CoordParam::CoordParam ( )

Definition at line 13 of file 800_CoordParam.cpp.

3.1.2.2 CoordParam::∼CoordParam ( )

Definition at line 18 of file 800_CoordParam.cpp.

3.1.2.3 CoordParam::CoordParam (const **CoordParam** & *rhs*) `[private]`

3.1.3 Member Function Documentation

3.1.3.1 void CoordParam::build (const string & *infile*)

Definition at line 23 of file 800_CoordParam.cpp.

3.1.3.2 void CoordParam::calc_volume ( )

Definition at line 142 of file 800_CoordParam.cpp.

3.1.3.3 void CoordParam::check_all ( ) const

Definition at line 160 of file 800_CoordParam.cpp.

3.1.3.4 void CoordParam::get_ndim (int & *ndim*) const

Definition at line 150 of file 800_CoordParam.cpp.

3.1.3.5 void CoordParam::get_vol (double & *vol*) const

Definition at line 155 of file 800_CoordParam.cpp.

3.1.3.6 void CoordParam::get_xmax (vector< double > & *xmax*) const

Definition at line 101 of file 800_CoordParam.cpp.

3.1.3.7 void CoordParam::get_xmin (vector< double > & *xmin*) const

Definition at line 88 of file 800_CoordParam.cpp.

3.1.3.8 CoordParam& CoordParam::operator= (const CoordParam & *rhs*) `[private]`

3.1.3.9 void CoordParam::print_info ( ) const

Definition at line 65 of file 800_CoordParam.cpp.

3.1.3.10 void CoordParam::set_xmax (vector< double > & *xmax*)

Definition at line 128 of file 800_CoordParam.cpp.

3.1.3.11 void CoordParam::set_xmin (vector< double > & *xmin*)

Definition at line 114 of file 800_CoordParam.cpp.

3.1.3.12 void CoordParam::transform_t_to_x (const vector< double > & *tvec,* vector< double > & *xvec*) const

Definition at line 75 of file 800_CoordParam.cpp.

3.1.4 Member Data Documentation

3.1.4.1 vector<double> CoordParam::length_ `[private]`

Definition at line 34 of file 800_CoordParam.h.

3.1.4.2 int CoordParam::ndim_ `[private]`

Definition at line 30 of file 800_CoordParam.h.

3.1.4.3 double CoordParam::vol_ `[private]`

Definition at line 31 of file 800_CoordParam.h.

3.1.4.4 vector< double > CoordParam::xmax_ `[private]`

Definition at line 33 of file 800_CoordParam.h.

3.1.4.5 vector< double > CoordParam::xmin_ `[private]`

Definition at line 32 of file 800_CoordParam.h.

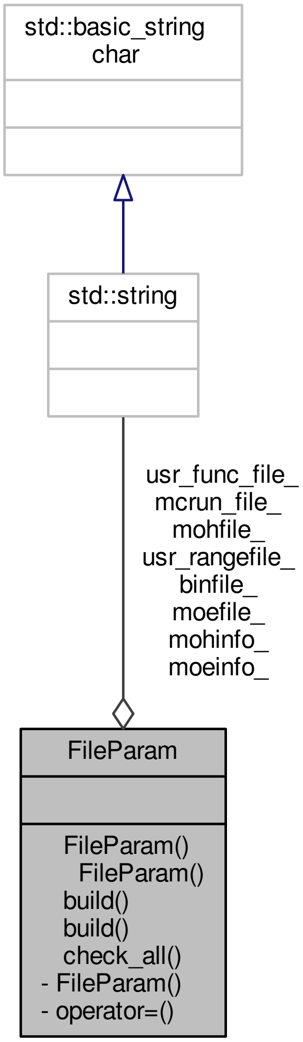
The documentation for this class was generated from the following files:

- 800_CoordParam.h
- 800_CoordParam.cpp

3.2 FileParam Class Reference

`#include <900_FileParam.h>`

Collaboration diagram for FileParam:



Public Member Functions

- FileParam ()
- ∼FileParam ()
- void build ()
- void build (const string &gridfile, const string &mohfile, const string &moefile, const string &usr_rangefile, const string &mcrun_file, const string &binfile)
- void check_all () const

Public Attributes

- string mohinfo_

- string moeinfo_
- string mohfile_
- string moefile_
- string usr_rangefile_
- string mcrun_file_
- string binfile_
- string usr_func_file_

Private Member Functions

- FileParam (const FileParam &rhs)
- FileParam & operator= (const FileParam &rhs)

3.2.1 Detailed Description

Definition at line 8 of file 900_FileParam.h.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 FileParam::FileParam ( )

Definition at line 10 of file 900_FileParam.cpp.

3.2.2.2 FileParam::∼FileParam ( )

Definition at line 15 of file 900_FileParam.cpp.

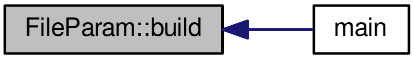
3.2.2.3 FileParam::FileParam (const FileParam & *rhs*) `[private]`

3.2.3 Member Function Documentation

3.2.3.1 void FileParam::build ( )

Definition at line 20 of file 900_FileParam.cpp.

Here is the caller graph for this function:



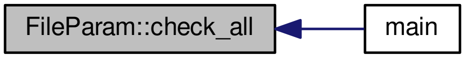
3.2.3.2 void FileParam::build (const string & *gridfile,* const string & *mohfile,* const string & *moefile,* const string & *usr_rangefile,* const string & *mcrun_file,* const string & *binfile*)

Definition at line 33 of file 900_FileParam.cpp.

3.2.3.3 void FileParam::check_all ( ) const

Definition at line 46 of file 900_FileParam.cpp.

Here is the caller graph for this function:



3.2.3.4 **FileParam**& FileParam::operator= (const **FileParam** & *rhs*) `[private]`

3.2.4 Member Data Documentation

3.2.4.1 string FileParam::binfile_

Definition at line 26 of file 900_FileParam.h.

3.2.4.2 string FileParam::mcrun_file_

Definition at line 25 of file 900_FileParam.h.

3.2.4.3 string FileParam::moefile_

Definition at line 23 of file 900_FileParam.h.

3.2.4.4 string FileParam::moeinfo_

Definition at line 21 of file 900_FileParam.h.

3.2.4.5 string FileParam::mohfile_

Definition at line 22 of file 900_FileParam.h.

3.2.4.6 string FileParam::mohinfo_

Definition at line 20 of file 900_FileParam.h.

3.2.4.7 string FileParam::usr_func_file_

Definition at line 27 of file 900_FileParam.h.

3.2.4.8 string FileParam::usr_rangefile_

Definition at line 24 of file 900_FileParam.h.

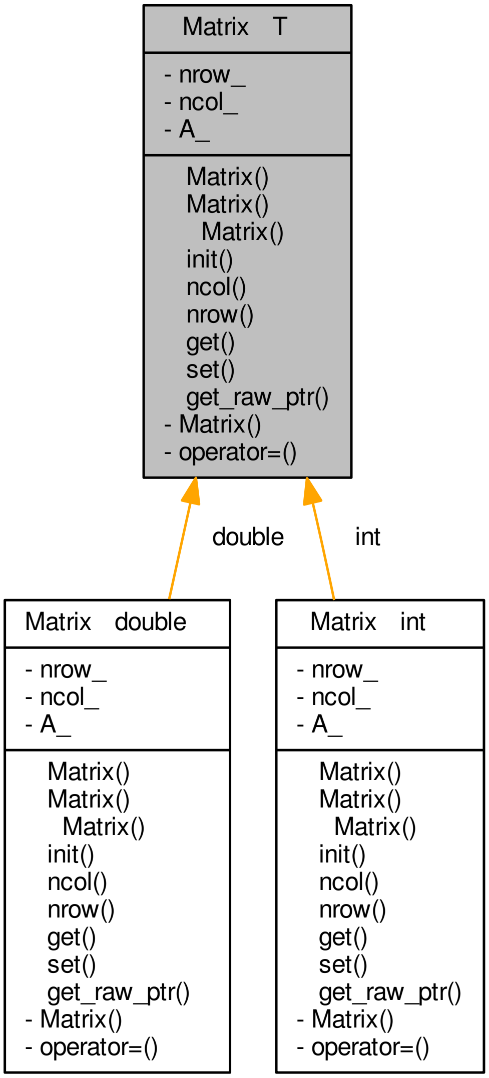
The documentation for this class was generated from the following files:

- 900_FileParam.h
- 900_FileParam.cpp

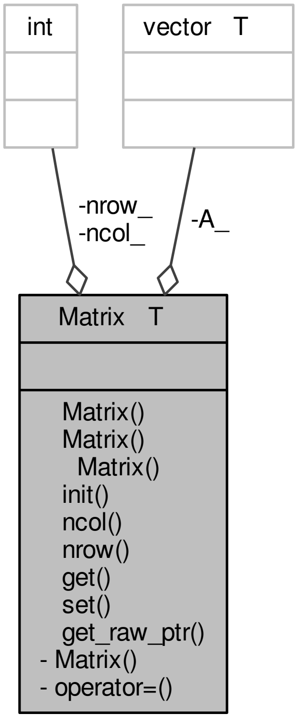
3.3 Matrix< T > Class Template Reference

`#include <Matrix.h>`

Inheritance diagram for Matrix< T >:



Collaboration diagram for Matrix< T >:



Public Member Functions

- Matrix ()
- Matrix (const int &nrow, const int &ncol)
- ∼Matrix ()
- void init (const int &nrow, const int &ncol)
- int ncol () const
- int nrow () const
- void get (const int &irow, const int &icol, T &val) const
- void set (const int &irow, const int &icol, const T &val)
- void get_raw_ptr (T ∗&ptr)

Private Member Functions

- Matrix (const Matrix &rhs)
- Matrix & operator= (const Matrix &rhs)

Private Attributes

- int nrow_
- int ncol_
- vector< T > A_

3.3.1 Detailed Description

template<class T>class Matrix< T >

Definition at line 14 of file Matrix.h.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 template<class T> Matrix< T >::Matrix ( ) `[inline]`

Definition at line 16 of file Matrix.h.

3.3.2.2 template<class T> Matrix< T >::Matrix (const int & *nrow*, const int & *ncol*) `[inline]`

Definition at line 17 of file Matrix.h.

3.3.2.3 template<class T> Matrix< T >::∼Matrix ( ) `[inline]`

Definition at line 20 of file Matrix.h.

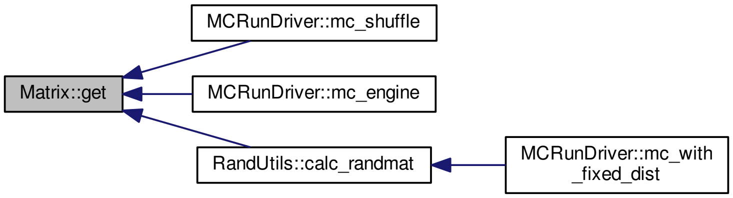
3.3.2.4 template<class T> Matrix< T >::Matrix (const Matrix< T > & *rhs*) `[private]`

3.3.3 Member Function Documentation

3.3.3.1 template<class T> void Matrix< T >::get (const int & *irow*, const int & *icol*, T & *val*) const `[inline]`

Definition at line 37 of file Matrix.h.

Here is the caller graph for this function:



3.3.3.2 template<class T> void Matrix< T >::get_raw_ptr (T ∗& *ptr*) `[inline]`

Definition at line 54 of file Matrix.h.

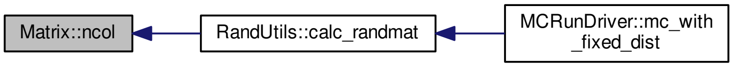
3.3.3.3 template<class T> void Matrix< T >::init (const int & *nrow*, const int & *ncol*) `[inline]`

Definition at line 22 of file Matrix.h.

3.3.3.4 template<class T> int **Matrix**< T >::ncol ( ) const [inline]

Definition at line 30 of file Matrix.h.

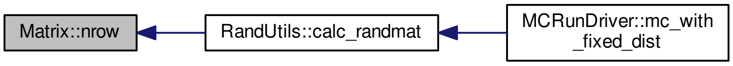
Here is the caller graph for this function:



3.3.3.5 template<class T> int **Matrix**< T >::nrow ( ) const [inline]

Definition at line 33 of file Matrix.h.

Here is the caller graph for this function:

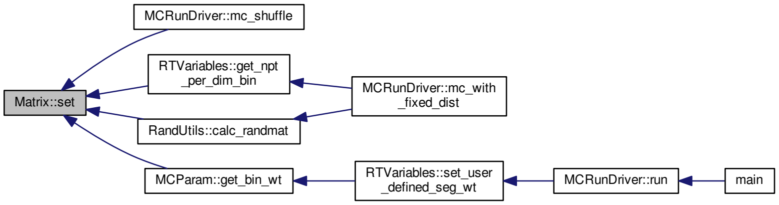


3.3.3.6 template<class T> **Matrix& Matrix**< T >::operator= (const **Matrix**< T > & *rhs*) [private]

3.3.3.7 template<class T> void **Matrix**< T >::set (const int & *irow,* const int & *icol,* const T & *val*) [inline]

Definition at line 46 of file Matrix.h.

Here is the caller graph for this function:



3.3.4 Member Data Documentation

3.3.4.1 template<class T> vector<T> **Matrix**< T >::A_ [private]

Definition at line 64 of file Matrix.h.

3.3.4.2 template<class T> int Matrix< T >::ncol_ `[private]`

Definition at line 63 of file Matrix.h.

3.3.4.3 template<class T> int Matrix< T >::nrow_ `[private]`

Definition at line 62 of file Matrix.h.

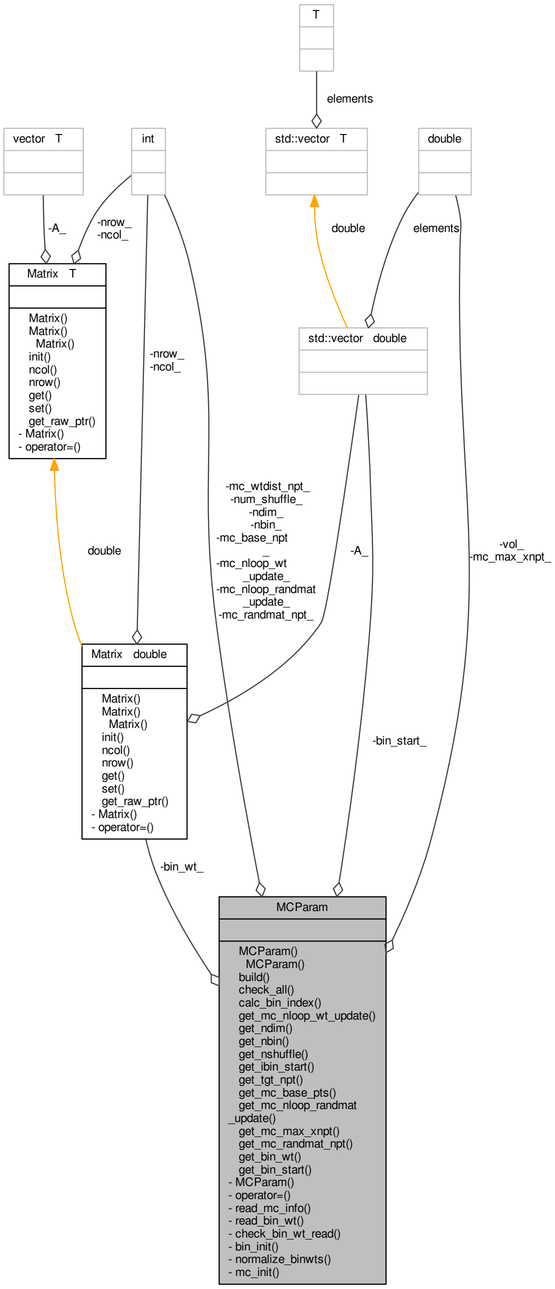
The documentation for this class was generated from the following file:

- Matrix.h

3.4 MCParam Class Reference

```
#include <500_MCParam.h>
```

Collaboration diagram for MCParam:



Public Member Functions

- MCParam ()
- ∼MCParam ()
- void build (const FileParam &myFile, const UserParam &myUser)
- void check_all () const
- void calc_bin_index (const vector< double > &tvec, vector< int > &tvec_binidx) const

- void get_mc_nloop_wt_update (int &nloop) const
- void get_ndim (int &ndim) const
- void get_nbin (int &nbin) const
- void get_nshuffle (int &nshuffle) const
- void get_ibin_start (const int &ibin, double &bin_val) const
- void get_tgt_npt (int &tgt_npt) const
- void get_mc_base_pts (int &base_pts) const
- void get_mc_nloop_randmat_update (int &mc_nloop_randmat_update) const
- void get_mc_max_xnpt (double &max_xnpt) const
- void get_mc_randmat_npt (int &mc_randmat_npt) const
- void get_bin_wt (Matrix< double > &some_mat) const
- void get_bin_start (vector< double > &bin_start) const

Private Member Functions

- MCParam (const MCParam &rhs)
- MCParam & operator= (const MCParam &rhs)
- void read_mc_info (const string &mcrun_file)
- void read_bin_wt (const string &binfile)
- void check_bin_wt_read (const int &ndim_local, const int &nbin_local) const
- void bin_init ()
- void normalize_binwts ()
- void mc_init ()

Private Attributes

- int ndim_
- int nbin_
- int num_shuffle_
- int mc_randmat_npt_
- int mc_base_npt_
- int mc_wtdist_npt_
- int mc_nloop_randmat_update_
- int mc_nloop_wt_update_
- double mc_max_xnpt_
- double vol_
- vector< double > bin_start_
- Matrix< double > bin_wt_

3.4.1 Detailed Description

Definition at line 13 of file 500_MCParam.h.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 MCParam::MCParam ( )

Definition at line 18 of file 500_MCParam.cpp.

3.4.2.2 MCParam::∼MCParam ( )

Definition at line 23 of file 500_MCParam.cpp.

3.4.2.3 MCParam::MCParam (const **MCParam** & *rhs*) `[private]`

3.4.3 Member Function Documentation

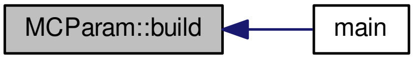
3.4.3.1 void MCParam::bin_init ( ) `[private]`

Definition at line 169 of file 500_MCParam.cpp.

3.4.3.2 void MCParam::build (const **FileParam** & *myFile,* const UserParam & *myUser*)

Definition at line 28 of file 500_MCParam.cpp.

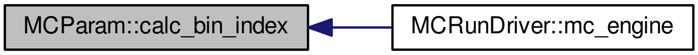
Here is the caller graph for this function:



3.4.3.3 void MCParam::calc_bin_index (const vector< double > & *tvec,* vector< int > & *tvec_binidx*) const

Definition at line 306 of file 500_MCParam.cpp.

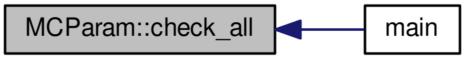
Here is the caller graph for this function:



3.4.3.4 void MCParam::check_all ( ) const

Definition at line 372 of file 500_MCParam.cpp.

Here is the caller graph for this function:



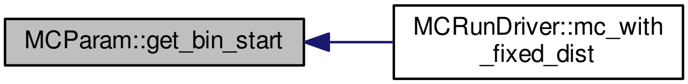
3.4.3.5 void MCParam::check_bin_wt_read (const int & *ndim_local,* const int & *nbin_local*) const `[private]`

Definition at line 346 of file 500_MCParam.cpp.

3.4.3.6 void MCParam::get_bin_start (vector< double > & *bin_start*) const

Definition at line 294 of file 500_MCParam.cpp.

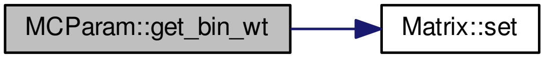
Here is the caller graph for this function:



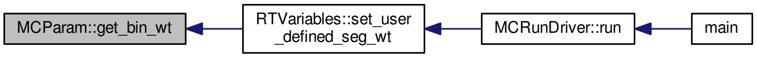
3.4.3.7 void MCParam::get_bin_wt (**Matrix**< double > & *some_mat*) const

Definition at line 283 of file 500_MCParam.cpp.

Here is the call graph for this function:



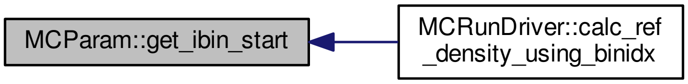
Here is the caller graph for this function:



3.4.3.8 void MCParam::get_ibin_start (const int & *ibin,* double & *bin_val*) const

Definition at line 248 of file 500_MCParam.cpp.

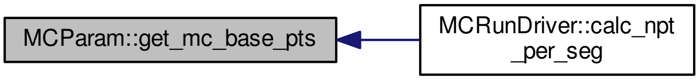
Here is the caller graph for this function:



3.4.3.9 void MCParam::get_mc_base_pts (int & *base_pts*) const

Definition at line 258 of file 500_MCParam.cpp.

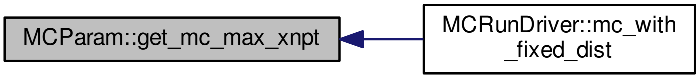
Here is the caller graph for this function:



3.4.3.10 void MCParam::get_mc_max_xnpt (double & *max_xnpt*) const

Definition at line 273 of file 500_MCParam.cpp.

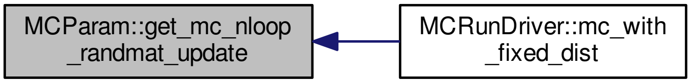
Here is the caller graph for this function:



3.4.3.11 void MCParam::get_mc_nloop_randmat_update (int & *mc_nloop_randmat_update*) const

Definition at line 263 of file 500_MCParam.cpp.

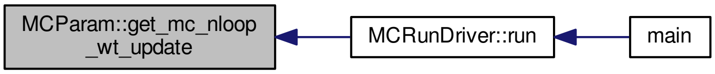
Here is the caller graph for this function:


MCParam::get_mc_nloop_randmat_update
MCRunDriver::mc_with_fixed_dist

3.4.3.12 void MCParam::get_mc_nloop_wt_update (int & *nloop*) const

Definition at line 233 of file 500_MCParam.cpp.

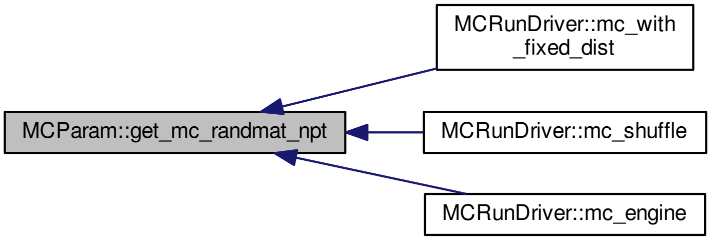
Here is the caller graph for this function:



3.4.3.13 void MCParam::get_mc_randmat_npt (int & *mc_randmat_npt*) const

Definition at line 268 of file 500_MCParam.cpp.

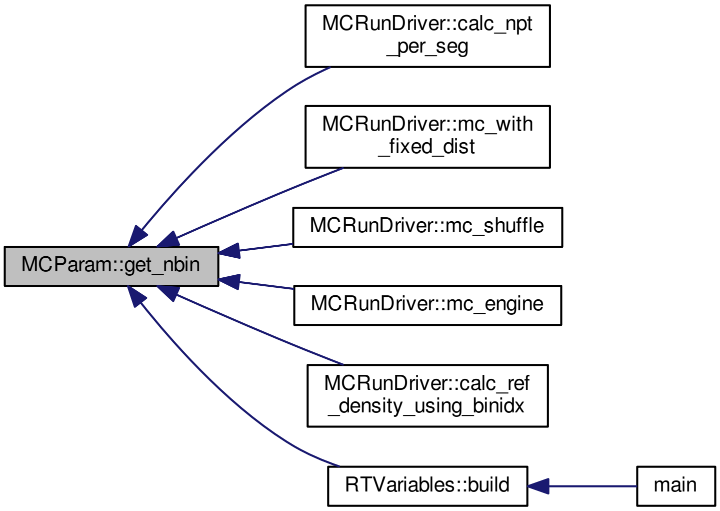
Here is the caller graph for this function:



3.4.3.14 void MCParam::get_nbin (int & *nbin*) const

Definition at line 243 of file 500_MCParam.cpp.

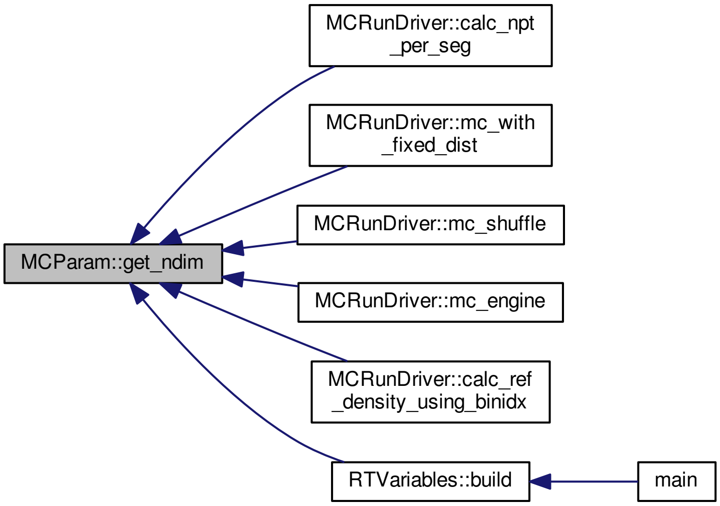
Here is the caller graph for this function:


MCRunDriver::calc_npt_per_seg
MCRunDriver::mc_with_fixed_dist
MCRunDriver::mc_shuffle
MCParam::get_nbin
MCRunDriver::mc_engine
MCRunDriver::calc_ref_density_using_binidx
RTVariables::build
main

3.4.3.15 void MCParam::get_ndim (int & *ndim*) const

Definition at line 238 of file 500_MCParam.cpp.

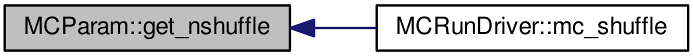
Here is the caller graph for this function:


MCRunDriver::calc_npt_per_seg
MCRunDriver::mc_with_fixed_dist
MCRunDriver::mc_shuffle
MCParam::get_ndim
MCRunDriver::mc_engine
MCRunDriver::calc_ref_density_using_binidx
RTVariables::build
main

3.4.3.16 void MCParam::get_nshuffle (int & *nshuffle*) const

Definition at line 278 of file 500_MCParam.cpp.

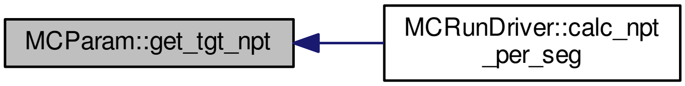
Here is the caller graph for this function:



3.4.3.17 void MCParam::get_tgt_npt (int & *tgt_npt*) const

Definition at line 253 of file 500_MCParam.cpp.

Here is the caller graph for this function:



3.4.3.18 void MCParam::mc_init ( ) `[private]`

Definition at line 218 of file 500_MCParam.cpp.

3.4.3.19 void MCParam::normalize_binwts ( ) `[private]`

Definition at line 188 of file 500_MCParam.cpp.

3.4.3.20 **MCParam**& MCParam::operator= (const **MCParam** & *rhs*) `[private]`

3.4.3.21 void MCParam::read_bin_wt (const string & *binfile*) `[private]`

Definition at line 110 of file 500_MCParam.cpp.

3.4.3.22 void MCParam::read_mc_info (const string & *mcrun_file*) `[private]`

Definition at line 61 of file 500_MCParam.cpp.

3.4.4 Member Data Documentation

3.4.4.1 vector< double > MCParam::bin_start_ `[private]`

Definition at line 45 of file 500_MCParam.h.

3.4.4.2 **Matrix**< double > MCParam::bin_wt_ `[private]`

Definition at line 46 of file 500_MCParam.h.

3.4.4.3 int MCParam::mc_base_npt_ `[private]`

Definition at line 39 of file 500_MCParam.h.

3.4.4.4 double MCParam::mc_max_xnpt_ `[private]`

Definition at line 43 of file 500_MCParam.h.

3.4.4.5 int MCParam::mc_nloop_randmat_update_ `[private]`

Definition at line 41 of file 500_MCParam.h.

3.4.4.6 int MCParam::mc_nloop_wt_update_ `[private]`

Definition at line 42 of file 500_MCParam.h.

3.4.4.7 int MCParam::mc_randmat_npt_ `[private]`

Definition at line 38 of file 500_MCParam.h.

3.4.4.8 int MCParam::mc_wtdist_npt_ `[private]`

Definition at line 40 of file 500_MCParam.h.

3.4.4.9 int MCParam::nbin_ `[private]`

Definition at line 36 of file 500_MCParam.h.

3.4.4.10 int MCParam::ndim_ `[private]`

Definition at line 35 of file 500_MCParam.h.

3.4.4.11 int MCParam::num_shuffle_ `[private]`

Definition at line 37 of file 500_MCParam.h.

3.4.4.12 double MCParam::vol_ `[private]`

Definition at line 44 of file 500_MCParam.h.

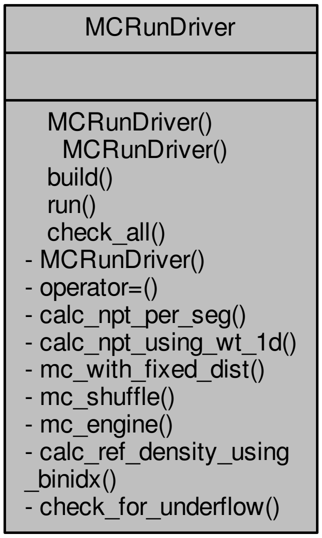
The documentation for this class was generated from the following files:

- 500_MCParam.h
- 500_MCParam.cpp

3.5 MCRunDriver Class Reference

`#include <100_MCRunDriver.h>`

Collaboration diagram for MCRunDriver:



Public Member Functions

- MCRunDriver ()
- ∼MCRunDriver ()
- void build ()
- void run (const FileParam &myFile, const UserParam &myUser, const MCParam &myMCParam, RTVariables &myRTV)
- void check_all () const

Private Member Functions

- MCRunDriver (const MCRunDriver &rhs)
- MCRunDriver & operator= (const MCRunDriver &rhs)
- void calc_npt_per_seg (RTVariables &myRTV, const MCParam &myMCParam)

- void calc_npt_using_wt_1d (RTVariables &myRTV, const int &ndim, const int &nbin, const int &mc_base_pts, const int &mc_tgt_npt, const int &idim)
- void mc_with_fixed_dist (const UserParam &myUser, const MCParam &myMCParam, RTVariables &myRTV, RandUtils &myRNG)
- void mc_shuffle (const UserParam &myUser, const MCParam &myMCParam, RTVariables &myRTV, const RandUtils &myRNG, Matrix< double > &randmat)
- void mc_engine (const UserParam &myUser, const MCParam &myMCParam, RTVariables &myRTV, Matrix< double > &randmat)
- void calc_ref_density_using_binidx (const MCParam &myMCParam, const RTVariables &myRTV, const vector< int > &idx, double &rho)
- void check_for_underflow (double &val)

3.5.1 Detailed Description

Definition at line 17 of file 100_MCRunDriver.h.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 MCRunDriver::MCRunDriver ( )

Definition at line 20 of file 100_MCRunDriver.cpp.

3.5.2.2 MCRunDriver::∼MCRunDriver ( )

Definition at line 25 of file 100_MCRunDriver.cpp.

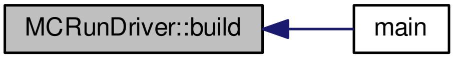
3.5.2.3 MCRunDriver::MCRunDriver (const **MCRunDriver** & *rhs*) `[private]`

3.5.3 Member Function Documentation

3.5.3.1 void MCRunDriver::build ( )

Definition at line 30 of file 100_MCRunDriver.cpp.

Here is the caller graph for this function:

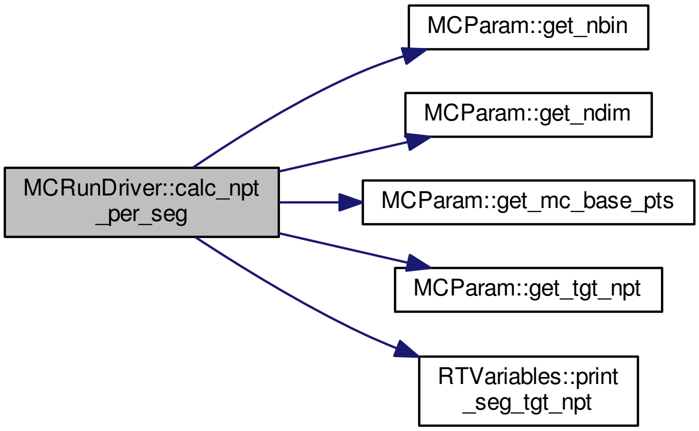


3.5.3.2 void MCRunDriver::calc_npt_per_seg (**RTVariables** & *myRTV,* const **MCParam** & *myMCParam*) `[private]`

Calculates the number of points per segment of ρ

Definition at line 64 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:

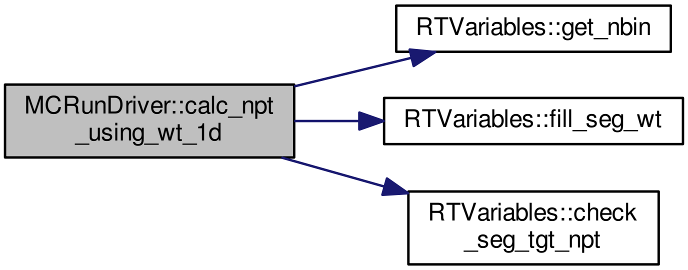


3.5.3.3 void MCRunDriver::calc_npt_using_wt_1d (**RTVariables** & *myRTV,* const int & *ndim,* const int & *nbin,* const int & *mc_base_pts,* const int & *mc_tgt_npt,* const int & *idim*) `[private]`

Determines the number of points to assign to a segment of ρ based on the weights

Definition at line 83 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:

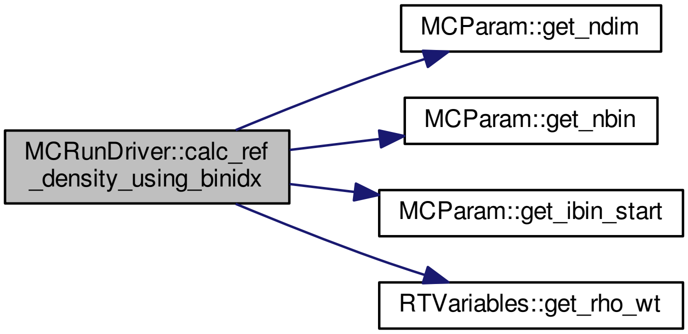


3.5.3.4 void MCRunDriver::calc_ref_density_using_binidx (const **MCParam** & *myMCParam,* const **RTVariables** & *myRTV,* const vector< int > & *idx,* double & *rho*) `[private]`

Calculate the N-dimensional density for a point in space from ρ of each dimension.

Definition at line 237 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:



3.5.3.5 void MCRunDriver::check_all ( ) const

Assertions of the MCRunDriver object.

Definition at line 274 of file 100_MCRunDriver.cpp.

3.5.3.6 void MCRunDriver::check_for_underflow (double & *val*) [private]

Check for underflow of a value

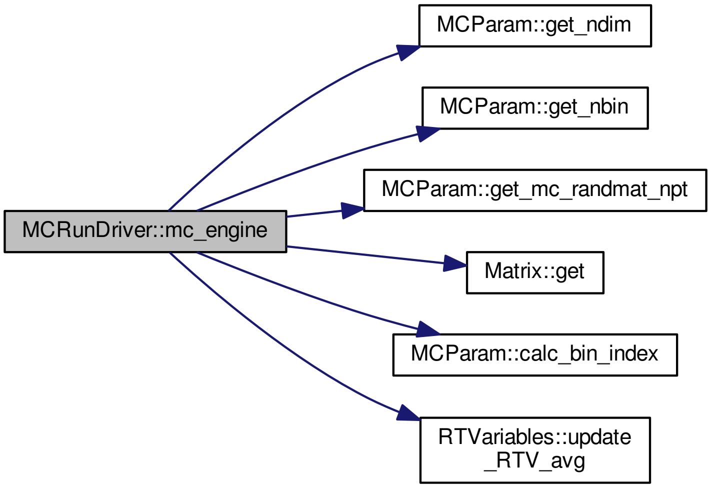
Definition at line 263 of file 100_MCRunDriver.cpp.

3.5.3.7 void MCRunDriver::mc_engine (const UserParam & *myUser*, const **MCParam** & *myMCParam*, **RTVariables** & *myRTV*, **Matrix**< double > & *randmat*) [private]

Main Monte Carlo engine were the random matrix is now used to generate N-Dimensional points in space that will be evaluated within User defined function. The value of the function at each position is then updated in the evaluation of the integral.

Definition at line 180 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:

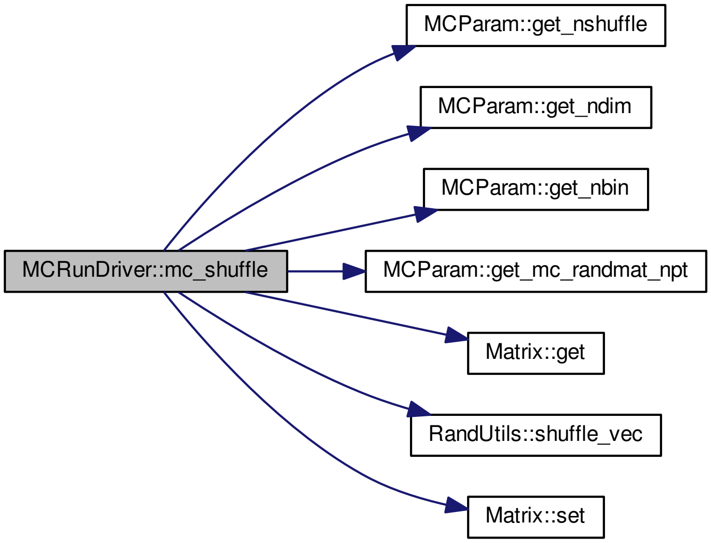


3.5.3.8 void MCRunDriver::mc_shuffle (const UserParam & *myUser*, const **MCParam** & *myMCParam*, **RTVariables** & *myRTV*, const **RandUtils** & *myRNG*, **Matrix**< double > & *randmat*) `[private]`

Take the random matrix and shuffle the ρ along the dimension in each shuffle loop.

Definition at line 145 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:

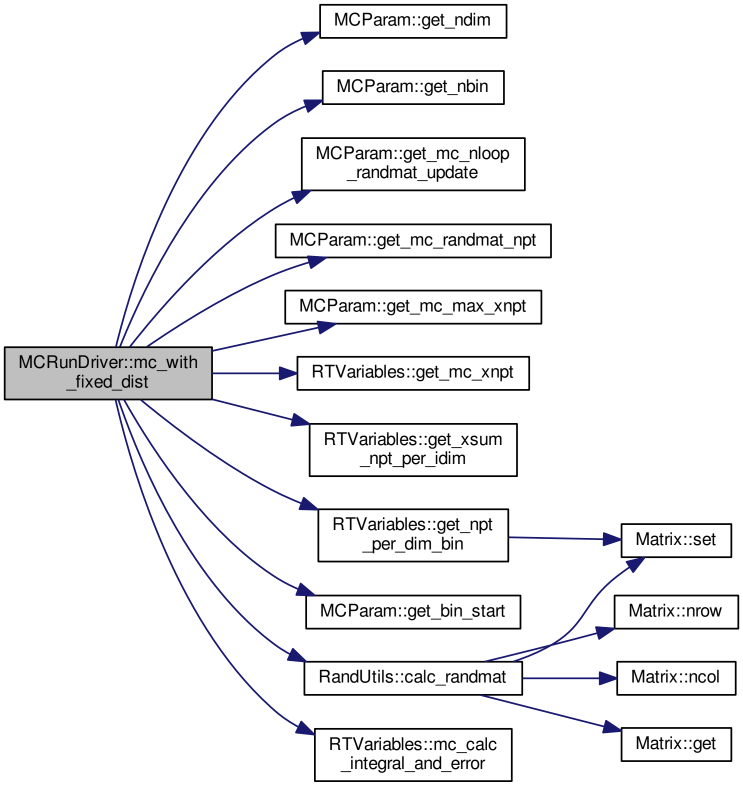


3.5.3.9 void MCRunDriver::mc_with_fixed_dist (const UserParam & *myUser,* const **MCParam** & *myMCParam,* **RTVariables** & *myRTV,* **RandUtils** & *myRNG*) `[private]`

Using the ρ constructed, evaluate a random matrix that contains position values with the same distribution of ρ within each loop. During each loop pass the random matrix to be used for evaluation of the integral. Updating the integral value and error until the max number of Monte Carlo points has been achieved.

Definition at line 104 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:


MCParam::get_ndim
MCParam::get_nbin
MCParam::get_mc_nloop_randmat_update
MCParam::get_mc_randmat_npt
MCParam::get_mc_max_xnpt
MCRunDriver::mc_with_fixed_dist
RTVariables::get_mc_xnpt
RTVariables::get_xsum_npt_per_idim
RTVariables::get_npt_per_dim_bin
Matrix::set
MCParam::get_bin_start
Matrix::nrow
RandUtils::calc_randmat
Matrix::ncol
RTVariables::mc_calc_integral_and_error
Matrix::get

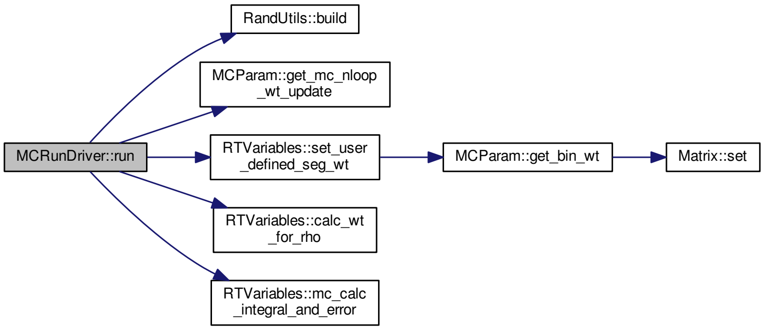
3.5.3.10 **MCRunDriver&** MCRunDriver::operator= (const **MCRunDriver** & *rhs*) `[private]`

3.5.3.11 void MCRunDriver::run (const **FileParam** & *myFile,* const UserParam & *myUser,* const **MCParam** & *myMCParam,* **RTVariables** & *myRTV*)

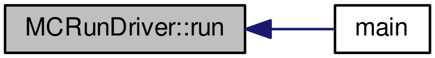
Main run driver that calculates a probabilty distribution, ρ from weights for each Monte Carlo cycles to perform

Definition at line 35 of file 100_MCRunDriver.cpp.

Here is the call graph for this function:


RandUtils::build
MCParam::get_mc_nloop _wt_update
MCRunDriver::run
RTVariables::set_user _defined_seg_wt
MCParam::get_bin_wt
Matrix::set
RTVariables::calc_wt _for_rho
RTVariables::mc_calc _integral_and_error

Here is the caller graph for this function:



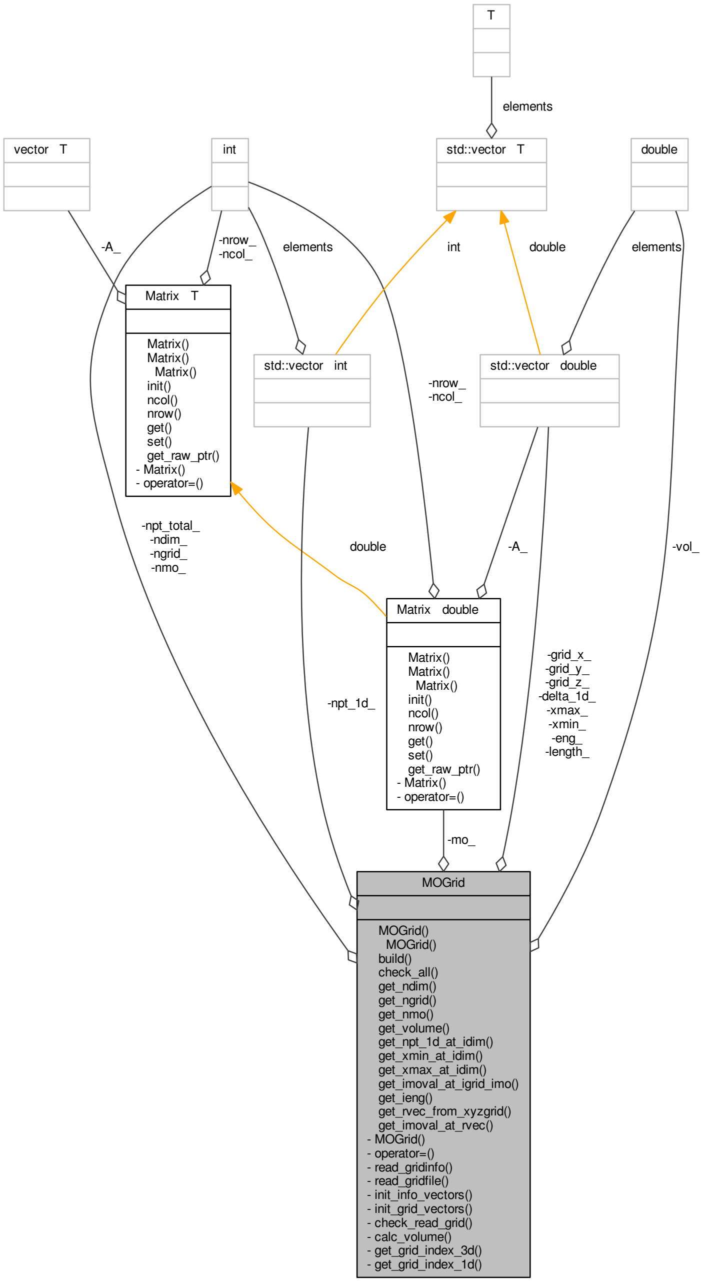
The documentation for this class was generated from the following files:

- 100_MCRunDriver.h
- 100_MCRunDriver.cpp

3.6 MOGrid Class Reference

`#include <800_MOGrid.h>`

Collaboration diagram for MOGrid:



Public Member Functions

- MOGrid ()
- ∼MOGrid ()
- void build (const string &infofile, const string &gridfile)
- void check_all () const
- void get_ndim (int &ndim) const

- void get_ngrid (int &ngrid) const
- void get_nmo (int &nmo) const
- void get_volume (double &vol) const
- void get_npt_1d_at_idim (int &idim, int &npt_val) const
- void get_xmin_at_idim (int &idim, double &xmin_val) const
- void get_xmax_at_idim (int &idim, double &xmax_val) const
- void get_imoval_at_igrid_imo (const int &igrid, const int &imo, double &mo_val) const
- void get_ieng (const int &imo, double &ieng) const
- void get_rvec_from_xyzgrid (const int &xyz, vector< double > &rvec) const
- void get_imoval_at_rvec (const int &imo, const vector< double > &rvec, double &moval) const

Private Member Functions

- MOGrid (const MOGrid &rhs)
- MOGrid & operator= (const MOGrid &rhs)
- void read_gridinfo (const string &infile)
- void read_gridfile (const string &infile)
- void init_info_vectors ()
- void init_grid_vectors ()
- void check_read_grid (const int &ngrid_local, const int &ndim_local, const int &nmo_local) const
- void calc_volume ()
- void get_grid_index_3d (const vector< double > &rv, int &gidx) const
- void get_grid_index_1d (const int &idim, const double &x, int &idx) const

Private Attributes

- int ndim_
- int ngrid_
- int nmo_
- int npt_total_
- double vol_
- vector< int > npt_1d_
- vector< double > xmin_
- vector< double > xmax_
- vector< double > length_
- vector< double > delta_1d_
- vector< double > grid_x_
- vector< double > grid_y_
- vector< double > grid_z_
- vector< double > eng_
- Matrix< double > mo_

3.6.1 Detailed Description

Definition at line 11 of file 800_MOGrid.h.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 MOGrid::MOGrid ( )

Definition at line 23 of file 800_MOGrid.cpp.

3.6.2.2 MOGrid::∼MOGrid ( )

Definition at line 28 of file 800_MOGrid.cpp.

3.6.2.3 MOGrid::MOGrid (const **MOGrid** & *rhs*) `[private]`

3.6.3 Member Function Documentation

3.6.3.1 void MOGrid::build (const string & *infofile,* const string & *gridfile*)

Definition at line 33 of file 800_MOGrid.cpp.

3.6.3.2 void MOGrid::calc_volume ( ) `[private]`

Definition at line 173 of file 800_MOGrid.cpp.

3.6.3.3 void MOGrid::check_all ( ) const

Definition at line 314 of file 800_MOGrid.cpp.

3.6.3.4 void MOGrid::check_read_grid (const int & *ngrid_local,* const int & *ndim_local,* const int & *nmo_local*) const `[private]`

Definition at line 285 of file 800_MOGrid.cpp.

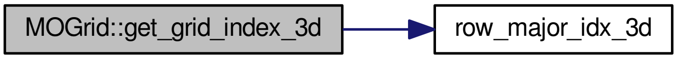
3.6.3.5 void MOGrid::get_grid_index_1d (const int & *idim,* const double & *x,* int & *idx*) const `[private]`

Definition at line 199 of file 800_MOGrid.cpp.

3.6.3.6 void MOGrid::get_grid_index_3d (const vector< double > & *rv,* int & *gidx*) const `[private]`

Definition at line 182 of file 800_MOGrid.cpp.

Here is the call graph for this function:



3.6.3.7 void MOGrid::get_ieng (const int & *imo,* double & *ieng*) const

Definition at line 253 of file 800_MOGrid.cpp.

3.6.3.8 void MOGrid::get_imoval_at_igrid_imo (const int & *igrid,* const int & *imo,* double & *mo_val*) const

Definition at line 226 of file 800_MOGrid.cpp.

3.6.3.9 void MOGrid::get_imoval_at_rvec (const int & *imo,* const vector< double > & *rvec,* double & *moval*) const

Definition at line 274 of file 800_MOGrid.cpp.

3.6.3.10 void MOGrid::get_ndim (int & *ndim*) const

Definition at line 233 of file 800_MOGrid.cpp.

3.6.3.11 void MOGrid::get_ngrid (int & *ngrid*) const

Definition at line 238 of file 800_MOGrid.cpp.

3.6.3.12 void MOGrid::get_nmo (int & *nmo*) const

Definition at line 243 of file 800_MOGrid.cpp.

3.6.3.13 void MOGrid::get_npt_1d_at_idim (int & *idim,* int & *npt_val*) const

Definition at line 259 of file 800_MOGrid.cpp.

3.6.3.14 void MOGrid::get_rvec_from_xyzgrid (const int & *xyz,* vector< double > & *rvec*) const

Definition at line 264 of file 800_MOGrid.cpp.

3.6.3.15 void MOGrid::get_volume (double & *vol*) const

Definition at line 248 of file 800_MOGrid.cpp.

3.6.3.16 void MOGrid::get_xmax_at_idim (int & *idim,* double & *xmax_val*) const

Definition at line 220 of file 800_MOGrid.cpp.

3.6.3.17 void MOGrid::get_xmin_at_idim (int & *idim,* double & *xmin_val*) const

Definition at line 215 of file 800_MOGrid.cpp.

3.6.3.18 void MOGrid::init_grid_vectors ( ) `[private]`

Definition at line 163 of file 800_MOGrid.cpp.

3.6.3.19 void MOGrid::init_info_vectors ( ) `[private]`

Definition at line 92 of file 800_MOGrid.cpp.

3.6.3.20 MOGrid& MOGrid::operator= (const MOGrid & *rhs*) `[private]`

3.6.3.21 void MOGrid::read_gridfile (const string & *infile*) `[private]`

Definition at line 102 of file 800_MOGrid.cpp.

3.6.3.22 void MOGrid::read_gridinfo (const string & *infile*) `[private]`

Definition at line 41 of file 800_MOGrid.cpp.

3.6.4 Member Data Documentation

3.6.4.1 vector<double> MOGrid::delta_1d_ `[private]`

Definition at line 48 of file 800_MOGrid.h.

3.6.4.2 vector<double> MOGrid::eng_ `[private]`

Definition at line 52 of file 800_MOGrid.h.

3.6.4.3 vector<double> MOGrid::grid_x_ `[private]`

Definition at line 49 of file 800_MOGrid.h.

3.6.4.4 vector<double> MOGrid::grid_y_ `[private]`

Definition at line 50 of file 800_MOGrid.h.

3.6.4.5 vector<double> MOGrid::grid_z_ `[private]`

Definition at line 51 of file 800_MOGrid.h.

3.6.4.6 vector<double> MOGrid::length_ `[private]`

Definition at line 47 of file 800_MOGrid.h.

3.6.4.7 Matrix<double> MOGrid::mo_ `[private]`

Definition at line 53 of file 800_MOGrid.h.

3.6.4.8 int MOGrid::ndim_ `[private]`

Definition at line 39 of file 800_MOGrid.h.

3.6.4.9 int MOGrid::ngrid_ `[private]`

Definition at line 40 of file 800_MOGrid.h.

3.6.4.10 int MOGrid::nmo_ `[private]`

Definition at line 41 of file 800_MOGrid.h.

3.6.4.11 vector<int> MOGrid::npt_1d_ `[private]`

Definition at line 44 of file 800_MOGrid.h.

3.6.4.12 int MOGrid::npt_total_ `[private]`

Definition at line 42 of file 800_MOGrid.h.

3.6.4.13 double MOGrid::vol_ `[private]`

Definition at line 43 of file 800_MOGrid.h.

3.6.4.14 vector<double> MOGrid::xmax_ `[private]`

Definition at line 46 of file 800_MOGrid.h.

3.6.4.15 vector<double> MOGrid::xmin_ `[private]`

Definition at line 45 of file 800_MOGrid.h.

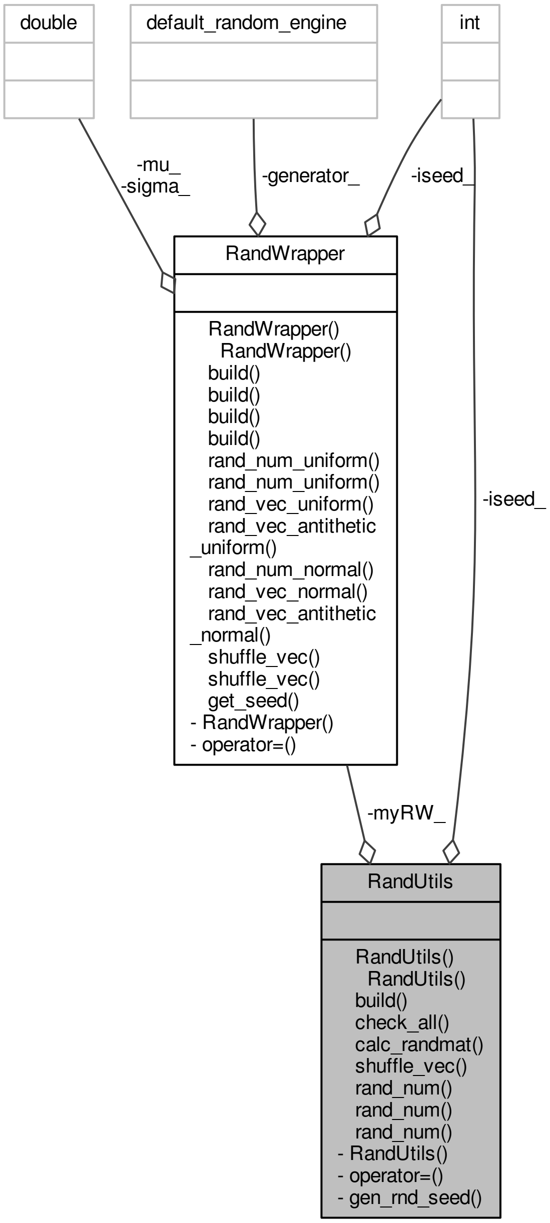
The documentation for this class was generated from the following files:

- 800_MOGrid.h
- 800_MOGrid.cpp

3.7 RandUtils Class Reference

`#include <900_RandUtils.h>`

Collaboration diagram for RandUtils:



Public Member Functions

- RandUtils ()
- ∼RandUtils ()
- void build ()
- void check_all () const
- void calc_randmat (const int &ndim, const int &nbin, const int &mc_randmat_npt, const vector< int > &xsum_npt_per_idim, const Matrix< int > &npt_per_dim_bin, const vector< double > &bin_start, Matrix< double

> &randmat)
- void shuffle_vec (vector< double > &some_vec) const
- void rand_num (double &r) const
- void rand_num (const double &xmin, const double &xmax, double &r) const
- void rand_num (const int &imin, const int &imax, int &r) const

Private Member Functions

- RandUtils (const RandUtils &rhs)=delete
- RandUtils & operator= (const RandUtils &rhs)=delete
- void gen_rnd_seed ()

Private Attributes

- RandWrapper myRW_
- int iseed_

3.7.1 Detailed Description

Definition at line 13 of file 900_RandUtils.h.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 RandUtils::RandUtils ( )

Definition at line 20 of file 900_RandUtils.cpp.

3.7.2.2 RandUtils::∼RandUtils ( )

Definition at line 25 of file 900_RandUtils.cpp.

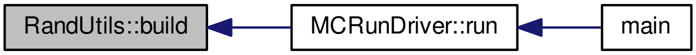
3.7.2.3 RandUtils::RandUtils (const **RandUtils** & *rhs*) [private], [delete]

3.7.3 Member Function Documentation

3.7.3.1 void RandUtils::build ( )

Definition at line 30 of file 900_RandUtils.cpp.

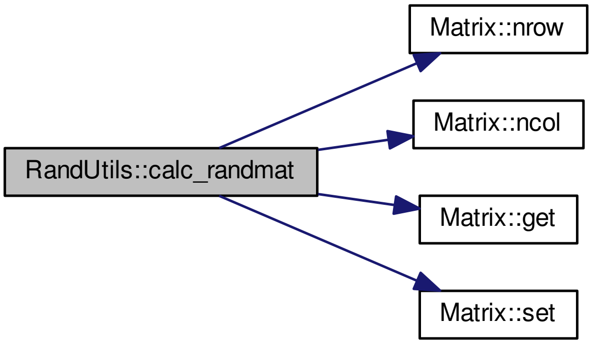
Here is the caller graph for this function:



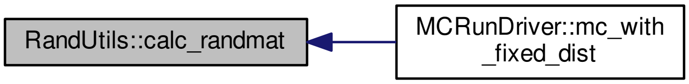
3.7.3.2 void RandUtils::calc_randmat (const int & *ndim,* const int & *nbin,* const int & *mc_randmat_npt,* const vector< int > & *xsum_npt_per_idim,* const **Matrix**< int > & *npt_per_dim_bin,* const vector< double > & *bin_start,* **Matrix**< double > & *randmat*)

Definition at line 70 of file 900_RandUtils.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



3.7.3.3 void RandUtils::check_all ( ) const

Definition at line 168 of file 900_RandUtils.cpp.

3.7.3.4 void RandUtils::gen_rnd_seed ( ) `[private]`

Definition at line 41 of file 900_RandUtils.cpp.

3.7.3.5 **RandUtils**& RandUtils::operator= (const **RandUtils** & *rhs*) `[private]`, `[delete]`

3.7.3.6 void RandUtils::rand_num (double & *r*) const

Definition at line 48 of file 900_RandUtils.cpp.

3.7.3.7 void RandUtils::rand_num (const double & *xmin,* const double & *xmax,* double & *r*) const

Definition at line 53 of file 900_RandUtils.cpp.

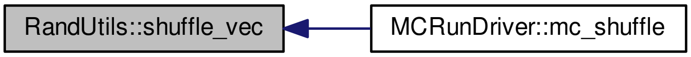
3.7.3.8 void RandUtils::rand_num (const int & *imin,* const int & *imax,* int & *r*) const

Definition at line 60 of file 900_RandUtils.cpp.

3.7.3.9 void RandUtils::shuffle_vec (vector< double > & *some_vec*) const

Definition at line 163 of file 900_RandUtils.cpp.

Here is the caller graph for this function:



3.7.4 Member Data Documentation

3.7.4.1 int RandUtils::iseed_ `[private]`

Definition at line 38 of file 900_RandUtils.h.

3.7.4.2 **RandWrapper** RandUtils::myRW_ `[private]`

Definition at line 37 of file 900_RandUtils.h.

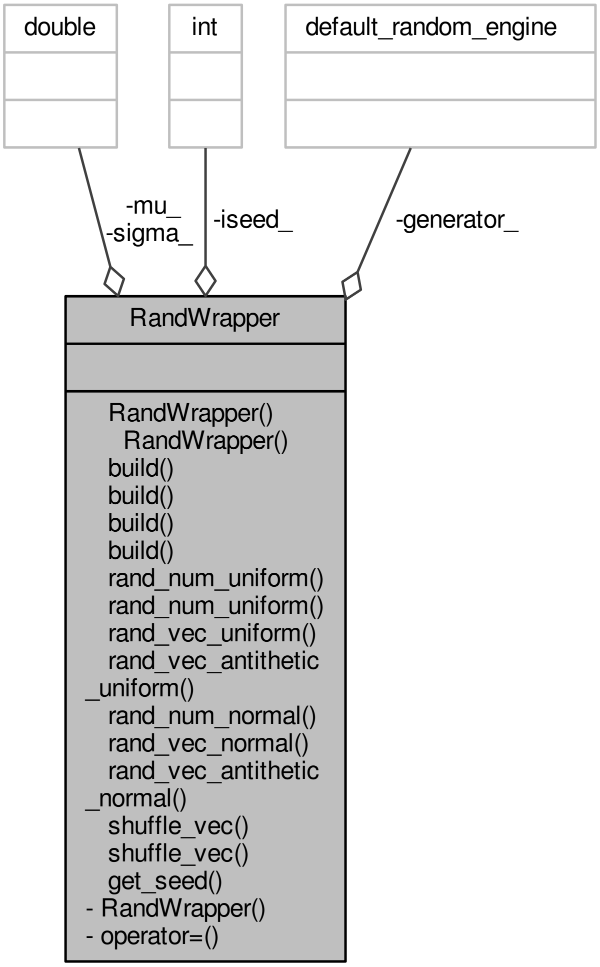
The documentation for this class was generated from the following files:

- 900_RandUtils.h
- 900_RandUtils.cpp

3.8 RandWrapper Class Reference

`#include <RandWrapper.h>`

Collaboration diagram for RandWrapper:



Public Member Functions

- RandWrapper ()
- ∼RandWrapper ()
- void build ()
- void build (const int &seed)
- void build (const double &mu, const double &sigma)
- void build (const int &seed, const double &mu, const double &sigma)
- void rand_num_uniform (double &r) const
- void rand_num_uniform (const int &imin, const int &imax, int &r) const
- void rand_vec_uniform (vector< double > &vec) const
- void rand_vec_antithetic_uniform (vector< double > &vec) const
- void rand_num_normal (double &r) const
- void rand_vec_normal (vector< double > &vec) const

- void rand_vec_antithetic_normal (vector$<$ double $>$ &vec) const
- void shuffle_vec (vector$<$ double $>$ &some_vec) const
- void shuffle_vec (vector$<$ int $>$ &some_vec) const
- void get_seed (int &seed) const

Private Member Functions

- RandWrapper (const RandWrapper &rhs)
- RandWrapper & operator= (const RandWrapper &rhs)

Private Attributes

- int iseed_
- double mu_
- double sigma_
- default_random_engine $*$ generator_

3.8.1 Detailed Description

Definition at line 11 of file RandWrapper.h.

3.8.2 Constructor & Destructor Documentation

3.8.2.1 RandWrapper::RandWrapper ( ) `[inline]`

Definition at line 13 of file RandWrapper.h.

3.8.2.2 RandWrapper::$\sim$RandWrapper ( )

Definition at line 10 of file RandWrapper.cpp.

3.8.2.3 RandWrapper::RandWrapper (const **RandWrapper** & *rhs*) `[private]`

3.8.3 Member Function Documentation

3.8.3.1 void RandWrapper::build ( )

Definition at line 16 of file RandWrapper.cpp.

3.8.3.2 void RandWrapper::build (const int & *seed*)

Definition at line 26 of file RandWrapper.cpp.

3.8.3.3 void RandWrapper::build (const double & *mu,* const double & *sigma*)

Definition at line 35 of file RandWrapper.cpp.

3.8.3.4 void RandWrapper::build (const int & *seed,* const double & *mu,* const double & *sigma*)

Definition at line 45 of file RandWrapper.cpp.

3.8.3.5 void RandWrapper::get_seed (int & *seed*) const

Definition at line 145 of file RandWrapper.cpp.

3.8.3.6 RandWrapper& RandWrapper::operator= (const RandWrapper & *rhs*) `[private]`

3.8.3.7 void RandWrapper::rand_num_normal (double & *r*) const

Definition at line 101 of file RandWrapper.cpp.

3.8.3.8 void RandWrapper::rand_num_uniform (double & *r*) const

Definition at line 56 of file RandWrapper.cpp.

3.8.3.9 void RandWrapper::rand_num_uniform (const int & *imin,* const int & *imax,* int & *r*) const

Definition at line 62 of file RandWrapper.cpp.

3.8.3.10 void RandWrapper::rand_vec_antithetic_normal (vector< double > & *vec*) const

Definition at line 115 of file RandWrapper.cpp.

3.8.3.11 void RandWrapper::rand_vec_antithetic_uniform (vector< double > & *vec*) const

Definition at line 79 of file RandWrapper.cpp.

3.8.3.12 void RandWrapper::rand_vec_normal (vector< double > & *vec*) const

Definition at line 107 of file RandWrapper.cpp.

3.8.3.13 void RandWrapper::rand_vec_uniform (vector< double > & *vec*) const

Definition at line 71 of file RandWrapper.cpp.

3.8.3.14 void RandWrapper::shuffle_vec (vector< double > & *some_vec*) const

Definition at line 135 of file RandWrapper.cpp.

3.8.3.15 void RandWrapper::shuffle_vec (vector< int > & *some_vec*) const

Definition at line 140 of file RandWrapper.cpp.

3.8.4 Member Data Documentation

3.8.4.1 default_random_engine* RandWrapper::generator_ `[private]`

Definition at line 35 of file RandWrapper.h.

3.8.4.2 int RandWrapper::iseed_ `[private]`

Definition at line 32 of file RandWrapper.h.

3.8.4.3 double RandWrapper::mu_ `[private]`

Definition at line 33 of file RandWrapper.h.

3.8.4.4 double RandWrapper::sigma_ `[private]`

Definition at line 34 of file RandWrapper.h.

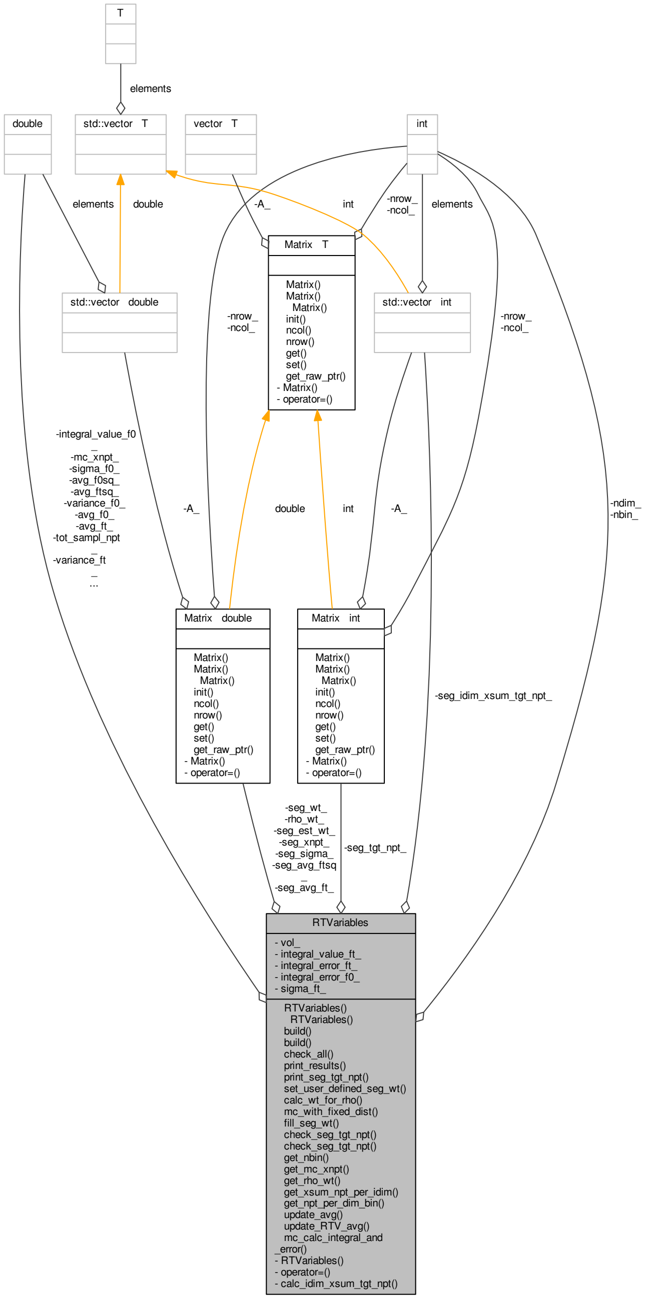
The documentation for this class was generated from the following files:

- RandWrapper.h
- RandWrapper.cpp

3.9 RTVariables Class Reference

`#include <400_RTVariables.h>`

Collaboration diagram for RTVariables:



Public Member Functions

- RTVariables ()
- ∼RTVariables ()
- void build (const FileParam &myFile, const UserParam &myUser, const MCParam &myMCParam)
- void build (const int &ndim, const int &nbin, const double &vol, const MCParam &myMCParam)
- void check_all () const

- void print_results () const
- void print_seg_tgt_npt () const
- void set_user_defined_seg_wt (const MCParam &myMCParam)
- void calc_wt_for_rho (const MCParam &myMCParam)
- void mc_with_fixed_dist (const MCParam &myMCParam, const UserParam &myUser)
- void fill_seg_wt (const int &idim, const int &ibin, const int &ndim, const int &nbin, const int &mc_base_pts, const int &mc_tgt_npt, int &xsum_npts, int &idx_max, double &max_wt)
- void check_seg_tgt_npt (const int &idim_max, const int &ibin_max, const int &xsum_npts, const int &mc_tgt-_npt)
- void check_seg_tgt_npt (const int &idim, const int &xsum_npts, const int &mc_tgt_npt)
- void get_nbin (int &nbin) const
- void get_mc_xnpt (double &mc_xnpt) const
- void get_rho_wt (const int &idim, const int &ibin, double &val) const
- void get_xsum_npt_per_idim (vector< int > &xsum_npt_per_idim) const
- void get_npt_per_dim_bin (Matrix< int > &npt_per_dim_bin) const
- void update_avg (const double &xnum, const double &xval, double &avg)
- void update_RTV_avg (const double &kernel_ft_1, const double &kernel_ft_2, const double &kernel_f0_1, const double &kernel_f0_2, const vector< int > &tvec_binidx)
- void mc_calc_integral_and_error (const UserParam &myUser, const MCParam &myMCParam)

Private Member Functions

- RTVariables (const RTVariables &rhs)
- RTVariables & operator= (const RTVariables &rhs)
- void calc_idim_xsum_tgt_npt (const int &idim)

Private Attributes

- int ndim_
- int nbin_
- double vol_
- double integral_value_ft_
- double integral_error_ft_
- double integral_value_f0_
- double integral_error_f0_
- double mc_xnpt_
- double avg_ft_
- double avg_ftsq_
- double avg_f0_
- double avg_f0sq_
- double variance_ft_
- double variance_f0_
- double sigma_ft_
- double sigma_f0_
- double tot_sampl_npt_
- vector< int > seg_idim_xsum_tgt_npt_
- Matrix< int > seg_tgt_npt_
- Matrix< double > seg_xnpt_
- Matrix< double > seg_avg_ft_
- Matrix< double > seg_avg_ftsq_
- Matrix< double > seg_sigma_
- Matrix< double > seg_wt_
- Matrix< double > seg_est_wt_
- Matrix< double > rho_wt_

3.9.1 Detailed Description

Definition at line 14 of file 400_RTVariables.h.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 RTVariables::RTVariables ( )

Definition at line 17 of file 400_RTVariables.cpp.

3.9.2.2 RTVariables::∼RTVariables ( )

Definition at line 22 of file 400_RTVariables.cpp.

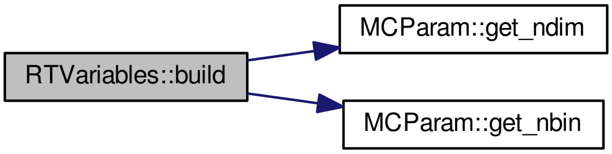
3.9.2.3 RTVariables::RTVariables (const **RTVariables** & *rhs*) `[private]`

3.9.3 Member Function Documentation

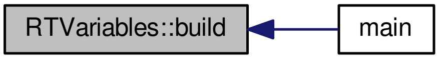
3.9.3.1 void RTVariables::build (const **FileParam** & *myFile,* const UserParam & *myUser,* const **MCParam** & *myMCParam*)

Definition at line 27 of file 400_RTVariables.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



3.9.3.2 void RTVariables::build (const int & *ndim,* const int & *nbin,* const double & *vol,* const **MCParam** & *myMCParam*)

Definition at line 79 of file 400_RTVariables.cpp.

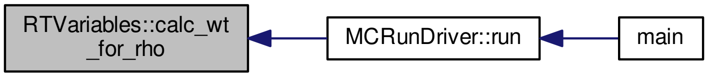
3.9.3.3 void RTVariables::calc_idim_xsum_tgt_npt (const int & *idim*) `[private]`

Definition at line 131 of file 400_RTVariables.cpp.

3.9.3.4 void RTVariables::calc_wt_for_rho (const **MCParam** & *myMCParam*)

Definition at line 143 of file 400_RTVariables.cpp.

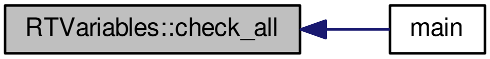
Here is the caller graph for this function:



3.9.3.5 void RTVariables::check_all ( ) const

Definition at line 429 of file 400_RTVariables.cpp.

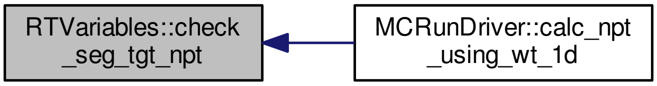
Here is the caller graph for this function:



3.9.3.6 void RTVariables::check_seg_tgt_npt (const int & *idim_max,* const int & *ibin_max,* const int & *xsum_npts,* const int & *mc_tgt_npt*)

Definition at line 188 of file 400_RTVariables.cpp.

Here is the caller graph for this function:



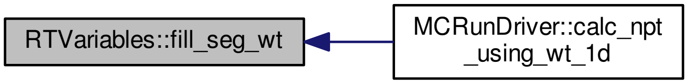
3.9.3.7 void RTVariables::check_seg_tgt_npt (const int & *idim*, const int & *xsum_npts*, const int & *mc_tgt_npt*)

Definition at line 202 of file 400_RTVariables.cpp.

3.9.3.8 void RTVariables::fill_seg_wt (const int & *idim*, const int & *ibin*, const int & *ndim*, const int & *nbin*, const int & *mc_base_pts*, const int & *mc_tgt_npt*, int & *xsum_npts*, int & *idx_max*, double & *max_wt*)

Definition at line 171 of file 400_RTVariables.cpp.

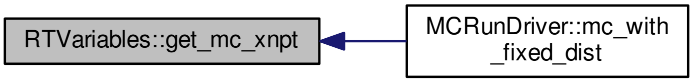
Here is the caller graph for this function:



3.9.3.9 void RTVariables::get_mc_xnpt (double & *mc_xnpt*) const

Definition at line 296 of file 400_RTVariables.cpp.

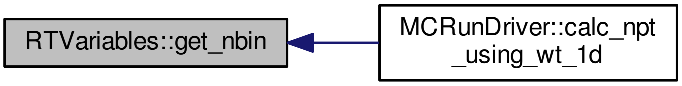
Here is the caller graph for this function:



3.9.3.10 void RTVariables::get_nbin (int & *nbin*) const

Definition at line 291 of file 400_RTVariables.cpp.

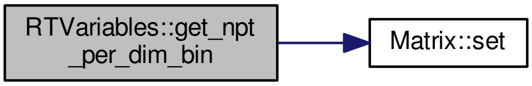
Here is the caller graph for this function:



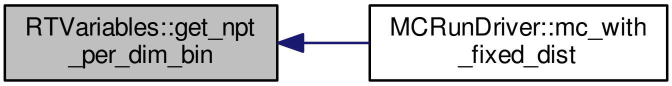
3.9.3.11 void RTVariables::get_npt_per_dim_bin (**Matrix**< int > & *npt_per_dim_bin*) const

Definition at line 318 of file 400_RTVariables.cpp.

Here is the call graph for this function:



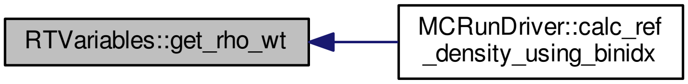
Here is the caller graph for this function:



3.9.3.12 void RTVariables::get_rho_wt (const int & *idim,* const int & *ibin,* double & *val*) const

Definition at line 301 of file 400_RTVariables.cpp.

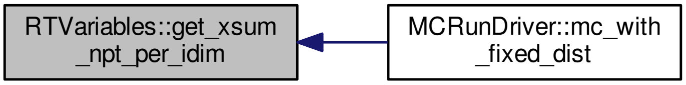
Here is the caller graph for this function:



3.9.3.13 void RTVariables::get_xsum_npt_per_idim (vector< int > & *xsum_npt_per_idim*) const

Definition at line 306 of file 400_RTVariables.cpp.

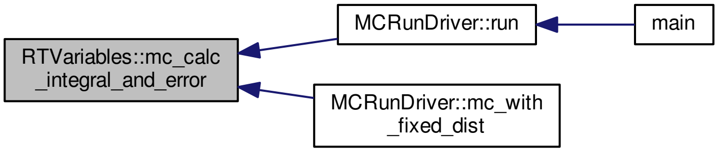
Here is the caller graph for this function:



3.9.3.14 void RTVariables::mc_calc_integral_and_error (const UserParam & *myUser,* const **MCParam** & *myMCParam*)

Definition at line 248 of file 400_RTVariables.cpp.

Here is the caller graph for this function:



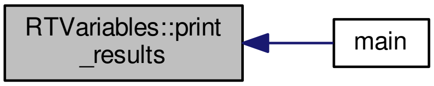
3.9.3.15 void RTVariables::mc_with_fixed_dist (const **MCParam** & *myMCParam,* const UserParam & *myUser*)

3.9.3.16 **RTVariables**& RTVariables::operator= (const **RTVariables** & *rhs*) `[private]`

3.9.3.17 void RTVariables::print_results ( ) const

Definition at line 365 of file 400_RTVariables.cpp.

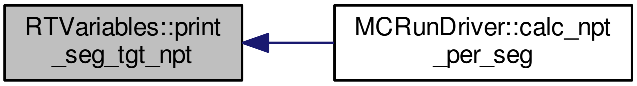
Here is the caller graph for this function:



3.9.3.18 void RTVariables::print_seg_tgt_npt ( ) const

Definition at line 234 of file 400_RTVariables.cpp.

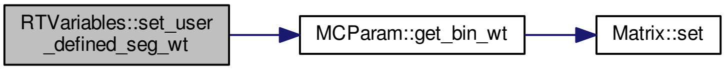
Here is the caller graph for this function:



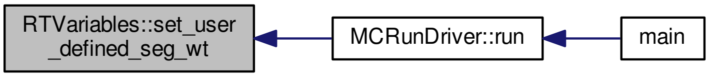
3.9.3.19 void RTVariables::set_user_defined_seg_wt (const **MCParam** & *myMCParam*)

Definition at line 166 of file 400_RTVariables.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



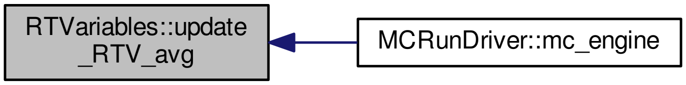
3.9.3.20 void RTVariables::update_avg (const double & *xnum,* const double & *xval,* double & *avg*)

Definition at line 330 of file 400_RTVariables.cpp.

3.9.3.21 void RTVariables::update_RTV_avg (const double & *kernel_ft_1,* const double & *kernel_ft_2,* const double & *kernel_f0_1,* const double & *kernel_f0_2,* const vector< int > & *tvec_binidx*)

Definition at line 339 of file 400_RTVariables.cpp.

Here is the caller graph for this function:



3.9.4 Member Data Documentation

3.9.4.1 double RTVariables::avg_f0_ [private]

Definition at line 76 of file 400_RTVariables.h.

3.9.4.2 double RTVariables::avg_f0sq_ [private]

Definition at line 77 of file 400_RTVariables.h.

3.9.4.3 double RTVariables::avg_ft_ [private]

Definition at line 74 of file 400_RTVariables.h.

3.9.4.4 double RTVariables::avg_ftsq_ [private]

Definition at line 75 of file 400_RTVariables.h.

3.9.4.5 double RTVariables::integral_error_f0_ [private]

Definition at line 72 of file 400_RTVariables.h.

3.9.4.6 double RTVariables::integral_error_ft_ [private]

Definition at line 70 of file 400_RTVariables.h.

3.9.4.7 double RTVariables::integral_value_f0_ [private]

Definition at line 71 of file 400_RTVariables.h.

3.9.4.8 double RTVariables::integral_value_ft_ [private]

Definition at line 69 of file 400_RTVariables.h.

3.9.4.9 double RTVariables::mc_xnpt_ [private]

Definition at line 73 of file 400_RTVariables.h.

3.9.4.10 int RTVariables::nbin_ [private]

Definition at line 67 of file 400_RTVariables.h.

3.9.4.11 int RTVariables::ndim_ [private]

Definition at line 66 of file 400_RTVariables.h.

3.9.4.12 Matrix<double> RTVariables::rho_wt_ [private]

Definition at line 93 of file 400_RTVariables.h.

3.9.4.13 Matrix<double> RTVariables::seg_avg_ft_ [private]

Definition at line 88 of file 400_RTVariables.h.

3.9.4.14 Matrix<double> RTVariables::seg_avg_ftsq_ [private]

Definition at line 89 of file 400_RTVariables.h.

3.9.4.15 Matrix<double> RTVariables::seg_est_wt_ [private]

Definition at line 92 of file 400_RTVariables.h.

3.9.4.16 vector<int> RTVariables::seg_idim_xsum_tgt_npt_ [private]

Definition at line 84 of file 400_RTVariables.h.

3.9.4.17 Matrix<double> RTVariables::seg_sigma_ [private]

Definition at line 90 of file 400_RTVariables.h.

3.9.4.18 Matrix<int> RTVariables::seg_tgt_npt_ [private]

Definition at line 86 of file 400_RTVariables.h.

3.9.4.19 Matrix<double> RTVariables::seg_wt_ [private]

Definition at line 91 of file 400_RTVariables.h.

3.9.4.20 Matrix<double> RTVariables::seg_xnpt_ [private]

Definition at line 87 of file 400_RTVariables.h.

3.9.4.21 double RTVariables::sigma_f0_ [private]

Definition at line 81 of file 400_RTVariables.h.

3.9.4.22 double RTVariables::sigma_ft_ `[private]`

Definition at line 80 of file 400_RTVariables.h.

3.9.4.23 double RTVariables::tot_sampl_npt_ `[private]`

Definition at line 82 of file 400_RTVariables.h.

3.9.4.24 double RTVariables::variance_f0_ `[private]`

Definition at line 79 of file 400_RTVariables.h.

3.9.4.25 double RTVariables::variance_ft_ `[private]`

Definition at line 78 of file 400_RTVariables.h.

3.9.4.26 double RTVariables::vol_ `[private]`

Definition at line 68 of file 400_RTVariables.h.

The documentation for this class was generated from the following files:

- 400_RTVariables.h
- 400_RTVariables.cpp

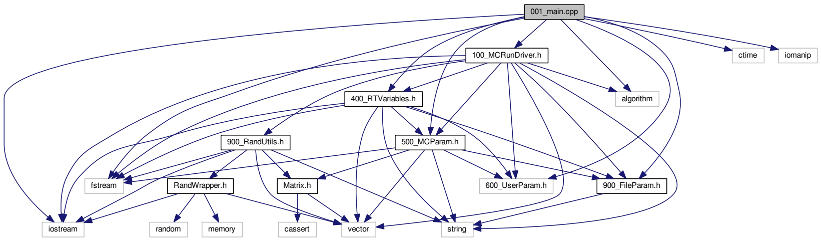
Chapter 4

File Documentation

4.1 001_main.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include <fstream>
#include <ctime>
#include <iomanip>
#include "900_FileParam.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
#include "400_RTVariables.h"
#include "100_MCRunDriver.h"
```

Include dependency graph for 001_main.cpp:



Functions

- int main ()

4.1.1 Function Documentation

4.1.1.1 int main ( )

Step1: Declaring all objects—// file param contains list of all files used user param wraps user defined functions mc-param has all run-time param for performing mc calc intermediate and final results are stored in run-time-variables (RTV) driver is the global object for running the entire mc calc declare as empty objects... we will build them in the next step.

Step2: build all objects—// first get file names, then get data from those files always use check_all to verify consistency information flow: file –> user_param –> mcrun_param –> RunTimeVariable –> driver

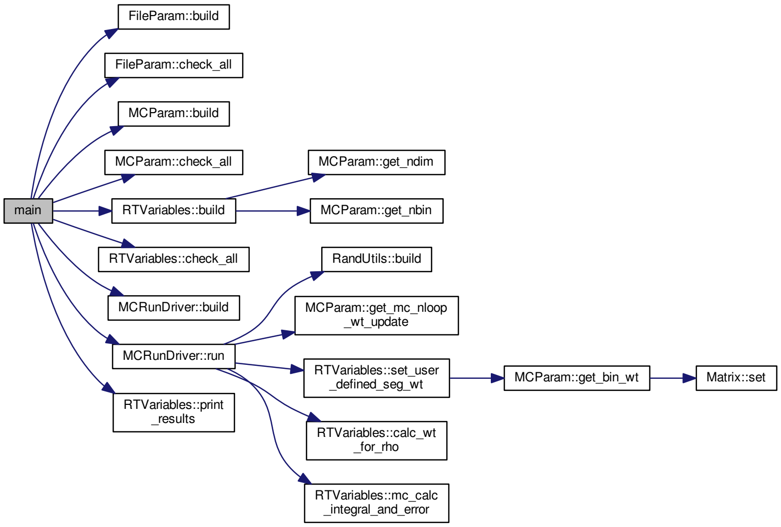
Step3: DO MC calculations-—//

Step4: print results

Step5: print time info and do normal termination

Definition at line 14 of file 001_main.cpp.

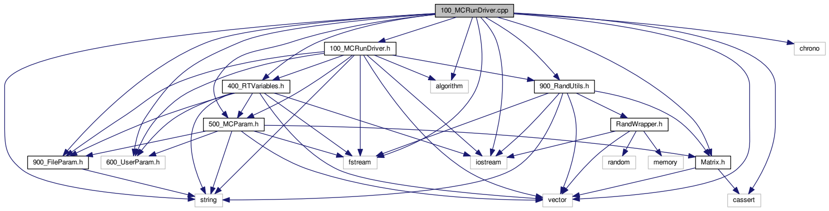
Here is the call graph for this function:



4.2 100_MCRunDriver.cpp File Reference

```
#include <iostream>
#include <cassert>
#include <string>
#include <fstream>
#include <algorithm>
#include <vector>
#include <chrono>
#include "Matrix.h"
#include "900_FileParam.h"
#include "900_RandUtils.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
#include "400_RTVariables.h"
#include "100_MCRunDriver.h"
```

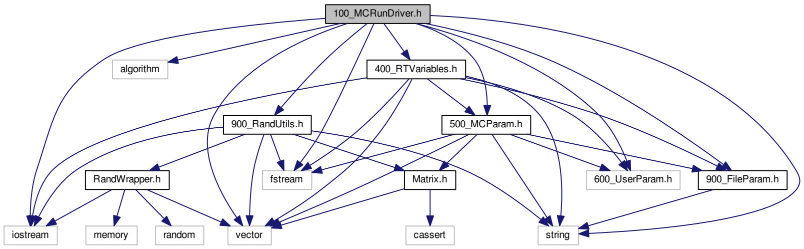
Include dependency graph for 100_MCRunDriver.cpp:



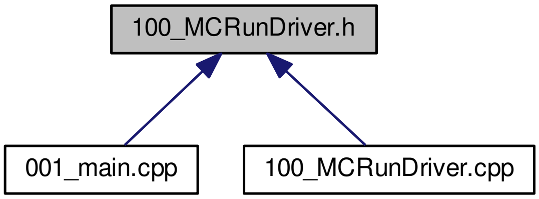
4.3 100_MCRunDriver.h File Reference

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <fstream>
#include "900_FileParam.h"
#include "900_RandUtils.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
#include "400_RTVariables.h"
```

Include dependency graph for 100_MCRunDriver.h:



This graph shows which files directly or indirectly include this file:



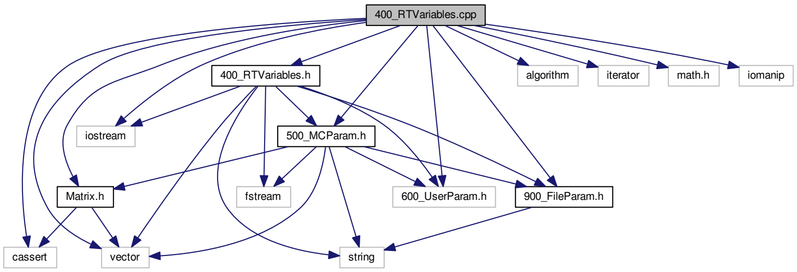
Classes

- class MCRunDriver

4.4 400_RTVariables.cpp File Reference

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator>
#include <math.h>
#include <cassert>
#include <iomanip>
#include "Matrix.h"
#include "900_FileParam.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
#include "400_RTVariables.h"
```

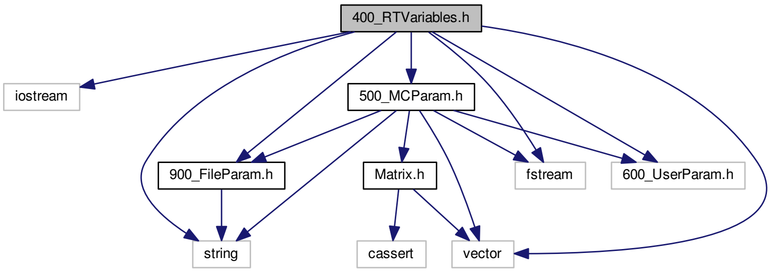
Include dependency graph for 400_RTVariables.cpp:



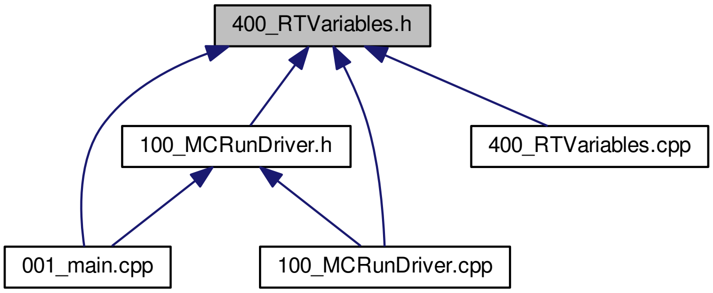
4.5 400_RTVariables.h File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include "900_FileParam.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
```

Include dependency graph for 400_RTVariables.h:



This graph shows which files directly or indirectly include this file:



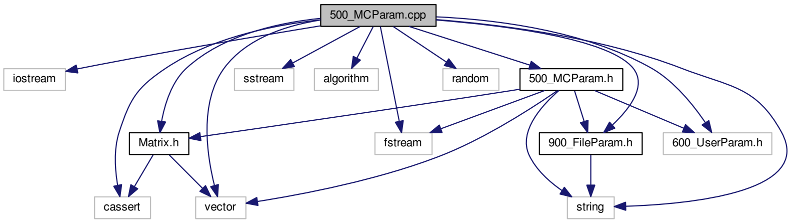
Classes

- class RTVariables

4.6 500_MCParam.cpp File Reference

```
#include <iostream>
#include <cassert>
#include <string>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <vector>
#include <random>
#include "Matrix.h"
#include "900_FileParam.h"
#include "600_UserParam.h"
#include "500_MCParam.h"
```

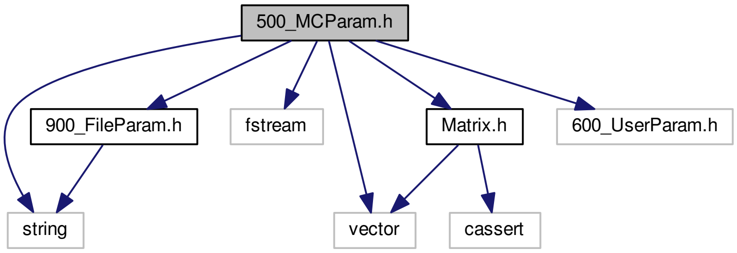
Include dependency graph for 500_MCParam.cpp:



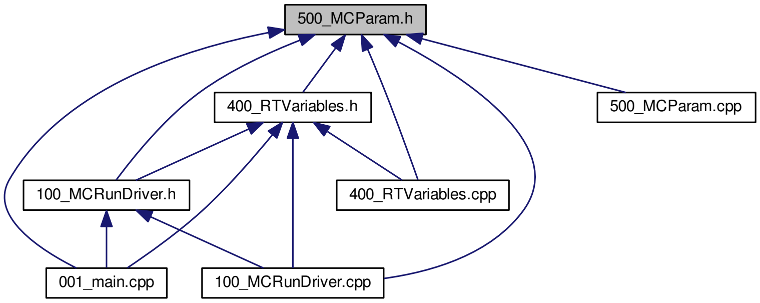
4.7 500_MCParam.h File Reference

```
#include <vector>
#include <string>
#include <fstream>
#include "Matrix.h"
#include "900_FileParam.h"
#include "600_UserParam.h"
```

Include dependency graph for 500_MCParam.h:



This graph shows which files directly or indirectly include this file:



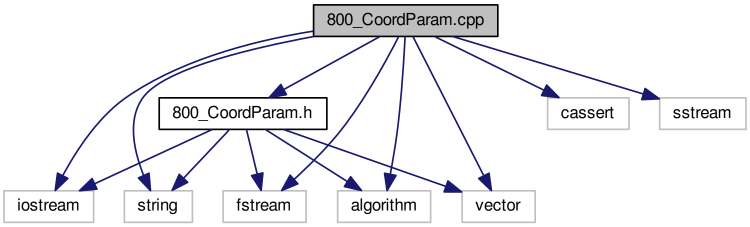
Classes

- class MCParam

4.8 800_CoordParam.cpp File Reference

```
#include <iostream>
#include <cassert>
#include <string>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <vector>
#include "800_CoordParam.h"
```

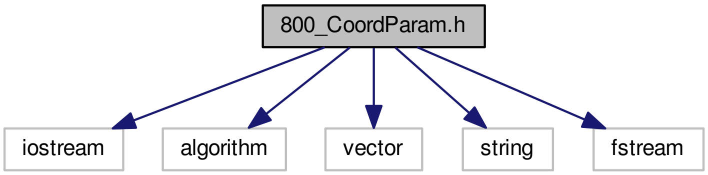
Include dependency graph for 800_CoordParam.cpp:



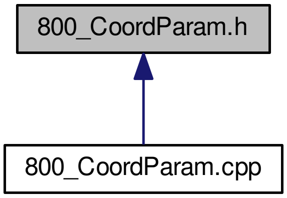
4.9 800_CoordParam.h File Reference

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <fstream>
```

Include dependency graph for 800_CoordParam.h:



This graph shows which files directly or indirectly include this file:



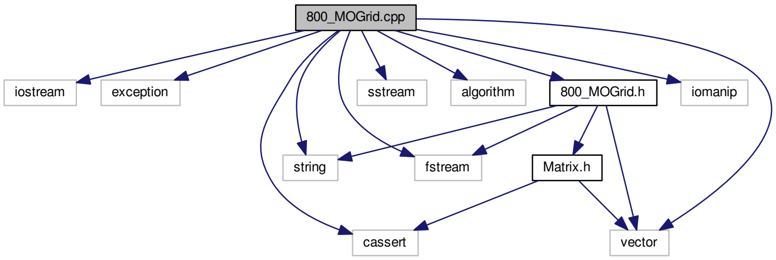
Classes

- class CoordParam

4.10 800_MOGrid.cpp File Reference

```
#include <iostream>
#include <exception>
#include <cassert>
#include <string>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <vector>
#include <iomanip>
#include "800_MOGrid.h"
```

Include dependency graph for 800_MOGrid.cpp:



Functions

- void col_major_idx_3d (const int &n1, const int &n2, const int &n3, const int &N1, const int &N2, const int &N3, int &index)
- void row_major_idx_3d (const int &n1, const int &n2, const int &n3, const int &N1, const int &N2, const int &N3, int &index)

4.10.1 Function Documentation

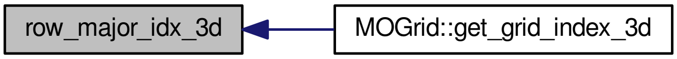
4.10.1.1 void col_major_idx_3d (const int & *n1*, const int & *n2*, const int & *n3*, const int & *N1*, const int & *N2*, const int & *N3*, int & *index*)

Definition at line 21 of file 999_col_major_idx.cpp.

4.10.1.2 void row_major_idx_3d (const int & *n1*, const int & *n2*, const int & *n3*, const int & *N1*, const int & *N2*, const int & *N3*, int & *index*)

Definition at line 39 of file 999_col_major_idx.cpp.

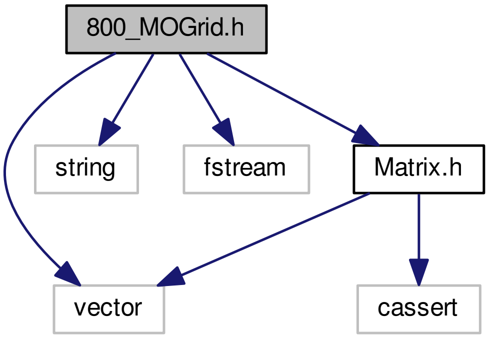
Here is the caller graph for this function:



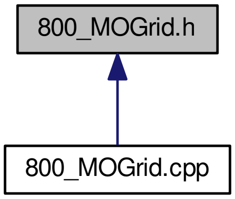
4.11 800_MOGrid.h File Reference

```
#include <vector>
#include <string>
#include <fstream>
#include "Matrix.h"
```

Include dependency graph for 800_MOGrid.h:



This graph shows which files directly or indirectly include this file:



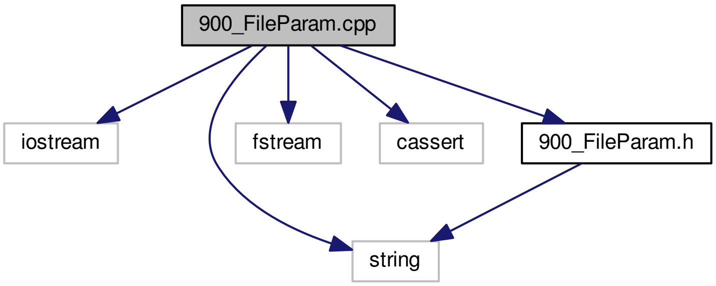
Classes

- class MOGrid

4.12 900_FileParam.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <cassert>
#include "900_FileParam.h"
```

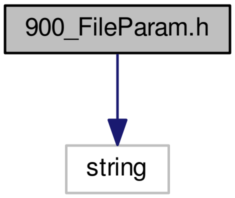
Include dependency graph for 900_FileParam.cpp:



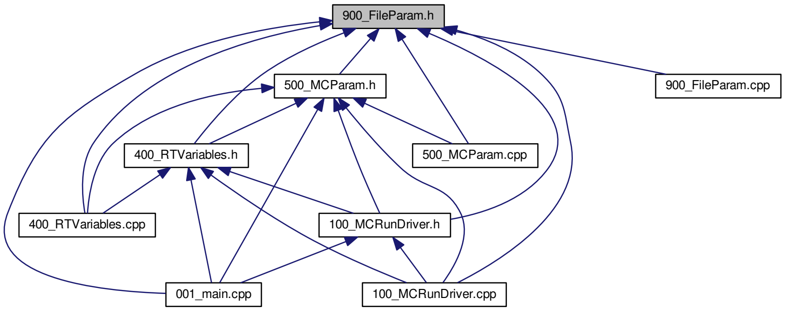
4.13 900_FileParam.h File Reference

```
#include <string>
```

Include dependency graph for 900_FileParam.h:



This graph shows which files directly or indirectly include this file:



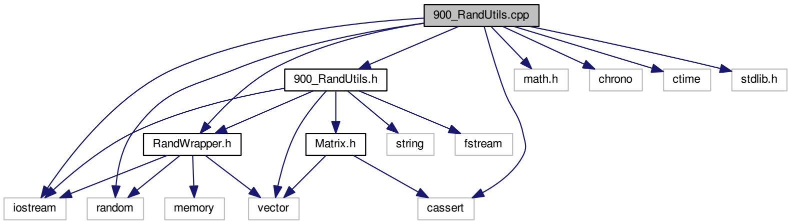
Classes

- class FileParam

4.14 900_RandUtils.cpp File Reference

```
#include <iostream>
#include <math.h>
#include <chrono>
#include <ctime>
#include <cassert>
#include <random>
#include <stdlib.h>
#include "RandWrapper.h"
#include "900_RandUtils.h"
```

Include dependency graph for 900_RandUtils.cpp:



Functions

- void col_major_idx_2d (const int &irow, const int &icol, const int &nrow, const int &ncol, int &index)

4.14.1 Function Documentation

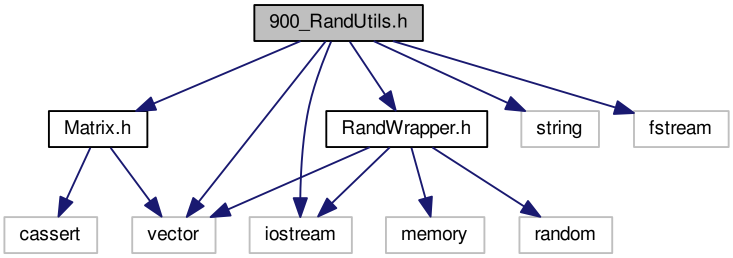
4.14.1.1 void col_major_idx_2d (const int & *irow,* const int & *icol,* const int & *nrow,* const int & *ncol,* int & *index*)

Definition at line 3 of file 999_col_major_idx.cpp.

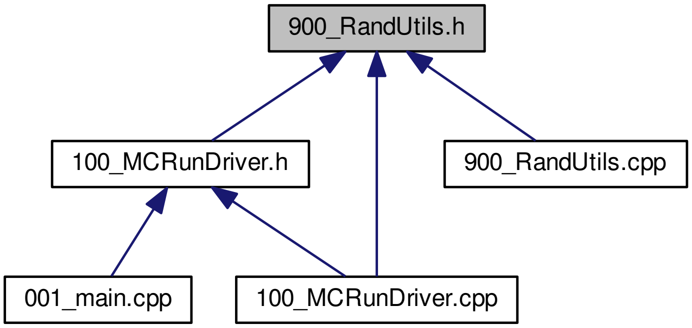
4.15 900_RandUtils.h File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include "Matrix.h"
#include "RandWrapper.h"
```

Include dependency graph for 900_RandUtils.h:



This graph shows which files directly or indirectly include this file:



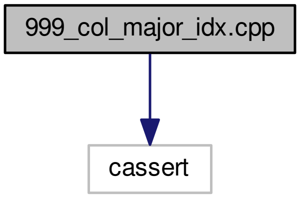
Classes

- class RandUtils

4.16 999_col_major_idx.cpp File Reference

`#include <cassert>`

Include dependency graph for 999_col_major_idx.cpp:



Functions

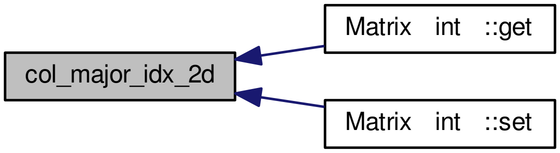
- void col_major_idx_2d (const int &irow, const int &icol, const int &nrow, const int &ncol, int &index)
- void col_major_idx_3d (const int &n1, const int &n2, const int &n3, const int &N1, const int &N2, const int &N3, int &index)
- void row_major_idx_3d (const int &n1, const int &n2, const int &n3, const int &N1, const int &N2, const int &N3, int &index)

4.16.1 Function Documentation

4.16.1.1 void col_major_idx_2d (const int & *irow,* const int & *icol,* const int & *nrow,* const int & *ncol,* int & *index*)

Definition at line 3 of file 999_col_major_idx.cpp.

Here is the caller graph for this function:



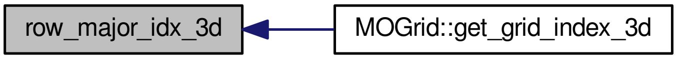
4.16.1.2 void col_major_idx_3d (const int & *n1,* const int & *n2,* const int & *n3,* const int & *N1,* const int & *N2,* const int & *N3,* int & *index*)

Definition at line 21 of file 999_col_major_idx.cpp.

4.16.1.3 void row_major_idx_3d (const int & *n1,* const int & *n2,* const int & *n3,* const int & *N1,* const int & *N2,* const int & *N3,* int & *index*)

Definition at line 39 of file 999_col_major_idx.cpp.

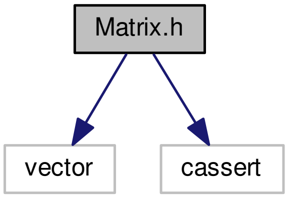
Here is the caller graph for this function:



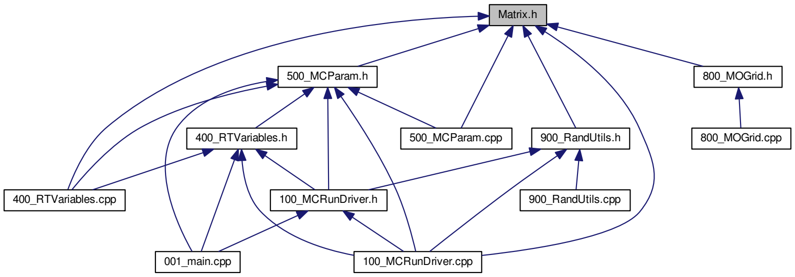
4.17 Matrix.h File Reference

```
#include <vector>
#include <cassert>
```

Include dependency graph for Matrix.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Matrix< T >

Functions

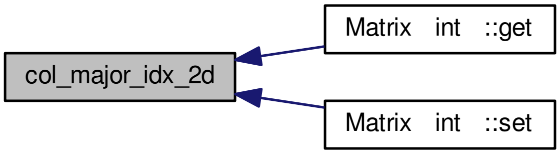
- void col_major_idx_2d (const int &irow, const int &icol, const int &nrow, const int &ncol, int &idx)

4.17.1 Function Documentation

4.17.1.1 void col_major_idx_2d (const int & *irow*, const int & *icol*, const int & *nrow*, const int & *ncol*, int & *idx*)

Definition at line 3 of file 999_col_major_idx.cpp.

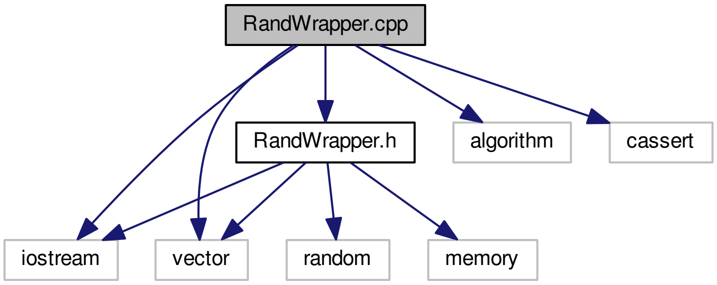
Here is the caller graph for this function:



4.18 RandWrapper.cpp File Reference

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cassert>
#include "RandWrapper.h"
```

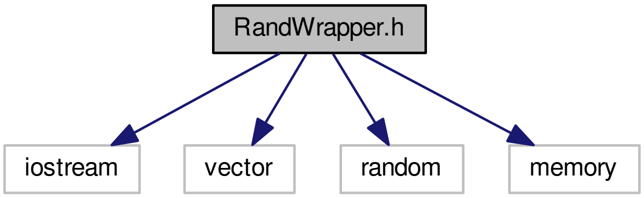
Include dependency graph for RandWrapper.cpp:



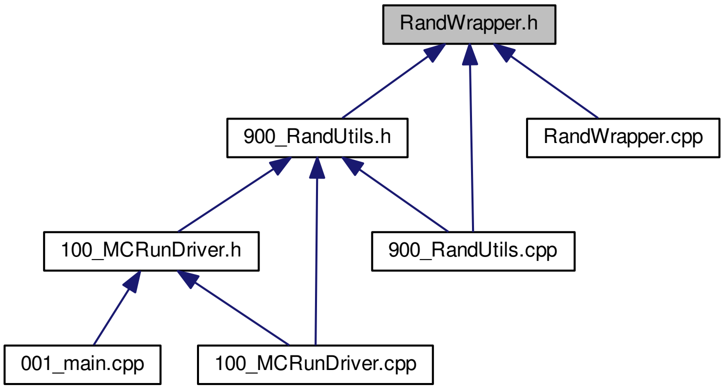
4.19 RandWrapper.h File Reference

```
#include <iostream>
#include <vector>
#include <random>
#include <memory>
```

Include dependency graph for RandWrapper.h:



This graph shows which files directly or indirectly include this file:



Classes

- class RandWrapper

Bibliography

[1] Isaiah; Bartlett Rodney J. Shavitt. *Many-Body Methods in Chemistry and Physics*. 2009.

[2] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, Inc., Mineola, first edition, 1996.

[3] A. Dreuw and M. Head-Gordon. Single-reference ab initio methods for the calculation of excited states of large molecules. *Chemical Reviews*, 105(11):4009–4037, 2005.

[4] A.L. Fetter and J.D. Walecka. *Quantum Theory of Many-Particle Systems*, 1971.

[5] M.E. Casida and M. Huix-Rotllant. Progress in time-dependent density-functional theory. *Annual Review of Physical Chemistry*, 63:287–323, 2012.

[6] P. Ring and P. Schuck. *The Nuclear Many-Body Problem*, 1980.

[7] X. Blase, I. Duchemin, and D. Jacquemin. The bethe-salpeter equation in chemistry: Relations with td-dft, applications and challenges. *Chemical Society Reviews*, 47(3):1022–1043, 2018.

[8] I. Duchemin, C.A. Guido, D. Jacquemin, and X. Blase. The bethe-salpeter formalism with polarisable continuum embedding: Reconciling linear-response and state-specific features. *Chemical Science*, 9(19):4430–4443, 2018.

[9] G. Onida, L. Reining, and A. Rubio. Electronic excitations: Density-functional versus many-body green's-function approaches. *Reviews of Modern Physics*, 74(2):601–659, 2002.

[10] D. Jacquemin, I. Duchemin, and X. Blase. Assessment of the convergence of partially self-consistent bse/ gw calculations. *Molecular Physics*, 114(7-8):957–967, 2016.

[11] R.J. Bartlett and D.S. Ranasinghe. The power of exact conditions in electronic structure theory. *Chemical Physics Letters*, 669:54–70, 2017.

[12] K. Sneskov and O. Christiansen. Excited state coupled cluster methods. *Wiley Interdisciplinary Reviews*, 2(4):566–584, 2012.

[13] C. Song and T.J. Martínez. Reduced scaling caspt2 using supporting subspaces and tensor hyper-contraction. *Journal of Chemical Physics*, 149(4), 2018.

[14] F. Neese, A. Hansen, and D.G. Liakos. Efficient and accurate approximations to the local coupled cluster singles doubles method using a truncated pair natural orbital basis. *Journal of Chemical Physics*, 131(6), 2009.

[15] H. Lischka, D. Nachtigallová, A.J.A. Aquino, P.G. Szalay, F. Plasser, F.B.C. MacHado, and M. Barbatti. Multireference approaches for excited states of molecules. *Chemical Reviews*, 118(15):7293–7361, 2018.

[16] C.J. Stein and M. Reiher. Automated selection of active orbital spaces. *Journal of Chemical Theory and Computation*, 12(4):1760–1771, 2016.

[17] P. Pulay. A perspective on the caspt2 method. *International Journal of Quantum Chemistry*, 111(13):3273–3279, 2011.

[18] D. Roca-Sanjuán, F. Aquilante, and R. Lindh. Multiconfiguration second-order perturbation theory approach to strong electron correlation in chemistry and photochemistry. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2(4):585–603, 2012.

[19] F. Neese, A. Hansen, F. Wennmohs, and S. Grimme. Accurate theoretical chemistry with coupled pair models. *Accounts of Chemical Research*, 42(5):641–648, 2009.

[20] F. Neese, F. Wennmohs, and A. Hansen. Efficient and accurate local approximations to coupled-electron pair approaches: An attempt to revive the pair natural orbital method. *Journal of Chemical Physics*, 130(11), 2009.

[21] S. Ghosh, P. Verma, C.J. Cramer, L. Gagliardi, and D.G. Truhlar. Combining wave function methods with density functional theory for excited states. *Chemical Reviews*, 118(15):7249–7292, 2018.

[22] G.H. Booth and A. Alavi. Approaching chemical accuracy using full configuration-interaction quantum monte carlo: A study of ionization potentials. *J. Chem. Phys.*, 132(17), 2010.

[23] J.F. Stanton and R.J. Bartlett. The equation of motion coupled-cluster method. a systematic biorthogonal approach to molecular excitation energies, transition probabilities, and excited state properties. *The J. Chem. Phys.*, 98(9):7029–7039, 1993.

[24] C. Wagner, J. Schuster, and A. Schleife. Strain and screening: Optical properties of a small-diameter carbon nanotube from first principles. *Phys. Rev. B*, 99(7), 2019.

[25] S. Sharifzadeh. Many-body perturbation theory for understanding optical excitations in organic molecules and solids. *Journal of Physics Condensed Matter*, 30(15), 2018.

[26] M. Govoni and G. Galli. Large scale gw calculations. *J. Chem. Theory Comput.*, 11(6):2680–2696, 2015.

[27] M.L. Tiago and J.R. Chelikowsky. First-principles gw-bse excitations in organic molecules. *Solid State Communications*, 136(6):333–337, 2005.

[28] D. Wing, J.B. Haber, R. Noff, B. Barker, D.A. Egger, A. Ramasubramaniam, S.G. Louie, J.B. Neaton, and L. Kronik. Comparing time-dependent density functional theory with many-body perturbation theory for semiconductors: Screened range-separated hybrids and the gw plus bethe-salpeter approach. *Physical Review Materials*, 3(6), 2019.

[29] Z.-H. Yang, A. Pribram-Jones, K. Burke, and C.A. Ullrich. Direct extraction of excitation energies from ensemble density-functional theory. *Phys. Rev. Lett.*, 119(3), 2017.

[30] C.A. Ullrich and Z.-H. Yang. Excitons in time-dependent density-functional theory. *Topics in Current Chemistry*, 368(3):185–218, 2016.

[31] R.E. Stratmann, G.E. Scuseria, and M.J. Frisch. An efficient implementation of time-dependent density-functional theory for the calculation of excitation energies of large molecules. *J. Chem. Phys.*, 109(19):8218–8224, 1998.

[32] C. Riplinger and F. Neese. An efficient and near linear scaling pair natural orbital based local coupled cluster method. *J. Chem. Phys.*, 138(3), 2013.

[33] F. Neese, F. Wennmohs, and A. Hansen. Efficient and accurate local approximations to coupled-electron pair approaches: An attempt to revive the pair natural orbital method. *J. Chem. Phys.*, 130(11), 2009.

[34] J.M. Elward, J. Hoffman, and A. Chakraborty. Investigation of electron-hole correlation using explicitly correlated configuration interaction method. *Chemical Physics Letters*, 535:182–186, 2012.

[35] J.M. Elward, B. Thallinger, and A. Chakraborty. Calculation of electron-hole recombination probability using explicitly correlated hartree-fock method. *J. Chem. Phys.*, 136(12):182–186, 2012.

[36] J.M. Elward and A. Chakraborty. Effect of heterojunction on exciton binding energy and electron-hole recombination probability in cdse/zns quantum dots. *J. Chem. Theory Comput.*, 11(2):462–471, 2015.

[37] M.G. Bayne, J.A. Scher, B.H. Ellis, and A. Chakraborty. Linked-cluster formulation of electron-hole interaction kernel in real-space representation without using unoccupied states. *J. Chem. Theory Comput.*, 14(7):3656–3666, 2018.

[38] H.-V. Nguyen, T.A. Pham, D. Rocca, and G. Galli. Improving accuracy and efficiency of calculations of photoemission spectra within the many-body perturbation theory. *Phys. Rev. B*, 85(8), 2012.

[39] B. Peng and K. Kowalski. Highly efficient and scalable compound decomposition of two-electron integral tensor and its application in coupled cluster calculations. *J. Chem. Theory Comput.*, 13(9):4179–4192, 2017.

[40] D. Neuhauser, Y. Gao, C. Arntsen, C. Karshenas, E. Rabani, and R. Baer. Breaking the theoretical scaling limit for predicting quasiparticle energies: The stochastic gw approach. *Phys. Rev. Lett.*, 113(7), 2014.

[41] E. Rabani, R. Baer, and D. Neuhauser. Time-dependent stochastic bethe-salpeter approach. *Phys. Rev. B*, 91(23), 2015.

[42] D. Neuhauser, E. Rabani, and R. Baer. Expeditious stochastic approach for mp2 energies in large electronic systems. *J. Chem. Theory Comput.*, 9(1):24–27, 2013.

[43] Y. Gao, D. Neuhauser, R. Baer, and E. Rabani. Sublinear scaling for time-dependent stochastic density functional theory. *J. Chem. Phys.*, 142(3), 2015.

[44] V. Vlček, R. Baer, and D. Neuhauser. Stochastic time-dependent dft with optimally tuned range-separated hybrids: Application to excitonic effects in large phosphorene sheets. *J. Chem. Phys.*, 150(18), 2019.

[45] M.G. Bayne, Y. Uchida, J. Eller, C. Daniels, and A. Chakraborty. Construction of explicitly correlated geminal-projected particle-hole creation operators for many-electron systems using the diagrammatic factorization approach. *Phys. Rev. A*, 94(5), 2016.

[46] G. Gidofalvi and D.A. Mazziotti. Active-space two-electron reduced-density-matrix method: Complete active-space calculations without diagonalization of the n -electron hamiltonian. *J. Chem. Phys.*, 129(13), 2008.

[47] B. Nichols and V.A. Rassolov. Description of electronic excited states using electron correlation operator. *J. Chem. Phys.*, 139(10), 2013.

[48] S.A. Varganov and T.J. Martínez. Variational geminal-augmented multireference self-consistent field theory: Two-electron systems. *J. Chem. Phys.*, 132(5):69–92, 2010.

[49] P.-O. Löwdin. Partitioning technique, perturbation theory, and rational approximations. *International Journal of Quantum Chemistry*, 21(1):69–92, 1982.

[50] X. Zhu, M.S. Hybertsen, and P. Littlewood. Electron-hole system revisited: A variational quantum monte carlo study. *Physical Review B - Condensed Matter and Materials Physics*, 54(19):13575–13580, 1996.

[51] Woggon Ulrike. *Optical Properties of Semiconductor Quantum Dots.* Springer, Berlin, Heidelberg, 1996.

[52] R.D. Mattuck. *A Guide to Feynman Diagrams in the Many-Body Problem.* Dover Publications, 1976.

[53] B.H. Ellis, S. Aggarwal, and A. Chakraborty. Development of the multicomponent coupled-cluster theory for investigation of multiexcitonic interactions. *Journal of Chemical Theory and Computation*, 12(1):188–200, 2016.

[54] B.H. Ellis and A. Chakraborty. Investigation of many-body correlation in biexcitonic systems using electron-hole multicomponent coupled-cluster theory. *Journal of Physical Chemistry C*, 121(2):1291–1298, 2017.

[55] Solomon Kullback. *Information Theory and Statistics.* Dover Publications, 1968.

[56] L. Kong, F.A. Bischoff, and E.F. Valeev. Explicitly correlated r12/f12 methods for electronic structure. *Chemical Reviews*, 112(1):75–107, 2012.

[57] A. Sirjoosingh, M.V. Pak, K.R. Brorsen, and S. Hammes-Schiffer. Quantum treatment of protons with the reduced explicitly correlated hartree-fock approach. *Journal of Chemical Physics*, 142(21), 2015.

[58] A. Chakraborty, M.V. Pak, and S. Hammes-Schiffer. Inclusion of explicit electron-proton correlation in the nuclear-electronic orbital approach using gaussian-type geminal functions. *Journal of Chemical Physics*, 129(1), 2008.

[59] Michael G. Bayne and Arindam Chakraborty. Development of composite control-variate stratified sampling approach for efficient stochastic calculation of molecular integrals. *arXiv e-prints*, page arXiv:1804.01197, Apr 2018.

[60] Y. Shao, Z. Gan, E. Epifanovsky, A.T.B. Gilbert, M. Wormit, J. Kussmann, A.W. Lange, A.Behn, J. Deng, and *et. al*. Advances in molecular quantum chemistry contained in the q-chem 4 program package. *Molecular Physics*, 113(2):184–215, 2015.

[61] J. Schirmer. Beyond the random-phase approximation: A new approximation scheme for the polarization propagator. *Physical Review A*, 26(5):2395–2416, 1982.

[62] M. Wormit, D.R. Rehn, P.H.P. Harbach, J. Wenzel, C.M. Krauter, E. Epifanovsky, and A. Dreuw. Investigating excited electronic states using the algebraic diagrammatic construction (adc) approach of the polarisation propagator. *Molecular Physics*, 112(5-6):774–784, 2014.

[63] Y. Noguchi, O. Sugino, M. Nagaoka, S. Ishii, and K. Ohno. A gwbethe-salpeter calculation on photoabsorption spectra of (cdse) 3 and (cdse) 6 clusters. *Journal of Chemical Physics*, 137(2), 2012.

[64] N.T. Maitra. Undoing static correlation: Long-range charge transfer in time-dependent density-functional theory. *J. Chem. Phys.*, 122(23), 2005.

[65] N.T. Maitra and D.G. Tempel. Long-range excitations in time-dependent density functional theory. *J. Chem. Phys.*, 125(18), 2006.

[66] Y. Shigeta, K. Hirao, and S. Hirata. Exact-exchange time-dependent density-functional theory with the frequency-dependent kernel. *Phys. Rev. A*, 73(1), 2006.

[67] M. Thiele and S. Kümmel. Frequency dependence of the exact exchange-correlation kernel of time-dependent density-functional theory. *Phys. Rev. Lett.*, 112(8), 2014.

[68] M.A.L. Marques, N.T. Maitra, F.M.S. Nogueira, E.K.U. Gross, and A. Rubio. *Fundamentals of Time-Dependent Density Functional Theory*. Lecture Notes in Physics. Springer Berlin Heidelberg, 2012.

[69] A.J. Haes, S. Zou, G.C. Schatz, and R.P. Van Duyne. Nanoscale optical biosensor: Short range distance dependence of the localized surface plasmon resonance of noble metal nanoparticles. *Journal of Physical Chemistry B*, 108(22):6961–6968, 2004.

[70] A.O. Govorov and H.H. Richardson. Generating heat with metal nanoparticles. *Nano Today*, 2(1):30–38, 2007.

[71] P. Tallury, A. Malhotra, L.M. Byrne, and S. Santra. Nanobioimaging and sensing of infectious diseases. *Advanced Drug Delivery Reviews*, 62(4-5):424–437, 2010.

[72] R. Sukirtha, K.M. Priyanka, J.J. Antony, S. Kamalakkannan, R. Thangam, P. Gunasekaran, M. Krishnan, and S. Achiraman. Cytotoxic effect of green synthesized silver nanoparticles using melia azedarach against in vitro hela cell lines and lymphoma mice model. *Process Biochemistry*, 47(2):273–279, 2012.

[73] N.G. Khlebtsov and L.A. Dykman. Optical properties and biomedical applications of plasmonic nanoparticles. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 111(1):1–35, 2010.

[74] P. Kumar, M. Govindaraju, S. Senthamilselvi, and K. Premkumar. Photocatalytic degradation of methyl orange dye using silver (ag) nanoparticles synthesized from ulva lactuca. *Colloids and Surfaces B: Biointerfaces*, 103:658–661, 2013.

[75] F.J. Beck, A. Polman, and K.R. Catchpole. Tunable light trapping for solar cells using localized surface plasmons. *Journal of Applied Physics*, 105(11), 2009.

[76] M.-G. Kang, T. Xu, H.J. Park, X. Luo, and L.J. Guo. Efficiency enhancement of organic solar cells using transparent plasmonic ag nanowire electrodes. *Advanced Materials*, 22(39):4378–4383, 2010.

[77] Y. Xiong, R. Long, D. Liu, X. Zhong, C. Wang, Z.-Y. Li, and Y. Xie. Solar energy conversion with tunable plasmonic nanostructures for thermoelectric devices. *Nanoscale*, 4(15):4416–4420, 2012.

[78] G. Zengin, M. Wersäll, S. Nilsson, T.J. Antosiewicz, M. Käll, and T. Shegai. Realizing strong light-matter interactions between single-nanoparticle plasmons and molecular excitons at ambient conditions. *Physical Review Letters*, 114(15), 2015.

[79] J.L. West and N.J. Halas. Engineered nanomaterials for biophotonics applications: Improving sensing, imaging, and therapeutics. *Annual Review of Biomedical Engineering*, 5:285–292, 2003.

[80] A.D. McFarland and R.P. Van Duyne. Single silver nanoparticles as real-time optical sensors with zeptomole sensitivity. *Nano Letters*, 3(8):1057–1062, 2003.

[81] P.K. Jain, X. Huang, I.H. El-Sayed, and M.A. El-Sayed. Noble metals on the nanoscale: Optical and photothermal properties and some applications in imaging, sensing, biology, and medicine. *Accounts of Chemical Research*, 41(12):1578–1586, 2008.

[82] J. Feng, V.S. Siu, A. Roelke, V. Mehta, S.Y. Rhieu, G.T.R. Palmore, and D. Pacifici. Nanoscale plasmonic interferometers for multispectral, high-throughput biochemical sensing. *Nano Letters*, 12(2):602–609, 2012.

[83] S. Eustis and M.A. El-Sayed. Why gold nanoparticles are more precious than pretty gold: Noble metal surface plasmon resonance and its enhancement of the radiative and nonradia-

tive properties of nanocrystals of different shapes. *Chemical Society Reviews*, 35(3):209–217, 2006.

[84] L.M. Liz-Marzán. Tailoring surface plasmons through the morphology and assembly of metal nanoparticles. *Langmuir*, 22(1):32–41, 2006.

[85] Y. Sonnefraud, A.L. Leen Koh, D.W. McComb, and S.A. Maier. Nanoplasmonics: Engineering and observation of localized plasmon modes. *Laser and Photonics Reviews*, 6(3):277–295, 2012.

[86] N.J. Halas, S. Lal, W.-S. Chang, S. Link, and P. Nordlander. Plasmons in strongly coupled metallic nanostructures. *Chemical Reviews*, 111(6):3913–3961, 2011.

[87] P. Geisler, E. Krauss, G. Razinskas, and B. Hecht. Transmission of plasmons through a nanowire. *ACS Photonics*, 4(7):1615–1620, 2017.

[88] M.W. Knight, N.K. Grady, R. Bardhan, F. Hao, P. Nordlander, and N.J. Halas. Nanoparticle-mediated coupling of light into a nanowire. *Nano Letters*, 7(8):2346–2350, 2007.

[89] P.K. Jain and M.A. El-Sayed. Plasmonic coupling in noble metal nanostructures. *Chemical Physics Letters*, 487(4-6):153–164, 2010.

[90] A. Varas, P. García-González, J. Feist, F.J. García-Vidal, and A. Rubio. Quantum plasmonics: from jellium models to ab initio calculations. *Nanophotonics*, 5(3):409–426, 2016.

[91] X. López Lozano, C. Mottet, and H.-Ch. Weissker. Effect of alloying on the optical properties of ag-au nanoparticles. *Journal of Physical Chemistry C*, 117(6):3062–3068, 2013.

[92] D.C. Marinica, A.K. Kazansky, P. Nordlander, J. Aizpurua, and A.G. Borisov. Quantum plasmonics: Nonlinear effects in the field enhancement of a plasmonic nanoparticle dimer. *Nano Letters*, 12(3):1333–1339, 2012.

[93] M.D. Malinsky, K.L. Kelly, G.C. Schatz, and R.P. Van Duyne. Chain length dependence and sensing capabilities of the localized surface plasmon resonance of silver nanoparticles

chemically modified with alkanethiol self-assembled monolayers. *Journal of the American Chemical Society*, 123(7):1471–1482, 2001.

[94] C.M. Aikens, S. Li, and G.C. Schatz. From discrete electronic states to plasmons: Tddft optical absorption properties of agn (n = 10, 20, 35, 56, 84, 120) tetrahedral clusters. *Journal of Physical Chemistry C*, 112(30):11272–11279, 2008.

[95] K.L. Kelly, E. Coronado, L.L. Zhao, and G.C. Schatz. The optical properties of metal nanoparticles: The influence of size, shape, and dielectric environment. *Journal of Physical Chemistry B*, 107(3):668–677, 2003.

[96] M.B. Ross, C.A. Mirkin, and G.C. Schatz. Optical properties of one-, two-, and three-dimensional arrays of plasmonic nanostructures. *Journal of Physical Chemistry C*, 120(2):816–830, 2016.

[97] M. Shabaninezhad and G. Ramakrishna. Theoretical investigation of size, shape, and aspect ratio effect on the lspr sensitivity of hollow-gold nanoshells. *Journal of Chemical Physics*, 150(14), 2019.

[98] G.-T. Bae and C.M. Aikens. Time-dependent density functional theory studies of optical properties of ag nanoparticles: Octahedra, truncated octahedra, and icosahedra. *Journal of Physical Chemistry C*, 116(18):10356–10367, 2012.

[99] H.E. Johnson and C.M. Aikens. Electronic structure and tddft optical absorption spectra of silver nanorods. *Journal of Physical Chemistry A*, 113(16):4445–4450, 2009.

[100] B. Gao, K. Ruud, and Y. Luo. Plasmon resonances in linear noble-metal chains. *Journal of Chemical Physics*, 137(19), 2012.

[101] P. Scholz, S. Schwieger, B. Ashall, D. Zerulla, and E. Runge. The influence of wire shape on surface plasmon mode distribution. *Applied Physics B: Lasers and Optics*, 93(1):111–115, 2008.

[102] E.B. Guidez and C.M. Aikens. Diameter dependence of the excitation spectra of silver and gold nanorods. *Journal of Physical Chemistry C*, 117(23):12325–12336, 2013.

[103] R.D. Senanayake, D.B. Lingerfelt, G.U. Kuda-Singappulige, X. Li, and C.M. Aikens. Real-time tddft investigation of optical absorption in gold nanowires. *Journal of Physical Chemistry C*, 123(23):14734–14745, 2019.

[104] M.J. Deible, M. Kessler, K.E. Gasperich, and K.D. Jordan. Quantum monte carlo calculation of the binding energy of the beryllium dimer. *Journal of Chemical Physics*, 143(8), 2015.

[105] B.S. Fales, Y. Shu, B.G. Levine, and E.G. Hohenstein. Complete active space configuration interaction from state-averaged configuration interaction singles natural orbitals: Analytic first derivatives and derivative coupling vectors. *Journal of Chemical Physics*, 147(9), 2017.

[106] J.J. Eriksen and J. Gauss. Generalized many-body expanded full configuration interaction theory. *Journal of Physical Chemistry Letters*, 10(24):7910–7915, 2019.

[107] J.J. Eriksen and J. Gauss. Many-body expanded full configuration interaction. ii. strongly correlated regime. *Journal of Chemical Theory and Computation*, 15(9):4873–4884, 2019.

[108] B.S. Fales and T.J. Martínez. Efficient treatment of large active spaces through multi-gpu parallel implementation of direct configuration interaction. *Journal of Chemical Theory and Computation*, 16(3):1586–1596, 2020.

[109] B.S. Fales, E.G. Hohenstein, and B.G. Levine. Robust and efficient spin purification for determinantal configuration interaction. *Journal of Chemical Theory and Computation*, 13(9):4162–4172, 2017.

[110] J. Van De Bovenkamp and F.B. Van Duijneveldt. Mrci calculations on the helium dimer employing an interaction optimized basis set. *Journal of Chemical Physics*, 110(23):11141–11151, 1999.

[111] K.K. Sunil, K.D. Jordan, and R. Shepard. Application of the mc scf method to the $\pi \implies \pi^*$ excitation energies of ethylene. *Chemical Physics*, 88(1):55–64, 1984.

[112] R.E. Thomas, Q. Sun, A. Alavi, and G.H. Booth. Stochastic multiconfigurational self-consistent field theory. *Journal of Chemical Theory and Computation*, 11(11):5316–5325, 2015.

[113] C.M. Aikens and M.S. Gordon. Influence of multi-atom bridging ligands on the electronic structure and magnetic properties of homodinuclear titanium molecules. *Journal of Physical Chemistry A*, 109(51):11885–11901, 2005.

[114] A. Fernando and C.M. Aikens. Ab initio electronic structure study of a model water splitting dimer complex. *Physical Chemistry Chemical Physics*, 17(48):32443–32454, 2015.

[115] B.G. Levine, J.D. Coe, and T.J. Martínez. Optimizing conical intersections without derivative coupling vectors: Application to multistate multireference second-order perturbation theory (ms-caspt2). *Journal of Physical Chemistry B*, 112(2):405–413, 2008.

[116] B.S. Fales, S. Seritan, N.F. Settje, B.G. Levine, H. Koch, and T.J. Martínez. Large-scale electron correlation calculations: Rank-reduced full configuration interaction. *Journal of Chemical Theory and Computation*, 14(8):4139–4150, 2018.

[117] E.B. Guidez and C.M. Aikens. Plasmon resonance analysis with configuration interaction. *Physical Chemistry Chemical Physics*, 16(29):15501–15509, 2014.

[118] F. Alkan and C.M. Aikens. Understanding plasmon coupling in nanoparticle dimers using molecular orbitals and configuration interaction. *Physical Chemistry Chemical Physics*, 21(41):23065–23075, 2019.

[119] R.L. Gieseking, M.A. Ratner, and G.C. Schatz. Semiempirical modeling of ag nanoclusters: New parameters for optical property studies enable determination of double excitation contributions to plasmonic excitation. *Journal of Physical Chemistry A*, 120(26):4542–4549, 2016.

[120] G. Piccini, R.W.A. Havenith, R. Broer, and M. Stener. Gold nanowires: A time-dependent density functional assessment of plasmonic behavior. *Journal of Physical Chemistry C*, 117(33):17196–17204, 2013.

[121] S. Bernadotte, F. Evers, and C.R. Jacob. Plasmons in molecules. *Journal of Physical Chemistry C*, 117(4):1863–1878, 2013.

[122] G. Barcaro, L. Sementa, A. Fortunelli, and M. Stener. Optical properties of silver nanoshells from time-dependent density functional theory calculations. *Journal of Physical Chemistry C*, 118(23):12450–12458, 2014.

[123] O. Baseggio, M. De Vetta, G. Fronzoni, M. Stener, L. Sementa, A. Fortunelli, and A. Calzolari. Photoabsorption of icosahedral noble metal clusters: An efficient tddft approach to large-scale systems. *Journal of Physical Chemistry C*, 120(23):12773–12782, 2016.

[124] B. Gao, K. Ruud, and Y. Luo. Shape-dependent electronic excitations in metallic chains. *Journal of Physical Chemistry C*, 118(24):13059–13069, 2014.

[125] Z. Ma, P. Wang, and Y. Pei. Geometric structure, electronic structure and optical absorption properties of one-dimensional thiolate-protected gold clusters containing a quasi-face-centered-cubic (quasi-fcc) au-core: A density-functional theoretical study. *Nanoscale*, 8(38):17044–17054, 2016.

[126] K. Zhang and H. Zhang. Plasmon coupling in gold nanotube assemblies: Insight from a time-dependent density functional theory (tddft) calculation. *Journal of Physical Chemistry C*, 118(1):635–641, 2014.

[127] K.L.D.M. Weerawardene, H. Häkkinen, and C.M. Aikens. Connections between theory and experiment for gold and silver nanoclusters. *Annual Review of Physical Chemistry*, 69:205–229, 2018.

[128] A. Fernando, K.L.D.M. Weerawardene, N.V. Karimova, and C.M. Aikens. Quantum mechanical studies of large metal, metal oxide, and metal chalcogenide nanoparticles and clusters. *Chemical Reviews*, 115(12):6112–6216, 2015.

[129] N. Durante, A. Fortunelli, M. Broyer, and M. Stener. Optical properties of au nanoclusters from td-dft calculations. *Journal of Physical Chemistry C*, 115(14):6277–6282, 2011.

[130] G. Barcaro, M. Broyer, N. Durante, A. Fortunelli, and M. Stener. Alloying effects on the optical properties of ag-au nanoclusters from tddft calculations. *Journal of Physical Chemistry C*, 115(49):24085–24091, 2011.

[131] M.A. Mosquera, B. Fu, K.L. Kohlstedt, G.C. Schatz, and M.A. Ratner. Wave functions, density functionals, and artificial intelligence for materials and energy research: Future prospects and challenges. *ACS Energy Letters*, 3(1):155–162, 2018.

[132] U. Bozkaya and C.D. Sherrill. Orbital-optimized mp2.5 and its analytic gradients: Approaching ccsd(t) quality for noncovalent interactions. *Journal of Chemical Physics*, 141(20), 2014.

[133] J. Lee and M. Head-Gordon. Regularized orbital-optimized second-order møller-plesset perturbation theory: A reliable fifth-order-scaling electron correlation model with orbital energy dependent regularizers. *Journal of Chemical Theory and Computation*, 14(10):5203–5219, 2018.

[134] F. Pavošević, B.J.G. Rousseau, and S. Hammes-Schiffer. Multicomponent orbital-optimized perturbation theory methods: Approaching coupled cluster accuracy at lower cost. *Journal of Physical Chemistry Letters*, 11(4):1578–1583, 2020.

[135] A.I. Krylov, C.D. Sherrill, E.F.C. Byrd, and M. Head-Gordon. Size-consistent wave functions for nondynamical correlation energy: The valence active space optimized orbital coupled-cluster doubles model. *Journal of Chemical Physics*, 109(24):10669–10678, 1998.

[136] U. Bozkaya, J.M. Turney, Y. Yamaguchi, H.F. Schaefer III, and C.D. Sherrill. Quadratically convergent algorithm for orbital optimization in the orbital-optimized coupled-cluster doubles method and in orbital-optimized second-order mller-plesset perturbation theory. *Journal of Chemical Physics*, 135(10), 2011.

[137] D. Hait and M. Head-Gordon. Excited state orbital optimization via minimizing the square of the gradient: General approach and application to singly and doubly excited states via density functional theory. *Journal of Chemical Theory and Computation*, 16(3):1699–1710, 2020.

[138] H.G.A. Burton and A.J.W. Thom. General approach for multireference ground and excited states using nonorthogonal configuration interaction. *Journal of Chemical Theory and Computation*, 15(9):4851–4861, 2019.

[139] X. Liu, S. Fatehi, Y. Shao, B.S. Veldkamp, and J.E. Subotnik. Communication: Adjusting charge transfer state energies for configuration interaction singles: Without any parameterization and with minimal cost. *Journal of Chemical Physics*, 136(16), 2012.

[140] L.N. Tran, J.A.R. Shea, and E. Neuscamman. Tracking excited states in wave function optimization using density matrices and variational principles. *Journal of Chemical Theory and Computation*, 15(9):4790–4803, 2019.

[141] H.-Z. Ye and T. Van Voorhis. Half-projected σ self-consistent field for electronic excited states. *Journal of Chemical Theory and Computation*, 2019.

[142] L. Zhao and E. Neuscamman. An efficient variational principle for the direct optimization of excited states. *Journal of Chemical Theory and Computation*, 12(8):3436–3440, 2016.

[143] S. Seritan, C. Bannwarth, B.S. Fales, E.G. Hohenstein, S.I.L. Kokkila-Schumacher, N. Luehr, Jr Snyder, J.W., C. Song, A.V. Titov, I.S. Ufimtsev, and T.J. Martínez. Terachem: Accelerating electronic structure and ab initio molecular dynamics with graphical processing units. *The Journal of chemical physics*, 152(22):224110, 2020.

[144] E.B. Guidez and C.M. Aikens. Theoretical analysis of the optical excitation spectra of silver and gold nanowires. *Nanoscale*, 4(14):4190–4198, 2012.

[145] B. Stroustrup. What is object-oriented programming? *IEEE Software*, 5(3):10–20, 1988.

[146] P. Kunwar, J. Hassinen, G. Bautista, R.H.A. Ras, and J. Toivonen. Sub-micron scale patterning of fluorescent silver nanoclusters using low-power laser. *Scientific Reports*, 6, 2016.

[147] P. Kunwar, J. Hassinen, G. Bautista, R.H.A. Ras, and J. Toivonen. Direct laser writing of photostable fluorescent silver nanoclusters in polymer films. *ACS Nano*, 8(11):11165–11171, 2014.

[148] L. A. LaJohn, P. A. Christiansen, R. B. Ross, T. Atashroo, and W. C. Ermler. Ab initio relativistic effective potentials with spin–orbit operators. iii. rb through xe. *J. Chem. Phys.*, 87(5):2812–2824, 1987.

[149] Lin-gun Liu and William A. Bassett. Compression of ag and phase transformation of nacl. *Journal of Applied Physics*, 44(4):1475–1479, 1973.

[150] Yihan Shao, Zhengting Gan, Evgeny Epifanovsky, Andrew T.B. Gilbert, Michael Wormit, Joerg Kussmann, Adrian W. Lange, Andrew Behn, Jia Deng, Xintian Feng, Debashree Ghosh, Matthew Goldey, Paul R. Horn, Leif D. Jacobson, Ilya Kaliman, Rustam Z. Khaliullin, Tomasz Kuś, Arie Landau, Jie Liu, Emil I. Proynov, Young Min Rhee, Ryan M. Richard, Mary A. Rohrdanz, Ryan P. Steele, Eric J. Sundstrom, H. Lee Woodcock III, Paul M. Zimmerman, Dmitry Zuev, Ben Albrecht, Ethan Alguire, Brian Austin, Gregory J. O. Beran, Yves A. Bernard, Eric Berquist, Kai Brandhorst, Ksenia B. Bravaya, Shawn T. Brown, David Casanova, Chun-Min Chang, Yunqing Chen, Siu Hung Chien, Kristina D. Closser, Deborah L. Crittenden, Michael Diedenhofen, Robert A. DiStasio Jr., Hainam Do, Anthony D. Dutoi, Richard G. Edgar, Shervin Fatehi, Laszlo Fusti-Molnar, An Ghysels, Anna Golubeva-Zadorozhnaya, Joseph Gomes, Magnus W.D. Hanson-Heine, Philipp H.P. Harbach, Andreas W. Hauser, Edward G. Hohenstein, Zachary C. Holden,

Thomas-C. Jagau, Hyunjun Ji, Benjamin Kaduk, Kirill Khistyaev, Jaehoon Kim, Jihan Kim, Rollin A. King, Phil Klunzinger, Dmytro Kosenkov, Tim Kowalczyk, Caroline M. Krauter, Ka Un Lao, Adèle D. Laurent, Keith V. Lawler, Sergey V. Levchenko, Ching Yeh Lin, Fenglai Liu, Ester Livshits, Rohini C. Lochan, Arne Luenser, Prashant Manohar, Samuel F. Manzer, Shan-Ping Mao, Narbe Mardirossian, Aleksandr V. Marenich, Simon A. Maurer, Nicholas J. Mayhall, Eric Neuscamman, C. Melania Oana, Roberto Olivares-Amaya, Darragh P. O'Neill, John A. Parkhill, Trilisa M. Perrine, Roberto Peverati, Alexander Prociuk, Dirk R. Rehn, Edina Rosta, Nicholas J. Russ, Shaama M. Sharada, Sandeep Sharma, David W. Small, Alexander Sodt, Tamar Stein, David Stück, Yu-Chuan Su, Alex J.W. Thom, Takashi Tsuchimochi, Vitalii Vanovschi, Leslie Vogt, Oleg Vydrov, Tao Wang, Mark A. Watson, Jan Wenzel, Alec White, Christopher F. Williams, Jun Yang, Sina Yeganeh, Shane R. Yost, Zhi-Qiang You, Igor Ying Zhang, Xing Zhang, Yan Zhao, Bernard R. Brooks, Garnet K.L. Chan, Daniel M. Chipman, Christopher J. Cramer, William A. Goddard III, Mark S. Gordon, Warren J. Hehre, Andreas Klamt, Henry F. Schaefer III, Michael W. Schmidt, C. David Sherrill, Donald G. Truhlar, Arieh Warshel, Xin Xu, Alán Aspuru-Guzik, Roi Baer, Alexis T. Bell, Nicholas A. Besley, Jeng-Da Chai, Andreas Dreuw, Barry D. Dunietz, Thomas R. Furlani, Steven R. Gwaltney, Chao-Ping Hsu, Yousung Jung, Jing Kong, Daniel S. Lambrecht, WanZhen Liang, Christian Ochsenfeld, Vitaly A. Rassolov, Lyudmila V. Slipchenko, Joseph E. Subotnik, Troy Van Voorhis, John M. Herbert, Anna I. Krylov, Peter M.W. Gill, and Martin Head-Gordon. Advances in molecular quantum chemistry contained in the q-chem 4 program package. *Molecular Physics*, 113(2):184–215, 2015.

[151] E. Hofman, R.J. Robinson, Z.-J. Li, B. Dzikovski, and W. Zheng. Controlled dopant migration in cds/zns core/shell quantum dots. *Journal of the American Chemical Society*, 139(26):8878–8885, 2017.

[152] Donald A. McQuarrie. *Statistical Mechanics*. 1975.

Curriculum Vitae

PETER F. MCLAUGHLIN

@ **E-mail:** mclaughlin.peterf@gmail.com linkedin.com/in/peterfmclaughlin

RESEARCH EXPERIENCE

Doctoral Research

Research Advisor: Prof Arindam Chakraborty

Department of Chemistry, Syracuse University, Syracuse, NY

Oct 2018 – Present

- Developed a frequency-dependent explicitly-correlated ansatz for excited state calculations in order to investigate many-electron systems
- Programmed Monte Carlo integration methods with the goal of achieving computationally efficient algorithms
- Investigated thermodynamic properties of dopant migration/ejection within modeled system
- Utilized real-space representation to perform excited state calculations for many-electron systems
- Programmed and developed software for large coding projects
- Experience of using electron-structure packages: Q-CHEM, NWCHEM, TERACHEM, GAUSSIAN

Masters Research

Department of Chemistry, Syracuse University, Syracuse, NY

Aug 2013 – Oct 2018

- Examined the solid state photopolymerization of host-guest inclusion crystals to create isolated long chain polyenes
- Investigated solid state structure and chemical composition utilizing Raman and IR spectroscopy for analysis of newly synthesized polyenes
- Studied electronic absorption of polyenes towards understanding triplet states

Immersive Seminar

Research Advisor: Prof Jason Ribblett

Virginia B. Ball Center for Creative Inquiry, Ball State University, Muncie, IN

Aug 2012 – Dec 2012

- Guided the creation of interactive multimedia textbook for application for the fifth-grade physical science classroom following the Indiana state standards
- *Chemistry Matters*, 1.0.1; Frog Baby Apps LLC: Apple iTunes Store, 2014.
- Regulated the scientific content including text and laboratory experiments

Undergraduate Summer Research

Research Advisor: Prof E. Michael Perdue

Department of Chemistry, Ball State University, Muncie, IN

May 2012 – Aug 2012

- Analyzed the pH dependence for the removal of boric acid using electrodialysis in fresh and saltwater conditions

WORK EXPERIENCE

Laboratory Prep Ball State University, Muncie, IN Aug 2011 – Dec 2012

- Prepared laboratory experiments to be used for chemistry labs for department

Quality Control Quaker Chemical, Crawfordsville, IN May 2010 – Jul 2010

- Performed quality control experiments on liquids and oils used in the industrial process of steel

ACADEMIC HISTORY

PhD in Chemistry

Syracuse University Syracuse, NY

Oct 2018 – Aug 2020

Supervisor: Prof Arindam Chakraborty

- Specialization: Computational Quantum Chemistry

MPhil in Chemistry

Syracuse University Syracuse, NY

Aug 2013 – Apr 2015

- Specialization: Experimental Physical Chemistry

BS in Chemistry

Ball State University Muncie, IN

Aug 2009 – Dec 2012

- *ACS* Certified

PROGRAMMING SKILLS

Python	●●●●●
Fortran	●●●●●
Numpy, Scipy	●●●●●
Linux	●●●●○
C++	●●●○○
Latex	●●●○○
BitBucket	●●●○○
Git	●●○○○

TECHNICAL SKILLS

Object-Oriented Programming

Test-Driven Development

Monte Carlo Integration

Algorithm Development

Quantum and Classical Mechanics

High-Performance Computing

Differential Equations

Linear Algebra

ACHIEVEMENTS

- Eagle Scout
- Science Student of the Month (Ball State University, April 2012)

AFFILIATIONS & MEMBERSHIPS

- American Chemical Society
- American Physical Society
- Phi Mu Alpha Fraternity

PUBLICATIONS

1. **McLaughlin, P. F.**; Chakraborty, A. Compact real-space representation of excited states using frequency-dependent explicitly-correlated electron-hole interaction kernel. *J. Chem. Theory Comput.*, Accepted, 2020.
2. Hofman, E.; Li, Z.; Davis, A.H.; Khammang, A.; Wright, J. T.; **McLaughlin, P.F.**; Franck, J.M.; Chakraborty, A.; Meulenberg, R.W.; Zheng, W. Decoupling and Coupling of the Host-Dopant Interaction by Manipulating Dopant Movement in Core/Shell Quantum Dots. *J. Phys. Chem. Lett.*, Accepted, 2020.
3. **McLaughlin, P. F.**; Chakraborty, A. Construction of an orbital-optimized dressed particle-hole basis resulting in compression of the configruation space for the efficient calculation of electron-correlation in silver nanowires. *J. Chem. Theory Comput.*, In Preparation, 2020.
4. Spanedda, N.; **McLaughlin, P. F.**; Chakraborty, A. Development of Permutation Sampling Monte Carlo method. *J. Chem. Theory Comput.*, In Preparation, 2020.

PRESENTATIONS

- **McLaughlin, Peter F.**; Chakraborty, Arindam. Penn Conference in Theoretical Chemistry, Chemical Dynamics in the Condensed Phase, Philadelphia, PA, United States, August 12-14, 2019.
- **McLaughlin, Peter F.**; Chakraborty, Arindam. The 51st Midwest Theoretical Chemistry Conference, Notre Dame, IN, United States, June 6-8, 2019.
- **McLaughlin, Peter F.**; Bayne, Michael; Chakraborty, Arindam. Abstracts of Papers, 256th ACS National Meeting & Exposition, Boston, MA, United States, August 19-23, 2018; PHYS-485
- **McLaughlin, Peter F.**; Scher, Jeremy; Bayne, Michael; Chakraborty, Arindam. Abstracts of Papers, 256th ACS National Meeting & Exposition, Boston, MA, United States, August 19-23, 2018; COMP-6
- **McLaughlin, Peter F.**; Hudson, Bruce S. The Minnesota Workshop on ab Initio Modeling in Solid State Chemistry with CRYSTAL, Minneapolis, MN, United States, July 9-14, 2017.
- **McLaughlin, Peter F.**; Hudson, Bruce S. From Abstracts of Papers, 252nd ACS National Meeting & Exposition, Philadelphia, PA, United States, August 21-25, 2016 (2016), POLY-519.
- **McLaughlin, Peter F.**; Hudson, Bruce S. From Abstracts of Papers, 251st ACS National Meeting & Exposition, San Diego, CA, United States, March 13-17, 2016 (2016), PHYS-486.
- **McLaughlin, Peter F.**; Hudson, Bruce S. From Abstracts of Papers, Pacifichem, 2015 International Chemical Congress of Pacific Basin Society Honolulu, HI, United States, December 15-20, 2015 (2015), PHYS-1129. (international conference)
- **McLaughlin, Peter F.**; Hudson, Bruce S. From Abstracts of Papers, 250th ACS National Meeting & Exposition, Boston, MA, United States, August 16-20, 2015 (2015), PHYS-557.
- **McLaughlin, Peter F.**; Lashua, Amanda F.; Hudson, Bruce S. From Abstracts of Papers, 248th ACS National Meeting & Exposition, San Francisco, CA, United States, August 10-14, 2014 (2014), PHYS-500.

TEACHING EXPERIENCE

Syracuse University 📅 Aug 2012 – Aug 2018

Guest Lecturer

Physical Chemistry Lecture I Thermodynamics

●Taught one lecture review of elementary chemical kinetics and complex reaction mechanisms for exam preparation for students

Teaching Assistant

Physical Chemistry Lecture I Thermodynamics

●Held office hours for students, graded assignments and exams, reviewed and proctored exams

Physical Chemistry Lecture II Quantum Mechanics

●Generated answer keys for homework assignments for students, held office hours for students, graded assignments and exams, reviewed and proctored exams

Physical Chemistry Laboratory I Thermodynamics

●Designed laboratory experiments including pre-lab assignments, procedures and post-lab work due to acquisition of new laboratory equipment. Taught laboratory section, held office hours for students and graded reports Instructed students how to use laboratory instruments including UV-VIS, fluorescence, and solution and bomb calorimeters.

Physical Chemistry Laboratory II Quantum Mechanics

●Constructed laboratory experiments including lecture, pre-lab assignments, procedures and post-lab work. Taught laboratory section, held office hours for students and graded reports. Instructed students how to use laboratory instruments including UV-VIS, fluorescence, liquid and solid Raman, FT-IR for gases and ATR-IR for liquids, and NMR. Introduced students to Gaussian09 for computational studies that supplemented their laboratory experiments.

General Chemistry

●Held office hours for students

Ball State University 📅 Aug 2011 – Dec 2012

Teaching Assistant

General Chemistry for Pre-Nursing

●Taught laboratory section, held office hours for students and graded reports

Analytical Chemistry

●Aided students during laboratory experiments including supervising and answering questions