

© 2019 by Alexander Doran. All rights reserved.

ADVANCEMENTS IN MONTE CARLO MANY BODY METHODS

BY

ALEXANDER DORAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Chemistry
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor So Hirata, Chair
Professor Martin Gruebele
Professor Nancy Makri
Professor Kenneth S Schweizer

Abstract

Many-body methods are one of the most powerful tools that may be brought to bear to solve the electronic structure problem. While both of the following theories provide a complete path to obtain any quantity, many-body perturbation theory is principally used to provide molecular energies, while many-body Green's function provides electron binding energies. Importantly, these methods are systematically improvable, which means there is a prescribed route to increase the accuracy of their results. They are also size consistent. Many-body methods are not without weakness, though, the chief of which is their considerable expense. For example, the operational cost to compute the k -order energy correction with many-body perturbation theory (MBPT) is $O(n^{k+3})$, with n being proportional to the molecular size. Additionally, many-body methods are typically cast into complex series of dense matrix-matrix multiplications, which is difficult to parallelize to millions of processors. Because of this, their enormous expense is difficult to mitigate with the highly parallel architecture of modern supercomputing resources. Recently, the Hirata lab has pioneered stochastic versions of MBPT and many-body Green's functions (MBGF), known as MC-MP and MC-GF, collectively MC-MB, respectively, wherein the correlation energy or electron binding energies are obtained through Monte Carlo integration. The motivation to create stochastic implementations of MBPT and MBGF was to prioritize parallelizability, as Monte Carlo integration is trivially parallel, so that modern supercomputer may be used effectively in the challenge of applying many-body methods to chemical relevance. Here, an overview of the current state of the MC-MP and MC-GF methods, including the pioneering work, advancements that I have completed, and several new developments, is archived.

Second- and third-order Monte Carlo perturbation theory (MC-MP2 and MC-MP3) are two of the earliest MC-MB methods to have been developed. The algorithmic details of these methods are considered due to their relevance within the MC-MB family of methods. One significant methodological development for the MC-MP methods, the implementation of fourth-order Monte Carlo perturbation theory (MC-MP4), is presented. The development of MC-MP4 represents a major milestone in the MC-MB family of methods timeline, as MP4 is the lowest level of perturbation theory capable of producing benchmark quality energies. Two essential studies on the computational characterization of MC-MP2 and MC-MP3 are presented. In

these studies, the cost of the methods is established by numerical experiments. The cost to obtain a result with a specified target relative statistical uncertainty is established to be $O(n^3)$ and $O(n^4)$ for MC-MP2 and MC-MP3, respectively. These cost functions are both two ranks lower than the cost functions of their conventional counterparts and establish the long term viability of MB-MB methods.

Second-order Monte Carlo Green’s function (MC-GF2) was one of the methods pioneered during the early work on MC-MB methods. However, the early implementation of MC-GF2 was reserved to be only applicable within the frequency-independent diagonal approximation. Recently, the MC-GF family of methods was extended in several directions. The first was to increase the rank of the self-energy to third-order producing the third-order Monte Carlo Green’s function (MC-GF3) method. The diagonal approximation for both the MC-GF2 and MC-GF3 methods have been lifted by implementing an efficient algorithm for the calculation of the self-energy matrix. The frequency-independent approximation has been lifted by expanding the self-energy in a truncated Taylor series, which in turn is used to approximate the zeros of the inverse Dyson equation. The derivatives needed to construct the truncated Taylor series are efficiently computed by the MC-GF algorithm. The computational characteristics of MC-GF3 were extensively studied during this work. The most important among these properties was the establishment of the cost function to obtain binding energies within the diagonal frequency-independent and frequency-independent approximations. This experiment is extended here to establish the same for MC-GF2. For the scaling studied, the MC-GF methods are found to have a cost scaling one rank lower than their conventional counterparts.

While MC-MB methods have favorable cost scaling with respect to system size, the naive implementation of these methods is computationally demanding. As such, there has been a significant amount of work to accelerate the convergence of MC-MB methods. Five algorithms that alter the original naive MC-MB framework are discussed. (1) The redundant-walker algorithm accelerates the integration of the electronic coordinates, in which more than the minimally required number of electron pairs are propagated, and the relevant energy correction is sampled at all possible permutations of the electron pairs, for all MC-MB methods. Because of disparities in cost between the various steps of MC-MB methods, the redundant-walker algorithm is able to achieve a boost in the sampling rate. Recent analyses of the redundant-walker algorithm suggest that it is in part responsible for the low scaling of MC-MB methods. (2) MC-MB methods are particularly well-suited to utilize many GPUs. This is because GPUs can push the redundant-walker algorithm harder, and thereby attain a higher than expected speedup. (3) Two strategies to integrate the imaginary time-coordinates are presented. In the first strategy, the imaginary-time coordinates are integrated by quadrature, while in the second strategy, they are integrated by direct Monte Carlo. The integration of these degrees of freedom by stochastic means proves to be significantly more efficient, with speedups of

8.4 and 221 for second- and third-order methods being observed. (4) An algorithm to sample the electronic degrees of freedom directly is presented. This is in contrast to the use of the Metropolis algorithm which has thus far been used to sample these coordinates. The new algorithm eliminates autocorrelation in all integrated quantities, thereby reducing their uncertainties. Speedups of up to 3.21 and 1.38 are observed for MC-MP2 and MC-MP3. This algorithm also provides many qualitative improvements to the MC-MB family of methods. (5) Control variates are applied to the MC-MP family of methods. The application of these control variance produces speedups of 13.9, 17.11, and 58.29 for MC-MP2, MC-MP3, and MC-MP4, respectively. Of these algorithms, the first is apart of the seminal work on MC-MB methods. The last two are recently completed.

To my loving wife Lily Doran.

Table of Contents

Chapter 1	Introduction	1
1.1	Many-body Perturbation Theory	4
1.2	Many-body Green's Function	8
1.3	Monte Carlo Integration	11
1.3.1	Importance Sampling	13
1.3.2	Sampling Algorithms	14
1.4	Computational Details	16
Chapter 2	Monte Carlo Many-body Perturbation Theory Methods	17
2.1	Second-order Monte Carlo Perturbation Theory	17
2.1.1	Derivation of the Integrands	18
2.1.2	The Electronic Importance Function	20
2.1.3	Algorithm Operational Costs	21
2.1.4	Illustrative Calculations	23
2.2	The Redundant-walker Algorithm	23
2.2.1	Algorithm Description	24
2.2.2	Results	26
2.3	Scaling of MC-MP2	27
2.4	Diagrammatic Derivation of MC-MB Methods	32
2.5	Third-order Monte Carlo Perturbation Theory	34
2.5.1	Algorithm Description	35
2.5.2	Illustrative Calculations	37
2.5.3	Scaling of MC-MP3	38
2.5.4	Conclusions	40
2.6	Forth-order Monte Carlo Perturbation Theory	41
2.6.1	Algorithm Description	41
2.6.2	Illustrative Calculations	45
2.6.3	Conclusions	46
2.7	Conclusions	46
Chapter 3	Monte Carlo Many-body Green's Function	48
3.1	Methods	48
3.1.1	Second-order Self-energy	48
3.1.2	Third-order Self-energy	49
3.1.3	Non-diagonal Self-energy and the Redundant-walker Algorithm	52
3.1.4	Frequency-dependent Self-energy	55
3.1.5	Statistical Uncertainty	57
3.2	Computational Efficiency and Applicability	57
3.2.1	Computer Implementation	58
3.2.2	Non-diagonal, Frequency-dependent Self-energy Calculation	58
3.2.3	System-size Dependence of Cost	59
3.2.4	Illustrative Calculations	64

3.3	Conclusions	69
Chapter 4	Acceleration Technique for Monte Carlo Many-body Methods	70
4.1	GPU-acceleration of MC-MB methods	70
4.1.1	Multilevel GPU Parallelism	72
4.1.2	Performance Analysis	78
4.1.3	Conclusions	82
4.2	Integration of the Imaginary Time Coordinates	83
4.2.1	Algorithm Description	83
4.2.2	Performance Analysis	84
4.2.3	Conclusions	85
4.3	Direct Sampling of the Electronic Coordinates	86
4.3.1	Algorithm Description	87
4.3.2	Performance Analysis	92
4.3.3	Conclusions	96
4.4	Control Variates for MC-MP Methods	97
4.4.1	Control Variates	97
4.4.2	Control Variates for MC-MP Methods	99
4.4.3	Performance Analysis	102
4.4.4	Conclusions	104
4.5	Conclusions	105
Chapter 5	Conclusions	106
Chapter 6	Bibliography	108
Appendix A	Calculations for Imaginary-time Integration	113
Appendix B	Calculations for System-size Dependence of Cost of MC-MP2	114
Appendix C	Calculations for System-size Dependence of Cost of MC-MP3	116
Appendix D	Calculations for System-size Dependence of Cost of MC-GF2	117
Appendix E	Calculations for System-size Dependence of Cost of MC-GF3	119
Appendix F	Format for the Integrands	121
Appendix G	Integrands for MC-MP3	122
Appendix H	Integrands for MC-MP4	123
Appendix I	Integrands for MC-GF3	131
Appendix J	Diagrams for MC-MP4	134

Chapter 1

Introduction

Many-body methods[1–6] are one of the most powerful tools to investigate the electronic structure of molecules and solids. Many-body methods include coupled-cluster[1, 2, 7], many-body perturbation theory (MBPT)[1, 2, 5, 8], and many-body Green’s functions (MBGF)[1, 5, 6, 9]. Typically, the total energies and related properties are of interest when employing the first two theories, while in the latter, electron binding energies are of interest.

The usefulness of many-body methods is predicated on two of their properties. The first is that these methods are systematically improvable to the exact solution to the Schrödinger wave equation. This stands in contrast to the situation for density functional theory (DFT) methods, the most commonly used method to investigate electronic structures, in which no clear path to produce exact solutions is known. The second is that these methods are size consistent, i.e., the predictions that these methods provide scale correctly with system size. This is essential when considering systems that approach the thermodynamic limit, i.e., having an infinite size. Clearly, any method that is not size consistent is nonsensical for such a system.

Despite their advantages, many-body methods have drawbacks, the greatest of which is their enormous cost. For example, the operational cost to compute the k -order MBPT energy correction scales as $O(n^{k+3})$ [2], assuming the use of canonical HF orbitals. The steep operational cost is compounded by many-body methods requiring large arrays of intermediates. Both of these obstacles are difficult to treat with supercomputing resources. These methods are often cast into the form of complex sequences of tensor contractions. Such operations are challenging to treat with modern supercomputers, which provide massively parallel computing environments, as matrix multiplication, which underlies tensor contractions, has a poor parallel efficiency when executed on millions of CPUs. The poor scaling is a result of communications between nodes becoming increasingly costly. In order to make many-body methods more suitable for modern supercomputers, one must, therefore, change the underlying algorithmic structure to be more appropriate for the hardware.

One algorithm that is well suited to the architecture of modern supercomputers is Monte Carlo (MC) integration. This is because MC integration is trivially parallel, because as many simulations as there are processors available may be run independently. At the conclusion of the independent simulations, their results

are combined to produce the final answer. The penalty for introducing such a high level of parallelizability is that the results of MC simulations have an associated statistical uncertainty that decays as the square root of the simulation length.

Recently, the Hirata lab has proposed a novel family of methods that integrate the order-by-order corrections to MBPT and MBGF by MC. These methods are deemed Monte Carlo perturbation theory (MC-MP) and Monte Carlo Green’s functions (MC-GF), which may be collectively referred to as the Monte Carlo many-body (MC-MB) family of methods. These methods, by design, have a naturally low operational cost to perform a single MC step, negligible memory requirements, have a high parallel efficiency, and a reasonably straight forward algorithmic implementation.

One of the most notable features of MC-MB methods is that they present a cost scaling with system size with a slower rate of growth than their conventional counterparts.[10–13] Two facts temper the lower scaling. The first is numerical experiments establish the cost scalings, and thus in some regard, they may not reflect the asymptotic cost. The second is that the cost functions have large prefactors, and therefore, for small molecules, the conventional methods outperform the MC-MB methods. One section of this work is convened with alleviating the second concern. Nevertheless, the lower cost scaling functions establish the long term viability of these methods.

The ease of implementation has allowed for a wide range of MC-MB methods to be developed. Including new developments presented here, up to the fourth-order MBPT energy[14, 15] may be obtained by MC-MP methods, while MC-GF may perform up to third-order MBGF calculations[13, 16]. The MC-MB family of methods also contain MC-MP2-F12 and MC-GF2-F12 methods[11, 12], which provide a correction to the finite basis set approximation. Additionally, these methods, while typically developed for molecular problems, have been shown to be implementable for solids[17]. Analogous methods have been developed for vibrational MBPT and MBGF.[18, 19]

The MC-MB methods discussed here are in contrast to other MC implementations of many-body methods[20–27]. The most significant difference between the families of methods is that the methods under consideration here do not require two-electron integrals in any basis. The MC-MB methods are also significantly simpler to understand, being based on simple MC integration, rather than a random state vectors approach.[28–32] This has the obvious benefit of allowing novel features to be explored more rapidly.

The MC-MB methods may also be contrasted against other work that seeks to overcome the great computational expense of many-body methods. One such group of methods is the tensor hypercontraction[33–45] (THC) methods by Martínez and coworkers. These methods scale lower ($O(n^4)$ for THC-MP2 and THC-MP3; proposed to scale as $O(n^{k+1})$ otherwise[33]) than conventional perturbation theory. However,

the reduced scaling comes at the cost of introducing systematic error into the correlation energy. MC-MB methods, although having uncertainty, are free from systematic error, and achieve scalings that are either equal or better than that of the comparable THC-MP methods.

The remainder of this chapter presents a brief introduction to MBPT, MBGF, and MC integration, as well as some notes on the computational resources used to perform the calculations presented. The remainder of this document is broken into three main chapters.

In Chapter 2, the MC-MP methods are discussed chronologically, and through this, the basic framework of the MC-MB methods is established. The MC-MP methods currently include MC-MP2, MC-MP3, and the recently developed MC-MP4. These methods have primarily established the common algorithmic framework used by all MC-MB methods. During the presentation of the algorithmic framework, one of the most vital underlying ideas used by MC-MB methods, the redundant-walker algorithm, is introduced. Two strategies of enumerating the integrands needed by MC-MB methods are touched upon. The cost scaling with problem size is presented for MC-MP2 and MC-MP3. Additionally, illustrative calculations for all three MC-MP methods are presented.

Chapter 3 covers the MC-GF family of methods, which currently include MC-GF2 and MC-GF3. During the development of MC-GF3, the capabilities of the MC-GF methods were greatly enhanced. The enhancements include algorithms to compute the derivatives of the self-energy, to lift the diagonal approximation, and to lift the frequency-independent approximation. The scalings with respect to problem size of both the MC-GF2 and MC-GF3 methods are presented. Illustrative calculations are restricted mainly to MC-GF3, as they better showcase the newly developed enhancements of the MC-GF methods.

Chapter 4 presents four techniques to accelerate the convergence of both MC-MP and MC-GF calculations. The first section covers the GPU-acceleration of MC-MB methods. The second section discusses two different strategies to perform a subset of the integrations (those over the imaginary-time coordinates) in MC-MB methods. The third section presents an algorithm to perform the integration of the electronic coordinates by direct MC rather than by Metropolis MC. The advantage of this technique lies in that the sequence of samples of the energy are uncorrelated. This produces an estimate of the energy that is free from autocorrelation and thus has a lower uncertainty. The final section presents an application of an MC variance reduction technique, control variates, for MC-MB methods.

1.1 Many-body Perturbation Theory

In time-independent perturbation theory[1, 2] a solution to the time-independent Schrödinger wave equation,

$$H |\Psi_m\rangle = E_m |\Psi_m\rangle, \quad (1.1)$$

is sought. The wave function and energy are expanded into perturbation series,

$$|\Psi_m\rangle = |\Psi_m^{(0)}\rangle + \lambda |\Psi_m^{(1)}\rangle + \lambda^2 |\Psi_m^{(2)}\rangle + \dots, \quad (1.2)$$

$$E_m = E_m^{(0)} + \lambda E_m^{(1)} + \lambda^2 E_m^{(2)} + \dots, \quad (1.3)$$

where $|\Psi_m^{(n)}\rangle$ and $E_m^{(n)}$ are the n^{th} -order components of the wave function and energy corresponding to state m respectively, and λ is an order parameter. Since MC-MP methods are currently only able to compute corrections to the ground state, the state subscripts will be dropped. The Hamiltonian operator, $\hat{H}$ is partitioned into,

$$\hat{H} = \hat{H}_0 + \lambda \hat{V}, \quad (1.4)$$

where $\hat{H}_0$ is the reference Hamiltonian and $\hat{V}$ is the perturbation operator. The solutions to the reference Hamiltonian problem are known,

$$\hat{H}_0 |\Phi_n\rangle = E_n^{(0)} |\Phi_n\rangle. \quad (1.5)$$

Inserting all of the above into the time-independent Schrödinger wave equation and grouping terms based on the total power of the order parameter, it reasonably straightforward to show that n^{th} -order component of the energy may be expressed as

$$E^{(n)} = \langle \Psi^{(0)} | \hat{V} | \Psi^{(n-1)} \rangle. \quad (1.6)$$

The energy expression can be simplified by employing the Wigner $2n + 1$ rule[46], which states that the highest order wave function needed to evaluate up to $2n + 1$ order energy is n . Employing this rule the

energy expression becomes

$$E^{(2n)} = \left\langle \Psi^{(n-1)} \left| \hat{V} \right| \Psi^{(n)} \right\rangle - \sum_{j=1}^{n-1} \sum_{l=1}^n E^{2n-j-l} \left\langle \Psi^{(j)} \left| \Psi^{(l)} \right\rangle, \quad (1.7)$$

$$E^{(2n+1)} = \left\langle \Psi^{(n)} \left| \hat{V} \right| \Psi^{(n)} \right\rangle - \sum_{j=1}^n \sum_{l=1}^n E^{2n-j-l+1} \left\langle \Psi^{(j)} \left| \Psi^{(l)} \right\rangle. \quad (1.8)$$

From either Eq. (1.6) or Eq. (1.7), it is apparent that to obtain the first-order correction to the energy, only the zeroth-order wave function is needed. However, since $E^{(1)}$ is accounted for at the HF level, the second-order energy correction is the first non-trivial correction to the energy. In order to evaluate the energies beyond first-order, expressions for the order-by-order components of the wave function must be obtained. A single simple expression for $|\Psi^{(n)}\rangle$ does not exist, and therefore each must be obtained sequentially. The first-order correction to the wave function is

$$|\Psi^{(1)}\rangle = \sum_k \frac{V_{k0}}{E_0^{(0)} - E_k^{(0)}} |\Phi_k\rangle. \quad (1.9)$$

Inserting this into either Eq. (1.6) or Eq. (1.7) furnishes the expression for $E^{(2)}$,

$$E^{(2)} = \sum_k \frac{V_{0k} V_{k0}}{E_0^{(0)} - E_k^{(0)}}. \quad (1.10)$$

By employing the Slater–Condon rules[1, 47, 48] and assuming that the reference Hamiltonian corresponds to a restricted Hartree-Fock (HF) calculation, $E^{(2)}$ becomes

$$E^{(2)} = \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} \frac{2 \langle ij|ab \rangle \langle ab|ij \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} - \frac{\langle ij|ab \rangle \langle ab|ji \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}, \quad (1.11)$$

where ‘occ’ and ‘vir’ denote sums that run over spatial occupied and virtual orbitals respectively, ϵ_i is the i^{th} HF orbital energy and

$$\langle pq|rs \rangle = \int \frac{\phi_p^*(\mathbf{r}_1) \phi_q^*(\mathbf{r}_2) \phi_r(\mathbf{r}_1) \phi_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2, \quad (1.12)$$

where ϕ_p is the p^{th} HF molecular orbital.

The first-order wave function is sufficient to furnish the third-order energy, although the required algebra is more involved. Similarly, producing expressions for the higher-order energies becomes more error-prone due to the meticulous care that must exist during their derivation. More so, relying on Slater-Condon rules to produce expressions in terms of two-electron integrals become increasingly complex.

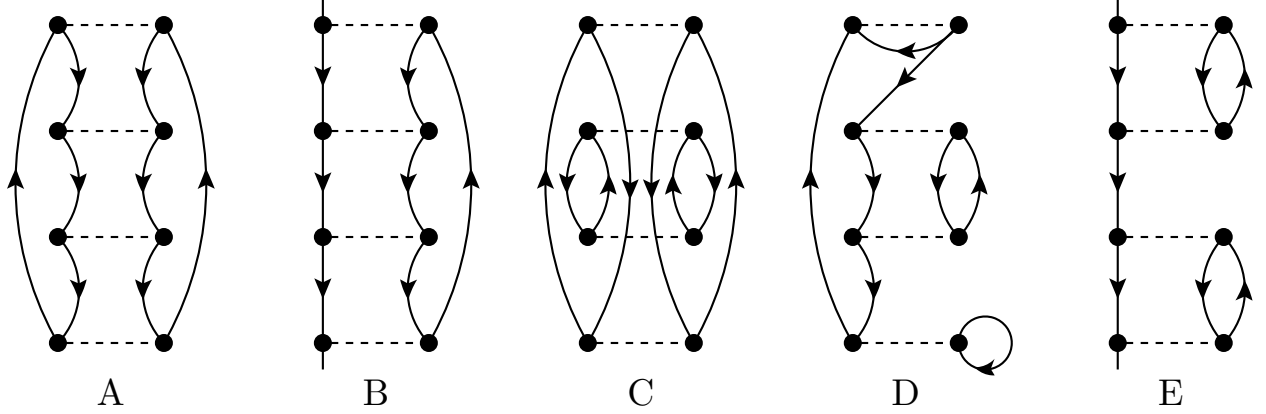


Figure 1.1: (A) A closed diagram. (B) An open diagram. (C) A disconnected diagram. (D) A diagram containing a bubble (bottom vertex) and an oyster (top vertex). (E) A reducible diagram.

Many-body perturbation theory[1, 2, 5, 8] provides a more robust framework to produce evaluable expressions for perturbation theory. The principle difference between the perturbation theory presented above and MBPT is the reliance on second-quantization or diagrams during the derivation of the energy expressions. The diagrammatic techniques, being vastly more useful for the following MC-MB methods, are considered.

The energy corrections in MBPT may be expressed in terms of Feynman–Goldstone diagrams[1–5, 49], which are assembled from three components. The first is the interaction vertex, which is visually represented by two nodes connected by a dashed line, with each node having an outgoing and incoming line. An interaction vertex has no left-right symmetry, so the two nodes may be freely permuted (see Figure 1.2 for an example of this). Algebraically, an interaction vertex corresponds to a two-electron repulsion integral.

The second of the elements is propagator lines, which connect interaction vertices. Propagator lines serve to carry orbital information. Downward lines correspond to occupied (or hole) states while upward lines correspond to virtual (or particle) states.

The final of the elements is resolvent lines, which occur between every adjacent pair of interaction vertices and intersect all relevant propagator lines. The resolvent lines are typically not visually represented, and their existence is made to be implicit. They are algebraically interpreted into energy denominators.

The Feynman–Goldstone diagrams defining $E^{(n)}$ consists of all closed connected diagrams with n interaction vertices and no bubbles or oysters.

In a closed diagram, all propagator lines must originate and terminate at an interaction vertex. This is contrasted by an open diagram, in which two or more propagator lines either originate or terminate at no vertex. Such a propagator line is referred to as a stub. Figure 1.1.A and 1.1.B provide an example of an open and a closed diagram.

Table 1.1: Rule to interpret closed Feynman–Goldstone diagrams into algebraic expressions assuming a restricted HF reference.

- | | |
|---|---|
| 1 | Assign an occupied orbital index, $i, j, \dots$, to each downward line. |
| 2 | Assign a virtual orbital index, $a, b, \dots$, to each upward line. |
| 3 | Associate a factor of $\langle \text{left} - \text{out} \text{ right} - \text{out} \text{left} - \text{in} \text{ right} - \text{in} \rangle$ with each interaction vertex. |
| 4 | Associate a factor of $(\sum_{\text{occupied}} - \sum_{\text{virtual}})^{-1}$ with each resolvent line, where the sums run over all occupied (virtual) orbital indices intersecting the resolvent line. |
| 5 | Sum over all indices associated with propagator lines. |
| 6 | Multiply by a factor of $(2)^l (-1)^{(h+l)}$ where l is the number of closed loops and h is the number of hole lines. |
| 7 | Multiply by a factor of $1/2$ if the diagram has left-right symmetry. |

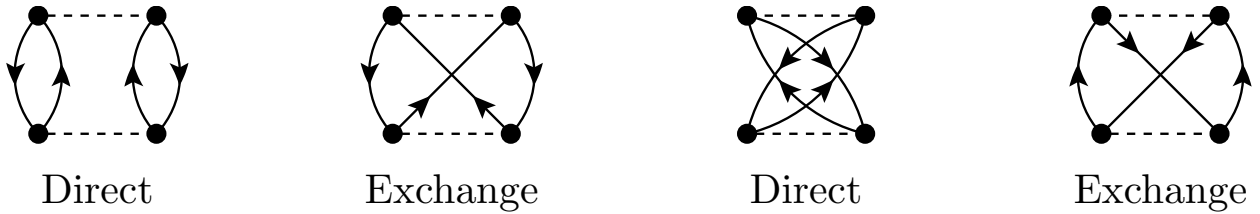


Figure 1.2: The diagrams defining the MP2 energy shown in two way. The two diagrams on the left are equivalent to the two diagrams on the right.

A connected diagram is perhaps best defined as a diagram that is not disconnected. A disconnected diagram is one in which there are two or more parts of the diagram which are not connected by any propagator lines. The exclusion of disconnected diagrams is a result of the linked cluster theorem[50, 51], perhaps one of the most famous results in many-body theories. Figure 1.1.C shows an example of a disconnected diagram.

A bubble or oyster is a propagator line that originates and terminates from the same interaction vertex. An example of a diagram containing a bubble and an oyster is shown Figure 1.1.D

The diagrams defining $E^{(2)}$ are presented in two ways in Figure 1.2. The first set is how the diagrams are typically represented, while the second set is included to demonstrate the permutational equivalency of an interaction vertex. The diagrams defining $E^{(3)}$ may be found in the Section 2.5, while those defining $E^{(4)}$ may be found in the appendix.

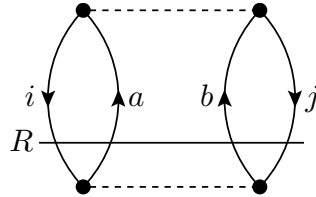


Figure 1.3: The diagram corresponding to the direct contribution to $E^{(2)}$ labeled according to Table 1.1. The resolvent line is explicitly shown and labeled by R .

The rules, assuming an restricted HF reference, to interpret Feynman–Goldstone diagrams into algebraic expressions are listed in Table 1.1. As an example of their application, the interpretation of the diagram corresponding to the direct contribution to $E^{(2)}$ is considered. This diagram, with the resolvent line explicitly shown, labeled according to Table 1.1 is shown in Figure 1.3. This diagram is then interpreted into the algebraic expression,

$$E^{(2)} = 2 \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} \frac{\langle ij|ab \rangle \langle ab|ij \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \quad (1.13)$$

Similarly, the diagram corresponding to the exchange term is interpreted to be

$$E^{(2)} = - \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} \frac{\langle ij|ab \rangle \langle ab|ij \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \quad (1.14)$$

The sum Eq. (1.13) and (1.14) reproduces Eq. (1.11) exactly.

1.2 Many-body Green’s Function

The central object in MBGF is the Green’s function[1, 5, 6, 9] (or propagator), $\mathbf{G}(\omega)$, which has the important property of its poles corresponding to electron ionization potentials and electron affinities. The Green’s function is defined by the Dyson equation,

$$\mathbf{G}(\omega) = \mathbf{G}_0(\omega) + \mathbf{G}_0(\omega)\mathbf{\Sigma}(\omega)\mathbf{G}(\omega), \quad (1.15)$$

where $\mathbf{\Sigma}(\omega)$ is the n_{orb} -by- n_{orb} self-energy matrix, and $\mathbf{G}_0(\omega)$ is the n_{orb} -by- n_{orb} zeroth-order Green’s function matrix, whose definition is

$$\{\mathbf{G}_0(\omega)\}_{pq} = (\omega - \epsilon_p)\delta_{pq}. \quad (1.16)$$

Equation (1.15) is typically manipulated into the inverse Dyson equation,

$$\mathbf{G}^{-1}(\omega) = \mathbf{G}_0^{-1}(\omega) - \mathbf{\Sigma}(\omega). \quad (1.17)$$

The importance of the inverse Dyson equation is that it exposes the inverse Green’s function whose zeros correspond to orbital binding energies. The advantage of working with the inverse Green’s function is apparent when considering that locating zeros numerically is much easier than determining the exact locations



Figure 1.4: The diagrams defining $\Sigma^{(2)}$.

of poles. The zeros of the inverse Green's function occur when its determinate,

$$|\mathbf{G}^{-1}(\omega)| = |\mathbf{G}_0^{-1}(\omega) - \Sigma(\omega)| = 0, \quad (1.18)$$

is equal to zero.

The self-energy is expanded in a perturbation series, i.e.,

$$\Sigma(\omega) = \Sigma^{(2)}(\omega) + \Sigma^{(3)}(\omega) + \Sigma^{(4)}(\omega) + \dots. \quad (1.19)$$

Since calculating the exact self-energy is not feasible, methods in MBGF are defined by how they approximate the self-energy. Truncation of the self-energy at second order produces the method known as GF2[1]. Similarly, truncation at third order produces the method known as GF3[52–54]. Continuing in this fashion produces the GF n family of methods, in which the self-energy is approximated by all self-energy contributions up to n^{th} order. Renormalized methods[9, 55–58], such as the algebraic diagrammatic construction[56] family methods, do not employ a hard truncation of the self-energy. Instead, they use an approximate self-energy that is complete through some order but also retains terms from all higher orders.

In the GF n family of methods, the order-by-order contributions to the self-energy are typically expressed in terms of Feynman–Goldstone diagrams, although other means may be used to produce the relevant expressions[6, 59]. The diagrams defining $\Sigma^{(n)}$ consist of all open irreducible connected diagrams with n interaction vertices, two stubs, and no bubbles or oysters. An irreducible diagram contains a propagator line that, if removed, would create a disconnected diagram. An example of an irreducible diagram may be found in Figure 1.1.E.

The rules to interpret diagrams for the self-energy are the same as in MBPT, with two modifications. First, the stubs in the diagrams are labeled with general orbital indices $\{p, q, \dots\}$. No sum is taken on these indices as they specify which element of the self-energy is being calculated. The second is that a fictitious propagator line is drawn between the two stubs, which contributes to the denominators as if it were a real propagator line. This line produces a factor of ω in the denominators rather than an HF orbital energy.

The diagrams defining the second-order self-energy are shown in Figure 1.4. These diagrams, using the rules in Table 1.1 with the two modifications mentioned above are interpreted into

$$\Sigma_{pq}^{(2:1)} = -2 \sum_{ij}^{\text{occ}} \sum_a^{\text{vir}} \frac{\langle ij|qa \rangle \langle pa|ij \rangle}{\epsilon_i + \epsilon_j - \omega - \epsilon_b}, \quad (1.20)$$

$$\Sigma_{pq}^{(2:2)} = \sum_{ij}^{\text{occ}} \sum_a^{\text{vir}} \frac{\langle ij|qa \rangle \langle ap|ij \rangle}{\epsilon_i + \epsilon_j - \omega - \epsilon_b}, \quad (1.21)$$

$$\Sigma_{pq}^{(2:3)} = 2 \sum_i^{\text{occ}} \sum_{ab}^{\text{vir}} \frac{\langle pi|ab \rangle \langle ab|qi \rangle}{\epsilon_i + \omega - \epsilon_a - \epsilon_b}, \quad (1.22)$$

$$\Sigma_{pq}^{(2:4)} = - \sum_i^{\text{occ}} \sum_{ab}^{\text{vir}} \frac{\langle ip|ab \rangle \langle ab|qi \rangle}{\epsilon_i + \omega - \epsilon_a - \epsilon_b}. \quad (1.23)$$

The diagrams defining the third-order self-energy may be found in Figure 3.1 in Chapter 3. Self-energy contributions beyond third order are not considered in this work.

Besides the truncation of the self-energy, which is the most severe approximation invoked in this work[9], there are two other important approximations to be considered. The first is the diagonal approximation, in which off-diagonal elements of the self-energy are considered to be unimportant, and thus neglected. The adoption of this approximation simplifies Eq. (1.18) to be

$$\omega - \epsilon_p - \Sigma_{pp}(\omega) = 0. \quad (1.24)$$

The second approximation is the frequency-independent approximation, in which the self-energy is assumed to be nearly constant in the neighborhood of an HF orbital energy. In this approximation Eq. (1.18) is simplified to be

$$|\mathbf{G}_0^{-1}(\epsilon_p) - \mathbf{\Sigma}(\epsilon_p)| = 0. \quad (1.25)$$

These approximations may be combined to produce the frequency-independent diagonal approximation. In this approximation, electron binding energies are given by

$$\omega - \epsilon_p - \Sigma_{pp}(\epsilon_p) = 0. \quad (1.26)$$

The frequency-independent and diagonal approximation are valid[9] when the state of interest corresponds to a Koopmans like state, which is a state where an electron is either removed or added, but otherwise, the occupancy remains unchanged. Shake-up states, states in which the occupancy is modified in addition to

the electron attachment or detachment, cannot be accurately described in either of these approximations. Although perhaps more importantly, shake-up states cannot be described by methods that employ a hard truncation of the self-energy.

1.3 Monte Carlo Integration

In this section, a simple, although abstract, MC integration problem is considered. The abstract MC integration problem's purpose is two-fold. First, it will serve a basic introduction to MC integration. For a more in-depth discussion of MC integration, there are many excellent texts that may be consulted.[60–62] Second, it will be of great use when explaining strategies that may be applied to an MC integration problem. This is because the need to carry potentially cumbersome equations while explaining an otherwise uncomplicated strategy will be avoided. One of the most basic integrals encountered in the theory of probability is to find the expected value of some function, $f(\mathbf{x})$, when $\mathbf{x}$ is distributed according to some probability density function (PDF), $\rho(\mathbf{x})$. This integral is written as

$$I = \int f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} \quad (1.27)$$

$$= E(f(\mathbf{x}))|_{\mathbf{x} \sim \rho(\mathbf{x})}. \quad (1.28)$$

The PDF must be a positive semi-definite function and satisfy

$$1 = \int \rho(\mathbf{x})d\mathbf{x}. \quad (1.29)$$

Typically, if $\mathbf{x}$ is low in dimensionality (less than 6-10[60]) and the functions f and ρ are well behaved, the value of the integral in Eq. (1.27) may be easily obtained with a quadrature rule. However, in many cases, a quadrature rule is an ineffective means to obtain the value of the integral.

To obtain the value of the integral, its probabilistic nature can be exploited. The expectation value at its most basic level conveys the average of the function evaluated at a large number of points distributed according to $\rho(\mathbf{x})$. In MC integration this exact process is performed, and assuming the function is evaluated at N points, an estimate of the integral, $\hat{I}^{[N]}$, is produced by

$$I \approx \hat{I}^{[N]} = \frac{1}{N} \sum_{n=1}^N f(\mathbf{x}^{[n]}), \quad (1.30)$$

where the values of $\mathbf{x}^{[n]}$ are generated randomly. There are many strategies that may be used to obtain a

large number of points distributed according to the PDF. The two most relevant strategies for the remaining work are shortly considered. First, several other important quantities in MC integration are noted.

Because of the probabilistic nature of MC integration, the estimate of the integral has an associated uncertainty. The uncertainty of the estimate of I , $s^{[N]}$, is given by

$$s^{[N]} = \sqrt{\frac{\sigma^2}{N-1}}, \quad (1.31)$$

where the variance, σ^2 , is given by

$$\sigma^2 = \int (f(\mathbf{x}) - I)^2 \rho(\mathbf{x}) d\mathbf{x} \quad (1.32)$$

and the standard deviation, σ , is the square root of the variance. Assuming the sequence of points, $\mathbf{x}^{[n]}$, are independent, an unbiased estimate of the variance is given by

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N \left(f(x^{[n]}) - \hat{I}^{[N]} \right)^2. \quad (1.33)$$

Equation (1.31) contains one of the most important details about MC integration, which is that the uncertainty decreases as the square root of the number of MC samples. This is one weakness of MC as it implies that to reduce the uncertainty by a factor of ten, the number of samples must be increased 100-fold. Thus if the standard deviation is a few orders of magnitude greater than the desired level of uncertainty, which is typically determined by I , then the MC integration of Eq. (1.28) maybe intractable. Because of this, a significant portion of many texts on MC integration are dedicated to variance reduction techniques. This weakness naturally leads to considering the cost of an MC simulation, which is the product of the number of requisite MC samples multiplied by the time required to produce a single sample, T , i.e.,

$$c = TN. \quad (1.34)$$

Given a desired level of uncertainty, the number of samples required to reach said desired uncertainty may be estimated from Eq. (1.31). After solving Eq. (1.31) for N , the cost of an MC integration may be written as

$$c = T \frac{\sigma^2}{s_{\text{desired}}^2}. \quad (1.35)$$

In many situation the desired level of uncertainty may be unspecified as its value implicitly decided by the

problem at hand. In this scenario, it is often more convenient to express the cost of an MC simulation as

$$c = T\sigma^2. \quad (1.36)$$

The definition of cost leads to a convenient quantity to compare two different MC algorithms that compute the same quantity. The MC efficiency, which is the ratio of the two different algorithms costs, is defined as

$$\eta = \frac{T_{\text{ref}}\sigma_{\text{ref}}^2}{T_{\text{alt}}\sigma_{\text{alt}}^2}, \quad (1.37)$$

where the subscripts ‘ref’ and ‘alt’ label the two algorithms as the reference and alternate algorithms, respectively. The MC efficiency measures how many times faster (or slower) the alternate algorithm is with respect to the reference algorithm.

1.3.1 Importance Sampling

In many MC integration problems outside of statistics, the PDF, $\rho(\mathbf{x})$, may not be explicitly defined. If the integral problem is over some closed finite interval, then a uniform PDF over the same interval can be implied. However, if any of the bounds of the integral problem are infinite, then either an explicit PDF must be teased out of the integrand by splitting it into two parts, or alternatively, one may introduce one via importance sampling.

In addition to being a convenient method to impose a PDF on an integration problem, importance sampling is one of the most powerful variance reduction techniques that can be employed to accelerate an MC integration. When using importance sampling one introduces an auxiliary PDF, $\omega(\mathbf{x})$, often called the importance function, to the integral problem such that

$$I = \int f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} \quad (1.38)$$

$$= \int \frac{f(\mathbf{x})\rho(\mathbf{x})}{\omega(\mathbf{x})}\omega(\mathbf{x})d\mathbf{x} \quad (1.39)$$

$$= \text{E} \left(\frac{f(\mathbf{x})\rho(\mathbf{x})}{\omega(\mathbf{x})} \right) \Big|_{\mathbf{x} \sim \omega(\mathbf{x})}. \quad (1.40)$$

Thus the introduction of the auxiliary PDF changes the estimate of I from Eq. (1.28) to Eq. (1.40). The importance function needs to satisfy a few conditions. [60] First, it needs to be a valid PDF, i.e., it should be positive semi-definite and its integral should be one. Second, the importance function needs to be greater than zero whenever $f(\mathbf{x})\rho(\mathbf{x})$ is greater than zero. Third, if any of the bounds of the integration problem are

infinity, the importance function should have tails that are at least as heavy as $f(\mathbf{x})\rho(\mathbf{x})$, that is, it should have either a slower or the same rate of decay as $f(\mathbf{x})\rho(\mathbf{x})$. Finally, the importance function should cancel any singularities present in $f(\mathbf{x})\rho(\mathbf{x})$.

Critically, with importance sampling the variance becomes

$$\sigma^2 = \int \left(\frac{f(\mathbf{x})\rho(\mathbf{x})}{\omega(\mathbf{x})} - I \right)^2 \omega(\mathbf{x}) d\mathbf{x} \quad (1.41)$$

$$= \int \frac{f(\mathbf{x})^2 \rho(\mathbf{x})^2}{\omega(\mathbf{x})} d\mathbf{x} - I^2. \quad (1.42)$$

Since the importance function is user-defined, the variance as written in Eq. (1.41) may be altered by modifying the importance function. Typically, one searches for a function that is as close as possible to $f(\mathbf{x})\rho(\mathbf{x})$ in order to minimize the variance.

1.3.2 Sampling Algorithms

How to sample from a PDF, whether naturally occurring or introduced by importance sampling, has not been addressed. Here, the inverse-CDF (or quantile method) and Metropolis–Hastings algorithm[63, 64], which are the two most relevant sampling methods for the MC-MB family of methods, will be considered. Other sampling methods, such as Gibbs sampling[65] or Hamiltonian MC[66], will not be considered, as they are not currently used by any MC-MB method. Random number generation (RNG) will be considered no farther than to say that most of the MC-MB methods use the Mersenne–Twister algorithm[67], due to the complexity of the subject, and the widespread availability of high-quality random number generators. Furthermore, specialized methods to sample from standard PDF efficiently, such as the Box-Muller transformation[68], which is used to sample normal distributions, will also not be considered for the same reasons given for RNG.

The inverse-CDF method is one of the most fundamental algorithms to generate a point distributed according to any one-dimensional PDF. Given a one-dimensional PDF, $w(x)$, and its associated cumulative density function (CDF), $W(x)$, a point x distributed according to $w(x)$ is generated by solving

$$W(x) - p = 0, \quad (1.43)$$

where p is a $U(0, 1)$ random variable. If $W(x)$ is invertible, hence the methods namesake, then x is given by

$$x = W^{-1}(p), \quad (1.44)$$

where $W^{-1}(p)$ is the quantile function of $W(x)$. If the quantile function is not readily available, a solution to Eq. (1.43) may be sought numerically through a root finding algorithm.

The inverse-CDF method may be used to sample PDFs whose dimensionality is greater than 1. First, it is recommended to separate the PDF into as many low-dimensional component distributions as possible. Sampling from components which are single-dimensional is trivial, as it is a straight forward application of the inverse-CDF methods. Component PDFs with a dimensionality greater than one may be sampled in a sequential process employing marginal and conditional distributions, although this process rapidly becomes cumbersome with increasing dimensionality.

Often, particularly in the case of high dimensional integration problems where MC proves itself to be most useful, the PDF can not easily be decomposed into a product of one- or low-dimensional functions, and thus the inverse-CDF method is not tractable. The Metropolis-Hastings algorithm, from here on out simply referred to as the Metropolis algorithm, can sample any valid PDF. This is accomplished by making a distorted random walk through coordinate space. A two-step process accomplishes the random walk. In the first step, a trial position $\mathbf{x}^{[T]}$ is generated by randomly offsetting the previous position, $\mathbf{x}^{[n-1]}$, i.e.,

$$\mathbf{x}^{[T]} = \mathbf{x}^{[n-1]} + \Delta\mathbf{x}, \quad (1.45)$$

where $\Delta\mathbf{x}$ is a random offset. The second step decides if the random move is accepted. The move is accepted with probability

$$P(\text{Accept } \mathbf{x}^{[T]}) = \min \left(1, \frac{\omega(\mathbf{x}^{[T]})}{\omega(\mathbf{x}^{[n-1]})} \right), \quad (1.46)$$

where $\omega(\mathbf{x})$ is the relevant PDF. If the move is accepted then $\mathbf{x}^{[n]} = \mathbf{x}^{[T]}$, otherwise $\mathbf{x}^{[n]} = \mathbf{x}^{[n-1]}$.

When using the Metropolis algorithm, a burn-in period where a large number of points are discarded before accumulating MC samples must be performed. The burn-in period chiefly accomplishes two tasks. First, it may serve to randomize the coordinates of $\mathbf{x}$. This is useful if the coordinates of $\mathbf{x}$ are initialized to some convenient known value, such as 0 or the atom centers in a molecule. Secondly, it allows the average value of the offset to be set in order to tune the accept-reject ratio, that is, the ratio of the number of moves that are accepted versus the number that are rejected. An ideal value of the accept-reject ratio is approximately 50%.[60] The length of a burn-in may be reduced by using more sophisticated schemes to initialize the coordinates of $\mathbf{x}$ and by using a value of the average offset which is known to produce an ideal accept-reject ratio for a similar problem.

The sequence of points produced by the Metropolis algorithm is not independent since the value of

$\mathbf{x}^{[n]}$ clearly depends on the value of $\mathbf{x}^{[n-1]}$. Instead, they are correlated. Because of this, the estimate of the variance as given in Eq. (1.33) is no longer valid. The estimate of the variance must account for the autocorrelation between the MC samples. In the MC-MB family of methods, the blocking algorithm of Flyvbjerg and Petersen[69] is used to ensure that the statistical uncertainty s and variance $\hat{\sigma}^2$ are accounting for all autocorrelations. They are given by

$$s(N_b) = \hat{\sigma}(N_b) \sqrt{\frac{N_b}{N}}, \quad (1.47)$$

$$\hat{\sigma}^2(N_b) = \frac{N_b}{N} \sum_{n=1}^{N/N_b} \left\{ \frac{1}{N_b} \sum_{n'=1}^{N_b} f(\mathbf{x}^{[(n-1)N_b+n']}) - \hat{f}^{[N]} \right\}^2, \quad (1.48)$$

where N_b is the block size. The statistical uncertainties with block sizes $N_b = 2^b$ for b ranging from 0 to 10 are typically accumulated concurrently for MC-MB methods. The statistical uncertainty accounting for all autocorrelations corresponds to the smallest value of N_b where $\hat{\sigma}^2$ plateaus, which normally occurs at $b = 7$ for MC-MB methods.

1.4 Computational Details

The computations in this manuscript were performed on three machines, an Acer Predator G3-605 workstation, the Gellmann cluster, and Blue Waters at National Center for Supercomputing Applications (NCSA). The specifications of these three machines are cataloged here so that they may be referred to with ease during the rest of the manuscript.

The Acer Predator G3-605 workstation consists of a 4-core Intel Core i5-4440 CPU, 8 GB of RAM, and an NVIDIA GeForce GTX 745 GPU.

The Gellmann cluster consists of eight compute nodes. Each compute node of the Gellmann cluster consists of two 8-core AMD 6136 CPUs and 64 GB of RAM.

The Blue Waters Supercomputer at NCSA is built from Cray XE and Cray XK nodes. Its Cray XE nodes have two 8-core AMD 6276 CPUs and 64 GB of RAM per node, whereas the Cray XK nodes consist of a single 8-core AMD 6276 CPU, 32 GB of RAM, and an NVIDIA K20X GPU per node. Each XE node supplies sixteen CPU threads of compute and has a peak performance of 313 GFLOPS while an XK node uses a single CPU thread to orchestrate the MC calculation on the GPU, which provides a peak performance of 1310 GFLOPS.

Chapter 2

Monte Carlo Many-body Perturbation Theory Methods

This chapter broadly discusses the historical developments for MC-MB methods as well as more recent ones. The historical events include a description of two of the earliest MC-MB methods to be produced, MC-MP2 and MC-MP3. A detailed derivation of the integrands used in MC-MP2 are considered. This derivation connects the sum of electron-repulsion integral representation typically encountered in MBPT methods to the integral expression used by the MC-MP methods. After this derivation, a more powerful tool to obtain the integrands are considered. In the discussion of these methods, two critical developments that were required to make them viable are naturally encountered. The first was a suitable function to sample the electronic degrees of freedom. The second is the redundant-walker algorithm. All of the MC-MB methods currently employ both of these developments. Recent work includes the development of MC-MP4, which is a milestone achievement for the MC-MB family of methods. This is because MP4 is the lowest rank of perturbation theory that is capable of producing benchmark quality energies, and thus the MC-MB family of methods is now capable of performing such studies. Additional recent work includes computational studies that establish the cost scaling with system size to perform MC-MP2 and MC-MP3 computations.

2.1 Second-order Monte Carlo Perturbation Theory

MC-MP2[14] was the first MC-MB method to be proposed by the Hirata lab. It established the essential framework common to all of the MC-MB methods and, therefore, a detailed analysis of MC-MP2 is presented as it highlights some of the advantages of the MC-MB family of methods as compared to their conventional counterparts. Although more expeditious approaches are used to derive the equations of more recently developed MC-MB methods, the original derivation of MC-MP2 is presented as it exhibits the connection between the integral equations used in the MC-MB methods and the sum over two-electron integral equations of their conventional counterparts. Following the derivation, the algorithmic analysis of MC-MP2 displays the low operational cost and memory footprint required by MC-MB methods as compared to conventional algorithms. Finally, several of the original calculations employing MC-MP2, which established the method's

viability, are reproduced and discussed.

2.1.1 Derivation of the Integrands

The integral equations used in MC-MP2 were originally derived by starting from Almlöf's Laplace transform MP2 (LT-MP2) formalism[70]. In LT-MP2, a Laplace transform,

$$\frac{1}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} = - \int_0^\infty d\tau e^{(\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b)\tau}, \quad (2.1)$$

is applied to the energy denominators in Eq. (1.13) and Eq. (1.14). As the following manipulations may be identically applied to the exchange contribution, the derivation proceeds with only the direct contribution to $E^{(2)}$. After applying the Laplace transform one obtains

$$E_{\text{direct}}^{(2)} = -2 \sum_{i,j,a,b} \langle ij|ab \rangle \langle ab|ij \rangle \int_0^\infty d\tau e^{(\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b)\tau}. \quad (2.2)$$

The integrals in the two-electron integrals are then exposed and exchanged with the summations over molecular orbitals (MO) to obtain

$$\begin{aligned} E_{\text{direct}}^{(2)} = -2 \iiint d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \int_0^\infty d\tau \sum_{i,j,a,b} & \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \phi_a(\mathbf{r}_1) \phi_b(\mathbf{r}_2) \\ & \times \phi_a^*(\mathbf{r}_3) \phi_b^*(\mathbf{r}_4) \phi_i(\mathbf{r}_3) \phi_j(\mathbf{r}_4) \\ & \times \frac{1}{r_{12}} \frac{1}{r_{34}} e^{(\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b)\tau}. \end{aligned} \quad (2.3)$$

Finally, all terms are grouped by their indices to produce

$$E_{\text{direct}}^{(2)} = -2 \iiint d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \int_0^\infty d\tau \frac{G^+(\mathbf{r}_3, \mathbf{r}_1, \tau) G^-(\mathbf{r}_1, \mathbf{r}_3, -\tau) G^+(\mathbf{r}_4, \mathbf{r}_2, \tau) G^-(\mathbf{r}_2, \mathbf{r}_4, -\tau)}{r_{12} r_{34}}, \quad (2.4)$$

where $G^\pm(\mathbf{r}', \mathbf{r}, \tau)$ terms are identified as the traces of the advanced and retarded HF Green's function in real space and imaginary-time and are defined as

$$G^-(\mathbf{r}', \mathbf{r}, \tau) = \sum_i^{\text{occ}} \phi_i(\mathbf{r}') \phi_i^*(\mathbf{r}) e^{-\epsilon_i \tau}, \quad (2.5)$$

$$G^+(\mathbf{r}', \mathbf{r}, \tau) = \sum_a^{\text{vir}} \phi_a(\mathbf{r}')^* \phi_a(\mathbf{r}) e^{-\epsilon_a \tau}. \quad (2.6)$$

After applying the same logic to the exchange contribution, the second-order energy correction is obtained

as

$$E^{(2)} = \iiint \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \mathrm{d}\mathbf{r}_3 \mathrm{d}\mathbf{r}_4 \int_0^\infty \mathrm{d}\tau \frac{1}{r_{12}} \frac{1}{r_{34}} \left(G^+(\mathbf{r}_3, \mathbf{r}_1, \tau) G^-(\mathbf{r}_2, \mathbf{r}_3, -\tau) G^+(\mathbf{r}_4, \mathbf{r}_2, \tau) G^-(\mathbf{r}_1, \mathbf{r}_4, -\tau) \right. \\ \left. - 2G^+(\mathbf{r}_3, \mathbf{r}_1, \tau) G^-(\mathbf{r}_1, \mathbf{r}_3, -\tau) G^+(\mathbf{r}_4, \mathbf{r}_2, \tau) G^-(\mathbf{r}_2, \mathbf{r}_4, -\tau) \right) \quad (2.7)$$

$$= \iiint \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \mathrm{d}\mathbf{r}_3 \mathrm{d}\mathbf{r}_4 \int_0^\infty \mathrm{d}\tau f^{(2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \tau). \quad (2.8)$$

An estimate of the second-order correlation energy is obtained by subjecting the integrals over the electron coordinates to an MC integration with a judiciously chosen importance function. The remaining integration over the imaginary-time coordinate introduced by the Laplace transformation was carried out using a quadrature rule during MC-MP2's proposal. Thus, an estimate of the second-order correlation energy after N MC step, $\hat{E}^{(2)[N]}$, is formed by

$$\hat{E}^{(2)[N]} = \frac{1}{N} \sum_{n=1}^N \frac{\tilde{f}^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]})}{\omega^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]})}, \quad (2.9)$$

$$= \frac{1}{N} \sum_{n=1}^N I^{(2)[n]} \quad (2.10)$$

where $\mathbf{r}_1^{[n]}$ is the coordinate of the first electron during the n^{th} MC step, $\omega^{(2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$ is the importance function for the electron coordinates (discussed shortly), and

$$\tilde{f}^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}) = \sum_q^{N_\tau} \nu_q f^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \tau_q), \quad (2.11)$$

where (ν_q, τ_q) are the weights and points of a suitable quadrature rule, typically, a 21-pt Gauss-Kronrod quadrature rule[71], and $f^{(2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \tau)$ is the integrand of Eq. (2.7). Alternatively, the integration of the imaginary-time coordinates by direct MC has recently been proposed. If the imaginary-time coordinates are integrated by direct MC, $\tilde{f}$ takes the form of

$$\tilde{f}^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}) = \frac{f^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \tau^{[n]})}{\omega_\tau(\tau^{[n]})}, \quad (2.12)$$

where $\omega_\tau(\tau)$ is the importance function for the imaginary-time coordinates and is an exponential distribution. As the two integration schemes are similar at the formalism level, the integration of the imaginary-time coordinates for all of the remaining MC-MB methods will adopt whichever technique is more appropriate, and the other will be omitted to avoid being needlessly tedious. In Section 4.2, the full implementation

details on the use of MC integration of the imaginary-time coordinates and how to shift between the two integrations schemes are presented. A comparison between the application of a quadrature rule or MC for these integrations is also presented.

2.1.2 The Electronic Importance Function

The importance function for the electron coordinates is

$$\omega^{(2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4) = \omega(\mathbf{r}_1, \mathbf{r}_2)\omega(\mathbf{r}_3, \mathbf{r}_4), \quad (2.13)$$

where $\omega(\mathbf{r}', \mathbf{r})$ is the importance function for a pair of electrons. A significant contribution from the original work on MC-MP2 was the preceding derivation of the integral equations for the second-order correlation energy. In retrospect, an equally important contribution was the proposal to form the importance function for the electronic coordinates as the product of importance functions for the pairs of electrons. An importance function of this form has been used in all of the MC-MB methods produced by the Hirata Lab. Its pervasiveness is because it may be naturally extended to form importance functions for higher-order integration problems by multiplying an appropriate number of electron pair importance functions together. Additionally, as presented in Section 2.2, forming importance functions as the product of electron pair importance functions gives rise to a crucial acceleration technique for MC-MB methods. The importance function for a pair of electrons is

$$\omega(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{N_g} \frac{g(\mathbf{r}_1)g(\mathbf{r}_2)}{r_{12}}, \quad (2.14)$$

where N_g is a normalization constant and $g(\mathbf{r})$ is typically a sum of S-type atom centered Gaussians. The normalization constant, N_g , is

$$N_g = \iint d\mathbf{r}_1 d\mathbf{r}_2 \frac{g(\mathbf{r}_1)g(\mathbf{r}_2)}{r_{12}}. \quad (2.15)$$

If $g(\mathbf{r})$ is a sum of S-type Gaussians, the integrals needed to compute Eq. (2.15) are easily obtained as they are S-type two-electron integrals[72]. In principle, $g(\mathbf{r})$ can be any function that roughly approximates the electron density. Using the electron density itself was examined during the proposal of MC-MP2[14]. It was found to perform less well than a simple sum of Gaussians. The lesser performance was attributed to the electron density being insufficiently diffuse for the integration of the virtual orbitals in the traces of the retarded HF Green's functions.

No other forms of $g(\mathbf{r})$, besides the sum of S-type Gaussians or the electron density, have been investigated. This is partially due to the difficulty of performing the integration in Eq. (2.15) needed to normalize the importance function. At the same time, the sum of S-type Gaussians has proved to be a robust and flexible importance function and there has been little need to examine other functional forms of $g(\mathbf{r})$.

In MC-MB methods, generating electron coordinates distributed according to Eq. (2.14) has historically been carried out using the Metropolis algorithm. Recently, I have developed a new algorithm in which the electron coordinates are sampled directly. The details of this algorithm are presented latter in Section 4.3 due to its complexity. Which sampling technique that was used for a calculation will be noted during the presentation of the calculation’s results.

2.1.3 Algorithm Operational Costs

By design, at the algorithmic level, MC-MP2, and indeed all MC-MB methods, is radically different from its conventional counterpart. This difference leads to two of the most advantageous features of MC-MP2. They are that the operational cost and memory footprint both have low scalings with respect to system size. The low scaling allows nearly any computer to be able to perform an MC step in an MC-MP2 calculation, although the overall cost of a calculation, due to the magnitude of the variance, maybe too significant for some machines. However, this is offset by the embarrassingly parallel nature of MC integration, as while any single machine may not be able to complete a calculation, many such machines may. Furthermore, the low scalings are inherited by all MC-MB methods. The scalings are established by breaking down the MC-MP2 algorithm into its elementary steps. The following analysis is based on the original presentation of MC-MP2 and assumes that the electron coordinates are sampled using the Metropolis algorithm and that the imaginary-time coordinate is integrated using a quadrature rule.

In the first step of an MC-MP2 calculation, the electron coordinates are propagated by the Metropolis algorithm. This step carries a cost of $O(n)$ because the evaluation of $g(\mathbf{r})$ in the electron pair importance functions is proportional to the system size. No intermediate arrays are required in this step, and thus its memory cost is constant.

The remaining steps are only evaluated if at least one of the two-electron pairs was moved during the Metropolis update. If both coordinates remained stationary, the energy contribution from the previous MC step is memoized and can be added to the MC estimate without recalculation. Under this occurrence, the calculation then returns to the first step.

In the second step, the atomic orbital (AO) amplitudes are calculated. Because there are n_{bf} AO amplitudes to evaluate, the operational cost of this step scales linearly with system size. For the same

Table 2.1: The operational and memory costs for each step in the MC-MP2 algorithm.

Step	Task	Operational Cost ^a	Memory Cost ^a
1	Update Coordinates	$O(n)$	$O(1)$
2	Calculate AOs	$O(n)$	$O(n)$
3	Calculate MOs	$O(n^2)$	$O(n^2)$
4	Calculate $G^\pm$	$O(n)$	$O(1)$
5	Calculate Energy	$O(1)$	$O(1)$

^a n is the problem size ($n = n_{\text{bas}} \propto n_{\text{ele}}$).

reason, an array of length $O(n)$ is required to store the AO amplitudes.

In the third step, the AO-to-MO amplitude transformation is carried out. As there are $n_{\text{bf}}n_{\text{orb}}$ MO coefficients, an $O(n^2)$ sized array is required to store them. The AO-to-MO transformation is a matrix-vector multiply, and it is therefore assigned a cost of $O(n^2)$.

The fourth and fifth steps are looped on the number of quadrature points used to integrate the imaginary-time coordinate. The exponentials required to compute the traces of the HF Green’s functions are precomputed at the start of the MC calculation and stored in an array of size $O(nN_\tau)$.

In the fourth step, the traces of the HF Green’s functions are computed. As can easily be seen from their definitions (Eq. (2.5) and Eq. (2.6)), the evaluation of the traces of the HF Green’s functions is proportional to either the number of occupied or virtual MO orbitals. On this basis, this step is assigned a cost of $O(n)$. No new intermediate arrays are required during this step.

In the final step, the integrand of Eq. (2.7) is evaluated. This consists of simply performing a handful of multiplications and is performed in constant time.

A summary of the operational and memory costs of all of the algorithmic steps in an MC-MP2 calculation is given in Table 2.1. From the analysis, it is observed that the most expensive step carries an operational cost of $O(n^2)$. For small molecules, it is not unusual for the evaluation of the AO amplitudes to be the most computationally demanding step as the exponential function is evaluated many times. Additionally, the largest required array is the array of MO coefficients, which is of size $O(n^2)$. All other values are either stored in arrays of lesser size or are used without storage. As will be seen during the development of higher-order MC-MP methods, steps 1 through 4 remain virtually unchanged, thus so do their operational and memory costs. The only modified step is the evaluation of the energy, which remains constant with molecular size.

It must be cautioned the operational cost does not describe the true scaling of MC-MP2 because this analysis does not consider the impact of the variance. The true scaling was established via numerical studies later and is discussed herein Section 2.3.

Table 2.2: Illustrative calculations performed by MC-MP2 at the cc-pVDZ level of chemistry invoking the frozen core approximation. The calculations were performed using a single core on the Acer workstation. All simulations are 2^{27} MC steps long. Energies and uncertainties are given in E_h and times are given in seconds.

Molecule	n_b	$E^{(2)}$	$\widehat{E}^{(2)}$	σ	Time
Water	24	-0.201627	-0.2011	0.0023	1357
Water dimer	48	-0.408032	-0.3934	0.0093	2386
Water trimer	72	-0.620396	-0.618	0.023	3732
Benzene	114	-0.783937	-0.814	0.052	6658

2.1.4 Illustrative Calculations

Table 2.2 presents a small set of calculations on several small water clusters ($r_{OH} = 0.9573\text{\AA}$) and benzene ($r_{CC} = 1.3970\text{\AA}$, $r_{CH} = 1.0869\text{\AA}$) carried out at the MC-MP2/cc-pVDZ level of chemistry. For the smallest system, water, MC-MP2 is able to achieve an uncertainty on the order of mE_h in approximately 10^8 MC steps. The MC estimate of $E^{(2)}$ for water is in good agreement with the exact result. As the molecular size increases, the error between the MC estimates of $E^{(2)}$ and the exact values becomes more substantial, but no error is unreasonable given its corresponding uncertainty. It is also noted, that the time and the uncertainty both increase with molecular size. The increasing time is well expected based on the algorithmic analysis of MC-MP2. However, the growth in the simulation time is roughly linear with system size rather than quadratic. This indicates that the rate-limiting step of MC-MP2, when applied to small molecules, is the calculation of the AO amplitudes as noted during the operational cost analysis.

Despite the ability of MC-MP2 to calculate $E^{(2)}$, the simulation time, the time required to reach an acceptable level of uncertainty, can hardly be considered an improvement over conventional MP2, as the exact values of $E^{(2)}$ were all obtained in less than a minute on the same machine. However, these simulations are disadvantaged as they are serial, which neglects the embarrassingly parallel nature in which MC integration can be performed, which is one of MC’s greatest advantages. The embarrassingly parallel nature of MC integration means that the reduction in wall time by the use of more compute for MC-MP2 simulation is limited by only the amount of available compute. Nevertheless, the cost of a simple MC-MP2 simulation demanded the proposal of acceleration techniques to make the method more tractable. One such technique is presented in the next section, while newer techniques may be found in Chapter 4.

2.2 The Redundant-walker Algorithm

The results of the MC-MP2 illustrative calculations presented in Table 2.2 demonstrated that MC-MP2 was a viable method. Simultaneously, they also demonstrated that the uncertainty of the second-order correla-

tion energy can be quite large for even moderately sized molecules. This caused the illustrative MC-MP2 calculations to be significantly more costly than their conventional counterparts. As such, the convergence of MC-MP2 needed to be massively accelerated. The redundant-walker algorithm[73], proposed by Willow *et al.*, provided a large portion of this much-needed performance increase. Similar to the importance function for a pair of electrons, the redundant-walker algorithm has been used in all of the MC-MB methods produced in the Hirata lab. As such, its importance cannot be understated and it is presented here rather than in Chapter 4 with more recently developed acceleration techniques.

2.2.1 Algorithm Description

In MC-MP2, the importance function for the electron coordinates is the product of two functions, each of which is the importance function for an electron pair. The two pairs of electrons in MC-MP2 are propagated separately by the Metropolis algorithm. With the redundant-walker algorithm, instead of propagating the minimally required two pairs of electrons, m ($m \gg 2$) pairs of electrons are propagated. Using the m pairs of electrons, the contribution to $E^{(2)}$ from an individual MC step is evaluated at all $m(m-1)/2$ unique combinations of the redundant walkers. Thus the energy contribution from the n^{th} MC step becomes

$$I^{(2)[n]} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \frac{\tilde{f}^{(2)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]})}{\omega^{(2)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]})}, \quad (2.16)$$

where $(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]})$ are the first and second coordinates of the i^{th} walker during the n^{th} MC step. Because Eq. (2.16) is an average of $O(m^2)$ values, during the proposal of the redundant-walker algorithm, it was implicitly hypothesized that the variance would be compressed by a factor of $O(m^2)$ for a calculation using m walkers. The relatively innocuous change made to the definition of $I^{(2)[n]}$ affects every algorithmic step of MC-MP2. The overall acceleration can only be determined when considering the impact of the redundant-walker algorithm on both the operational cost and the reduction of the variance.

Table 2.3 updates the operational and memory costs of MC-MP2 when using the redundant-walker algorithm. The scaling of each step with respect to system size is left unchanged by the redundant-walker algorithm, and the scaling with respect to the number of walkers is examined.

The first three steps of MC-MP2, the propagation of the electron coordinates, and the computation of the AO and MO amplitudes scale as $O(mn_{\text{bf}})$, $O(mn_{\text{bf}})$, and $O(mn_{\text{bf}}^2)$ respectively under the redundant-walker algorithm. This is simply because there are now m walkers to propagate, and that the AO and MO amplitudes must be calculated at m walker coordinates. Similarly, the intermediate arrays required for the first and second steps have a linear dependence on the number of redundant walkers. During the third step,

Table 2.3: The operational and memory costs for each step in the redundant walker accelerated MC-MP2 algorithm.

Step	Task	Operational Cost ^a	Memory Cost ^a
1	Update Coordinates	$O(mn)$	$O(m)$
2	Calculate AOs	$O(mn)$	$O(mn)$
3	Calculate MOs	$O(mn^2)$	$O(n^2) + O(mn)$
4	Calculate $G^\pm$	$O(m^2n)$	$O(m^2)$
5	Calculate Energy	$O(m^2)$	$O(m^2)^b$

^a m is the number of electron pairs and n is the problem size ($n = n_{\text{bas}} \propto n_{\text{ele}}$). ^bCost is $O(1)$ if the energies are not memoized.

the array of MO coefficients is unmodified, but there are now m sets of n_{orb} MO amplitudes to be stored. This is the origin of the second term in the memory cost of this step.

The calculation of the traces of the HF Green's functions scales as $O(m^2n_{\text{bf}})$ because each of the traces requires coordinates from two redundant walkers and thus each trace must be evaluated at $m(m-1)/2$ coordinate pairs lead to the $O(m^2)$ cost. For the same reason, arrays of size $O(m^2)$ are required to store the traces of the HF Green's functions. The evaluation of the energy with the redundant-walker algorithm scales as $O(m^2)$ as can be trivially observed from Eq. (2.16). If the energies are memoized, an array of $O(m^2)$ is required for this step.

With the impact of the redundant-walker algorithm on the operational cost and the variance determined, the overall acceleration is examined. The cost of an MC simulation, as stated in Eq. (1.36), is the product of the operational cost and the variance. Inserting the operational cost and the variance for MC-MP2 with the redundant-walker algorithm gives

$$\text{Cost} = (\text{Operational Cost}) \cdot (\text{Variance}) \quad (2.17)$$

$$\approx (O(m^2n) + O(mn^2)) \frac{\sigma_0^2}{O(m^2)}, \quad (2.18)$$

where only steps with a total polynomial dependence of three are used to approximate the operational costs. When using the redundant-walker algorithm, if the number of walkers is made to be proportional to the problem size, i.e., $m \propto n$, the cost becomes

$$\text{Cost} \approx (O(n^3)) \frac{\sigma_0^2}{O(n^2)} \quad (2.19)$$

$$\approx (O(n)) \sigma_0^2. \quad (2.20)$$

Thus the redundant-walker algorithm reduces the cost scaling of MC-MP2 by one rank by trading an increase in the operational cost for a greater reduction in the variance. The scaling of the variance with problem size

Table 2.4: Calculations on benzene using redundant walker accelerated MC-MP2 at the cc-pVDZ level of chemistry invoking the frozen core approximation. The calculations were carried out on a single node of the Gellmann cluster. All simulations are of length 2^{24} MC steps. Energies and uncertainties are given in E_h .

Walkers	$\widehat{E}^{(2)}$	σ	Steps/s ^a	Samples/s ^a	η
2	-0.81	0.14	$1.48 \cdot 10^5$	$1.48 \cdot 10^5$	2.287
4	-0.747	0.066	$7.96 \cdot 10^4$	$4.78 \cdot 10^5$	0.925
8	-0.775	0.030	$3.43 \cdot 10^4$	$9.60 \cdot 10^5$	0.445
16	-0.798	0.014	$1.23 \cdot 10^4$	$1.48 \cdot 10^6$	0.275
32	-0.7793	0.0071	$3.82 \cdot 10^3$	$1.89 \cdot 10^6$	0.223
64	-0.7850	0.0043	$1.08 \cdot 10^3$	$2.17 \cdot 10^6$	0.282
128	-0.7852	0.0020	$2.90 \cdot 10^2$	$2.36 \cdot 10^6$	0.224
256	-0.7831	0.0010	$7.45 \cdot 10^1$	$2.43 \cdot 10^6$	0.236

^a Rate for a 16-way parallel MC-MP2 calculation.

remains unknown in Eq. (2.20). It’s establishment is discussed in Section 2.3.

There is also a large degree of acceleration to be obtained for a fixed problem size with the redundant-walker algorithm. When using a low number of redundant walkers, the operational cost associated with increasing the number of redundant walkers scales linearly, as the evaluation of the AO (not included in Eq. (2.17)) and MO amplitudes tend to dominate the time per step. Thus, the $O(m^2)$ samples are produced at a $O(m)$ increase in cost, and the overall sampling rate is increased by $O(m)$. As the number of redundant walkers becomes large, the rate-limiting step shifts to the evaluation of the traces of the HF Green’s functions. In this regime, the cost to increase the number of redundant walkers increases as $O(m^2)$ and no sampling boost occurs.

Since the redundant-walker algorithm requires the number of walkers to be proportional to the system size, it is recommended to establish a linear function that causes a maximal sampling rate to occur to achieve the best performance, although any linear function is technically valid. An optimal function can be established from pilot calculations on several molecules that monitor the sampling rate of simulations as a function of the number of redundant walkers.

2.2.2 Results

Table 2.4 presents a series of calculations on benzene ($r_{CC} = 1.3970\text{\AA}$, $r_{CH} = 1.0869\text{\AA}$) as a function of the number of redundant walkers to demonstrate the acceleration provided by the redundant-walker algorithm. Most importantly, it is observed that the standard deviation of the accelerated calculations decreases as predicted from Eq. (2.17).

Turning to the sampling rate, it is found to grow dramatically when the number of walkers is low. As the number of walkers is increased to larger values, the sampling rate boost becomes lesser and tends toward saturation. The sampling rate is observed to increase by a factor of ~ 16 when the number of walkers is

increased from 2 to 256. These trends match the simple analysis of the performance boost for a fixed problem size provided by the redundant-walker algorithm.

Similarly, the MC efficiency also stalls at roughly the same time the sampling rate begins to saturate. From the MC efficiency, it is observed that the calculations using between 16 and 256 walkers are approximately an order of magnitude more efficient than the calculation with the minimal number of walkers. This is a substantial performance increase over the unaccelerated calculation. The stagnation in the MC efficiency appears to be quite troubling when compared to the continually increasing sampling rate. The cause of the stagnation is assigned to increasing autocorrelation present in the uncertainty as the number of redundant walkers is increased. A recently developed technique, the direct sampling of the electronic importance function, which eliminates the autocorrelation, helps to alleviate this stagnation.

It is cautioned that this is a single example of the acceleration provided by the redundant-walker algorithm. The optimum number of redundant walkers to achieve the best efficiency strongly depends on the hardware, basis set, and other acceleration techniques used.

As can be seen from Table 2.4, the redundant-walker algorithm provides a much-needed performance boost for MC-MP2. Further, the redundant-walker algorithm is naturally extended to higher-order methods, and such extensions will be presented as needed. In fact, without the redundant-walker algorithm, the verification of higher-order MC-MB methods would have been significantly more challenging, if not impossible, due to increasing computational demands. Overall, the redundant-walker algorithm has made the application of MC-MP2 more feasible as well as making the development of higher-order methods possible.

2.3 Scaling of MC-MP2

One of the most critical properties to establish for any algorithm is how the algorithms run time changes with problem size. Because MC-MP2 is an MC method, the cost function is established through the scaling of the MC efficiency. As discussed in Chapter 1, the MC efficiency is the product of the operational cost to perform a single MC step and the variance of the problem at hand. The scaling of the operational cost for MC-MP2 was trivially ascertained during the methods proposal, but the scaling of the variance was unable to be established. A numerical study investigating the scaling of the variance was performed during MC-MP2’s proposal but failed to establish the scaling of the variance due to the inability to examine all but the smallest of molecules.

During the proposal of the GPU-accelerated implementation of MC-MP2, MC-MP2 was sufficiently accelerated to perform a numerical study of an adequate extent to establish the scaling of the variance with

Section 2.3 is based off of Reference [10].

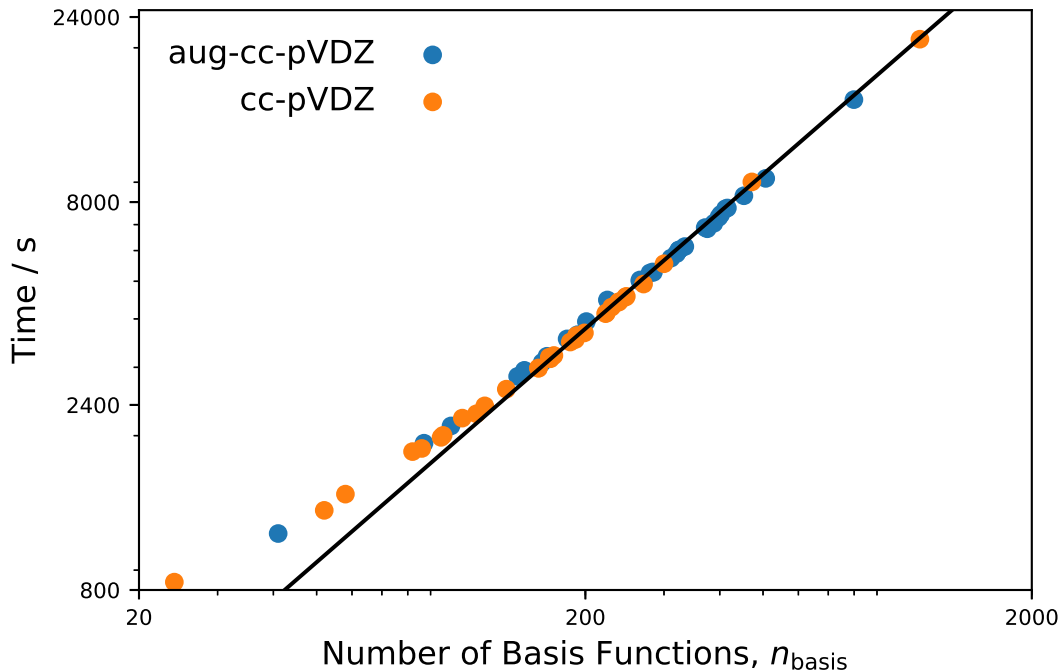


Figure 2.1: The wall time required to perform 16384-step MC-MP2 calculations on molecules ranging in size from water to taxol (tetrahydrocannabinol) using either the cc-pVDZ (aug-cc-pVDZ) basis set as a function of the number of basis functions. The calculations were performed using 512 redundant walkers on single XK node using the GPU-accelerated MC-MP2 algorithm.

respect to system size. At the same time, the ability of the redundant-walker algorithm to make the time per MC step to effectively grow linearly with system size was confirmed.

The numerical experiment was carried out using the cc-pVDZ and aug-cc-pVDZ basis sets to ensure that employing a larger basis set does not affect the rank of the scaling of the variance. The molecules in the study ranged in size from water ($n_{\text{ele}} = 10$) to taxol ($n_{\text{ele}} = 452$) and tetrahydrocannabinol ($n_{\text{ele}} = 172$) for the cc-pVDZ and aug-cc-pVDZ basis sets respectively. All calculations in the study were executed for 16384 MC steps using 512 redundant walkers on a single XK node using the GPU-accelerated implementation of MC-MP2. The raw values of the calculations in this experiment are cataloged in Appendix B.

The log of the simulation time as a function of the log of the number of basis functions, n_{bf} , is presented in Figure 2.1. Given the sufficiently large number of redundant walkers used in the study, the simulation time is observed to grow linearly with respect to the number of basis functions. The defect occurring at a low number of basis functions is due to providing the GPU with an inadequate amount of work, which causes the time per MC step to be greater than expected. This phenomenon is discussed in Section 4.1.

Figure 2.2 presents the log of the relative uncertainty as a function of the log of the number of electrons.

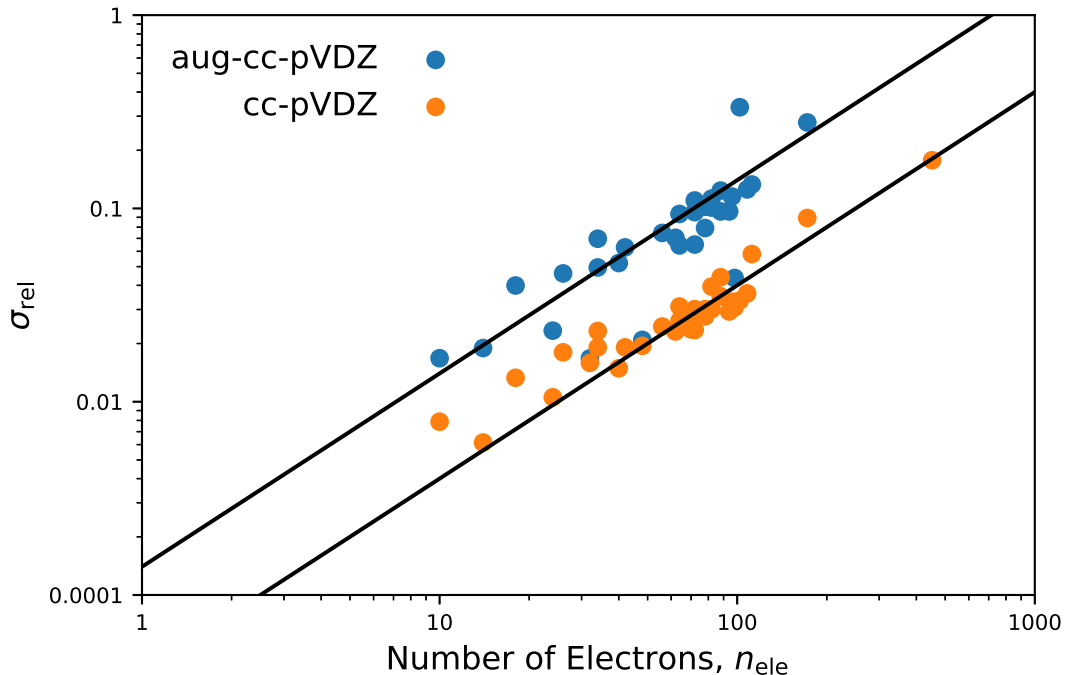


Figure 2.2: The same calculations as Figure 2.1, but showing the relative uncertainty as functions of the number of electrons.

Based on the figure, the relative uncertainty is established to grow linearly with system size.

Combining the conclusions of Figure 2.2 and Figure 2.1, the scaling of the MC efficiency for MC-MP2 to reach a target relative uncertainty is established to be $O(n_{\text{ele}}^3)$. Figure 2.3 presents the MC efficiency as a function of the number of electrons directly and confirms this conclusion. This is a significantly more favorable cost than the $O(n_{\text{ele}}^5)$ scaling of a conventional MP2 implementation[74]. The lower scaling of MC-MP2, however, is tempered by a large prefactor attached to its cost function.

From Figure 2.2, it was observed that the relative uncertainty of the aug-cc-pVDZ calculations were systematically greater than that of the cc-pVDZ calculations. This trend was more fully examined by performing a series of MC-MP2 calculations on benzene using increasingly larger basis sets. These calculations used the same parameters as the preceding study. Figure 2.4 presents the uncertainty as a function of the number of basis functions. Based on this figure, the cost to increase the basis set size, for a fixed number of electrons is observed to be $O(n_{\text{bf}}^{0.75})$. This implies that the cost to increase the basis size for a fixed number of electrons scales as $O(n_{\text{bf}}^{2.5})$. Similar to the cost function to increase the system size, the cost to increase the number of basis functions assuming a fixed number of electrons is more favorable than of a conventional MP2 algorithm, whose cost increase as $O(n_{\text{bf}}^4)$ [74] under these conditions.

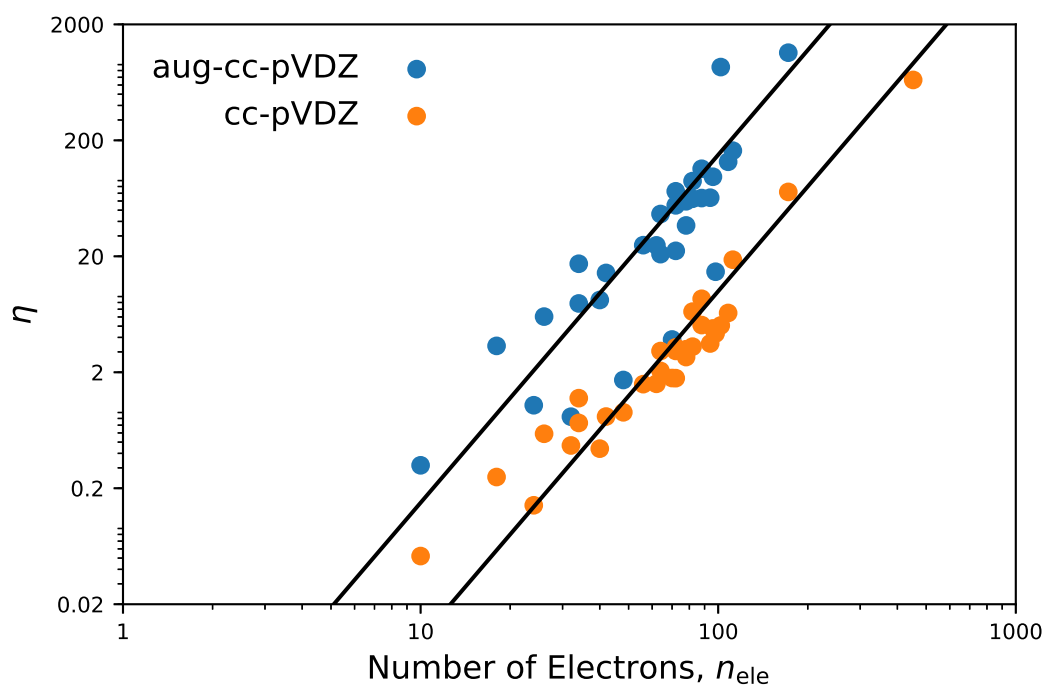


Figure 2.3: The same calculations as Figure 2.1, but showing the MC efficiency as functions of the number of electrons.

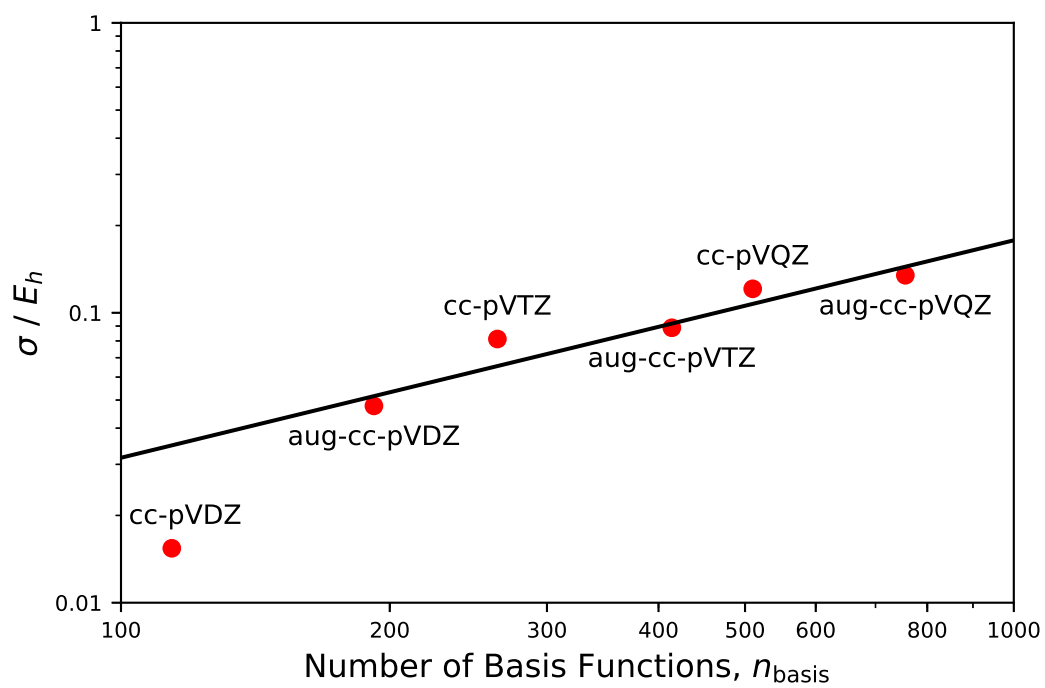


Figure 2.4: The uncertainty of the energy for 16384-step MC-MP2 calculations on benzene as a function of the number of basis functions for basis sets of increasing size. The calculations were performed using 512 redundant walkers on single Blue Waters XK node using the GPU-accelerated MC-MP2 algorithm.

This study critically established the scaling of MC-MP2 with respect to system size and to an increase in basis size for a fixed number of electrons. Both cost scalings are more favorable than those presented by their conventional counterparts. At the time of this study, it was noted that these lower scalings must be tempered by a large prefactor on the cost functions, which makes the crossover point where MC-MP2 will outperform a conventional MP2 algorithm difficult to determine. Additionally, this type of numerical experiment set the standard used to establish the cost function of the other MC-MB methods in the Hirata lab[10–13].

2.4 Diagrammatic Derivation of MC-MB Methods

The Laplace transform formalism used to derive the integral equations for the second-order correlation energy is suitable to produce integral equations for higher-order correlation energies. Briefly, as the equations for the n^{th} -order energy contain $n - 1$ energy denominators, $n - 1$ Laplace transforms are applied energy expression. After applying a suitable number of Laplace transforms, the integrations and summations are interchanged to produce products of the traces of HF Green’s functions.

However, even at third order, proceeding with the Laplace transform formalism becomes cumbersome due to several reasons. Foremost is that the integral equations grow in complexity as the perturbation rank is increased. This is compounded by a rapid increase in the number of diagrams (or equations) defining a theory as the perturbation rank is increased. One strategy that may be used to combat the latter issue is to employ a computer-aided derivation of the integral equations. If this strategy is used, producing sums of two-electron integrals and then transforming them into the integrals equations is also cumbersome. Thus, it is useful to change to a more direct and elegant method in which the Feynman–Goldstone diagrams defining a particular theory are directly interpreted to produce the integral equations for MC-MB methods.

Producing the integral equations in this manner begins by enumerating all of the diagrams defining the desired theory. The rules provided in Chapter 1 to enumerate the diagrams for a particular theory are left unchanged. After enumeration, the diagrams are interpreted directly into integral equations by the rules provided in Table 2.5. In the context of MC-MB methods, this set of rules was first recognized by Willow and Hirata during the development of MC-MP3[15], but was limited to be only applicable to closed Goldstone diagrams. During the development of MC-GF3[13], I expanded the rule set to be relevant to either closed or open Goldstone diagrams.

Rules five and six in Table 2.5 are only required to interpret open self-energy diagrams as found in MBGF methods. The fourth and fifth rules in Table 2.5 may be alternatively combine into a single rule.

Table 2.5: Rules to interpret Feynman–Goldstone diagrams into integral equations for MC-MB methods.

1	Assign each end of a vertex an electron position, $\mathbf{r}_i$.
2	Assign each adjacent vertex pair an imaginary-time coordinate τ_i . Imaginary times are chronologically ordered from bottom to top with $\tau_0 = 0$.
3	Associate a factor of r_{ij}^{-1} with each adjacent vertex pair $\mathbf{r}_i$ and $\mathbf{r}_j$.
4	Associate a factor of $G^-(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o)$ with each downward line with origin $\mathbf{r}_o, \tau_o$ and destination $\mathbf{r}_d, \tau_d$.
5	Associate a factor of $G^+(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o)$ with each upward line with origin $\mathbf{r}_o, \tau_o$ and destination $\mathbf{r}_d, \tau_d$.
6	Associate a factor of $\phi_p^*(\mathbf{r}_o)e^{\omega\tau_o}$ with each stub originating at vertex $\mathbf{r}_o, \tau_o$.
7	Associate a factor of $\phi_q(\mathbf{r}_d)e^{-\omega\tau_d}$ with each stub arriving at vertex $\mathbf{r}_d, \tau_d$.
8	Integrate over all electron coordinates.
9	Integrate over all imaginary-time coordinates.
10	Multiply by $(-1)^{l+n_\tau} 2^l$ where n_τ is the number of imaginary-time coordinates, and l is the number of closed loops.
11	Multiply by a factor of $\frac{1}{2}$ if the diagram has left-right symmetry.
4+5	Associate a factor of $G(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o)$ with each line with origin $\mathbf{r}_o, \tau_o$ and destination $\mathbf{r}_d, \tau_d$.

The combine rule is given at the bottom of Table 2.5. In the combine rule, the lines in a diagram are interpreted into factors of $G(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o)$, which is defined as

$$G(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o) = \theta(\tau_d - \tau_o)G^+(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o) - \theta(\tau_o - \tau_d)G^-(\mathbf{r}_d, \mathbf{r}_o, \tau_d - \tau_o), \quad (2.21)$$

where $\theta(\tau)$ is the Heaviside theta function which is defined as

$$\theta(\tau) = \begin{cases} 0 & \tau \leq 0 \\ 1 & \tau > 0 \end{cases}. \quad (2.22)$$

As a brief example of the application of these rules, the diagram defining the direct contribution to MBPT(2) is considered. Figure 2.5 presents this diagram labeled according to the rules given in Table 2.5, and the combine 4+5 rule. Constructing the resulting integral equation produces

$$E_{\text{direct}}^{(2)} = -1^{2+1}2^2\frac{1}{2} \iiint \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \int d\tau \frac{G(\mathbf{r}_3, \mathbf{r}_1, \tau)G(\mathbf{r}_1, \mathbf{r}_3, -\tau)G(\mathbf{r}_4, \mathbf{r}_2, \tau)G(\mathbf{r}_2, \mathbf{r}_4, -\tau)}{r_{12}r_{34}} \quad (2.23)$$

$$= -2 \iiint \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 \int d\tau \frac{G^+(\mathbf{r}_3, \mathbf{r}_1, \tau)G^-(\mathbf{r}_1, \mathbf{r}_3, -\tau)G^+(\mathbf{r}_4, \mathbf{r}_2, \tau)G^-(\mathbf{r}_2, \mathbf{r}_4, -\tau)}{r_{12}r_{34}} \quad (2.24)$$

where in going to the second line the Heaviside theta functions in G have been applied. This equation is

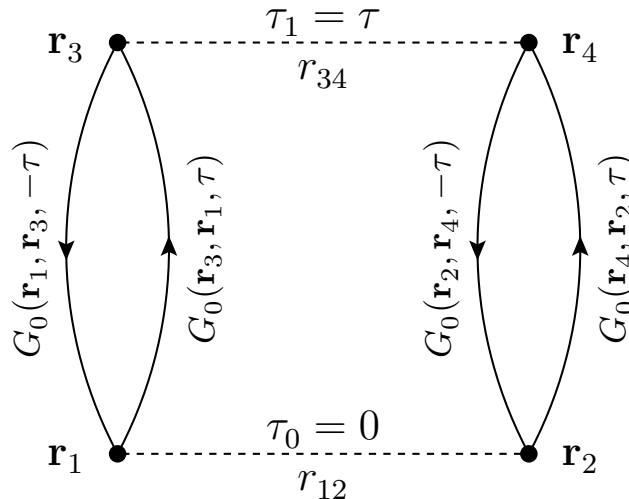


Figure 2.5: The diagram associated with the direct contribution to MBPT(2) labeled according to Table 2.5

exactly identical Eq. (2.4) derived through the Laplace transform formalism.

The rules to interpret diagrams directly into MC integral equations established by Willow and later refined by myself during the development of MC-GF3 have been essential for the development of all MC-MB methods beyond or at third order. They complement another area of research that I have worked on, the automatic generation of arbitrarily high-order Goldstone diagrams[75]. The combination of these two advancements, in principle, allows for high-order MC-MB methods to be created, although, as will be demonstrated during the discussion on MC-MP4, other factors limit their development in practice. The direct interpretation of diagrams will be used to develop the remaining MC-MB methods.

2.5 Third-order Monte Carlo Perturbation Theory

During the development of MC-MP2, it was suggested that the algorithmic techniques and desirable properties of MC-MP2 would apply to higher-order methods. Algorithmic techniques of MC-MP2 that were predicted to be applicable were the electronic importance function, the redundant-walker algorithm, and the general computational framework. As a direct consequence of this, favorable properties predicted to be inherited include the low operational and memory costs required by an MC-MP2 calculation. Additionally, tools used to establish the integral equations and cost scaling functions of MC-MB methods have been discussed. Thus, it is appropriate to examine MC-MP methods beyond second order for the remainder of the chapter. The first realization of such an MC-MB method occurred with the proposal and implementation of MC-MP3 by Willow and Hirata[15]. During the following brief on MC-MP3, the inheritability of the favorable qualities of MC-MP2 by higher-order methods is demonstrated. Several calculations on small molecules

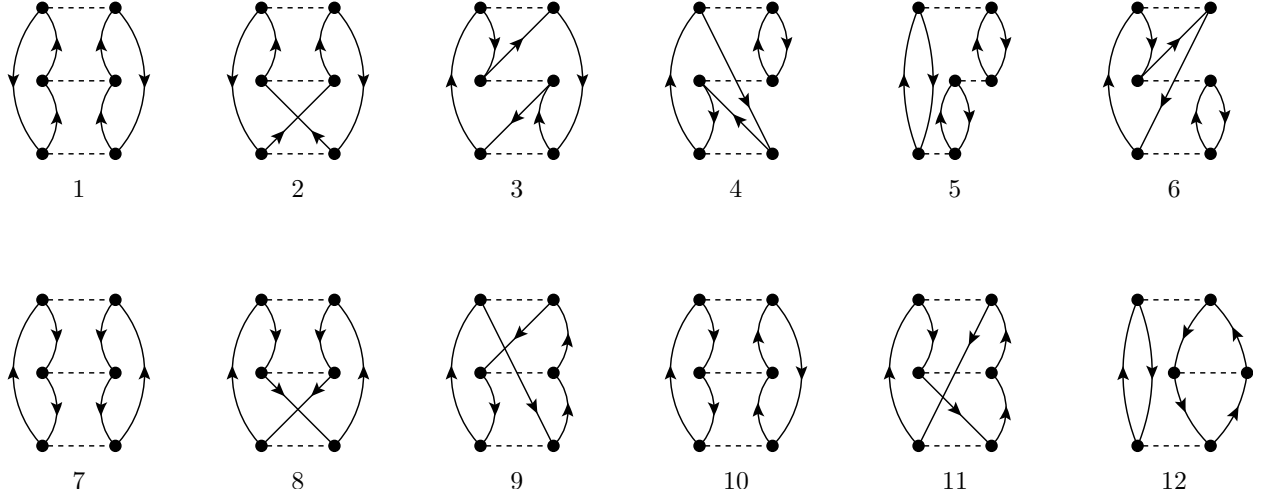


Figure 2.6: The 12 Goldstone diagrams defining the third-order MBPT correction to the energy.

that were used to establish the fundamental viability of MC-MP3 are reproduced and examined. After, the cost scaling function to compute $E^{(3)}$ with respect to system size using MC-MP3 is established for the first time via a numerical experiment.

2.5.1 Algorithm Description

The rules outlined in the previous section make the derivation of the integral equations needed for MC-MP3 trivial. The Goldstone diagrams defining the third-order correction to the energy are presented in Figure 2.6. These diagrams produce 20-dimensional integral equations, 18 dimensions of which are over real-space electron coordinates while the two remaining integrations are over the imaginary-time coordinates. The integrands of these integrals are given in Appendix G. The third-order correlation energy is given by

$$E^{(3)} = \int \cdots \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 d\mathbf{r}_5 d\mathbf{r}_6 \iint_0^\infty d\tau_1 d\tau_2 f^{(3)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \tau_1, \tau_2), \quad (2.25)$$

where $f^{(3)}$ is a sum over the 12 integrand given in Appendix G. The integrals over the real-space electron coordinates in Eq. (2.25) are subject to an MC integration. The integration of the imaginary-time coordinates may be carried out by either a quadrature rule or by MC integration. Assuming the use of a quadrature rule to integrate the imaginary-time coordinates an estimate of the third-order correlation energy is produced by

$$\hat{E}^{(3)[n]} = \frac{1}{N} \sum_{n=1}^N \frac{\tilde{f}^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]})}{\omega^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]})} \quad (2.26)$$

$$= \frac{1}{N} \sum_{n=1}^N I^{(3)[n]}, \quad (2.27)$$

where

$$\tilde{f}^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}) = \sum_{p,q} \nu_p \nu_q f^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \tau_p, \tau_q), \quad (2.28)$$

where ν_p, τ_p are the weights and points of the quadrature rule used to integrate the imaginary-time coordinates. The importance function for the electronic coordinates, $w^{(3)}$, is the third-order extension of the electronic importance function used in MC-MP2. It is the product of three electron pair importance functions,

$$\omega^{(3)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6) = \omega(\mathbf{r}_1, \mathbf{r}_2) \omega(\mathbf{r}_3, \mathbf{r}_4) \omega(\mathbf{r}_5, \mathbf{r}_6). \quad (2.29)$$

The redundant-walker algorithm is readily extended to accelerate the convergence of MC-MP3 calculations. As in the application of the redundant-walker algorithm to MC-MP2, m walkers are propagated. In MC-MP3, three electron pairs are needed to evaluate the integrand of the third-order correlation energy. Thus with m walkers, $O(m^3)$ samples are formed to construct $I^{(3)[n]}$ during each MC step. Under the redundant-walker algorithm, $I^{(3)[n]}$ is modified to be

$$I^{(3)[n]} = \frac{1}{m(m-1)(m-2)} \sum_{i \neq j \neq k}^m \frac{\tilde{f}^{(3)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]})}{\omega^{(3)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]})}. \quad (2.30)$$

As in MC-MP2, careful analysis of the redundant-walker algorithm is required to understand its impacts on the sampling rate and the MC efficiency for MC-MP3. The cost scaling of all of the algorithmic steps when applying the redundant-walker algorithm to MC-MP3 are the same as in MC-MP2, save one, the evaluation of the energy. In MC-MP3, the evaluation of the energy, Eq. (2.30), carries a cost of $O(m^3)$. Using an analysis similar to the one used for MC-MP2, the cost of an MC-MP3 calculation is

$$\text{Cost} = (\text{Operational Cost}) \cdot (\text{Variance}) \quad (2.31)$$

$$\approx (O(m^2 n) + O(mn^2) + O(m^3)) \frac{\sigma_0^2}{O(m^3)}, \quad (2.32)$$

where steps with a total polynomial dependence of less than three have been neglected when approximating the time per step. As in MC-MP2, the number of redundant walkers is made to be proportional to the

problem size. This leads to the expression for the cost to become

$$\text{Cost} \approx O(n^3) \frac{\sigma_0^2}{O(n^3)} \quad (2.33)$$

$$\approx \sigma_0^2. \quad (2.34)$$

The rank of the cost function is reduced with respect to problem size as in MC-MP2. However, it is reduced by two ranks rather than one.

Essential to the above derivation is that the number of electron pairs be increased proportionally to the problem size. An optimal linear function to accomplish this is clearly hardware dependent. Such a function may be easily established from short calculations on molecules of varying sizes. The short calculation are used to determine the number of electron pairs required to achieve a fixed sampling rate (number of evaluations of $\tilde{f}^{(3)}$ per unit time) for each molecule. This data in turn is used to determine a linear function for the number of electron pairs as a function of number of basis function by standard regression techniques.

Since all of the steps for MC-MP2 and MC-MP3 when using the redundant-walker algorithm, except for the evaluation of the energy, are identical, the calculation of $E^{(2)}$ and $E^{(3)}$ are typically carried out using the same trajectory of walkers. This choice causes the estimates of $E^{(2)}$ and $E^{(3)}$ to be correlated. Rather than compute the covariance between the two quantities, it is easier to form an estimate of $E^{(2)} + E^{(3)}$ directly, as this estimate naturally incorporates the correlation.

2.5.2 Illustrative Calculations

During its proposal, several calculations on small molecules were performed to assess the viability of MC-MP3. The results of a reproduction of such calculations on water ($r_{\text{OH}} = 0.9573\text{\AA}$), nitrogen ($r_{\text{NN}} = 1.0975\text{\AA}$), oxygen ($r_{\text{OO}} = 1.2080\text{\AA}$), methane ($r_{\text{CH}} = 1.0848\text{\AA}$), and benzene ($(r_{\text{CC}} = 1.3970\text{\AA}, r_{\text{CH}} = 1.0869\text{\AA})$) at the cc-pVDZ level of chemistry are given in Table 2.6. All of the reproduced results were performed using a single node on the Gellmann cluster and use 32 redundant walkers.

The set of illustrative calculations produces several results of note for the MC-MP3 algorithm. The first is that MC-MP3 can produce an accurate estimate of the third-order correlation energy. Most of the calculations in Table 2.6 achieve either sub or near mE_h uncertainties. The only exception to this is the calculation on benzene, whose uncertainty is on the scale of ten's of mE_h . MC-MP3, and indeed all higher-order MC-MB methods, need to achieve an accuracy on the order of mE_h or better as large uncertainties pollute the total correlation energy, and thus render the calculations meaningless. Benzene makes a clear example of this, as the estimate of $E^{(3)}$ is approximately three times larger than the exact value determined

Table 2.6: Illustrative calculations performed by MC-MP3 at the cc-pVDZ level of chemistry invoking the frozen core approximation. A quadrature rule was used to integrate the imaginary-time coordinates. All simulations are of length 2^{24} MC steps. Energies and uncertainties are given in E_h and times are given in hours.

Molecule	$E^{(3)}$	$\widehat{E}^{(3)}$	σ	Time
Water	-0.007002	-0.00753	0.00044	44.0
Nitrogen	0.004410	0.00403	0.00061	64.7
Oxygen	0.008147	0.00764	0.0008	43.9
Methane	-0.020237	-0.0186	0.0012	49.1
Benzene	-0.033962	-0.106	0.041	69.4

by conventional means. It should be noted that the lack of accuracy for this particular calculation is not particularly troubling, as new the acceleration techniques discussed in Chapter 4 make MC-MP3 significantly more efficient.

Next, the time required to perform the MC-MP3 simulations is examined. The calculations in Table 2.4 and Table 2.6 were performed using the same basis set, number of steps, and computer, abet for MC-MP2 and MC-MP3 calculations respectively. In particular, the calculation using 32 redundant walkers from Table 2.4 is of interest, as is the calculation for benzene in Table 2.6. This pair of calculations will serve as a case study to examine the time requirements of MC-MP3 as compared to MC-MP2. Using MC-MP2, 1.2 hours were required to form an estimate of $E^{(2)}$ for benzene from 2^{24} MC steps. This estimate had an uncertainty of $0.0071 E_h$. Using MC-MP3 to form an estimate of $E^{(3)}$ with the same number of MC steps required 69.4 hours. The estimate of $E^{(3)}$ had an uncertainty of $0.041 E_h$. Unsurprisingly, the MC-MP3 calculation is more expensive than the comparable MC-MP2 calculation. The greater expense is in the sense that it takes more time to perform a similar number of MC steps and that the uncertainties of the results from the MC-MP3 calculation are greater than those from an MC-MP2 calculation. This result is examined in a more general fashion in the next section.

2.5.3 Scaling of MC-MP3

Here, for the first time, the cost function of MC-MP3 is established via a numerical experiment. This consists of two parts. In the first part, the ability to achieve a fixed sampling rate by increasing the making the number of redundant walkers proportional to the problem size is verified. In the second more critical part, the variance of 18 molecules ranging in size from acetylene ($n_{\text{ele}} = 14$ to pentacene ($n_{\text{ele}} = 146$) at the cc-pVDZ level of chemistry is computed. The experiment required the use of several recently developed acceleration techniques. Namely, the electron coordinates were sampled directly rather than with the Metropolis algorithm, and the integration of the imaginary-time coordinates was performed using an MC

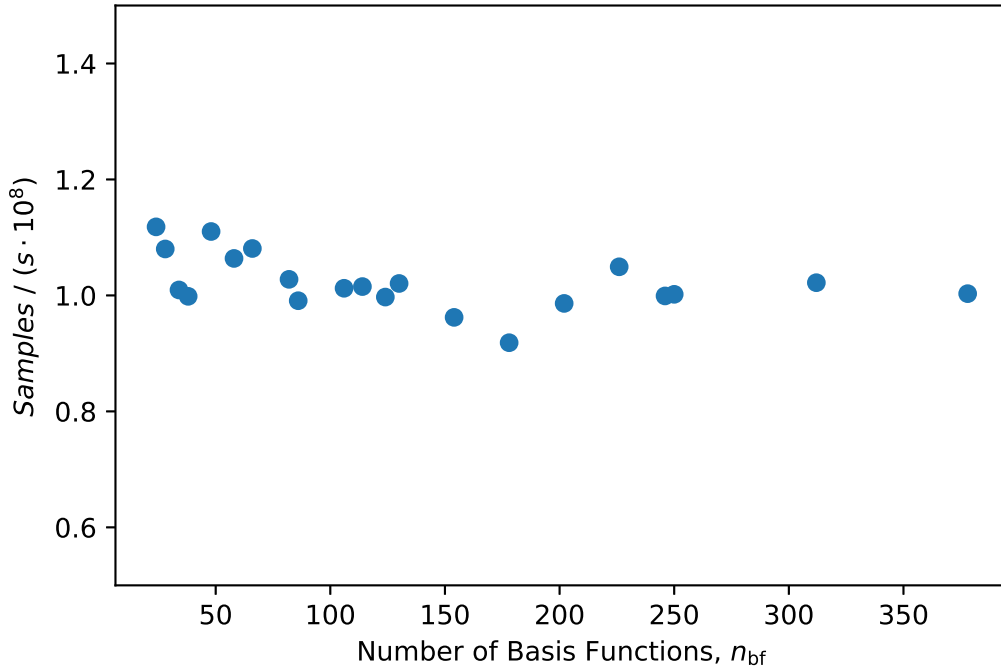


Figure 2.7: The sampling rate of MP-MP3 calculations on molecules ranging in size from acetylene to pentacene as a function of the number of basis functions. The number of electron pairs is equal to $\lfloor 0.55n_{bf} + 52.88 \rfloor$.

integration. All calculations involved in the establishment of the scaling of the variance used 128 redundant walkers. All calculations in this study were performed on a single node of the Gellmann cluster. The exact calculation results may be found in Appendix C.

Figure 2.7 presents the sampling rate of MC-MP3 calculations as function of the number of basis functions, n_{bf} , when the number of electron pairs is set by a function linear in n_{bf} . The linear function for this set of calculations is $\lfloor 0.55n_{bf} + 52.88 \rfloor$. It was prepared by determining the number of electron pairs need to achieve a sampling rate of 10^8 for MC-MP3 calculations on methane, benzene, and pentacene. The sampling rate is stable across the range of molecules with some slight deviations, which are to be expected, and thus it confirms the ability of MC-MP3 to achieve a fixed sampling rate.

The cost of MC-MP3 is established on the ability to reach a target relative uncertainty, and hence the scaling of the relative variance must be established. The relative variance as a function of the number of electrons is presented in Figure 2.8. The trend line is proportional to the forth power of the number of electrons, and thus the variance is taken to scale quartically with system size. In light of Figure 2.8 and Eq. (2.33) the scaling of MC-MP3 is established to be $O(n_{ele}^4)$.

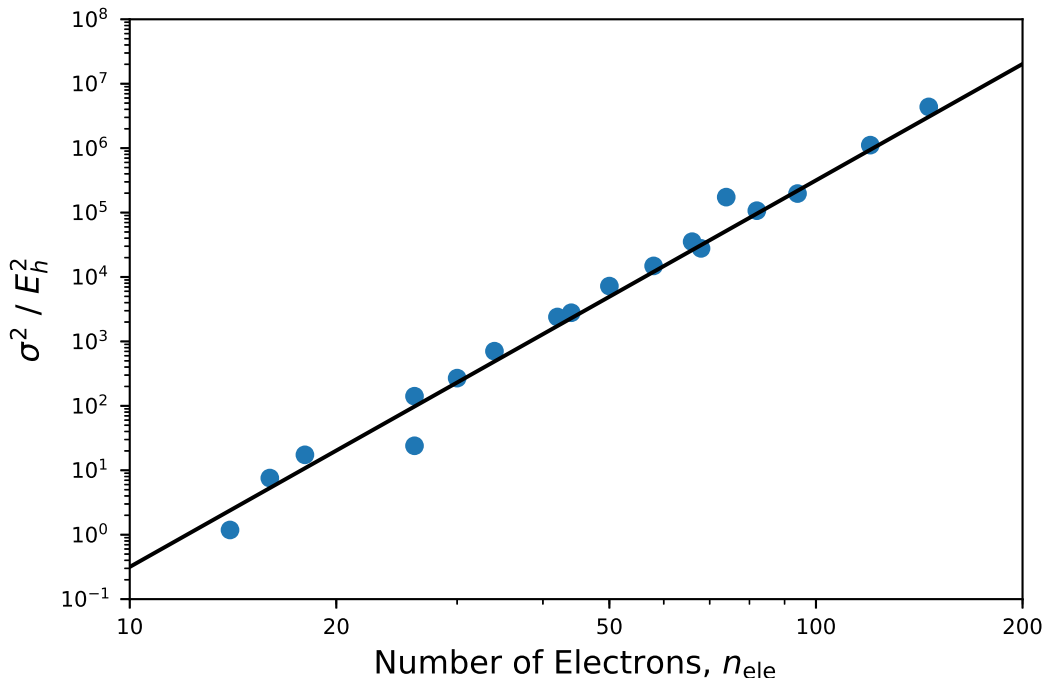


Figure 2.8: The variance of MP-MP3 calculations of molecules ranging in size from acetylene to pentacene as a function of the number of electrons when using 128 redundant walkers.

2.5.4 Conclusions

MC-MP3 was the first method to explore the feasibility of higher rank MC-MB methods. Its development confirmed that higher-order MC-MB methods would inherit the low operational cost and negligible memory footprint of MC-MP2. At the same time, it was observed that MC-MP3 was more costly than MC-MP2 as the uncertainties of $E^{(3)}$ are more significant than those of $E^{(2)}$. This is compounded by MC-MP3 having a more significant operational cost than MC-MP2, even though it does have a low operational cost in the absolute sense.

Here, a cost scaling function of $O(n^4)$ to reach a target relative uncertainty with MC-MP3 has been proposed. The proposed cost function scales two ranks lower than the cost function of the conventional MP3 algorithm, mirroring the results obtained with MC-MP2. These results indicate that the entire family of MC-MB methods may be more scalable than their conventional counterparts and warrants the development of higher-order methods.

2.6 Forth-order Monte Carlo Perturbation Theory

Recently, I have developed MC-MP4 in which $E^{(4)}$ is obtained stochastically. The development of MC-MP4 is an important milestone in the MC-MB hierarchy of methods, as MP4 is known to produce energies that are comparable in accuracy to CCSD(T)[2]. This trait alone demands the investigation of MC-MP4 as it is crucial to allow for high accuracy benchmark calculations to be performed with MC-MB methods. Additional importance is encountered when considering that MP4 has a steep $O(n_{\text{ele}}^7)$ operational cost, requires the use of large intermediate arrays, and is difficult to parallelize. In fact, the literature on efficient parallel MP4 algorithms is rather sparse, and, likely, the best parallel algorithms to perform the $O(n_{\text{ele}}^7)$ step of MP4 are the same as those used to perform the perturbative triples step of CCSD(T)[2, 76, 77]. The MC-MB family of methods sidesteps the steep operational cost and has a small memory footprint. Furthermore, in light of the favorable cost scaling functions of both MC-MP2 and MC-MP3, MC-MP4 will also likely have a lower cost scaling function, which remains to be determined, than its conventional counterpart. In light of all of the above, the development of MC-MP4 is essential.

The computational viability of MC-MP4, however, has proven to be significantly more challenging to establish than either MC-MP2 or MC-MP3. This is due to a two-part obstacle. The first is that for a converging perturbation series, the magnitude of the energy correction is known to decrease as the perturbation rank is increased[78–80]. The second is that the variance of the energy correction is observed to increase with the perturbation rank, as demonstrated when moving from MC-MP2 to MC-MP3. When combined, these features make it difficult to establish if a result is merely correct in the statistical sense, i.e., a t-test does not reject it, or it is convincing correct, i.e., being statistically correct but also presenting a small error from the exact value. The desire to develop MC-MP4 provided much of the motivation to develop many of the acceleration techniques presented in Chapter 4.

2.6.1 Algorithm Description

The integral equations for $E^{(4)}$ are enumerated from its diagrammatic expansion, which contains 300 Goldstone diagrams and is given in Appendix J. The diagrams are interpreted into 27-dimensional integral equations, of which 24 of the integrations are over the electronic coordinates, and the remaining three integrations are over the imaginary-time coordinates. Given the integrands given in Appendix H, $E^{(4)}$ is

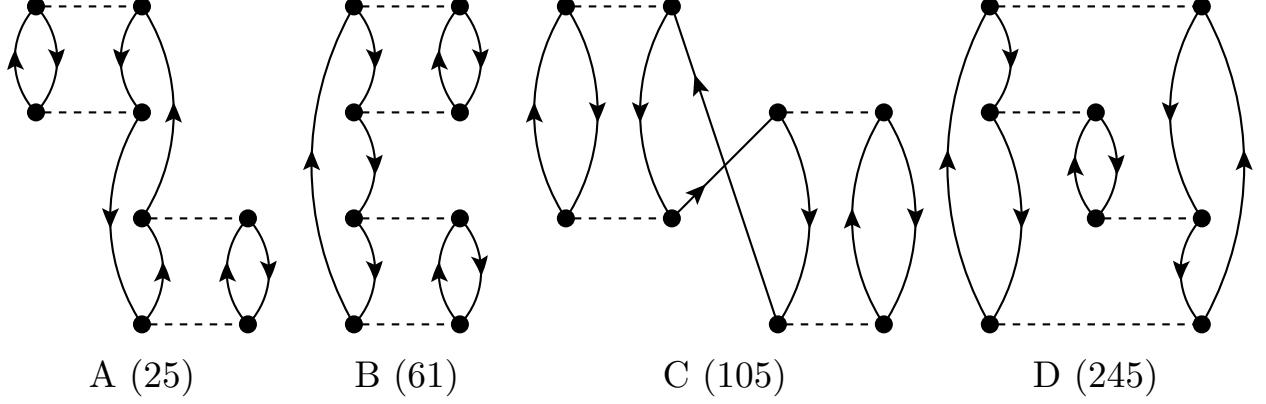


Figure 2.9: Examples of each of the four diagrammatic topologies that the MBPT(4) diagrams are subdivided into for fast evaluation by the redundant-walker algorithm. The numbers in parentheses denote the diagram's label in Appendix J.

obtained from

$$E^{(4)} = \int \cdots \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 d\mathbf{r}_5 d\mathbf{r}_6 d\mathbf{r}_7 d\mathbf{r}_8 \iiint_0^\infty d\tau_1 d\tau_2 d\tau_3 f^{(4)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \mathbf{r}_7, \mathbf{r}_8, \tau_1, \tau_2, \tau_3). \quad (2.35)$$

In MC-MP4, all of the integrals are subject to direct-MC integration, and thus an estimate of $E^{(4)}$ is obtained from

$$\hat{E}^{(4)[N]} = \frac{1}{N} \sum_{n=1}^N \frac{\tilde{f}^{(4)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \mathbf{r}_7^{[n]}, \mathbf{r}_8^{[n]})}{\omega^{(4)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \mathbf{r}_7^{[n]}, \mathbf{r}_8^{[n]})} \quad (2.36)$$

$$= \frac{1}{N} \sum_{n=1}^N I^{(4)[n]}, \quad (2.37)$$

where

$$\tilde{f}^{(4)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \mathbf{r}_7^{[n]}, \mathbf{r}_8^{[n]}) = \frac{\tilde{f}^{(4)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \mathbf{r}_7^{[n]}, \mathbf{r}_8^{[n]}, \tau_1^{[n]}, \tau_2^{[n]}, \tau_3^{[n]})}{\omega_\tau(\tau_1^{[n]})\omega_\tau(\tau_2^{[n]})\omega_\tau(\tau_3^{[n]})}, \quad (2.38)$$

where ω_τ is the importance function for the imaginary-time degrees of freedom. The forth-order importance function for the electronic coordinates is

$$\omega^{(4)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \mathbf{r}_7, \mathbf{r}_8) = \omega(\mathbf{r}_1, \mathbf{r}_2)\omega(\mathbf{r}_3, \mathbf{r}_4)\omega(\mathbf{r}_5, \mathbf{r}_6)\omega(\mathbf{r}_7, \mathbf{r}_8). \quad (2.39)$$

As in MC-MP2 and MC-MP3, the redundant-walker algorithm provides a critical acceleration to the convergence of MC-MP4 calculations. As is expected, in MC-MP4, the integrand is evaluated at all possible

permutations of the redundant walkers, of which there are $O(m^4)$, in order to accelerate the convergence of a simulation. Thus, the contribution to $E^{(4)}$ from a single MC step is given by

$$I^{(4)[n]} = \frac{1}{mP_4} \sum_{i \neq j \neq k \neq l} \frac{\tilde{f}^{(4)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]}, \mathbf{r}_{l1}^{[n]}, \mathbf{r}_{l2}^{[n]})}{\omega^{(4)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]}, \mathbf{r}_{l1}^{[n]}, \mathbf{r}_{l2}^{[n]})}. \quad (2.40)$$

When applying the redundant-walker algorithm to MC-MP4, the naive sum over all combinations of redundant walkers, as in Eq. (2.40), produces suboptimal computational performance. To achieve the best computational performance, the sum of diagrams in $f^{(4)}$ is exposed and exchanged with the sums over redundant walkers,

$$I_n^{(4)} = \frac{1}{mP_4} \sum_{d=1}^{300} \sum_{i \neq j \neq k \neq l} \frac{f^{(4:d)}}{\omega^{(4)} \omega_\tau \omega_\tau \omega_\tau} \quad (2.41)$$

$$= \frac{1}{mP_4} \sum_{d=1}^{300} \frac{f_{\text{RWA}}^{(4:d)}}{\omega_\tau \omega_\tau \omega_\tau}, \quad (2.42)$$

where the arguments to the functions have been omitted for compactness. The sums over redundant walkers for the d^{th} integrand, $f_{\text{RWA}}^{(4:d)}$, are structured depending on the topology of its associated diagram. There are four relevant topological classes, which are labeled A through D, that the diagrams are divided into. A diagram representative of each topological class may be found in Figure 2.9. Classes A through C each contains 36 diagrams with the remaining 192 diagrams belonging to Class D.

In the type A diagrams, when labeling the interaction vertices numerically from top to bottom, no propagator lines connect the first and fourth or the second and third interaction vertices. Because of this particular structure, a subset of the sums over redundant walkers may be carried out independently of the other sums. Using the particular type A diagram in Figure 2.9 as an example, the sums are structured to be

$$\begin{aligned} f_{\text{RWA}}^{(4:A)} = & \sum_{i \neq l} \frac{1}{\omega(\mathbf{r}_{i1}, \mathbf{r}_{i2}) \omega(\mathbf{r}_{l1}, \mathbf{r}_{l2})} \\ & \times \left[\left(\sum_{j \neq i, l} \frac{G^+(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{j1}^{[n]}, \tau_3) G^-(\mathbf{r}_{j1}^{[n]}, \mathbf{r}_{i1}^{[n]}, -\tau_3) G^-(\mathbf{r}_{j2}^{[n]}, \mathbf{r}_{i2}^{[n]}, -\tau_3) G^-(\mathbf{r}_{l1}^{[n]}, \mathbf{r}_{j2}^{[n]}, -\tau_1 - \tau_2)}{\omega(\mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]})} \right) \right. \\ & \times \left(\sum_{k \neq i, l} \frac{G^+(\mathbf{r}_{i2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \tau_2 + \tau_3) G^+(\mathbf{r}_{k1}^{[n]}, \mathbf{r}_{l1}^{[n]}, \tau_1) G^+(\mathbf{r}_{k2}^{[n]}, \mathbf{r}_{l2}^{[n]}, \tau_1) G^-(\mathbf{r}_{l2}^{[n]}, \mathbf{r}_{k2}^{[n]}, -\tau_1)}{\omega(\mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]})} \right) \\ & \left. - \left(\sum_j \frac{f^{(4:d)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{l1}^{[n]}, \mathbf{r}_{l2}^{[n]}, \tau_1, \tau_2, \tau_3)}{\omega(\mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]})^2} \right) \right], \quad (2.43) \end{aligned}$$

where the last term arises to enforce the condition $j \neq k$ present in the original sum. Because all of the

sum in the square braces may be evaluated independently, Eq. (2.43) may be evaluated at a cost of $O(m^3)$, a full rank lower than if the original summation structure was used. The type B and C diagrams have a similar topology to the type A diagrams, in that two pairs of interaction vertices are not connected by any propagator lines. As such, their evaluation with the redundant-walker algorithm may be similarly strength reduced, albeit with slightly modified ordering of the summations over redundant walkers.

In type D diagrams, every interaction vertex is connected to every other interaction vertex by at least one propagator line. No strength reduction is possible for this class of diagrams and $f_{\text{RWA}}^{(4:D)}$ is computed by

$$f_{\text{RWA}}^{(4:D)} = \sum_{i \neq j \neq k \neq l} \frac{f^{(4:d)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]}, \mathbf{r}_{l1}^{[n]}, \mathbf{r}_{l2}^{[n]}, \tau_1, \tau_2, \tau_3)}{\omega^{(4)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]}, \mathbf{r}_{k1}^{[n]}, \mathbf{r}_{k2}^{[n]}, \mathbf{r}_{l1}^{[n]}, \mathbf{r}_{l2}^{[n]})}. \quad (2.44)$$

The appearance of this behavior in the redundant-walker algorithm arising at the MP4 level of theory is quite notable. When developing conventional MBPT algorithms, the first time one must employ strength reducing strategies to achieve the best scaling with system size occurs at fourth-order[2], and thus in some ways, the appearance of the optimizations strategies is somewhat analogous. For conventional MBPT methods, the optimization strategies become more complex as the rank of the theory is increased.[81–86] If the development of MC-MB methods beyond fourth-order is pursued, it is likely that finding optimal summation orderings for the redundant-walker algorithm will similarly be more complicated.

The cost of an MC-MP4 calculation with the redundant-walker algorithm is now considered. The redundant-walker algorithm compresses the variance by a factor of $O(m^4)$ as the MC-MP4 integrand is sampled $O(m^4)$ times. The algorithmic complexity of all steps in MC-MP4 is the same as in MC-MP2 and MC-MP3, with the exception of the energy correction. The algorithmic complexity of computing the energy correction is $O(m^3 + m^4) = O(m^4)$, where the cubic term arises from the evaluation of type A, B, and C diagrams and the quartic term arises from the evaluation of type D diagrams. The cost function of MC-MP4 with the redundant-walker algorithm is

$$\text{Cost} = (\text{Operational Cost}) \cdot (\text{Variance}) \quad (2.45)$$

$$\approx (O(m^2n) + O(mn^2) + O(m^4)) \frac{\sigma_0^2}{O(m^4)}. \quad (2.46)$$

Formally, increasing the walker population as the 2/3 power of the problem size is sufficient to mask the operational cost by the variance compression. However, MC-MP2 and MC-MP3 both require the walker population to be linearly proportional to the problem size. Thus, to remain consistent, the electron pair population for MC-MP4 should be made linearly proportional to the problem size as well. Under this

Table 2.7: Illustrative calculations performed by MC-MP4 at the cc-pVDZ level of chemistry invoking the frozen core approximation. Energies and uncertainties are given in E_h .

Molecule	$E^{(4)}$	$\widehat{E}^{(4)}$	σ
Methane	-0.0059	-0.0028	0.0031
Ammonia	-0.0058	-0.0077	0.0023
Water	-0.0052	-0.0063	0.0012
Benzene	-0.0426	-0.13	0.78

condition the cost function becomes

$$\text{Cost} \approx (O(n^3) + O(n^4)) \frac{\sigma_0^2}{O(n^4)} \quad (2.47)$$

$$\approx \sigma_0^2. \quad (2.48)$$

As in MC-MP3, the cost scaling with system size of MC-MP4 is reduced by two ranks by the redundant-walker algorithm and all that remains is to establish the cost function of the variance.

2.6.2 Illustrative Calculations

The results from several MC-MP4 calculations on water ($r_{\text{OH}} = 0.9573\text{\AA}$), methane ($r_{\text{CH}} = 1.0848\text{\AA}$), ammonia ($r_{\text{NH}} = 1.0123\text{\AA}$), and benzene ($r_{\text{CC}} = 1.3970\text{\AA}$, $r_{\text{CH}} = 1.0869\text{\AA}$) are presented in Table 2.7. All calculations use the cc-pVDZ basis set, were run for 3481600 MC steps and utilized 64 redundant walkers. The calculations were performed on 10 Blue Waters XE nodes, each requiring ~ 5 wall-clock hours to complete.

The fourth-order correlation energy, as calculated by MC-MP4, is clearly in agreement with the deterministic results for the ten-electron molecules, although, the uncertainty is still significant. For benzene, the uncertainty is much larger than the energy correction, and therefore, the result is of limited use. Extending the number of MC samples by at least a factor of 100 would be required to provide more qualitative results.

However, $E^{(4)}$ is a relatively small component of the total energy correction. The uncertainties presented by $E^{(4)}$ and the sum of $E^{(2)} + E^{(3)} + E^{(4)}$ are nearly identical. As such, the large relative uncertainty may be masked when looking at their total, as can be seen in Table 2.8 for the ten-electron molecules. For benzene, the large uncertainty pollutes the result and makes the calculated correlation energy less useful, than say the MC-MP2 or MC-MP3 result.

The conclusions on the viability of MC-MP4 are quite similar to those drawn during the initial presentation of MC-MP3. MC-MP4 is more expensive than MC-MP3, in both the operational cost and the variance, just as MC-MP3 is more expensive than MC-MP2. As was the case then, the fact that MC-MP4 is more

Table 2.8: The same as Table 2.7, but for the total correlation-energy through fourth order. Energies and uncertainties are given in E_h .

Molecule	E	$\widehat{E}$	σ
Ammonia	-0.2061	-0.2086	0.0024
Methane	-0.1871	-0.1837	0.0031
Water	-0.2139	-0.2146	0.0013
Benzene	-0.8600	-0.87	0.78

expensive is unsurprising. The scaling of the uncertainty with problem size currently remains unexplored and needs to be established. The cost of the simulations in Table 2.7 were significant, and a numerical study of sufficient size to explore the cost scaling of the variance would be challenging to perform. A GPU-accelerated implementation of MC-MP4 will likely make the numerical study significantly more reasonable. Additionally, one of the newly developed acceleration techniques presented in Section 4.4 makes MC-MP4 more computationally feasible.

2.6.3 Conclusions

The development of MC-MP4 represents a significant milestone in the timeline of the MC-MB methods. In its development, a new characteristic of the redundant-walker algorithm, the need for strength reducing strategies, has emerged. With MC-MP4, it is now possible to obtain high accuracy energies using the MC-MB family of methods. However, the uncertainties presented for even small molecules are rather significant. Already, work to reduce the uncertainty from the outset of a calculation has been conducted. This work makes MC-MP4 significantly more viable. This penalty, as compared to a conventional MP4 algorithm, is also offset by MC-MP4’s ability to use hundreds or even thousands of processors readily. Further, porting the MC-MP4 algorithm to utilize GPU acceleration should be a relatively easy task and will significantly extend the applicability of MC-MP4. All of these aspects make MC-MP4 an excellent method to employ on the current and next generation of supercomputers.

2.7 Conclusions

This chapter largely followed the historical timeline for the development of the MC-MP methods to highlight the most important features of the MC-MB family methods. The feature includes the low operational cost to perform a single MC step, the trivial memory requirements for the MC-MB methods, as well as the high degree of parallel efficiency that these methods achieve. During this discussion, two critical innovations toward the MC-MB family of methods were encountered. The first was the importance function for the

electron coordinates proposed during the development of MC-MP2. This importance function critically cancels the Coulomb singularities in the integrands of the MC-MB methods. Additionally, from a historical perspective, the ability to extend this importance function to methods beyond second order has been of great value. The second was the redundant-walker algorithm, which provides a critical acceleration to the integration of the electronic coordinates in MC-MB methods. The use of the redundant-walker algorithm is essential to the development of MC-MB methods beyond second order.

Two new developments were also presented in this chapter. The first is the establishment of the cost scaling function of MC-MP3. In this study, the cost as a function of system size to reach a specified level of relative uncertainty is found to be $O(n^4)$. This cost is two ranks lower than a conventional MP3 implementation. The lower rank scaling of MC-MP2 and MC-MP3 than their conventional counterparts suggests that the entire family of MC-MP methods are likely more efficient than their conventional counterparts. The lower cost scaling is mitigated by the cost scaling function having large prefactors, making the point at which an MC-MP method is actually more efficient than its conventional counterpart difficult to ascertain. The second, and more important, development is the proposal and implementation of MC-MP4. The value of MC-MP4 lies in the fact that MP4 is the first perturbation theory method that is capable of providing benchmark quality energies for electronic structure problems.

Chapter 3

Monte Carlo Many-body Green's Function

This chapter examines the MC-GF family of methods. Historically, the MC-GF methods have been limited to MC-GF2 within the frequency-independent diagonal approximation. Recently, the usability of the MC-GF methods was greatly enhanced in many directions. The most important of the recent developments is lifting the truncation level of the self-energy to third order. The diagonal approximation has been lifted, and an efficient algorithm to calculate the entire self-energy matrix was proposed. The frequency-independent approximation was raised by approximating the self-energy in a truncated Taylor series. The truncated Taylor series representation of the self-energy is then used to locate a solution to the Dyson equation. During this development, the computational characteristics of the MC-GF methods were extensively studied. Rather than pursue a historical development of the MC-GF methods, as was presented for the MC-MP methods in the previous chapter, the algorithms for the MC-GF methods are presented, followed by the studies on the computational characteristics of the methods.

3.1 Methods

3.1.1 Second-order Self-energy

In 2013, Willow *et. al* developed Monte Carlo GF2 (MC-GF2)[16] in which the diagonal elements of the second-order self-energy are obtained stochastically. The integral equations defining the second-order were derived using the Laplace transform formalism at the time of MC-GF2's original proposal. Here, the integral equations are inferred using the rules listed in Table 2.5 and the diagrams in Figure 1.4. As the GF2 is a second-order method, the diagrams produce 13-dimensional equations, which may be written as

$$\Sigma_{pq}^{(2)}(\omega) = \iiint \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \mathrm{d}\mathbf{r}_3 \mathrm{d}\mathbf{r}_4 \int_0^\infty \mathrm{d}\tau f_{pq}^{(2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau) \mathbf{r}_{12}^{-1} \mathbf{r}_{34}^{-1}, \quad (3.1)$$

Chapter 3 is based off of Reference [13].

where the subscripts identify the element of self-energy matrix. The integrand of Eq. (3.1) is

$$f_{pq}^{(2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau) = \sum_{d=1}^4 f_{pq}^{(2:d)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau), \quad (3.2)$$

where d labels each diagram in Figure ???. The integrands from each diagram are

$$f_{pq}^{(2:1)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau) = 2G^-(\mathbf{r}_3, \mathbf{r}_1, -\tau)G^-(\mathbf{r}_4, \mathbf{r}_2, -\tau)G^+(\mathbf{r}_2, \mathbf{r}_4, \tau)\varphi_p(\mathbf{r}_1)\varphi_q(\mathbf{r}_3)e^{-\omega\tau}, \quad (3.3)$$

$$f_{pq}^{(2:2)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau) = -G^-(\mathbf{r}_3, \mathbf{r}_1, -\tau)G^-(\mathbf{r}_4, \mathbf{r}_2, -\tau)G^+(\mathbf{r}_2, \mathbf{r}_3, \tau)\varphi_p(\mathbf{r}_1)\varphi_q(\mathbf{r}_4)e^{-\omega\tau}, \quad (3.4)$$

$$f_{pq}^{(2:3)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau) = -2G^-(\mathbf{r}_4, \mathbf{r}_2, -\tau)G^+(\mathbf{r}_1, \mathbf{r}_3, \tau)G^+(\mathbf{r}_2, \mathbf{r}_4, \tau)\varphi_p(\mathbf{r}_1)\varphi_q(\mathbf{r}_3)e^{\omega\tau}, \quad (3.5)$$

$$f_{pq}^{(2:4)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \omega, \tau) = G^-(\mathbf{r}_4, \mathbf{r}_1, -\tau)G^+(\mathbf{r}_1, \mathbf{r}_3, \tau)G^+(\mathbf{r}_2, \mathbf{r}_4, \tau)\varphi_p(\mathbf{r}_2)\varphi_q(\mathbf{r}_3)e^{\omega\tau}. \quad (3.6)$$

The terms $G^\pm(\mathbf{r}, \mathbf{r}', \tau)$ have been defined previously in Eq. (2.5) and Eq. (2.6).

By applying the integration scheme of MC-MP2 to the above integration problem, an estimate of an element of the second-order self-energy is formed by

$$\widehat{\Sigma}_{pq}^{(2)[N]}(\omega) = \frac{1}{N} \sum_{n=1}^N I_{pq}^{(2)[n]}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \omega), \quad (3.7)$$

where

$$I_{pq}^{(2)[n]}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \omega) = \frac{\tilde{f}_{pq}^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \omega)}{\omega^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]})}. \quad (3.8)$$

The integrals over the imaginary-time coordinates were carried out by a quadrature rule in the original implementation of MC-GF2. However, the relevant results and illustrative calculations in this chapter were all produced by carrying out these integrals by direct-MC. Proceeding with the direct-MC integrations, $\tilde{f}_{pq}^{(2)}$ takes the definition

$$\tilde{f}_{pq}^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \omega) = \frac{f_{pq}^{(2)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \tau^{[n]}\omega)}{\omega_\tau(\tau^{[n]})}. \quad (3.9)$$

3.1.2 Third-order Self-energy

The third-order correction to the self-energy[6, 53, 54, 87, 88] is defined as

$$\Sigma_{pq}^{(3)}(\omega) = \sum_{d=1}^{84} \Sigma_{pq}^{(3:d)}(\omega), \quad (3.10)$$

where the sum runs over all third-order self-energy diagrams in the Goldstone style shown in Figure 3.1. As in MC-GF3, they are directly converted into algebraic expressions convenient for MC integrations according to the set of interpretation rules given in Table 2.5. As an example, diagram 35 of Figure 3.1 can be converted to an algebraic formula convenient for MC-GF3 according to the rules in Table 2.5 as follows. It is interpreted to be

$$\begin{aligned}
\Sigma_{pq}^{(3;35)}(\omega) = & (-1)^{1+1+2} 2^1 \int \cdots \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 d\mathbf{r}_5 d\mathbf{r}_6 \int_0^\infty d\tau_1 \int_0^\infty d\tau_2 \\
& \times \frac{1}{r_{12}} \frac{1}{r_{34}} \frac{1}{r_{56}} \\
& \times G^+(\mathbf{r}_5, \mathbf{r}_3, \tau_2) G^+(\mathbf{r}_3, \mathbf{r}_1, \tau_1) G^+(\mathbf{r}_6, \mathbf{r}_4, \tau_2) G^+(\mathbf{r}_4, \mathbf{r}_2, \tau_1) \\
& \times G^-(\mathbf{r}_2, \mathbf{r}_6, -\tau_1 - \tau_2) \varphi_p^*(\mathbf{r}_5) e^{i\omega(\tau_1 + \tau_2)} \varphi_q(\mathbf{r}_1) e^{-i\omega 0}.
\end{aligned} \tag{3.11}$$

The integrand of the above and the remaining 83 integrands for the third-order self-energy may be found in Appendix I.

Each element of the third-order self-energy matrix is ultimately written as

$$\begin{aligned}
\Sigma_{pq}^{(3)}(\omega) = & \int \cdots \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_3 d\mathbf{r}_4 d\mathbf{r}_5 d\mathbf{r}_6 \int_0^\infty d\tau_1 \int_0^\infty d\tau_2 \\
& \times f_{pq}^{(3)}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6, \tau_1, \tau_2, \omega)
\end{aligned} \tag{3.12}$$

$$\approx \widehat{\Sigma}_{pq}^{(3)[N]}(\omega), \tag{3.13}$$

where $f_{pq}^{(3)}$ is the sum of all integrands in Appendix I obtained by applying the aforementioned interpretation rules to all 84 third-order self-energy diagrams, and $\widehat{\Sigma}_{pq}^{(3)[N]}(\omega)$ is an estimate of the integral obtained after N steps of the MC-integration algorithm.

As was the case for the MC-MP methods, the MC integrations for the self-energy may be carried out in one of the following two ways. The integrations over the real-space electron coordinates are, of course, performed stochastically. The integrals over the imaginary-time coordinates, on the other hand, may be obtained with either a quadrature rule[14–17] or by direct-MC. As in MC-GF2, we shall proceed with only the direct-MC integration in this chapter. Therefore,

$$\widehat{\Sigma}_{pq}^{(3)[N]}(\omega) = \frac{1}{N} \sum_{n=1}^N I_{pq}^{(3)[n]}(\omega), \tag{3.14}$$

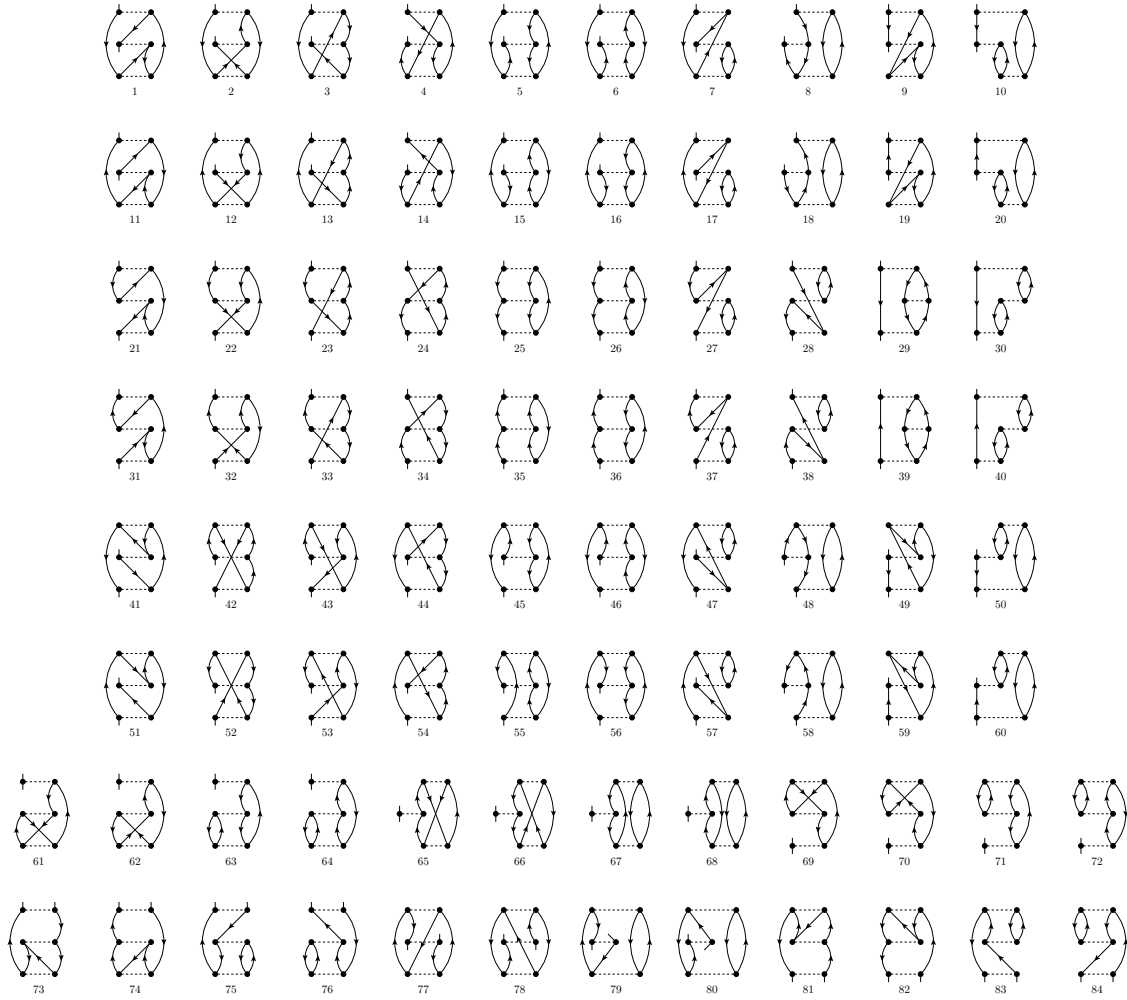


Figure 3.1: The 84 third-order self-energy diagrams in the Goldstone style.

with

$$I_{pq}^{(3)[n]}(\omega) = \frac{\tilde{f}_{pq}^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \omega)}{\omega^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]})}. \quad (3.15)$$

The integrations over imaginary-time variables are also carried out by the MC method. Hence, the definition of $\tilde{f}_{pq}^{(3)}$ in Eq. (3.15) becomes

$$\tilde{f}_{pq}^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \omega) = \frac{f_{pq}^{(3)}(\mathbf{r}_1^{[n]}, \mathbf{r}_2^{[n]}, \mathbf{r}_3^{[n]}, \mathbf{r}_4^{[n]}, \mathbf{r}_5^{[n]}, \mathbf{r}_6^{[n]}, \tau_1^{[n]}, \tau_2^{[n]}, \omega)}{w_\tau(\tau_1^{[n]})w_\tau(\tau_2^{[n]})}. \quad (3.16)$$

3.1.3 Non-diagonal Self-energy and the Redundant-walker Algorithm

As was discussed in the previous chapter, it is essential to use the redundant-walker convergence acceleration[73] to make MC-MP n and MC-GF n , particularly with $n \geq 3$ viable. With redundant walker acceleration scheme, Eq. (3.8) is adjusted to

$$I_{pq}^{(2)[n]}(\omega) = \frac{1}{mP_2} \sum_{i=1}^m \sum_{j \neq i}^m \frac{\tilde{f}_{pq}^{(2)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \omega)}{\omega^{(2)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]})}. \quad (3.17)$$

Similarly, Eq. (3.15) is modified to

$$I_{pq}^{(3)[n]}(\omega) = \frac{1}{mP_3} \sum_{i=1}^m \sum_{j \neq i}^m \sum_{k \neq i \neq j}^m \frac{\tilde{f}_{pq}^{(3)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]}, \omega)}{\omega^{(3)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]})}. \quad (3.18)$$

The acceleration of MC-GF methods by the redundant-walker algorithm, particularly if the full self-energy matrix is desired, introduces intricacies not encountered during the acceleration of MC-MP methods. The differences that arise between the application of the redundant-walker algorithm to the two families of methods is instructive to consider before resolving the additional complexities. Here, the differences are only explored in the context of MC-MP3 and MC-GF3 for two reasons. (1) The differences, their effects, and their resolution are the same at both the second- and third-order levels of theory. (2) The acceleration provided by the redundant-walker algorithm is more vital for the feasibility of the third-order methods, and thus, the differences that arise are naturally more impactful on their performance.

The key difference between MC-MP3 and MC-GF3 is that there are n_{orb}^2 integrals, $\bar{\Sigma}_{pq}$, in the latter, while there is only one, say, $\bar{E}^{(3)}$, in the former. In MC-MP3, the sampling rate is increased by a factor of mP_3 or $O(m^3)$ by this algorithm with m walkers, relative to the unaccelerated case. The cost grows by a factor of $O(n_{\text{orb}}^2 m + n_{\text{orb}} m^2 + m^3)$, where the first, second, and third terms arise from the steps evaluating the MO amplitudes, Green's function traces, and MC-MP3 summand (analogous to only one element of $I_{pq}^{[n]}$),

Table 3.1: Groups of the third-order self-energy diagrams.

Group	Diagrams
A	1–60
B	61–72
C	73–84
a	1–10
b	11–20
c	21–30
d	31–40
e	41–50
f	51–60
g	61–84

respectively. When the problem size is sufficiently large as compared to the number of walkers ($n_{\text{orb}} \gg m$), the first term dominates in the cost function, leading to the overall sampling efficiency boost of $O(m^2)$. If m is increased further, the rate-determining step shifts from the $O(n_{\text{orb}}^2 m)$ step to the $O(n_{\text{orb}} m^2)$ step and eventually to the $O(m^3)$ step, dampening the efficiency boost and ultimately saturating it. The optimal value of m is at the onset of the saturation, which is roughly proportional to n_{orb} . This analysis is the essentially the same as that presented in the previous chapter, where a large speedup and its saturation with m has been numerically demonstrated in the case of MC-MP2.[10]

In MC-GF2 or MC-GF3, a full self-energy matrix with its n_{orb}^2 elements is desired. A naive application of the foregoing algorithm to each individual element of the third-order self-energy would make the third step of evaluating $I_{pq}^{[n]}$ an $O(n_{\text{orb}}^2 m^3)$ process, negating any convergence acceleration. The costs of the first (MO-amplitude-evaluation) and second (Green’s function trace evaluation) steps remain unchanged at $O(n_{\text{orb}}^2 m)$ and $O(n_{\text{orb}} m^2)$. However, the full convergence acceleration can be restored by this algorithm in the following way:

First, the 84 diagrammatic contributions are divided into three groups (see Table 3.1): Group A consists of diagrams 1–60, group B diagrams 61–72, and group C diagrams 73–84. Group A diagrams are the ones with two stubs on two different vertices, group B diagrams have two stubs on the one end of the same vertex, and group C diagrams have two stubs on two different ends of the same vertex. The following ‘core matrix’ elements can then be defined in the sum of Eq. (3.18) for diagram 1 in group A as

$$g_{ij}^{(3:1)}(\tau_1^{[n]}, \tau_2^{[n]}, \omega) = \sum_{k \neq i \neq j}^m \frac{1}{\varphi_p(\mathbf{r}_{1i}^{[n]})\varphi_q(\mathbf{r}_{1j}^{[n]})} \frac{f_{pq}^{(3:1)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]}, \tau_1^{[n]}, \tau_2^{[n]}, \omega)}{w(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]})w(\mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]})w(\mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]})}, \quad (3.19)$$

for $i \neq j$ (hence $g_{ii}^{(3:1)} = 0$), where superscript ‘(3:d)’ designates diagram d of the third-order self-energy. Note that the left-hand side no longer carries the pq index. The same can be defined for all diagrams in

group A. For diagram 61 in group B, as an example, the core matrix element can similarly define as

$$g_{ii}^{(3:61)}(\tau_1^{[n]}, \tau_2^{[n]}, \omega) = \sum_{j \neq i}^m \sum_{k \neq i \neq j}^m \frac{1}{\varphi_p(\mathbf{r}_{1i}^{[n]})\varphi_q(\mathbf{r}_{1i}^{[n]})} \frac{f_{pq}^{(3:61)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]}, \tau_1^{[n]}, \tau_2^{[n]}, \omega)}{w(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]})w(\mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]})w(\mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]})}, \quad (3.20)$$

and $g_{ij}^{(3:61)} = 0$ for $i \neq j$. For diagram 73 in group C, in turn, the core matrix must be defined in a slightly different manner as

$$g_{ii}^{(3:73)}(\tau_1^{[n]}, \tau_2^{[n]}, \omega) = \sum_{j \neq i}^m \sum_{k \neq i \neq j}^m \frac{1}{\varphi_p(\mathbf{r}_{1i}^{[n]})\varphi_q(\mathbf{r}_{2i}^{[n]})} \frac{f_{pq}^{(3:73)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]}, \tau_1^{[n]}, \tau_2^{[n]}, \omega)}{w(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]})w(\mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]})w(\mathbf{r}_{1k}^{[n]}, \mathbf{r}_{2k}^{[n]})}, \quad (3.21)$$

and $g_{ij}^{(3:61)} = 0$ for $i \neq j$.

Then, using these $m \times m$ core matrices $\mathbf{g}$, rewrite Eq. (3.18) can be rewritten as

$$I_{pq}^{[n]}(\tau_1^{[n]}, \tau_2^{[n]}, \omega) = \frac{1}{w(\tau_1^{[n]})w(\tau_2^{[n]})} \frac{1}{mP_3} \sum_{i=1}^m \varphi_p(\mathbf{r}_{1i}^{[n]}) \left\{ \sum_{j=1}^m \left(g_{ij}^{(A)} + g_{ij}^{(B)} \right) \varphi_q(\mathbf{r}_{1j}^{[n]}) + \sum_{j=1}^m g_{ij}^{(C)} \varphi_q(\mathbf{r}_{2j}^{[n]}) \right\}, \quad (3.22)$$

with

$$g_{ij}^{(A)} = \sum_{d=1}^{60} g_{ij}^{(3:d)}; \quad g_{ij}^{(B)} = \sum_{d=61}^{72} g_{ij}^{(3:d)}; \quad g_{ij}^{(C)} = \sum_{d=73}^{84} g_{ij}^{(3:d)}. \quad (3.23)$$

The cost of the inner matrix multiplication (the summation over j) in Eq. (3.22) is $O(n_{\text{orb}}m^2)$. The cost of the outer matrix multiplication (the summation over i) is $O(n_{\text{orb}}^2m)$. The cost of constructing each of the core matrices $\mathbf{g}$ is $O(m^3)$, and the overall cost of the redundant-walker algorithm for the full self-energy-matrix evaluation is, therefore, $O(n_{\text{orb}}^2m + n_{\text{orb}}m^2 + m^3)$ as before. The same applies to the quadrature integrations in the τ dimensions. This is an example of common subexpression elimination followed by the simplest form of matrix chain ordering problem.

The redundant-walker algorithm is similarly adjusted for MC-GF2 without the diagonal approximation to the self-energy, but with the following simplifications. First, since there is no static contribution to the self-energy at second order, the diagrams defining the second-order self-energy are all essentially group A diagrams. Secondly, since the second-order diagrams require two redundant walkers to evaluate, their in no need to perform a summation to evaluate the core matrix. As an example, the core matrix for diagram 1 of Figure 1.4 is written as

$$g_{ij}^{(2:1)}(\tau_1^{[n]}, \omega) = \frac{1}{\varphi_p(\mathbf{r}_{1i}^{[n]})\varphi_q(\mathbf{r}_{1j}^{[n]})} \frac{f_{pq}^{(2:1)}(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]}, \mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]}, \tau_1^{[n]}, \omega)}{w(\mathbf{r}_{1i}^{[n]}, \mathbf{r}_{2i}^{[n]})w(\mathbf{r}_{1j}^{[n]}, \mathbf{r}_{2j}^{[n]})}. \quad (3.24)$$

The second-order self-energy Eq. (3.17) can be rewritten as

$$I_{pq}^{[n]}(\tau_1^{[n]}, \omega) = \frac{1}{w(\tau_1^{[n]})} \frac{1}{mP_2} \sum_{i=1}^m \varphi_p(\mathbf{r}_{1i}^{[n]}) \left\{ \sum_{j=1}^m g_{ij}^{(2)} \varphi_q(\mathbf{r}_{1j}^{[n]}) \right\}. \quad (3.25)$$

Because only two electron-coordinates are required to evaluate the second-order self-energy, the naive evaluation of the self-energy carries a cost of $O(n_{\text{orb}}^2 m^2)$. Using the core matrices, the evaluation of the second-order self-energy becomes dominated by Eq. (3.17), which carries a cost of $O(n_{\text{orb}}^2 m + n_{\text{orb}} m^2)$. While the total reduction in the cost as compared to the naive evaluation may not be as significant in the second-order case, it is still essential to maintain the efficacy of the redundant-walker algorithm.

3.1.4 Frequency-dependent Self-energy

Here, an efficient method of solving the Dyson equation [Eq. (1.18)] is proposed without resorting to the frequency-independent approximation, which is expected not only to improve the accuracy of Koopmans' electron binding energies but also to have roots corresponding to non-Koopmans states.

It approximates the self-energy (in the perturbation approximation of any order) by its Taylor series truncated at order k ,

$$\Sigma_{pq}(\omega) \approx \Sigma_{pq}(\omega_0) + \sum_{n=1}^k \frac{1}{n!} \frac{\partial^n \Sigma_{pq}(\omega)}{\partial \omega^n} \Big|_{\omega=\omega_0} (\omega - \omega_0)^n \quad (3.26)$$

$$\equiv \tilde{\Sigma}_{pq}(\omega; \omega_0, k), \quad (3.27)$$

and solving

$$|\omega \mathbf{I} - \mathbf{F} - \tilde{\Sigma}(\omega; \omega_0, k)| = 0 \quad (3.28)$$

by an iterative root-finding algorithm.[18] In our implementation, $\mathbf{F} + \tilde{\Sigma}$ is diagonalized and then the Taylor-series approximation of $\tilde{\Sigma}$ is updated by using the eigenvalue for the target root for the value of ω . This cycle is repeated until the target eigenvalue ceases to change. Note that the self-energy matrix and its derivatives are evaluated only once at ω_0 , which is typically one of the HF orbital energies.

When the diagonal approximation to the self-energy is also invoked, the Dyson equation for each Koopmans state becomes a k^{th} -order polynomial equation for ω ,

$$\omega - \epsilon_p - \tilde{\Sigma}_{pp}(\omega; \omega_0, k) = 0, \quad (3.29)$$

which can be solved numerically exactly and instantaneously by any of the standard root-finding methods. The aforementioned iterative algorithm works as well, yielding roots within 0.01 meV of the converged values typically in 3 cycles.[9]

The MC-GF n formalism and algorithm are particularly well suited for the self-energy-derivative calculation. The only factors in each integrand [see Eq. (3.3) through Eq. (3.6) and either Appendix I or Eq. (3.11)] that contain ω always take the form:

$$\frac{\partial^n}{\partial \omega^n} e^{\pm \omega \tau} = (\pm \tau)^n e^{\omega \tau}. \quad (3.30)$$

Then, the integration of all self-energy derivatives as well as the self-energy itself can be written in a common expression,

$$\begin{aligned} \frac{\partial^n}{\partial \omega^n} \Sigma_{pq}^{(k)}(\omega) &= \int \cdots \int d\mathbf{r}_1 d\mathbf{r}_2 \cdots d\mathbf{r}_{2k-1} d\mathbf{r}_{2k} \int \cdots \int_0^\infty d\tau_1 \cdots d\tau_{k-1} \\ &\times \frac{\partial^n}{\partial \omega^n} f_{pq}^{(k)}(\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_{2k-1}, \mathbf{r}_{2k}, \tau_1, \cdots, \tau_{k-1}, \omega), \end{aligned} \quad (3.31)$$

and use the aforementioned algorithms, sharing much of the computer code. Here, the ω -derivative of the second-order integrand is simply written as

$$\frac{\partial^n}{\partial \omega^n} f_{pq}^{(2)} = (-\tau_1)^n (f_{pq}^{(2:1)} + f_{pq}^{(2:2)}) + (\tau_1)^n (f_{pq}^{(2:3)} + f_{pq}^{(2:4)}). \quad (3.32)$$

Similarity, the ω -derivative of the third-order integrand is

$$\begin{aligned} \frac{\partial^n}{\partial \omega^n} f_{pq}^{(3)} &= (-\tau_2)^n h_{pq}^{(a)} + (\tau_2)^n h_{pq}^{(b)} + (-\tau_1 - \tau_2)^n h_{pq}^{(c)} + (\tau_1 + \tau_2)^n h_{pq}^{(d)} \\ &+ (-\tau_1)^n h_{pq}^{(e)} + (\tau_1)^n h_{pq}^{(f)} + \delta_{0n} h_{pq}^{(g)}, \end{aligned} \quad (3.33)$$

which encompasses $n = 0$ (the self-energy itself), where each $\mathbf{h}$ is the sum of integrands of a subgroup of diagrams sharing the same imaginary-time difference between the incoming and outgoing stubs (see Table 3.1):

$$h_{pq}^{(a)} = \sum_{d=1}^{10} f_{pq}^{(3:d)} ; h_{pq}^{(b)} = \sum_{d=11}^{20} f_{pq}^{(3:d)} ; \cdots ; h_{pq}^{(g)} = \sum_{d=61}^{84} f_{pq}^{(3:d)}. \quad (3.34)$$

Since the computation of Eq. (3.32) and Eq. (3.33) adds only $O(k)$ multiplications per MC step for the k^{th} derivatives, the self-energy-derivative calculation is nearly free of operation cost regardless of k .

3.1.5 Statistical Uncertainty

If the frequency-independent approximation to the self-energy is lifted and thus Σ is approximated by the k^{th} -order Taylor series $\tilde{\Sigma}(\omega)$, the statistical uncertainty of each element of the latter is calculated by the standard theory of error propagation. The covariance matrix, $\text{cov}[\mathbf{X}_{pq}, \mathbf{X}_{pq}]$, is accumulated for each element of the self-energy and self-energy-derivative matrices simultaneously according to the procedure outlined in the previous subsection, where

$$\mathbf{X}_{pq} = \left(\Sigma_{pq}, \frac{\partial \Sigma_{pq}}{\partial \omega}, \dots, \frac{\partial^k \Sigma_{pq}}{\partial \omega^k} \right)^T. \quad (3.35)$$

Then, rewriting Eq. (3.27) (omitting all nonessential arguments) as

$$\tilde{\Sigma}_{pq} = a_0 \Sigma_{pq} + a_1 \frac{\partial \Sigma_{pq}}{\partial \omega} + \dots + a_k \frac{\partial^k \Sigma_{pq}}{\partial \omega^k}, \quad (3.36)$$

with $a_n = (\omega - \omega_0)^n / n!$. The variance of the left-hand side is obtained by evaluating

$$\sigma_{pq}^2 = \sum_{i=0}^k \sum_{j=0}^k a_i (\text{cov}[\mathbf{X}_{pq}, \mathbf{X}_{pq}])_{ij} a_j. \quad (3.37)$$

This reduces to Eq. (1.48) when $k = 0$ (the frequency-independent approximation) as $(\text{cov}[\mathbf{X}_{pq}, \mathbf{X}_{pq}])_{00}$ is identified as σ_{pq}^2 of Eq. (1.48).

When the diagonal approximation to the self-energy is also lifted, an electron binding energy is obtained from the corresponding eigenvalue of $\mathbf{F} + \tilde{\Sigma}(\omega)$. Its statistical uncertainty is estimated by a numerical simulation instead of the standard error-propagation theory due to the inherent challenge of applying aforementioned to an eigenvalue finding algorithm. During the simulation, the covariance between different elements of the self-energy matrix is neglected because the resulting covariance matrix would be too large with $O(n_{\text{orb}}^4 k^2)$ elements. One thousand samples of $\mathbf{F} + \tilde{\Sigma}(\omega)$ with its pq^{th} elements artificially saddled with a Gaussian noise with variance σ_{pq}^2 are prepared. Then, these matrices are diagonalized and the statistical distribution of the target eigenvalue is measured to determine its variance numerically. In the diagonal approximation, $\bar{\sigma}_{pp}$ directly reports the statistical uncertainty of the electron binding energy. ^j

3.2 Computational Efficiency and Applicability

Since the MC-MP n and MC-GF n methods give the same numerical results (with statistical uncertainties) as the deterministic counterparts (if they exist), their merit lies in the fact that they become more efficient in

the large-problem-size limit on the largest, heterogeneous computers, thereby, broadening the methods’ applicability. Theoretically, this merit is argued to exist for three ultimate reasons: First, the MC-integration algorithm is fundamentally more efficient than quadrature as the integral dimension exceeds seven or so, making the MC-MP n and MC-GF n methods more scalable with the problem (molecule) size.[89] Second, the MC-integration algorithm is inherently parallel, making the methods more scalable with the computer hardware size and types, and thus with human implementation efforts. Third, the MC-integration algorithm uses a negligibly small memory space, allowing the so-called space-time trade-off,[90] which is one of the most important general techniques to enlarge applicability of calculations that are otherwise memory-limited. The MC-MP n and MC-GF n methods completely eliminate the need for evaluating and storing two-electron integrals either in the AO or MO basis and, in this regard, may be compared with Almlöf’s direct SCF method, which is another prime example of the space-time trade-off.[91] In the following, the various important aspects of the MC-GF3 method’s efficiency and its applications to large conjugated molecules are discussed.

3.2.1 Computer Implementation

GPU-accelerated renditions of the MC-GF2 and MC-GF3 programs have been created. The algorithmic details of these programs may be found in Chapter 4 of the current manuscript. All calculations in this study were performed on Blue Waters at NCSA. Unless otherwise noted, all timing measurements were based on calculations with the aug-cc-pVDZ basis set[92] and direct-MC integration in the imaginary-time dimensions. All calculations invoked the frozen core approximation.

3.2.2 Non-diagonal, Frequency-dependent Self-energy Calculation

Table 3.2 compares the time required to complete 128 MC steps of MC-GF3 calculations in various (diagonal and/or frequency-independent) approximations to the self-energy on one Blue Waters XE (CPU) or XK (GPU) node.

The time per MC step hardly increases by lifting the diagonal or frequency-independent approximation. This is expected from the algorithms described in the previous section. Common subexpression elimination and precalculation of the common subexpression (i.e., the core matrix) allows MC-GF3 (and MC-GF2) to obtain off-diagonal elements of the self-energy matrix without any significant addition of cost. Likewise, ω -derivatives of the self-energy up to any arbitrary high order can be accumulated concurrently by incurring only a few extra multiplications, and insofar as a low-order Taylor expansion of the self-energy is appropriate, MC-GF3 (and MC-GF2) can lift the frequency-independent approximation at essentially no cost. From Table

Table 3.2: The average wall time (in seconds) to complete 128 MC steps of MC-GF3.

Approximations ^a	Methane		Benzene		Pentacene	
	CPU ^b	GPU ^c	CPU ^b	GPU ^c	CPU ^b	GPU ^c
Diagonal, ω -independent	0.95	55.3	2.58	56.3	17.4	60.4
Diagonal	0.95	55.3	2.58	56.4	17.4	60.5
ω -independent	1.00	55.5	2.76	56.7	19.2	62.5
Null	1.02	56.0	3.05	57.4	22.8	64.5

^a ‘Diagonal’ means the diagonal approximation to the self-energy. ‘ ω -independent’ means the frequency-independent approximation to the self-energy. ‘Null’ means that the full non-diagonal, frequency-dependent self-energy is used with its frequency dependence approximated by a fifth-order Taylor expansion.

^b A single Blue Waters XE (CPU) node is used with 151, 208, or 397 redundant walkers for methane, benzene, and pentacene respectively.

^c A single Blue Waters XK (GPU) node is used with 1024 redundant walkers.

3.2, it can be shown that there is a slight decrease in the sampling rate for the larger molecules on the CPU when the off-diagonal elements are computed. This suggests that the number of walkers needs to be increased somewhat more aggressively, but still linearly, for non-diagonal calculations.

3.2.3 System-size Dependence of Cost

The system-size dependence (scaling) of the operation cost of MC-GF3 was discussed in the methods section of this chapter during the presentation of the redundant-walker algorithm. The operational costs for MC-GF2 were not discussed in less detail, since they are the same as MC-MP2 with the advent of the core matrices for the self-energy. By keeping the number of walkers proportional to the system size, the redundant-walker algorithm masks the operational costs by a corresponding decrease in the variance. Guidelines to appropriately select the number of walkers are easily established numerically and have been demonstrated in the context of the corresponding MC-MP methods. In order to establish the cost scaling functions for MC-GF2 and MC-GF3, the scaling of the variance must be established. Figures 3.2 and 3.3 (3.4 and 3.5) respectively plot the variance (σ^2) [Eq. (1.48)] of diagonal or non-diagonal MC-GF2 (MC-GF3) binding energies for the HOMO as a function of molecular size (as measured by the number of electrons, n) across the size of methane to pentacene. In each plot, the data are shown that are obtained with both the aug-cc-pVDZ and aug-cc-pVTZ basis sets and the frequency-independent approximation to the self-energy with 128 redundant walkers. See Appendix D and Appendix E for the variances and number of steps of these simulation.

Figure 3.2 shows the variance of the diagonal MC-GF2 methods as a function of the number of electrons. The variance is observed to grow as approximately $O(n^2)$. The operational cost scaling of an MC-GF2 simulations is assigned to be $O(n)$ as the MC-GF2 and MC-MP2 methods have identical operational cost

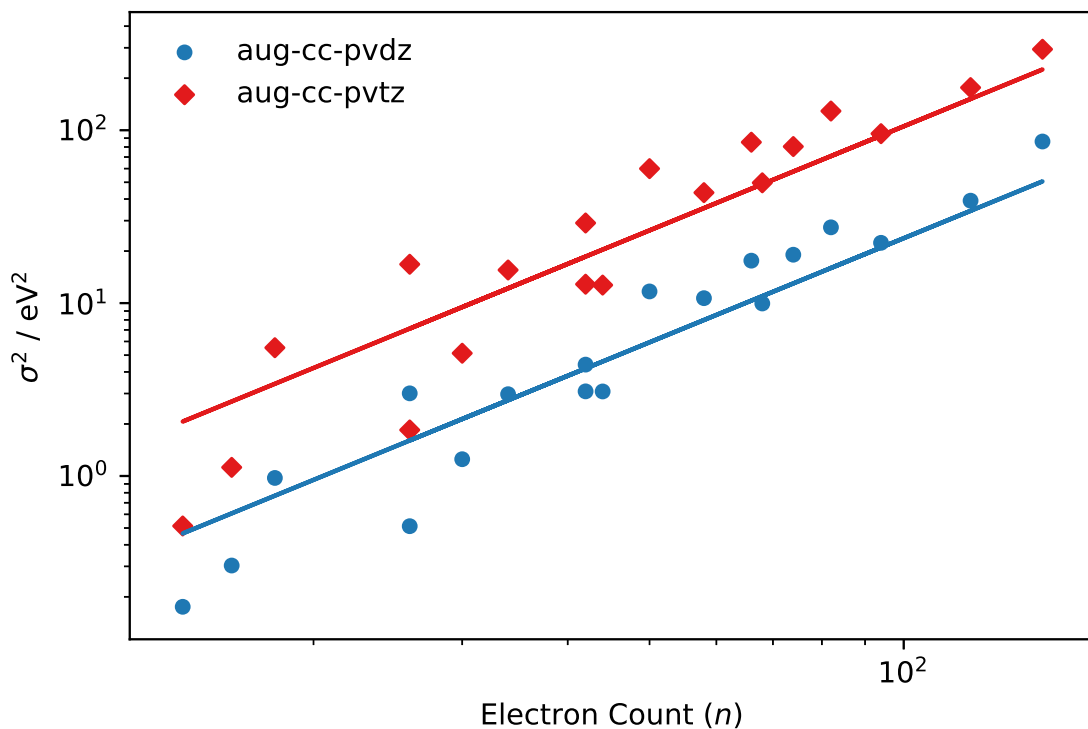


Figure 3.2: The variance, σ^2 , of binding energies obtained by diagonal, frequency-independent MC-GF2 calculations (obtained with the aug-cc-pVDZ or aug-cc-pVTZ basis set) as a function of the number of electrons, n . Functions proportional to n^2 are superimposed to guide the eyes. See Appendix D for the unprocessed data.

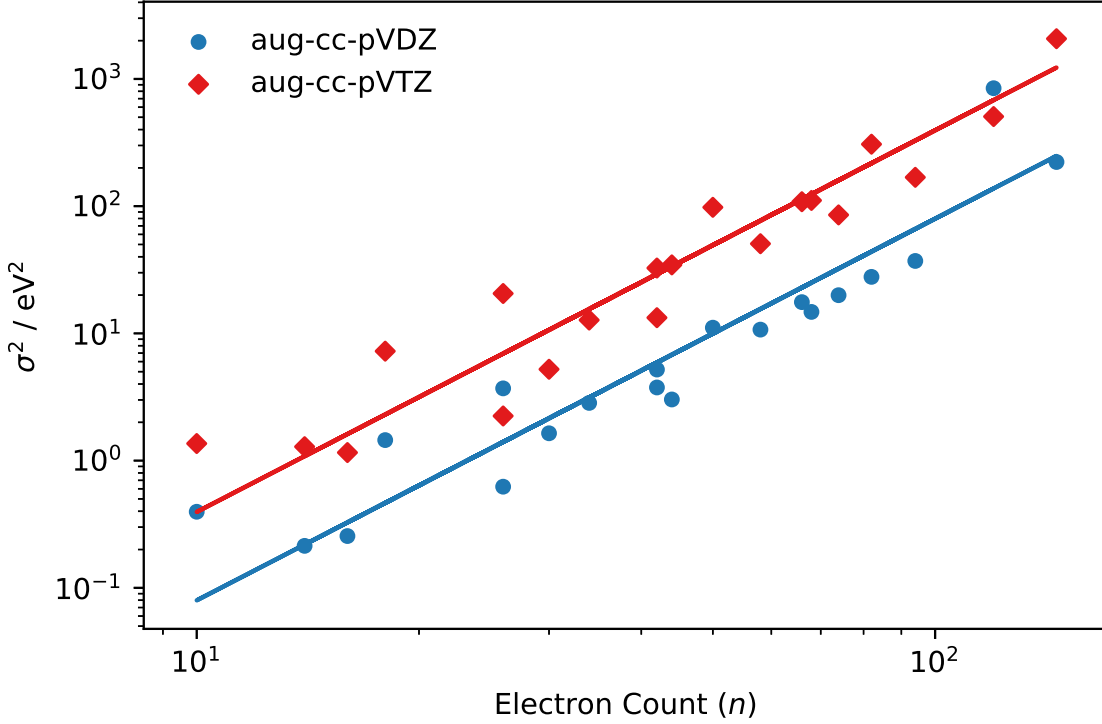


Figure 3.3: The variance, σ^2 , of binding energies obtained by non-diagonal, frequency-independent MC-GF2 calculations (obtained with the aug-cc-pVDZ or aug-cc-pVTZ basis set) as a function of the number of electrons, n . Functions proportional to n^3 are superimposed to guide the eyes. See Appendix D for the unprocessed data.

scalings. In combination, these two results indicate that the cost to obtain a binding energy with a prescribed target absolute uncertainty with diagonal MC-GF2 scales as approximately $O(n^3)$. Thus, diagonal MC-GF2 is observed to scale as one-rank lower than the conventional diagonal GF2 algorithm, whose cost scaling is $O(n^4)$.

Figure 3.3 is similar to Figure 3.2, except that it displays the variance of binding energies obtained by the non-diagonal MC-GF2 method rather than the diagonal variant. The variance of the binding energies obtained from the non-diagonal MC-GF2 methods is observed to scale as $O(n^3)$. The operational cost scaling of the non-diagonal MC-GF2 method is assigned to be $O(n)$ for the same reasons this cost was assigned to diagonal MC-GF2. The cost scaling to obtain a binding energy with a fixed target absolute uncertainty with the non-diagonal MC-GF2 methods is assigned to be approximately $O(n^4)$. This is a full rank lower than the conventional non-diagonal GF2 methods cost scaling of $O(n^5)$, paralleling the results for the diagonal MC-GF2 method.

In both the diagonal and non-diagonal MC-GF2 methods, enlarging the basis from aug-cc-pVDZ to aug-

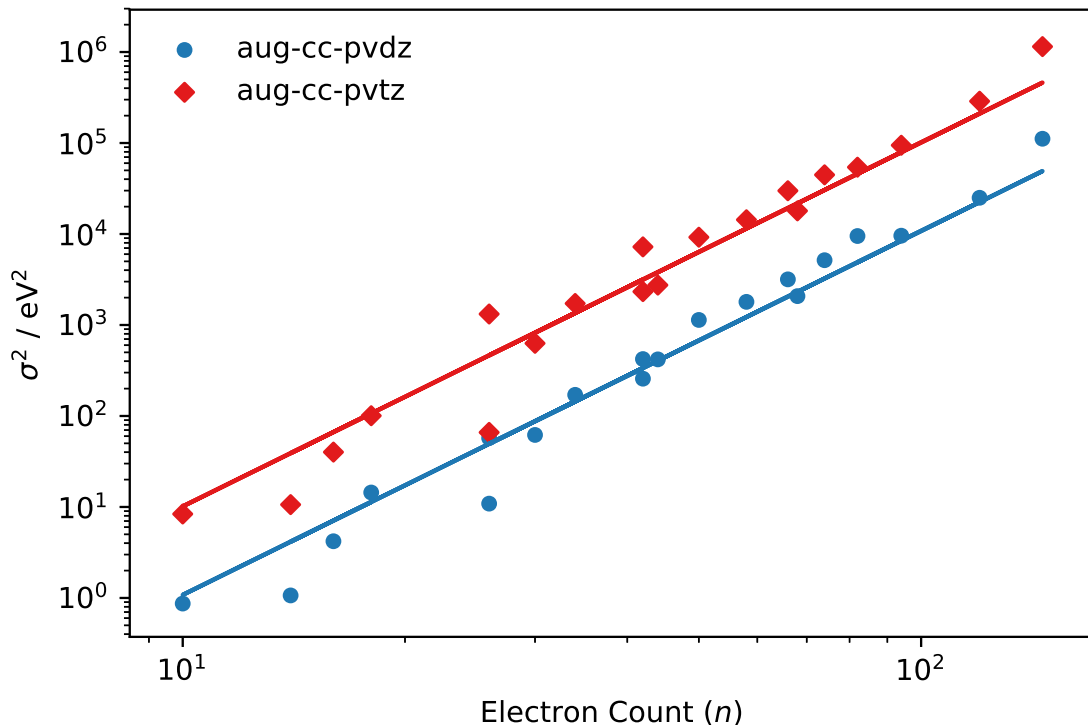


Figure 3.4: The variance, σ^2 , of binding energies obtained by diagonal, frequency-independent MC-GF3 calculations (obtained with the aug-cc-pVDZ or aug-cc-pVTZ basis set) as a function of the number of electrons, n . Functions proportional to n^4 are superimposed to guide the eyes. See Appendix E for the unprocessed data.

cc-pVTZ increasing the variance uniformly. In diagonal MC-GF2, the variance is increased by a factor of ~ 4.4 , while it is increased by a factor of ~ 5.0 for the non-diagonal method.

Figure 3.4 shows that the variance (σ^2) of the diagonal MC-GF3 method is observed to scale as approximately $O(n^4)$. This, when combined with the $O(n^0)$ size-dependence of the cost per MC sample, implies[10] that the overall scaling of cost to reach a given absolute statistical uncertainty in a binding energy is $O(n^4)$. This is lower than the $O(n^5)$ scaling for the deterministic algorithm of the diagonal MBGF(3) method. Although the stochastic algorithm has a much greater prefactor on the cost function than the deterministic algorithm, the former is expected to outperform the latter for sufficiently large molecules both in applicability and speed. Enlarging the basis set from aug-cc-pVDZ to aug-cc-pVTZ increases the statistical uncertainty uniformly by a factor ~ 9.4 .

Figure 3.5 paints a similar picture for the non-diagonal MC-GF3 method. The variance is observed to scale as approximately $O(n^5)$, which, when combined with the $O(n^0)$ size-dependence of the cost per MC sample, suggests an overall scaling $O(n^5)$. This is an improvement over the $O(n^6)$ scaling of the deterministic

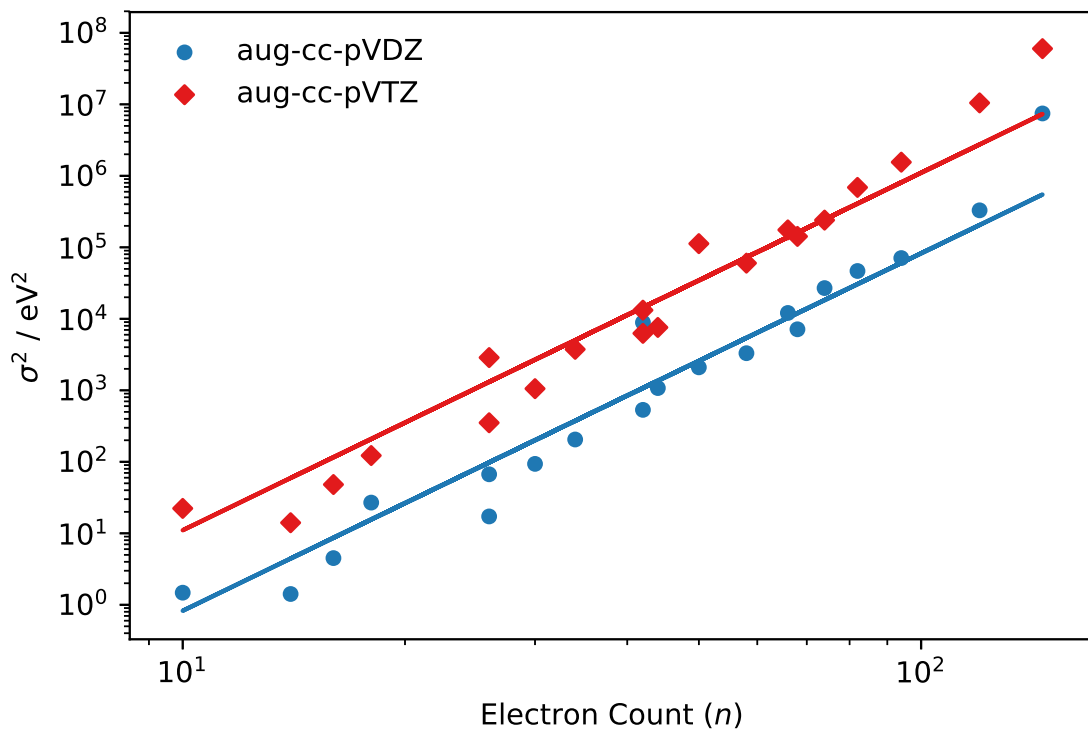


Figure 3.5: The variance, σ^2 , of binding energies obtained by non-diagonal, frequency-independent MC-GF3 calculations (obtained with the aug-cc-pVDZ or aug-cc-pVTZ basis set) as a function of the number of electrons, n . Functions proportional to n^5 are superimposed to guide the eyes. See Appendix E for the unprocessed data.

MBPT(3) method with non-diagonal self-energy. Increasing the basis-set size from the aug-cc-pVDZ to aug-cc-pVTZ causes the variance and thus the cost to uniformly increase by a factor of ~ 12 . Frequency-dependent self-energy calculations should not alter these conclusions.

It is caution that these cost scaling functions are tentative because of the limited range of molecular sizes used to measure them and the less-than-perfect fit of these data in the asymptotic limit. It seems sometimes possible to rationalize or predict the observed scaling of the MC-MP and MC-GF methods by analyzing theoretically their applications to an ensemble of non-interacting noble-gas atoms. However, we find it difficult to perform the same analysis of the reported algorithm as it involves optimizing the number of walkers according to the computer characteristics and problem size. The observed scaling functions are, therefore, gleaned from actual performance measurements and can change depending on computer hardware details and/or when even larger molecules are included in the measurement.

3.2.4 Illustrative Calculations

MC-GF2 and MC-GF3 calculations of the self-energy matrix and its derivatives up to the fifth order for methane ($r_{\text{C-H}} = 1.0935$) and benzene ($r_{\text{C-C}} = 1.397$, $r_{\text{C-H}} = 1.0869$) have been performed. The values of ω used are -14.769697 and -9.144574 eV, the HF orbital energies of the HOMO for methane and benzene, respectively. Both calculations employ the aug-cc-pVDZ basis set and 1024 redundant walkers. The calculation for methane was performed on 16 Blue Waters XK (GPU) nodes, carrying out 145408 MC steps across all nodes. The benzene calculation was run on 32 Blue Waters XK (GPU) nodes and 1064960 MC steps were completed across all nodes.

Tables 3.3 and 3.4 compare the sum of the MC-GF2 and GF3 corrections to the self-energy along with their derivatives up to the fifth order with the deterministic results for methane and benzene, respectively. For both molecules, the self-energy corrections (the zeroth-order derivatives) are in excellent agreement within 6 meV between the stochastic and deterministic calculations, which is more than enough precision in view of the intrinsic accuracy of these methods and experimental uncertainties. The estimated statistical uncertainties are consistent with these deviations, suggesting the absence of any bias in the MC-GF n results.

The first-order derivatives of the MC-GF self-energy are in even better agreement with the deterministic values; the relative errors of the first-order derivatives are significantly smaller than those of the self-energies for both molecules. Whether this is still the case at a frequency near a pole is unknown. Given the rapid convergence of the first-order derivatives, accurate estimates of pole-strengths, which are related to the first derivatives as $(1 - d\Sigma_{pp}(\omega)/d\omega)^{-1}$ in the diagonal approximation are given by, can be formed well in advance of bringing binding energies to convergence. Unlike the deterministic methods where pole strengths

Table 3.3: The n^{th} -order derivatives [in (eV) $^{1-n}$] of the sum of the second- and third-order corrections to the self-energy of the HOMO of methane calculated deterministically and stochastically (MC-GF). Values in parenthesis are the statistical uncertainties.

Order	Deterministic	Stochastic (MC-GF3)	$\sigma_{\text{rel}} / \%^a$
0	0.477 056 0	0.4825(74)	1.5
1	-0.076 955 6	-0.076 43(36)	0.48
2	0.003 313 9	0.003 289(79)	2.4
3	-0.000 447 8	-0.000 452(24)	5.3
4	0.000 053 3	0.000 058 2(82)	14
5	-0.000 007 7	-0.000 010 9(29)	27

^a Relative uncertainty.

Table 3.4: The same as Table 3.3 but for benzene.

Order	Deterministic	Stochastic (MC-GF3)	$\sigma_{\text{rel}} / \%^a$
0	-0.211 949	-0.207(26)	13
1	-0.107 160	-0.106 62(33)	0.31
2	0.003 695	0.003 551(84)	2.4
3	-0.001 171	-0.001 088(45)	4.1
4	0.000 206	0.000 152(28)	18
5	-0.000 080	-0.000 042(19)	45

^a Relative uncertainty.

and binding energies are both obtained at the conclusion of a calculation, the MC-GF2 and MC-GF3 methods can discern early non-Koopmans states whose diagonal pole strengths are 0.85 or less [58, 93, 94] and quickly switch to methods or algorithms specialized for such states.

Tables 3.5 and 3.6 present the binding energies of the HOMO (i.e., the vertical ionization energies) of methane and benzene, respectively, calculated by deterministic and stochastic (MC-GF3) third-order many-body Green’s function methods. Figures 3.6 and 3.7 give a greater detail of the frequency-independent MC-GF3 results by plotting the binding energies obtained with various truncation orders of the Taylor expansions of the self-energy.

In all cases, MC-GF3 reproduces the deterministic data within $1\bar{\sigma}$, underscoring the fact that the method has no bias and the statistical uncertainty is estimated conservatively. The statistical uncertainty does not increase much by lifting the frequency-independent approximation and suggests that lifting this approximation does not impact this quantity significantly. Similarly, although the off-diagonal elements are observed to make the statistical uncertainty rise more steeply with molecular size in Figure 3.5, they do not cause a prominent increase in the statistical uncertainty for small molecules such as methane but also for benzene. For the latter, this may be because the off-diagonal elements have negligible effect on the binding energy in the first place.

An examination of Figures 3.6 and 3.7 shows that the frequency-dependent effects are captured nearly

Table 3.5: The vertical ionization energy (in eV) of methane calculated by third-order many-body Green’s function theory implemented in the deterministic and stochastic (MC-GF3) algorithms. Values in parenthesis are the statistical uncertainties.

Approximations ^a	Deterministic	Stochastic (MC-GF3)
Diagonal, ω -independent	14.2934	14.2879(74)
Diagonal	14.3272	14.3219(74)
ω -independent	14.3057	14.2971(89)
Null	14.3382	14.3299(87)

^aSee the corresponding footnote of Table 3.2.

Table 3.6: The same as Table 3.5 but for benzene.

Approximations ^a	Deterministic	Stochastic (MC-GF3)
Diagonal, ω -independent	9.357	9.352(26)
Diagonal	9.336	9.332(26)
ω -independent	9.363	9.358(27)
Null	9.341	9.338(25)

^aSee the corresponding footnote of Table 3.2.

completely by the first derivatives of the self-energy at least for the Koopmans states of the two molecules studied. The inclusion of second and higher derivatives causes the binding energies to shift only by 0.1 meV, and can be avoided.

A calculation on C₆₀ fullerene was performed to examine the efficacy of MC-GF2 and MC-GF3 on larger systems. The diagonal elements of the second- and third-order self-energies associated with the lowest-lying set of degenerate HF virtual orbitals and the seven highest-lying sets of degenerate HF occupied orbitals were evaluated with the aug-cc-pVDZ basis set. The frequency-independent calculations were performed at the corresponding HF orbital energies given in Table 3.7 as the negative of the ‘HF’ column entries. The frequency-dependent calculations used the fifth-order Taylor expansions of the self-energies. The self-energy corrections and their derivatives were obtained from an average of 10⁹ MC steps using 512 redundant walkers. The whole calculation took only 534 hours (wall time) on 32 Blue Waters XE/XK CPUs/GPUs nodes, and is practically feasible.

The second-order (MC-GF2) binding energies are all converged to within a statistical uncertainty of 70 meV, although the uncertainties of most states are an order of magnitude smaller. For all three states for which experimental data are available, MC-GF2 corrects the Koopmans (HF) binding energies in the correct direction towards the experimental data. In all three cases, however, the corrections overshoot, not necessarily improving the HF binding energies, which is expected.[9]. Taking into account the frequency-dependence of the self-energy systematically brings the MC-GF2 binding energies into closer alignment with the experimental data, although this is likely coincidental considering the MC-GF3 results, which is now discussed.

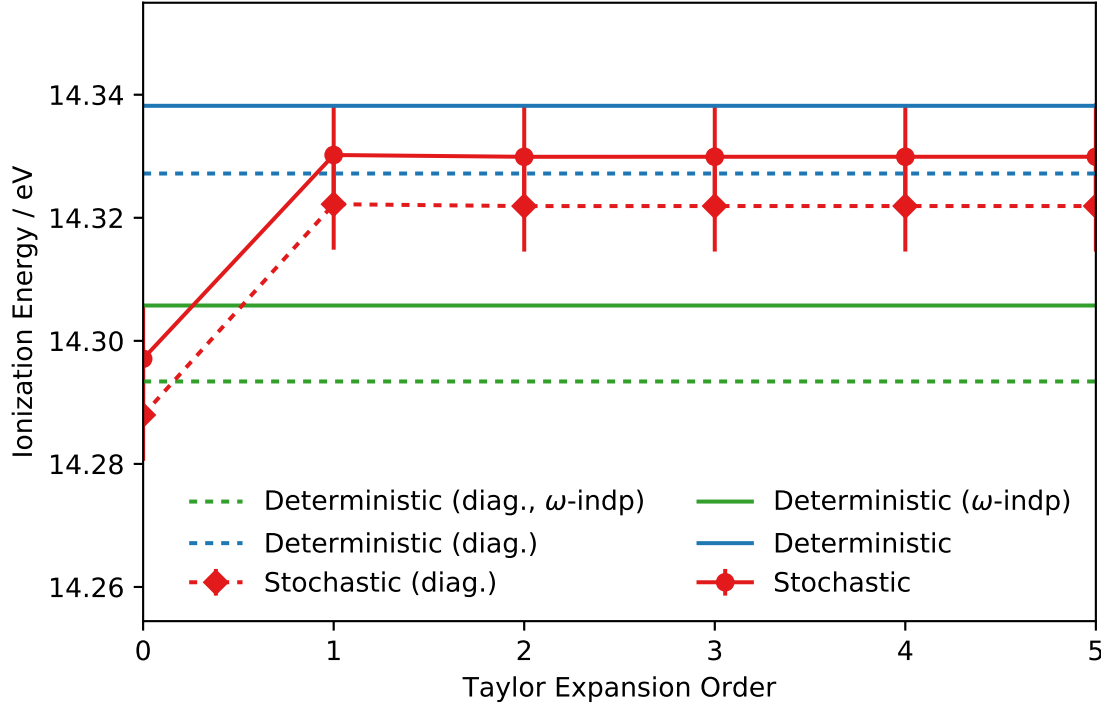


Figure 3.6: The vertical ionization energy of methane calculated by various third-order many-body Green’s function methods. The dashed lines plot either the deterministic or stochastic (MC-GF3) calculations in the diagonal (‘diag.’) approximation, while the solid lines are without this approximation. The blue (green) lines are the deterministic calculations with (without) the frequency-independent (‘ ω -indp.’) approximation. The red lines designate the stochastic (MC-GF3) calculations with the self-energy expanded in a Taylor series truncated after the order given in the abscissa (the plots at order zero correspond to the frequency-independent calculations).

Table 3.7: The binding energies (in eV) of C_{60} calculated by MC-GF2 and MC-GF3. Values in parenthesis are the statistical uncertainties.

State	HF	MC-GF2		MC-GF3		Experiment
		Diag., ω -indep. ^c	Diag. ^d	Diag., ω -indep. ^c	Diag. ^d	
t_{2u}^a	0.8499	3.419(15)	2.934(14)	1.41(59)	1.36(59)	2.68350(60) ^e
h_u^b	7.8547	6.7048(51)	6.9450(51)	9.00(29)	8.93(29)	7.64(2) ^f
h_g^b	9.6445	7.8391(44)	8.2501(45)	10.45(29)	10.41(29)	8.95 ^g
g_g^b	9.9049	7.6219(46)	8.1617(47)	10.78(29)	10.74(29)	
g_u^b	12.5948	8.8525(49)	9.9221(54)	12.82(36)	12.78(37)	
t_{2u}^b	13.0593	9.4247(63)	10.4343(81)	13.58(39)	13.58(56)	
h_u^b	13.6349	8.4790(46)	9.769(45)	14.12(28)	14.12(36)	
h_g^b	13.9870	8.7534(55)	10.054(70)	14.79(38)	14.77(24)	

^aVertical electron affinities.

^bVertical ionization energies.

^cDiagonal and frequency-independent approximations.

^dDiagonal approximation.

^eHuang *et al.*[95]

^fLichtenberger *et al.*[96]

^gLichtenberger *et al.*[97] Band centroid. No uncertainty reported.

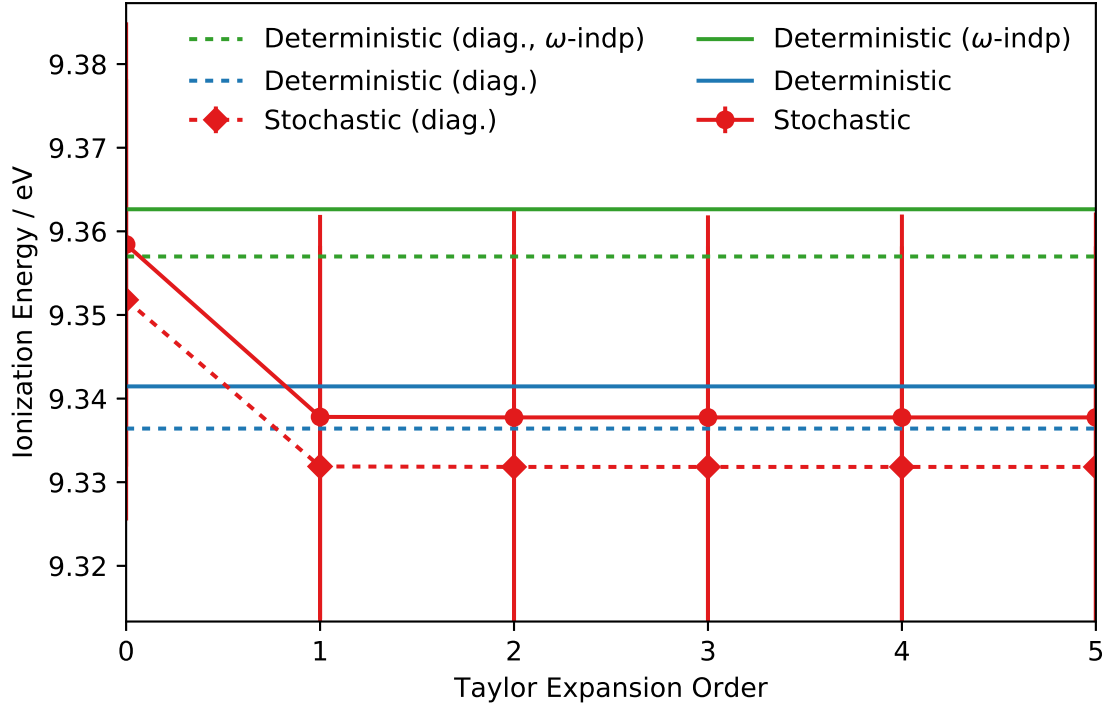


Figure 3.7: The same as Figure 3.6 but for benzene.

The third-order (MC-GF3) binding energies suffer from greater statistical uncertainties of a few tenths of an electron volt, which is sufficient for band assignments and assessments of methods' intrinsic accuracy, if not for prediction of more congested spectra. Whereas no issue has been observed that may be characterized as the sign problem in MC-MP n or MC-GF n , the greater statistical uncertainties with increasing perturbation order n may be considered as a manifestation of the problem. In all three states with experimental data, MC-GF3 shifts the MC-GF2 values in the correct direction, but again overshoots the experimental values even more significantly, not improving the MC-GF2 results. At this third order, lifting of the frequency-independent approximation does not necessarily bring the binding energies into better agreement with the experimental values. Correcting the basis-set incompleteness by the F12 method[12] is estimated to increase the first IE (the h_u state) by 0.47 eV, improving the GF2 results but bringing the GF3 data in worse agreement with the experiment. These findings are reminiscent of the non-monotonic convergence[59] of the MBGF(n) binding energies as a function of n especially at $1 \leq n \leq 3$. After the third order, the convergence is monotonic and smooth at least for the smallest basis set,[59] which makes higher-order MC-GF n ($n \geq 4$) more important and worthwhile for future developments.

3.3 Conclusions

In this chapter, the capabilities and efficiency of the stochastic many-body Green’s function methods in several directions have been enhanced in several directions.

The most critical is the formulation and implementation of MC-GF3 to compute the third-order self-energy. MC-GF3 as well as MC-GF2 can now calculate either a selected set of diagonal elements or all elements of the self-energy matrix efficiently. We have furthermore introduced into both methods the capability to calculate the frequency-derivatives of the self-energy up to any arbitrary order with essentially no added cost. These derivatives permit a Taylor series representation of the self-energy and evaluates the frequency-dependent effects on electron binding energies and other quantities such as pole strengths (not considered in this article).

The MC-GF2 and MC-GF3 programs were implemented into the multi-level parallel algorithm described in our earlier paper.[10] The asymptotic operation cost of the diagonal MC-GF3 method is observed to be $O(n^{4.5})$, where n is the number of electrons, which is considerably superior to the corresponding deterministic algorithm having the $O(n^5)$ size-dependence of cost. The cost of the non-diagonal MC-GF3 method is $O(n^{5.5})$, again a large improvement over the $O(n^6)$ cost of the deterministic counterpart. The memory cost of MC-GF3 is negligible. Combining the high parallel scaling and low cost scaling, MC-GF2 and GF3 should outperform their deterministic counterparts for larger molecules on supercomputers today and in the future; however, the crossover point is unknown at present.

Illustrative calculations on methane and benzene showed that electron binding energies can be reliably obtained (within a statistical uncertainty of 90 meV or less) by MC-GF3 in as few as 10^6 MC steps for benzene. An application on C_{60} demonstrated that MC-GF3 can compute the third-order self-energy within a few tenths of an electron volt (statistical uncertainty) after 10^9 MC steps.

Chapter 4

Acceleration Technique for Monte Carlo Many-body Methods

As has been demonstrated in the two preceding chapters, the MC-MB family of methods achieves favorable scalings with respect to system size compared to their conventional counterparts, when such results are available. At the same time, performing a calculation with these methods can be expensive due to a large prefactor on their cost functions. Fortunately, there is a wide range of strategies that may be applied to accelerate MC-MB methods, either by exploiting known physics or the functional form of the integrands, by improving algorithmic performance, or by employing a known MC acceleration technique. One acceleration technique, the redundant-walker algorithm, which falls in the first two categories, has been discussed extensively due to its tremendous success and historical significance. In this chapter, four techniques to accelerate the convergence of MC-MB calculations are discussed. These techniques have been mentioned several times in the preceding chapter, but have not been given an in-depth examination. They are the use of GPU accelerators, the introduction of multiple strategies to perform the integration of the imaginary-time coordinates, an algorithm to directly sample the electron-pair importance function, and the identification of useful control variate functions for the MC integration of the electronic coordinates. Together, these techniques represent a substantial performance increase for MC-MB methods and have drastically widened their area of applicability.

4.1 GPU-acceleration of MC-MB methods

In 2009, the fastest supercomputer was Cray XT5 Jaguar at Oak Ridge National Laboratory, which consisted of 224,162 AMD Opteron processor cores, according to the TOP500 ranking. In 2012, IBM Blue Gene/Q Sequoia at Lawrence Livermore National Laboratory took the top spot with 98,304 nodes, each with a 16-core PowerPC A2 central processing units (CPUs), for a total of 1,572,864 processor cores. In late 2012, Jaguar underwent an upgrade and became Cray XK7 Titan with an addition of a single NVIDIA K20X graphical processing units (GPUs) to each of its 18,688 nodes, reclaiming the first place in the ranking.

Section 4.1 is based off of Reference [10] and contains results from Reference [13].

Recently, in June of 2018, Summit, the successor to Titan at Oak Ridge National Laboratory, consisting of 4,608 nodes each configured with two 22-core Power PC CPUs and 6 NVIDIA V100 GPUs, came online. At the time of writing, Summit holds the top ranking in TOP500. Shortly after coming online, tentative plans for Summit’s replacement, Frontier, were announced with a proposed node configuration of four GPUs per CPU processor. There are two striking features to be observed from this timeline. First, supercomputers are being built with fewer nodes that contain an ever-increasing number of GPUs. Second, although perhaps not immediately apparent, is that the overwhelming majority of the computational power of the supercomputers equipped with GPUs is derived almost exclusively from the GPUs. These trends herald the modern and future supercomputing and workstation architectures.

Given the trend to employ GPUs as the main source of computational power on flagship supercomputers, it is obvious that any algorithm, including those in quantum-chemistry, must be able to execute efficiently on either many CPUs or GPUs. A great deal of effort has been applied to this very task of modernizing quantum chemistry methods.[98–108] Programs that rely heavily on standard numerical libraries, such as BLAS and LAPACK, are easily made to run on GPUs as GPU equivalents of these libraries exist. Programs that are reliant on subroutines that are not readily available in a GPU friendly format can be difficult to port. This is because in comparison to a CPU, which can perform complex instructions and supports overlapping and out-of-order executions as well as branch predictions, a GPU (also known as a coprocessor or accelerator) applies a relatively simple operation on a large amount of data at once, in the so-called SIMD (single-instruction, multiple-data) parallelism. The latter requires the expression of fine-grain data parallelism, which can be a highly non-trivial task. Therefore, any algorithm attempting to harness the full power of *many* GPUs may need to adopt multilevel parallelism, in which a fine-grain data-parallel part is handled by a single GPU, whereas a coarse-grain instruction-parallel part runs on many GPUs or many CPUs.

Being embarrassingly parallel by design (used here in a non-derogatory sense), MC-MB methods are easily made to execute efficiently on many CPUs,[73] by simply having all processes perform independent MC integrations and report their intermediate results to the master process occasionally. To translate an MC-MB algorithm into a form suitable for execution on a GPU, fine-grain data parallelism must be exposed. Fortunately, the redundant-walker algorithm consists of two equally important parts with disparate data-access patterns: (1) near-independent MC integrations using disjoint sets of electron pairs and (2) a maximal reuse of electron pairs in each independent MC integration.

Here, it is shown that the two algorithmic parts naturally expose multilevel parallelism, with part (1) ideally handled by coarse-grain instruction-parallelism on many CPUs or many GPUs involving few and

infrequent interprocessor communications, while part (2) can greatly benefit from fine-grain data-parallelism on a GPU with data on many-electron pairs rapidly computed and made available on shared memory to all processor cores (threads) that need them. Not only does this algorithmic design lead to a highly scalable MC-MB methods on many GPUs, but it also brings about a qualitative change in the behavior of the redundant-walker algorithm for the better; a GPU is able to support an astounding number of redundant walkers and provides a greater acceleration to the MC-MB algorithm than would be naively expected based on peak operational performance. This, in turn, is caused by near-perfect parallelization of the redundant-walker algorithm on a GPU until the number of electron pairs is large. GPU acceleration of the MC-MB algorithm has been studied most extensively in the context of MC-MP2 and MC-GF3. These studies show that an MC-MP2/cc-pVDZ calculation of a benzene dimer (228 basis functions), for instance, is found to be 11 times faster on a single GPU than on two CPUs (using all of their 16 cores). Similarly, an MC-GF3/aug-cc-pVDZ calculation of benzene is found to be 5 times faster. These high degrees of speedups are partly due to the fact that whereas the speedup from the redundant-walker algorithm for MC-MP2 and MC-GF3 calculations ceases to increase after 256 pairs on CPUs, as many as 2,048 pairs can contribute meaningfully on GPUs.

4.1.1 Multilevel GPU Parallelism

Before entering the discussion of the GPU algorithm for MC-MB methods, the different operational characteristics between CPUs and GPUs are briefly contrasted.

A CPU (often referred to as host for heterogeneous machines) can perform complex instructions efficiently, including overlapping (pipelined) and out-of-order executions. It can also make branch predictions. However, interprocessor communications are relatively slow (both in latency and bandwidth) because two CPUs typically do not share memory and are physically distant from each other. Therefore, an efficient algorithm on many CPUs ought to be coarse-grain embarrassingly parallel with as few and infrequent interprocessor communications as possible.

A GPU (often referred to as device) consists of several multiprocessors, each of which can execute a large number of concurrent calculations called threads. Several threads are grouped into a “warp” and several warps into a “block.” All threads in a block run on a single multiprocessor and have access to data on volatile shared memory or persistent global memory. A large number of threads should be spawned, so that no cores are idle, while thread load-balancing is automatically ensured by hardware. All threads in a warp are meant to execute a rather simple operation concurrently on many data. Although each thread in a warp has its own instruction pointer and two threads can, in principle, execute different instructions, if that takes

Table 4.1: The algorithmic steps of the generalized redundant walker accelerated MC-MB algorithm and their associated costs.

Step	Task	Cost ^a
1	Propagate pairs	$O(mn)$
2	Calculate AOs	$O(mn)$
3	Calculate MOs	$O(mn^2)$
4	Calculate $G^\pm$	$O(m^2n)$
5	Calculate Energy	$O(m^k)$

^a k is the order of the theory, m is the number of electron pairs and n is the problem size ($n = n_{\text{bas}} \propto n_{\text{ele}}$).

place, all other threads in the same warp must remain idle, a situation known as a “thread divergence.” In the worst-case scenario where all threads in a warp follow different execution paths, the computation is effectively serialized. An important consideration when using GPUs is, therefore, to avoid thread divergence, by exposing fine-grain SIMD parallelism in the algorithm. Furthermore, data transfers to/from CPU (host) from/to GPU (device) tend to have a high latency. So another important consideration is to minimize such transfers.

An ideal algorithm for *many* GPUs may, therefore, implement multilevel parallelism—coarse-grain instruction-parallelism across many CPUs or many GPUs and fine-grain data-parallelism on each GPU with built-in error/fault tolerance or possibly single-precision arithmetic (single-precision arithmetic is typically twice as fast as double-precision arithmetic on data-center-grade hardware and significantly faster on consumer-grade hardware). Data transfers between two CPUs, between a CPU and a GPU, and between two GPUs should be minimized.

To create an efficient GPU implementation of the MC-MB algorithm, all computing tasks associated with the redundant-walker algorithm are performed on a GPU. Table 4.1 lists the asymptotic cost function of each step. The cost is measured as a function of the order of the theory (k), number of electron pairs ($m \geq k$) and of the problem size (n) (which defines the system size and is equal to n_{bas} and proportional to n_{ele} or the number of atoms). Below, the GPU-parallelization strategy for each step is discussed.

Step 1 updates the coordinates of all of the electron-pairs. This is typically accomplished by the Metropolis algorithm in which m electron pairs make a distorted random walk. The cost of this step is dominated by the evaluation of the value of the weight function at new pair coordinates. It, therefore, involves $O(mn)$ exponential function calls. This step executes efficiently on either CPU (host) or GPU (device).

Step 2 calculates the amplitudes of all (n) AOs at m electron pair coordinates at an $O(mn)$ cost. Although the scaling is lower than some other steps, this step can be the hotspot unless m or n is large, because the prefactor on the cost function is the greatest as it involves expensive evaluations of exponential functions and spherical harmonics. To avoid thread divergence, all threads in a block (thus all threads in each warp

in the block) are tasked to calculate AOs that share the same spherical harmonics (e.g., d -type GTOs only), but at different electron coordinates concurrently, so that they perform an identical series of instructions and have the same memory-access pattern. The converse algorithm, in which all threads in a block calculate different types of GTOs concurrently at one electron's coordinates, causes a severe thread divergence. The AO amplitudes are stored in $n_{\text{bas}} \times m$ arrays $\mathbf{X}_i$ ($i = 1$ or 2), which persist in the GPU's global memory,

$$(\mathbf{X}_i)_{\mu k} = \chi_{\mu}(\mathbf{r}_{ik}^{[n]}), \quad (4.1)$$

where i labels electron 1 or 2 in the k^{th} electron pair ($1 \leq k \leq m$), μ designates an AO, and n counts the MC cycle.

The remainder of the MC-MB algorithm for GPUs relies heavily on matrix arithmetic. The matrix arithmetic, with the exception of perhaps the AO-to-MO transformation, is rather artificially imposed on the redundant-walker algorithm. However, the use of matrix arithmetic has the distinct advantage of allowing the use of high-performance linear algebra libraries designed explicitly for GPUs, such as cublas, to implement the MC-MB algorithm. Such high-performance libraries automatically account for differences in GPU hardware intricacies without the need for significant reconfiguration of the MC-MB algorithm and grant maximal performance. An additional benefit of coercing the redundant-walker algorithm into matrix expressions is that high-performance linear algebra libraries are also used for the standard CPU implementation of MC-MB methods, and greatly enhance its operational performance as well.

Step 3 generates the whole set of the occupied MO amplitudes in $n_{\text{occ}} \times m$ arrays Φ_i^- ($i = 1, 2$) and the virtual MO amplitudes in $n_{\text{vir}} \times m$ arrays Φ_i^+ ($i = 1, 2$):

$$(\Phi_i^-)_{pk} = \varphi_p(\mathbf{r}_{ik}^{[n]}), \quad 1 \leq p \leq n_{\text{occ}}, \quad (4.2)$$

$$(\Phi_i^+)_{pk} = \varphi_{n_{\text{occ}}+p}(\mathbf{r}_{ik}^{[n]}), \quad 1 \leq p \leq n_{\text{vir}}. \quad (4.3)$$

They are related to the AO amplitudes by

$$\Phi_i^- = \mathbf{C}^- \mathbf{X}_i, \quad (4.4)$$

$$\Phi_i^+ = \mathbf{C}^+ \mathbf{X}_i, \quad (4.5)$$

with

$$(\mathbf{C}^-)_{p\mu} = C_p^\mu, \quad (4.6)$$

$$(\mathbf{C}^+)_{p\mu} = C_{n_{\text{occ}}+p}^\mu. \quad (4.7)$$

The AO-to-MO transformation is, therefore, $O(mn^2)$ dense matrix multiplications. For large n , this step becomes the hotspot.

In step 4, $m \times m$ arrays of retarded and advanced Green's function traces are constructed:

$$\{\mathbf{G}_{ij}^-(\tau)\}_{kl} = (1 - \delta_{k,l})G^-(\mathbf{r}_{ik}^{[n]}, \mathbf{r}_{jl}^{[n]}, \tau), \quad (4.8)$$

$$\{\mathbf{G}_{ij}^+(\tau)\}_{kl} = (1 - \delta_{k,l})G^+(\mathbf{r}_{ik}^{[n]}, \mathbf{r}_{jl}^{[n]}, \tau), \quad (4.9)$$

where k and l designate an electron pair ($1 \leq k, l \leq m$), while i and j take the value of 1 or 2, corresponding to electron 1 or 2, respectively, in each pair. They are defined as

$$\mathbf{G}_{ij}^-(\tau) = ((\Phi_i^-(\tau))^\dagger \Phi_j^-(\tau)) \circ (\mathbf{1} - \mathbf{I}), \quad (4.10)$$

$$\mathbf{G}_{ij}^+(\tau) = ((\Phi_i^+(\tau))^\dagger \Phi_j^+(\tau)) \circ (\mathbf{1} - \mathbf{I}), \quad (4.11)$$

where $\mathbf{1}$ is an m -by- m matrix of ones, $\mathbf{I}$ is the m -by- m identity matrix, and $\circ$ indicates taking the Hadamard product (element wise multiplication). The remaining matrices $\Phi_j^-(\tau)$ and $\Phi_j^+(\tau)$ are defined as

$$\Phi_j^-(\tau) = \mathbf{T}^-(\tau)\Phi_j^-, \quad (4.12)$$

$$\Phi_j^+(\tau) = \mathbf{T}^+(\tau)\Phi_j^+, \quad (4.13)$$

with

$$\{\mathbf{T}^-(\tau)\}_{pq} = \delta_{pq}\sqrt{e^{-\epsilon_p\tau}}, \quad 1 \leq p, q \leq n_{\text{occ}}, \quad (4.14)$$

$$\{\mathbf{T}^+(\tau)\}_{pq} = \delta_{pq}\sqrt{e^{-\epsilon_{n_{\text{occ}}+p}\tau}}, \quad 1 \leq p, q \leq n_{\text{vir}}. \quad (4.15)$$

These Green's function arrays have the following symmetry:

$$\mathbf{G}_{ij}^-(\tau) = \{\mathbf{G}_{ji}^-(\tau)\}^\dagger, \quad (4.16)$$

$$\mathbf{G}_{ij}^+(\tau) = \{\mathbf{G}_{ji}^+(\tau)\}^\dagger. \quad (4.17)$$

Taking the Hadamard product by $\mathbf{1} - \mathbf{I}$ in Eq. (4.10) and Eq. (4.11) satisfies the constraint that the diagonals of the Green's function matrices are zero. Practically, this is accomplished by setting the diagonal of the matrix to be zero. Since $\mathbf{G}_{11}^{\pm}(\tau)$ and $\mathbf{G}_{22}^{\pm}(\tau)$ are symmetric matrices they should be constructed by subroutines that exploit this property such as CUBLASDYSRK (a symmetric matrix multiplication rank up subroutine). $\mathbf{G}_{12}^{\pm}(\tau)$ is obtained by a standard dense matrix multiplication whereas $\mathbf{G}_{21}^{\pm}(\tau)$ is obtained as the transpose of $\mathbf{G}_{12}^{\pm}(\tau)$ (assuming real orbitals). The cost of these multiplications are $O(m^2n)$, and hence this step can also be a hotspot when m is large.

In step 5, the contribution of the current MC step to the relevant energies is evaluated. Strategies to obtain values within MC-MP calculations are presented on a diagram-by-diagram basis. Differences in the strategies needed for MC-GF methods are noted but are otherwise easily inferred from the MC-MP strategies.

The contribution from the direct diagram (diagram 1 Figure 1.2) to the MP2 energy from a single MC step under the redundant-walker algorithm is obtained from

$$I^{(2:\text{direct})} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \sum_s w_s \frac{\{\mathbf{G}_{11}^{-}(-\tau_s)\}_{ij} \{\mathbf{G}_{22}^{-}(-\tau_s)\}_{ij} \{\mathbf{G}_{11}^{+}(\tau_s)\}_{ij} \{\mathbf{G}_{22}^{+}(\tau_s)\}_{ij}}{\omega^{(2)}(\mathbf{r}_{i1}^{[n]}, \mathbf{r}_{i2}^{[n]}, \mathbf{r}_{j1}^{[n]}, \mathbf{r}_{j2}^{[n]})}. \quad (4.18)$$

On a GPU, $I^{(2:\text{direct})}$ is efficiently obtained by first constructing the intermediate array

$$\tilde{\mathbf{f}}^{(2:\text{direct})} = -2 \sum_s w_s \mathbf{G}_{11}^{-}(-\tau_s) \circ \mathbf{G}_{22}^{-}(-\tau_s) \circ \mathbf{G}_{11}^{+}(\tau_s) \circ \mathbf{G}_{22}^{+}(\tau_s). \quad (4.19)$$

The intermediate for the exchange term is obtained analogously, and the two are summed together. With the intermediate, $I^{(2)}$ is obtained from

$$I^{(2)} = \vec{\omega}^T \tilde{\mathbf{f}}^{(2)} \vec{\omega}, \quad (4.20)$$

where the elements of column vector $\vec{\omega}$ are

$$\{\vec{\omega}\}_i = \omega(\mathbf{r}_{1i}, \mathbf{r}_{2i}). \quad (4.21)$$

The expressions to obtain $\tilde{\mathbf{f}}$ for in an MC-GF2 calculation are similar to that of Eq. (4.19), although only three $\mathbf{G}^{\pm}$ terms occur. To obtain the core matrix from $\tilde{\mathbf{f}}$, the expression following is used

$$\mathbf{g}^{(2)} = (\vec{\omega} \otimes \vec{\omega}) \circ \tilde{\mathbf{f}}^{(2)}. \quad (4.22)$$

A single diagram (diagram 1 of Figure 2.6) contributing to the MP3 energy serves as an example to demonstrate the set of matrix operations used to calculate a third-order diagrammatic contribution. Under the redundant-walker algorithm, the contribution from this diagram at a single set of imaginary-time point is

$$\begin{aligned}
I^{(3:1)} = \frac{1}{mP_3} \sum_{i \neq j \neq k}^m & G^+(\mathbf{r}_{1i}, \mathbf{r}_{1k}, \tau_1 + \tau_2) G^-(\mathbf{r}_{1k}, \mathbf{r}_{1j}, -\tau_2) G^-(\mathbf{r}_{1j}, \mathbf{r}_{1i}, -\tau_1) \\
& \times G^+(\mathbf{r}_{2i}, \mathbf{r}_{2j}, \tau_1) G^+(\mathbf{r}_{2j}, \mathbf{r}_{2k}, \tau_2) G^-(\mathbf{r}_{2k}, \mathbf{r}_{2i}, -\tau_1 - \tau_2) \\
& \div \omega^{(3)}(\mathbf{r}_{1i}, \mathbf{r}_{2i}, \mathbf{r}_{1j}, \mathbf{r}_{2j}, \mathbf{r}_{1k}, \mathbf{r}_{2k}).
\end{aligned} \tag{4.23}$$

The strategy for the efficient computation of a third-order diagrammatic contribution is more complex than the strategy used for second-order contributions. First, intermediate matrices that group the $\mathbf{G}^\pm$ terms with the same imaginary-time argument (the sign is ignored) are formed,

$$\mathbf{A}_{(1)} = \mathbf{G}_{11}^-(-\tau_1) \circ \mathbf{G}_{22}^+(\tau_1), \tag{4.24}$$

$$\mathbf{A}_{(2)} = \mathbf{G}_{11}^-(-\tau_2) \circ \mathbf{G}_{22}^+(\tau_2), \tag{4.25}$$

$$\mathbf{A}_{(1+2)} = \mathbf{G}_{11}^-(-\tau_1 - \tau_2) \circ \mathbf{G}_{22}^+(\tau_1 + \tau_2), \tag{4.26}$$

where the subscripts indicate which imaginary-time argument the intermediates capture. Once this set of intermediates are formed, the sum over one of the redundant walkers is performed. For MC-GF3, the sum must be the one not associated with the final indices of the core contribution, while for MC-MP3, the choice is arbitrary. Here, the sum over k is selected, and the resulting matrix operation is

$$\mathbf{B} = \mathbf{A}_{(1+2)} \mathbf{W} \mathbf{A}_{(2)}^T, \tag{4.27}$$

where $\mathbf{W}$ is a diagonal matrix formed from $\vec{\omega}$. Finally, $\tilde{\mathbf{f}}^{(3)}$ is obtained from

$$\tilde{\mathbf{f}}^{(3)} = \mathbf{A}_{(1)} \circ \mathbf{B}. \tag{4.28}$$

Either Eq. (4.20) or Eq. (4.22) is then applied to $\tilde{\mathbf{f}}^{(3)}$ to obtain the energy contribution for MC-MP3 or the core-contribution matrix for MC-GF3.

4.1.2 Performance Analysis

All of the following calculations were performed on Blue Waters at NCSA of the University of Illinois. Full specifications for the Cray XE and XK nodes are listed in Section 1.4. In this section, the number of CPUs refers to the number of AMD 6276 Interlagos processors, each containing 8 cores, on an XE node. The number of GPUs is equal to the number of the XK nodes used in a calculation; one CPU core on an XK node is given a minuscule amount of work required to orchestrate the computation. For convenience, the baseline performance of these computational resources are reiterated. An XE node has a peak performance of 0.314 TFLOPS while the NVIDIA K20X on an XK node has a peak performance of 1.31 TFLOPS. This suggests that a GPU (an XK node) is able to perform intrinsically roughly 4.2 ($\approx 1.31/0.314$) times many more operations than two CPUs (an XE node).

The unit of speed is taken as that of a 16-way parallel calculation on two CPUs (using all of their combined 16 cores) on an XE node. The speed is measured in terms of the number of samples produced in a unit wall-clock time. A sample for an MC-MP2 calculation is, in turn, a unique summand of the relevant target energy, whose number is $n_{\text{HW}}m(m-1)N$ for a calculation with N MC steps running on n_{HW} GPUs or CPU cores. Similarly, the number of samples produced in an MC-GF3 computation is $n_{\text{HW}}m(m-1)(m-2)N$.

Figure 4.1 compares the efficiency of the redundant-walker accelerated MC-MP2 algorithm on a GPU and on 8 CPUs (64 cores) as a function of the number of electron pairs (m) as measured by the sampling rate per XE or XK node. Given the excellent scalability across many CPUs or many GPUs, this figure will hardly change with the number of CPUs or GPUs.

Focusing on the dashed (CPU) curves, we find that the sampling rate increases linearly with the number of pairs (m). For small problems (water and benzene dimer with $n_{\text{bas}} = 24$ and 228, respectively), the speedup slows down at $m = 64$ or so, whereas it continues to grow for taxol ($n_{\text{bas}} = 1123$) up to $m = 512$ or more. These speedups are consistent with the analysis of the redundant-walker algorithm as presented in Section 2.2. On CPUs, Step 3 (MO construction) is the hotspot with an $O(mn^2)$ operation cost for large n (such as taxol), while the number of samplings is $O(m^2)$, leading to a net $O(m)$ increase in the sampling efficiency.[73] When n is small (such as water and benzene dimer), Step 6 (Green’s functions construction) with an $O(m^2n)$ cost surpasses Step 4 and becomes the hotspot as m increases, whereupon both the cost and benefit of the redundant-walker algorithm grow quadratically with m , giving no net increase in sampling efficiency, which is consistent with the examination of the redundant-walker algorithm given in Section 2.2. Nevertheless, even for water, the redundant-walker algorithm with $m = 64$ gives an order-of-magnitude speedup as compared with $m = 2$.

Shifting attention to the solid (GPU) curves, it is observed that the sampling rate increases *quadratically*

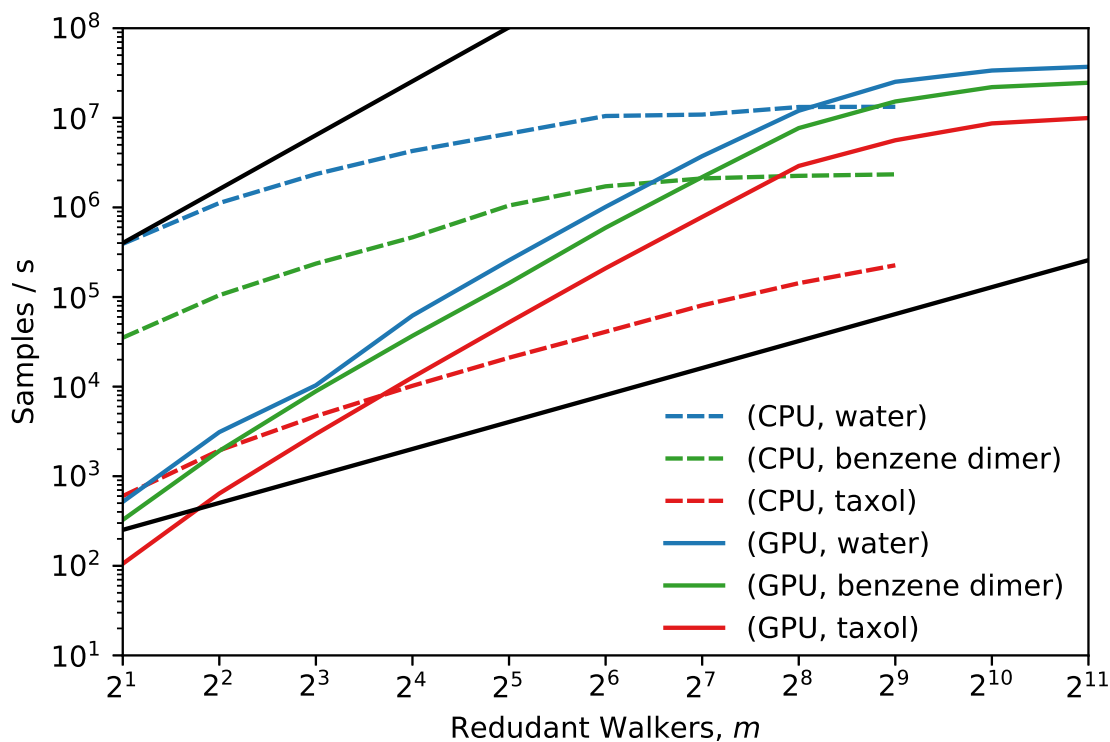


Figure 4.1: The efficiency of the redundant-walker algorithm as a function of the number of electron pairs (m) as measured by the number of samples produced per XE or XK node in MC-MP2/cc-pVDZ calculations on water ($n_{\text{bas}} = 24$), benzene dimer ($n_{\text{bas}} = 228$), or taxol ($n_{\text{bas}} = 1123$). The CPU calculations were performed on 8 XE nodes while the GPU calculations were performed on a single XK node. Functions proportional to the first and second power of the number of redundant walkers (m) are superimposed to guide the eyes.

up to $m \approx 256$, then *linearly* for $256 \leq m \leq 2048$, and the saturation of speedup is delayed until $m \approx 2048$ for all cases considered (water, benzene dimer, and taxol). This is due to the fact that all steps in Table 4.1 are nearly perfectly parallelized on a GPU for $m \leq 256$ and are completed in near-constant wall time because as many threads (up to tens of thousands) as there are arithmetic operations are spawned and they execute these operations concurrently in an automatically load-balanced manner. This makes the wall time spent by these steps independent of m , i.e., $O(1)$, while generating $O(m^2)$ samples, yielding a net $O(m^2)$ sampling efficiency increase. Only when the number of pairs (m) approaches 256 and the number of threads necessary to cover all the arithmetic tasks exceeds the number that can be run concurrently, does the wall time start to increase as $O(m)$. Thereupon, the speedup increases only linearly with m (as on CPUs) until it stalls near $m \approx 2048$, which is much greater than the saturation point on CPUs ($m \approx 64$ for water and benzene dimer).

Overall, the GPU-accelerated implementation of MC-MP2 achieves speedups of 2.8, 10.5, and 44.0 for water, benzene dimer, and taxol respectively when comparing a single XE node to a single XK node. These speedups, with the exception for water, are greater than the intrinsic increase in throughput of 4.2 that an XK node provides with respect to an XE node. These greater speedups are attributed to a GPU’s capability to provide a quadratic increase in the sampling rate when the number of redundant walkers is small and the delayed onset in the saturation of the sampling rate with the number of redundant walkers. The latter grants the GPU-accelerated MC-MP2 algorithm the ability to use nearly an order of magnitude more redundant walkers meaningfully as compared to the CPU implementation. Therefore, with respect to all of the above, a GPU is considered to bring about a qualitative change in redundant-walker algorithm’s behavior for the better.

An analysis analogous to the above, but for the application of GPU-acceleration to compute third-order quantities, was performed during the original presentation of MC-GF3. Figure 4.2 shows the sampling rate as a function of m for six sets of calculations. Each set of calculations was run to determine a diagonal self-energy of methane, benzene, or pentacene by MC-GF3 on a single XE (CPU) or XK (GPU) node.

The trends of the CPU results (the dashed curves in Figure 4.2) for MC-GF3 closely mirror those of MC-MP2: The sampling rate increases quadratically with the number of redundant walkers, m , when m is relatively small. The initial growth is quadratic, rather than linear as in MC-MP2, as $O(m^3)$ samples are produced at a cost of $O(m)$. For an intermediate value of m , the sampling rate continues to rise with m , albeit at a diminished rate. For a larger value of m , the sampling rate ceases to increase. The redundant-walker algorithm’s performance is observed to saturate at around $m = 2^7$ for methane, $m = 2^8$ for benzene, and $m = 2^9$ for pentacene, with only a slight increase upon a further increase in m .

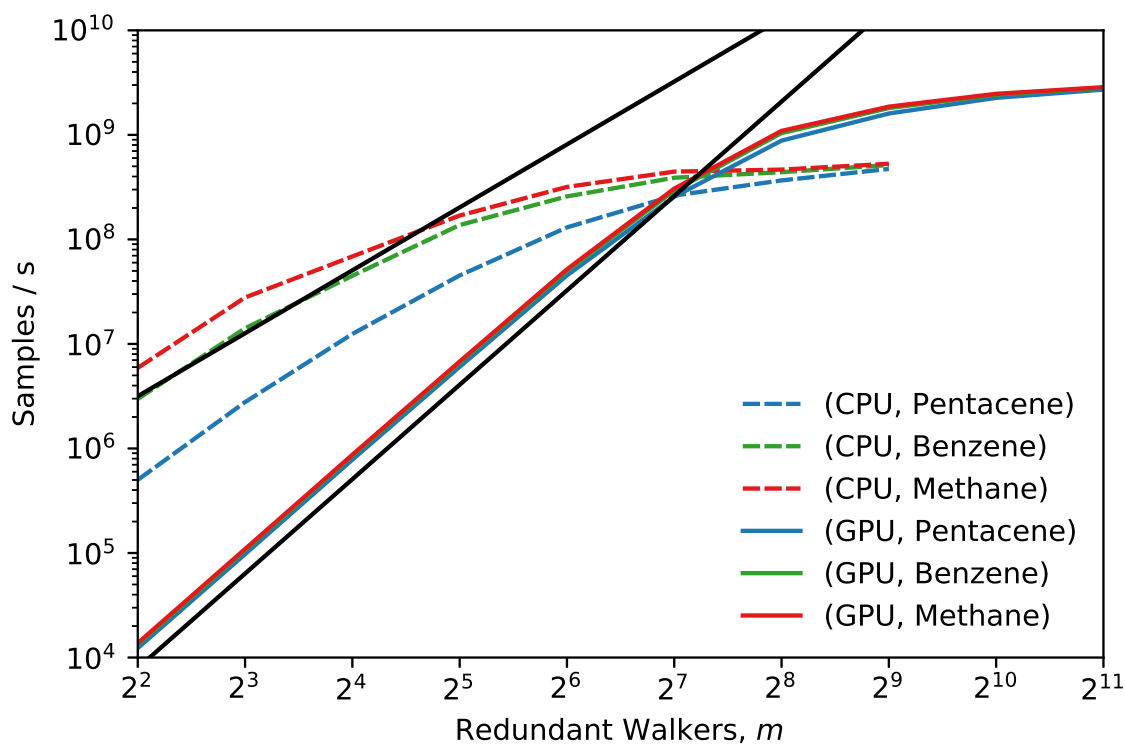


Figure 4.2: The sampling rate as a function of the number of redundant walkers (m) in a diagonal MC-GF3/aug-cc-pVDZ calculations on methane ($n_{\text{bas}} = 59$), benzene ($n_{\text{bas}} = 192$), or pentacene ($n_{\text{bas}} = 632$) on a single Blue Waters XE (CPU) or XK (GPU) node. Functions proportional to the cube and square of the number of redundant walkers (m) are superimposed to guide the eyes.

It is noted that the sampling rate for MC-GF3 converges to the same value, regardless of molecular size, which is in contrast to the behavior observed of MC-MP2 in Figure 4.1. This is because in MC-GF3 the rate-limiting steps when the number of walkers is large is the computation of the energy correction, whose cost is $O(m^3)$, while in MC-MP2 it is the computation of the traces of the Green’s functions whose cost is $O(m^2n)$. Clearly, the former has no dependence on molecular size while the latter does, explaining this observation.

In contrast to the CPU runs, the GPU-accelerated calculations display an $O(m^3)$ increase in the sampling rate until m becomes too large, whereupon the sampling rate quickly saturates. Furthermore, there is no dependence of the sampling rate on molecular size for any number of walkers. On average, the GPU-accelerated MC-GF3 program presents a speedup in the sampling rate by approximately 5.5 compared to the CPU implementation on a node-by-node basis. These desirable behaviors continue to mirror the GPU-accelerated MC-MP2 program and are explained by the same reasoning as in MC-MP2.[10].

4.1.3 Conclusions

The GPU-accelerated rendition of the MC-MB algorithm was initially conceived in response to the future outlook of high-performance computing, which indicates a trend for GPUs to be the primary source of computational power. Because of this outlook, existing and new algorithms must be adapted to execute on either CPUs or GPUs to maintain relevance. In this regard, the GPU implementation has succeeded remarkably well. More importantly, however, was the serendipitous discovery that the redundant-walker algorithm and the GPU programming paradigms mutually enhance each other. On Bluewaters, speedups greater than would be expected by comparing raw operational performances between the XK and XE nodes have been achieved for both second- and third-order methods. For MC-MP2, speedups by a factor of as high as 44 were observed, while for MC-GF3, the observed speedups were on average 5.5. *These larger than expected speedups are the result of the GPU programming paradigm of spawning a number of threads in proportion to the problem size and of automatically load-balancing them, which makes the efficiency of the redundant-walker algorithm qualitatively better on a GPU.* This paradigm makes it possible for the GPU-accelerated algorithms to use as many as 2048 redundant walkers effectively. On CPUs, such a large number of electron pairs cannot be used meaningfully for small molecules and is difficult to use for large molecules. Looking ahead, while the CPU-implementation of the MC-MB algorithm will continue to be the choice environment for developing and validating new methods, it is expected that the GPU-accelerated implementation will likely play a critical role in performing calculations on systems of chemical relevance.

4.2 Integration of the Imaginary Time Coordinates

In the original presentation of MC-MP2, the first step in deriving its integral equations was to adopt Almöf's LT-MP2 formalism. Later, during the proposal of MC-MP3, it was recognized that coordinates introduced by the LT-MP2 formalism are analogous to imaginary-time. Regardless of their origination, adopting the imaginary-time coordinates causes the energy denominator in the MBPT and MBGF energy expressions to become decoupled and is critical to expose the product of low-rank-sums integral expressions. The penalty of adopting this formalism is increasing the dimensionality of the resulting integration problem.

As mentioned several times in the preceding chapters, there are currently two approaches to perform the integrals over the imaginary-time coordinates. Historically, as in LT-MP2, the integration of the imaginary-time degrees of freedom has been performed using a quadrature rule. More recently, pioneered during the proposal of MC-GF3[13], a second approach of performing these integrations by MC was implemented and analyzed. The latter was found to be significantly more expeditious. The details of both approaches are summarized, and their performance in the context of MC-GF2 and MC-GF3 are considered.

4.2.1 Algorithm Description

As noted, during the proposal of MC-MP2, and additionally, MC-GF2, and MC-MP3, the imaginary-time coordinates introduced by the Laplace transform were integrated using a quadrature rule. In this paradigm, the integration of the imaginary-time coordinates for a n^{th} -order energy or self-energy is

$$\tilde{f}^{(n)} = \sum_{s_1, \dots, s_{n-1}}^{N_\tau} \left(f^{(n)}(\mathbf{r}_1, \dots, \mathbf{r}_{2n}, \tau^{[s_1]}, \dots, \tau^{[s_{n-1}]}) \prod_{i=1}^{n-1} \nu^{[s_i]} \right), \quad (4.29)$$

where $\tau^{[s]}$ and $\nu^{[s]}$ are the points and weights of a N_τ -point quadrature rule, historically a 21-pt Gauss-Kronrod quadrature.

For either of the second-order methods, the application of a quadrature rule to integrate the imaginary-time degrees of freedom is a logical choice. The integrations are single dimensional and can be performed almost exactly and relatively cheaply with a quadrature. However, in going beyond second-order, the curse of dimensionality arises.[60] For an n^{th} -order method, Eq. (4.29) becomes a sum of N_τ^{n-1} terms, which, in combination with the MC integration of the electronic coordinates, constitutes a large computational effort even at third-order. A quadrature rule with fewer points could alleviate the computational demands. In fact, in LT-MP2, the integrations are typically carried out using a quadrature rule with substantially fewer points.[109–112] For the MC-MB methods though, the systematic error introduced by less accurate rules is

Section 4.2 is based off of Reference [13].

not desirable. More so, applying a smaller quadrature rule only delays the onset of this obstacle rather than eliminating it.

During the development of MC-GF3, an alternate strategy to integrate the imaginary-time degrees of freedom by direct-MC was proposed. The contribution to an MC step under this schema is

$$\tilde{f}^{(n)} = \frac{f^{(n)}(\mathbf{r}_1, \dots, \mathbf{r}_n, \tau_1, \dots, \tau_{n-1})}{\prod_{i=1}^{n-1} \omega_\tau(\tau_i)}, \quad (4.30)$$

where the importance function for the imaginary-time coordinates is

$$\omega_\tau(\tau) = \lambda e^{-\lambda\tau}. \quad (4.31)$$

where the parameter λ is equal to $2(\epsilon_{\text{LUMO}} - \epsilon_{\text{HOMO}})$. Since the importance function is a simple exponential distribution, sampling is carried out trivially using the inverse-CDF method. Using MC to perform the integrations introduces additional uncertainty, which is offset by needing to evaluate the energy at only one set of imaginary-time coordinates per MC step. Additionally, stochastically integrating the imaginary coordinates is in principle exact, in contrast, to the use of a quadrature rule in which there will always be some systematic error.

4.2.2 Performance Analysis

Tables 4.2 and 4.3 compare the performance of the quadrature and direct-MC approaches to integrate the imaginary-time dimensions for MC-GF2 and MC-GF3 calculations. The quadrature used was the 21-point Gauss-Kronrod rule,[71] which was adopted in all of the previous MC-MB implementations.[14–16] In all cases, the MC imaginary-time integration is significantly more efficient than the quadrature rule. For MC-GF2, the stochastic integration is more efficient by a factor ranging from ~ 2 to ~ 8 . For MC-GF3, the stochastic integration is even more efficient, with the increase in efficiency ranging from 50 to 200. There are two competing factors in the efficiency, η , which are now discussed.

The first factor, the speedup, is the ratio of the times spent in one MC step of the two algorithms. Equation (4.29) and Equation (4.30) immediately predict that an upper bound for this ratio, in the case of 21 grid points, is 21 for MC-GF2 and $21 \times 21 = 441$ for MC-GF3. This is borne out in the observed speedups in Tables 4.2 and 4.3, which range between 6 and 20 for MC-GF2 and 406 and 440 for MC-GF3. The slight decrease in the speedup with increasing molecular size can be traced to the increased cost of the MO evaluation step, which does not depend on the imaginary-time integration details. The deviation from the upper bound is more significant for MC-GF2, as the evaluation of the second-order self-energy is

Table 4.2: Performance of the direct MC integration in the imaginary-time dimensions relative to the quadrature for the integration of the second-order self-energy. See Appendix A for the unprocessed data.

Molecule	Speedup ^a	Variance ^b	Efficiency ^c
Methane	9.65	2.01	4.80
Propane	20.12	2.39	8.40
Benzene	19.91	2.53	7.87
<i>n</i> -Nonane	6.10	2.58	2.36
<i>n</i> -Decane	6.08	2.49	2.44
Pentacene	6.70	3.24	2.07

^a The rate of reduction in wall time per MC step as measured by $T_{\text{ref}}/T_{\text{alt}}$ where the reference method is the quadrature and the alternative method is the direct MC integration.

^b The rate of increase in variance as measured by $\sigma_{\text{alt}}^2/\sigma_{\text{ref}}^2$.

^c The rate of increase in MC efficiency.

Table 4.3: The same as Table 4.2 but for MC-GF3.

Molecule	Speedup	Variance	Efficiency
Methane	440	1.95	226
Propane	416	3.10	134
Benzene	418	3.33	126
<i>n</i> -Nonane	406	3.15	129
<i>n</i> -Decane	412	1.98	209
Pentacene	407	7.17	56.7

relatively cheap, and thus, the increasing cost of the MO evaluation has a greater impact.

The second factor, the ratio of the variances, describes the inevitable increase in the statistical uncertainty by the MC imaginary-time integration. When the quadrature rule is used for imaginary-time integration, an 12(18)-dimensional MC integration is performed over the spatial coordinates of the electrons for MC-GF2 (MC-GF3). In contrast, the direct MC algorithm integrates all 13(20) dimensions stochastically. Clearly, the variance of the latter is greater than that of the former. Table 4.2 and Table 4.3 show that the variance indeed increases but only by a factor of 2 to 3 and 2 to 7 for MC-GF2 and MC-GF3 respectively, despite the fact that the direct MC integration samples only one imaginary-time point per MC step, while the quadrature samples 21 or 441 imaginary-time points per MC step for spatial integrations.

4.2.3 Conclusions

The results of the performance analysis for MC-GF2 and MC-GF3 support the original motivation of introducing the MC imaginary-time integration algorithm: A balanced treatment of different integration dimensions with comparable statistical uncertainties leads to superior overall performance. In other words, using a quadrature rule expends an inordinate amount of computational effort to numerically exactly integrate only $n - 1$ of the $7n - 1$ dimensions for an n^{th} theory, clearly passing the point of diminishing returns in overall performance. Using the more balanced approach, i.e. all MC integration, leads to an improvement

of up to 8.4 and 221 in the overall calculation rates with no loss of accuracy for MC-GF2 and MC-GF3. In fact, the improvement in integrating the imaginary-time coordinates was instrumental in allowing for the development of MC-MP4, and it is similarly expected to play as significant of a role in the development of MC-GF methods beyond third order.

4.3 Direct Sampling of the Electronic Coordinates

The importance function for a pair of electrons, proposed alongside MC-MP2, is ubiquitous in all of the MC-MB methods that have been developed by the Hirata lab. During the development of nearly all of the MC-MB methods, the importance function for a pair of electrons has been sampled using the Metropolis algorithm. The only exception is MC-MP4, which used the following. As discussed in Chapter 1, the Metropolis algorithm is a powerful tool that may be used to sample nearly any arbitrary distribution. However, the Metropolis algorithm, as it is a Markov sampling technique[60–62], inherits several less than desirable properties. One of the most significant is that Markov sampling techniques introduce autocorrelation into the variance[60–62], which causes it to increase. This, in turn, clearly makes computations more demanding.

Another difficulty that may be encountered when using the Metropolis algorithm[63, 64] is that the generated distribution is only guaranteed to converge to the desired distribution in the limit of an infinite number of samples[60–62]. This can become problematic when MC-MB methods are applied to systems with multiple well-separated molecules, as the trajectories of the walkers are likely to be non-ergodic. With the redundant-walker algorithm, this difficulty can be partially mitigated by using a sufficiently large number of walkers and evenly dispersing them across the multiple molecules. In this way, there are walkers present on all molecules, and walkers need not transition between separate molecules. Nevertheless, eliminating this trait *a priori* is still desirable.

Another advantage that direct sampling presents is an increase in the reliability of the estimate of the variance. In MC-MB methods when using Metropolis sampling, the blocking algorithm of Flyvberg[69] is used to account for the autocorrelation in the variance. The blocking algorithm effectively reduces the number of samples used to compute the variance by a factor of the block size, which can be quite large for MC-MB methods. This reduction in the number of effective samples causes the blocked variance to become unstable when the block size becomes large and the simulation length is short. Comparatively, the number of samples used to construct the estimate of the variance when using a direct sampling approach is the same as the simulation length. As such, not only is the estimate of the variance more stable, it is possible to construct a reliable estimate from a shorter simulation.

Given the number of methods[10–17, 73, 113] that employ the importance function for a pair of electrons, it is desirable to eschew the Metropolis algorithm in favor of a direct sampling approach. Here, an algorithm is proposed in which the importance function for a pair of electron for a pair of electrons is sampled directly. In the algorithm, several carefully chosen transformations are applied to the importance function which causes it to decouple into a product of one- and two-dimensional distributions. The one- and two-dimensional distributions are then trivially sampled either through readily available standard algorithms or through their quantile functions. Because this algorithm produces a sequence of independent random variables, all autocorrelations in the variance are entirely eliminated, which leads to a moderate performance increase. At the same time, because the importance function itself has not been modified, this technique is immediately applicable to all of the MC-MB methods produced by the Hirata lab.

4.3.1 Algorithm Description

In order to sample the importance function for a pair of electrons, Eq. (2.14), the two sums over the atom-centered Gaussians are rewritten into a single sum as

$$\omega(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{N_g} \frac{g(\mathbf{r}_1)g(\mathbf{r}_2)}{r_{12}} \quad (4.32)$$

$$= \frac{1}{N_g} \sum_{i,j}^{n_g} \frac{\varphi_i(\mathbf{r}_1)\varphi_j(\mathbf{r}_2)}{r_{12}} \quad (4.33)$$

$$= \sum_{i,j}^{n_g^2} P_{i,j} w_{i,j}(\mathbf{r}_1, \mathbf{r}_2) \quad (4.34)$$

$$= \sum_n^{n_g^2} P_n w_n(\mathbf{r}_1, \mathbf{r}_2), \quad (4.35)$$

where $\varphi_i(\mathbf{r})$ is the i^{th} S-type Gaussian in $g(\mathbf{r})$, $w_{i,j}(\mathbf{r}_1, \mathbf{r}_2)$ is an importance primitive, $P_{i,j} = N_{i,j}/N_g$ is the weight of an importance primitive in the importance function, and $N_{i,j}$ and N_g are the normalization constant for an importance primitive and the importance function respectively. The combine index $n = i * n_g + j$ is introduced for simplicity. An importance primitive is defined as

$$w_{i,j}(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{N_{i,j}} \frac{\phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}. \quad (4.36)$$

The definition of the importance primitive includes a factor of $N_{i,j}$ to make the integral of the importance primitives normalized to unity. Thus, each importance primitive is itself a valid PDF.

With the importance function for a pair of electrons in this form, sampling is carried out as such. First,

a single importance primitive is selected randomly with probability, P_n . A coordinate transformation and a coordinate substitution are then subsequently applied to the chosen importance primitive to decouple it into a product of functions that depend on either one or two coordinates. The coordinates of the electron pair are then generated based on the decoupled functions and then projected back into the original coordinate system.

The coordinate transformation moves the Gaussian functions of the importance primitive such that their centers are symmetric about the origin and aligned along the z -axis. The Gaussian function associated with the index i is specifically placed to be along the positive z -axis. Thus, the centers of the Gaussian in the transformed coordinate system are $(0, 0, \pm Z_n)$ where

$$Z_n = |\mathbf{R}_i - \mathbf{R}_j|/2. \quad (4.37)$$

This is accomplished by reorienting the entire coordinate system. The desired transformation is described by

$$\tilde{\mathbf{r}} = \hat{R}_y \left(-\arccos\left(\frac{\Delta Z_n}{2Z_n}\right) \right) \hat{R}_z \left(-\arctan\left(\frac{\Delta Y_n}{\Delta X_n}\right) \right) \hat{T}_{-\bar{\mathbf{R}}_n} \mathbf{r}, \quad (4.38)$$

where $\mathbf{r}$ is a point in the original coordinate system, $\tilde{\mathbf{r}}$ is a point in the alternate coordinate system, $\Delta X = X_i - X_j$, $\bar{\mathbf{R}} = (\mathbf{R}_i + \mathbf{R}_j)/2$, $\hat{T}_{\mathbf{v}}$ is a spacial translation by $\mathbf{v}$, and $\hat{R}_x(\theta)$ is a spacial rotation by θ about the x -axis. Transforming a point in the alternate coordinate system back to the original coordinate system is trivially accomplished by applying the inverse of the operator sequence in Eq. (4.38),

In the alternate coordinate system, an importance primitive is fully expressed as

$$w_n(\mathbf{r}_1, \mathbf{r}_2) = \frac{c_i c_j}{N_n} \frac{\exp(-\alpha_i(x_1^2 + y_1^2 + (z_1 - Z_n)^2)) \exp(-\alpha_j(x_2^2 + y_2^2 + (z_2 + Z_n)^2))}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}}. \quad (4.39)$$

Once in the alternate coordinate system, a coordinate substitution is applied to the importance primitive. The coordinate substitution expresses the electron pair in a weighted center-of-mass like representation. The key feature of this coordinate substitution is to make the inner-electronic distance a coordinate, rather than

a derived property. The coordinate substitution $\mathbf{F}(x, y, z, r, \theta, \phi)$ is given by

$$x_1 = x - \frac{\gamma}{\alpha} r \cos(\phi) \sin(\theta), \quad (4.40)$$

$$y_1 = y - \frac{\gamma}{\alpha} r \sin(\phi) \sin(\theta), \quad (4.41)$$

$$z_1 = z - \frac{\gamma}{\alpha} r \cos(\theta), \quad (4.42)$$

$$x_2 = x + \frac{\gamma}{\beta} r \cos(\phi) \sin(\theta), \quad (4.43)$$

$$y_2 = y + \frac{\gamma}{\beta} r \sin(\phi) \sin(\theta), \quad (4.44)$$

$$z_2 = z + \frac{\gamma}{\beta} r \cos(\theta), \quad (4.45)$$

where (x, y, z) is the center of the electron pair in Cartesian coordinates, r is half the inter-electronic distance, (ϕ, θ) are the azimuthal and polar angles which describe the orientation of the electron pair, and

$$\gamma^2 = \frac{\alpha\beta}{\alpha + \beta}. \quad (4.46)$$

Applying the coordinate substitution to Eq. (4.39) produces

$$\begin{aligned} \omega_n(\mathbf{r}_1, \mathbf{r}_2) &= \omega_n(\mathbf{F}(x, y, z, r, \theta, \phi)) |\mathbf{J}_{\mathbf{F}}(x, y, z, r, \theta, \phi)| \\ &= N\left(x; 0, \frac{1}{\sqrt{2(\alpha + \beta)}}\right) N\left(y; 0, \frac{1}{\sqrt{2(\alpha + \beta)}}\right) N\left(z; \frac{Z_n(\alpha - \beta)}{2(\alpha + \beta)}, \frac{1}{\sqrt{2(\alpha + \beta)}}\right) \\ &\quad \times U(\phi; 0, 2\pi) w_{r\theta}(r, \theta; Z_n\gamma), \end{aligned} \quad (4.47)$$

where $|\mathbf{J}_{\mathbf{F}}|$ is the determinant of the Jacobian of the coordinate substitution whose value is

$$|\mathbf{J}_{\mathbf{F}}(x, y, z, r, \theta, \phi)| = \frac{r^2 \sin(\theta)}{\gamma^3}. \quad (4.48)$$

The component functions of the importance primitive after the coordinate substitution are several normal distributions, $N(x; \mu, \sigma)$, whose PDF when with center μ and width σ is

$$N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (4.49)$$

a uniform distribution, $U(\phi; a, b)$, whose PDF when on the interval $[a, b]$ is

$$U(\phi; a, b) = \frac{1}{b-a} \quad b > a, \quad (4.50)$$

and $w_{r\theta}(r, \theta; \zeta)$, which is the joint distribution of the remaining two coordinates r and θ whose PDF is

$$w_{r\theta}(r, \theta; \zeta) = \frac{2\zeta e^{-r^2 - 2\zeta r \cos(\theta) - \zeta^2} r \sin(\theta)}{\sqrt{\pi} \operatorname{erf}(\zeta)}. \quad (4.51)$$

Thus, this substitution decouples an importance primitive into three coordinates that are normally distributed, one that is uniformly distributed, and leaves the remaining two coordinates jointly distributed according to Eq. (4.51). The coordinates that are normally distributed may be sampled by well-known algorithms[68, 114, 115] and the uniformly distributed coordinate may be sampled trivially.

Two approaches are required to sample the radial and azimuthal coordinates. The appropriate approach depends on the value of Z_n as Eq. (4.51) becomes indeterminate if the parameter $\zeta = 0$ as is the case if $Z_n = 0$. This condition occurs if the two Gaussian functions of the randomly selected importance primitive share the same center. The more general case of $Z_n \neq 0$ is given first and then followed by the special case of $Z_n = 0$.

If $Z_n \neq 0$, a two step approach is used to generate (r, θ) . First, r is generated from its marginal distribution. Then θ is generated from $w_{r\theta}(r, \theta; \zeta)$ given the generated value of r . The marginal distribution of r , $w_r(r; \zeta)$, is given by integrating θ out of Eq. (4.51) and is

$$w_r(r; \zeta) = \frac{e^{-(r+\zeta)^2} (e^{4\zeta r} - 1)}{\sqrt{\pi} \operatorname{erf}(\zeta)}. \quad (4.52)$$

To generate a value of r distributed according to Eq. (4.52), the inverse-CDF method is used.

The CDF of $w_r(r; \zeta)$ is

$$W_r(r; \zeta) = \frac{2\operatorname{erf}(\zeta) - \operatorname{erf}(r + \zeta) + \operatorname{erf}(r - \zeta)}{2\operatorname{erf}(\zeta)}. \quad (4.53)$$

Substitution of Eq. (4.53) into Eq. (1.43) produces an equation that is not directly solvable for r , and thus the quantile function is unavailable. Instead, a solution is found numerically using Halley's method. Halley's method is a cubically convergent iterative root finding algorithm.[116] The next value of x produced by

Halley's method is

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)] - f(x_n)f''(x_n)}, \quad (4.54)$$

where $f(x)$ is the left side of Eq. (1.43). Equation (4.54) requires the first and second derivatives of $f(x)$. Since the derivative of p with respect to r is zero, the derivatives of the left side Eq. (1.43) are the derivatives of W_r . The first derivative of W_r is trivially w_r . The second derivative is

$$\frac{d^2}{dr^2}W_r(r; \zeta) = \frac{2 \left(e^{-(r+\zeta)^2} (r + \zeta) - e^{-(r-\zeta)^2} (r - \zeta) \right)}{\sqrt{\pi} \operatorname{erf}(\zeta)}. \quad (4.55)$$

The expectation value of r distributed according to Eq. (4.52),

$$E[r]_{w_r} = \frac{\zeta}{\operatorname{erf}(\zeta)}, \quad (4.56)$$

is used as the initial guess for the search with Halley's method.

Given a fixed value of r , θ is distributed according to

$$w_\theta(\theta; r, \zeta) = \frac{1}{w_r(r; \zeta)} w_{r\theta}(r, \theta; \zeta), \quad (4.57)$$

where the factor of $w_r(r; \zeta)$ arises as a normalizing constant. Values of θ are also generated using the inverse-CDF method. The CDF of $w_\theta(\theta; r, \zeta)$ is

$$W_\theta(\tilde{\theta}; \tilde{r}, \zeta) = \frac{1}{w_r(r; \zeta)} \frac{e^{-(r^2+\zeta^2)} (e^{-2\zeta r \cos(\phi)} - e^{-2\zeta r})}{\sqrt{\pi} \operatorname{erf}(\zeta)}. \quad (4.58)$$

Substitution of Eq. (4.58) into Eq. (1.43) and solving for θ gives the quantile function,

$$\theta = \operatorname{arcsec} \left(\frac{2\zeta r}{2\zeta r - \log(p\sqrt{\pi} \operatorname{erf}(\zeta) w_r(r; \zeta) e^{(r+\zeta)^2} + 1)} \right). \quad (4.59)$$

The second approach, which is required if $Z_n = 0$ and hence the parameter ζ also becomes 0 causing the joint distribution of r and θ to become indeterminate, is now discussed. Taking the limit of Eq. (4.51) as ζ approaches 0 from the positive side (ζ is strictly greater than 0) produces

$$\lim_{\zeta \rightarrow 0^+} w_{r\theta}(r, \theta; \zeta) = w_\theta(\theta) \chi_2(r). \quad (4.60)$$

where $\chi_2(r)$ is a chi distribution with two degrees of freedom and whose PDF is

$$\chi_2(r) = r e^{-\frac{r^2}{2}}, \quad (4.61)$$

and

$$w_\theta(\theta) = \frac{\sin(\theta)}{2}. \quad (4.62)$$

The radial coordinate may be generated from standard algorithms while θ is trivially generated using its quantile function. The CDF of θ is

$$W_\theta(\theta) = \frac{1}{2}(1 - \cos(\theta)). \quad (4.63)$$

Insertion of Eq. (4.63) into Eq. (1.43) and solving for θ gives the quantile function,

$$\theta = \arccos(1 - 2p). \quad (4.64)$$

Once all of the coordinates have been generated in the center-of-mass like representation in the alternate coordinate space, the coordinates of the electron pair in the original coordinate space are obtained through Eq. (4.40) through Eq. (4.45) and then by applying the inverse of Eq. (4.38).

4.3.2 Performance Analysis

The primary motivation to propose and implement the direct-sampling algorithm is to eliminate the autocorrelation in the variance and thereby decrease the cost of MC-MB methods. Therefore, the MC efficiency for two methods, MC-MP2 and MC-MP3, when using the direct-sampling algorithm rather than the Metropolis algorithm is examined. The components of the MC efficiency are computed separately as the step rate may be obtained through relatively short simulations while the estimates of the variance for both the second- and third-order energy corrections are obtained from a single more lengthy MC-MP3 simulation. All calculations required were performed using a single executable capable of sampling with either the Metropolis algorithm or the proposed direct-sampling algorithm to ensure that no calculation is unfairly disadvantaged. This study examines four molecules, water ($r_{\text{OH}} = 0.9573\text{\AA}$), methane ($r_{\text{CH}} = 1.0848\text{\AA}$), ammonia ($r_{\text{NH}} = 1.0123\text{\AA}$), and D-leucine at the cc-pVDZ level of chemistry. Although the redundant-walker algorithm dictates the number of walkers be proportional to the system size, for convenience 128 redundant walkers are used for all

Table 4.4: The time in seconds required to execute a 128 MC step when using either the direct sampler or the Metropolis sampler. t_D (t_M) is the time per step for calculation using the direct (Metropolis) sampling algorithm.

Method	Molecule	t_D	t_M	$t_D - t_M$	t_M / t_D
MC-MP2	Methane	0.169	0.149	0.020	0.882
	Ammonia	0.143	0.126	0.017	0.879
	Water	0.127	0.114	0.013	0.899
	D-Leucine	1.014	0.987	0.027	0.974
MC-MP3	Methane	1.437	1.417	0.019	0.987
	Ammonia	1.403	1.379	0.024	0.983
	Water	1.372	1.359	0.013	0.991
	D-Leucine	2.398	2.378	0.020	0.992

simulations. All calculations required to establish the step rate were carried out on a single XE node using all of its processors. The calculations required to obtain the estimates of the variance were carried out using various numbers of XE nodes as required to obtain sufficiently converged estimates.

Table 4.4 presents the average time required to perform 128 MC steps when utilizing either the direct-sampling algorithm or the Metropolis algorithm. It is observed that the difference between the average sampling times is nearly constant across both methods. The sampling times is also nearly constant for each given molecule.

The ratio of the sampling times will clearly have a less significant impact on the MC efficiency when the generation of the electron coordinates requires a lesser fraction of the total MC step time. This is the case for D-leucine as the larger molecular size, as compared to the other molecules in the study, dictates that a greater fraction of each MC step is spent constructing the atomic and molecular orbital amplitudes. Similarly, the ratio is less significant for MC-MP3, as the construction of the third-order energy correction is more computationally demanding than the construction of the second-order energy correction. With both of these effects, the ratio of the time per MC step for an MC-MP3 calculation on the modestly sized D-leucine is nearly unity. As such, the increase in the time per MC step required by the direct-sampling algorithm is expected to be negligible for any calculation of practical interest.

Because the speedup of the direct-sampling algorithm is a result of eliminating autocorrelation in the variance, special attention is given to ensuring that the entirety of the autocorrelations for the calculations using the Metropolis algorithm is accounted for. The variance of the energy for the calculations using the Metropolis algorithm is estimated through the blocking algorithm of Flyvberg[69]. If the variance is plotted as a function of the number of blocking transformations, as in Figures (4.3) and (4.4), an accurate estimate of the variance occurs in the smooth plateau region that begins after the initial rise and ends when the estimate of the variance becomes unstable. For MC-MP2, this region begins after as few as 6 blocking transformations

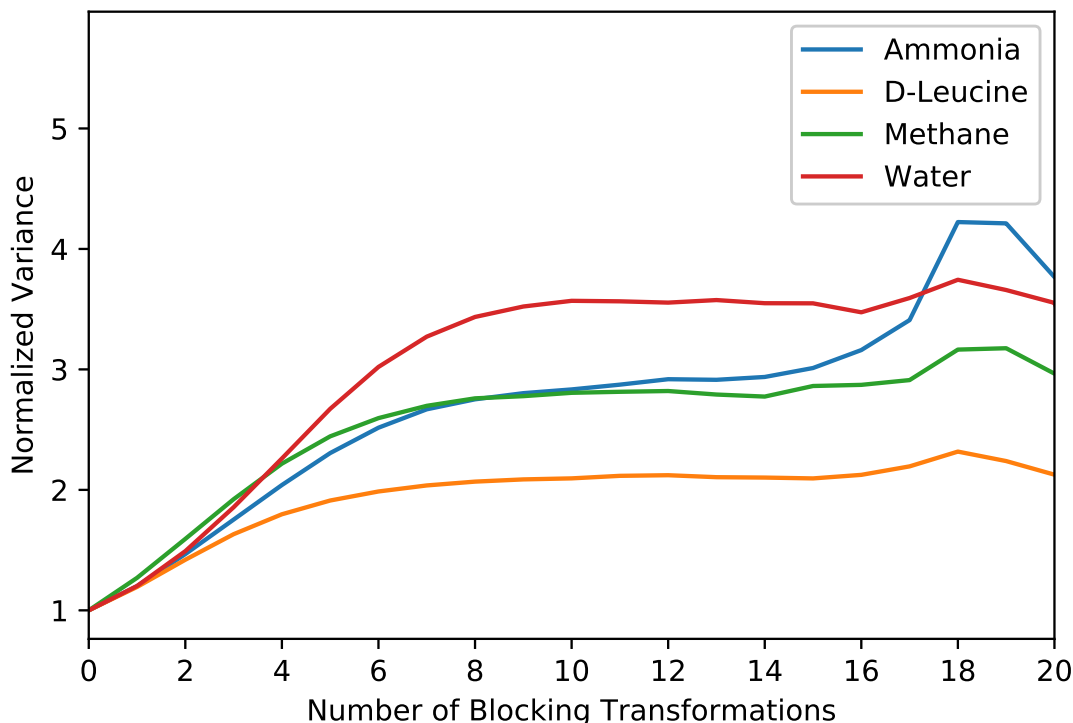


Figure 4.3: The blocked variance of as a function of block size for the second-order energy correction. The blocked variances (blocking transformations greater than 0) are normalized with respect to the unblocked variance (blocking transformations equal to zero).

and ends after approximately 15 transformations. Similarly, for MC-MP3, this region begins after as few as 5 blocking transformations and ends after approximately 10 transformations. The estimates of the variance after 10 blocking transformations for MC-MP2 and after 8 transformations for MC-MP3 are used in the remaining analysis.

The estimates of the variances and their ratios for all molecules at both levels of theory are given in Table 4.5. For MC-MP2, the estimates of the variance when using the direct-sampling algorithm are reduced by a factor ranging from 2 to 3.6 as compared to the those obtained with the Metropolis algorithm. Similarly, for MC-MP3, the estimates of the variance are decreased by a factor ranging from 1.24 to 1.4.

It is interesting to note that the relative magnitude of the autocorrelations for the third-order energy corrections are lesser the second-order energy corrections. This result parallels that fewer blocking transformations are required to account for the autocorrelation in the third-order energy corrections. This is likely because the integrand of the third-order energy corrections requires three redundant walkers as input rather than two. Given the greater number of walkers, there is a greater chance that at least one electron pair is

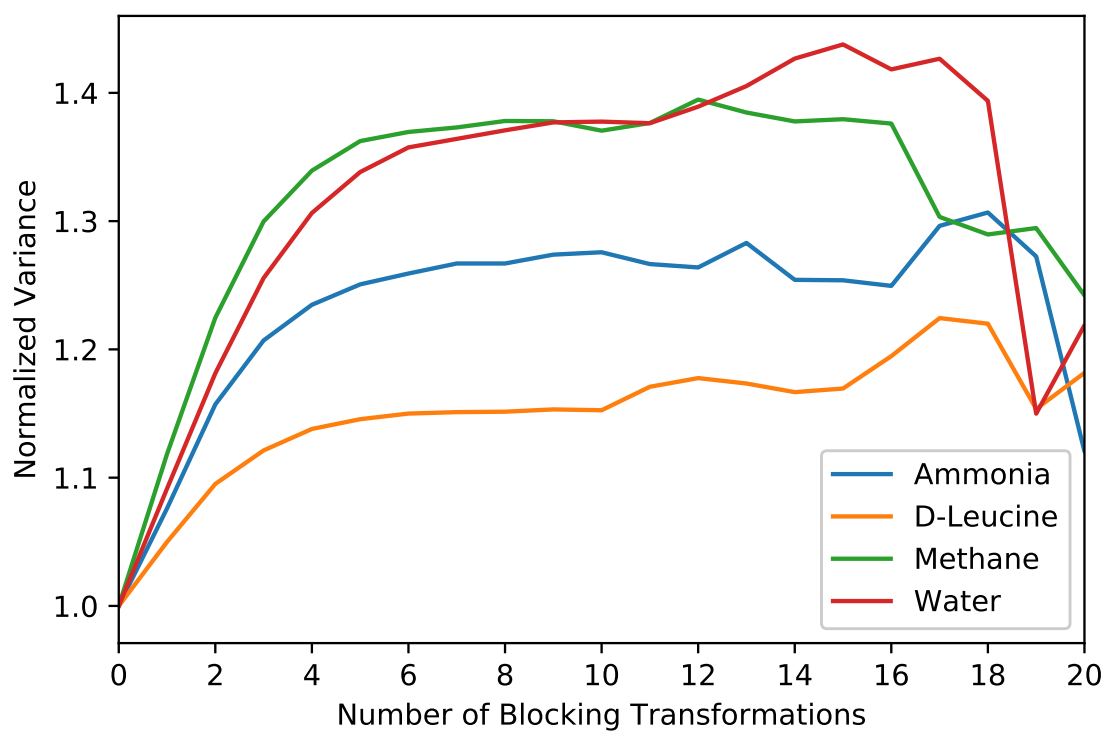


Figure 4.4: The same as Figure 4.3, but for MC-MP3 calculations.

Table 4.5: The variance in E_h^2 obtained when sampling when using the direct-sampling or the Metropolis algorithm.

Molecule	MC-MP2			MC-MP3		
	Direct	Metropolis	Ratio	Direct	Metropolis	Ratio
Methane	$1.91 \cdot 10^{-1}$	$5.35 \cdot 10^{-1}$	2.80	$6.48 \cdot 10^{-1}$	$8.83 \cdot 10^{-1}$	1.36
Ammonia	$1.69 \cdot 10^{-1}$	$4.79 \cdot 10^{-1}$	2.84	$4.27 \cdot 10^{-1}$	$5.46 \cdot 10^{-1}$	1.28
Water	$1.30 \cdot 10^{-1}$	$4.66 \cdot 10^{-1}$	3.57	$1.69 \cdot 10^{-1}$	$2.36 \cdot 10^{-1}$	1.40
D-Leucine	$5.29 \cdot 10^2$	$1.10 \cdot 10^3$	2.09	$2.96 \cdot 10^4$	$3.66 \cdot 10^4$	1.24

Table 4.6: The MC efficiency achieved by using the direct-sampling algorithm as compared to using the sampling using the Metropolis algorithm.

Molecule	MC-MP2	MC-MP3
Methane	2.47	1.34
Ammonia	2.49	1.26
Water	3.21	1.38
D-Leucine	2.03	1.23

moved when evaluating the third-order integrand. This, in turn, could cause a decrease in the correlation time and the magnitude of the autocorrelation.

Table 4.6 presents the MC efficiency which is computed using the results presented in Tables 4.4 and 4.5. The overall gain in efficiency closely mirrors the reduction in the estimates of the variances, albeit slightly reduced due to the lesser sampling rate. The gain in efficiency, while appearing moderate, is still taken to be a significant increase for the MC-MB methods, as it serves to reduce the massive prefactors on the cost functions of the MC-MB family of methods.

4.3.3 Conclusions

An algorithm to directly sample the importance function for a pair of electrons has been proposed and evaluated. The direct sampling approach provides more robust statistics which are guaranteed to produce the desired distribution. The statistics of MC-MB methods are also improved quantitatively by reducing the variance. More importantly, the direct-sampling algorithm is method agnostic and may replace the Metropolis algorithm in all MC-MB methods. This is crucial as it has been well established that the cost functions for MC-MB methods with system size have a more favorable scaling than their conventional counterparts, but are plagued by massive prefactors. Thus, any reduction in the prefactors, such as the one provided by the direct-sampling algorithm, serves to broaden the range of applicability for MC-MB methods.

4.4 Control Variates for MC-MP Methods

The previously presented acceleration techniques provide a substantial performance boost to MC-MB methods by increasing the computational power that may be brought to bear on a problem, by more efficiently integrating the imaginary-time coordinates, and by eliminating autocorrelation from the variance which in turn improves the resulting statistics. However, none of these approaches directly combat the substantial variance in MC-MB simulations, which is the main contributor to the enormous prefactor on the cost functions for these methods. In this section, an effective method to reduce the variance of MC-MP methods through the use of control variates is proposed and assessed.

4.4.1 Control Variates

Control variates are a well-known technique to reduce the variance for an MC integration problem. [60–62] Suppose an MC integration problem where the integral of $f(x)$ under the PDF, $\rho(x)$, is desired,

$$\mu_f = \int f(x)\rho(x)dx \quad (4.65)$$

$$\approx \frac{1}{N} \sum_{n=1}^N f(x_n) \Big|_{x \sim \rho(x)} \quad (4.66)$$

$$\approx \hat{\mu}_f. \quad (4.67)$$

Let there be a second related function, $g(x)$, in which the integral of $g(x)$ under $\rho(x)$, μ_g , is known exactly. Then, $g(x)$ may be used as a control variate for the MC integration of $f(x)$. If $g(x)$ is used as a control variate, then the control variate estimate for μ_f is

$$\tilde{\mu}_f = \frac{1}{N} \sum_{n=1}^N f(x_n) - \alpha(g(x_n) - \mu_g) \Big|_{x \sim \rho(x)}, \quad (4.68)$$

where α is an adjustable parameter and the tilde is used to indicate the control variate estimate of μ_f . The variance of the control variate estimate of μ_f is

$$\sigma_{cv:f}^2 = \sigma_f^2 + \alpha^2 \sigma_g^2 + 2\alpha \mathbf{K}_{f,g}. \quad (4.69)$$

The adjustable parameter α is determined by minimizing the expression for the variance of the control variate estimate of μ_f . It is easy to show that the optimum value of α is

$$\alpha = -\frac{\mathbf{K}_{f,g}}{\sigma_g^2}. \quad (4.70)$$

Thus the variance of the control variate estimate becomes

$$\sigma_{cvf}^2 = \sigma_f^2 - \frac{\mathbf{K}_{f,g}^2}{\sigma_g^2} \quad (4.71)$$

$$= (1 - \rho_{f,g}^2) \sigma_f^2. \quad (4.72)$$

where $\rho_{f,g}$ is the Pearson correlation coefficient between $f(x)$ and $g(x)$.

If multiple control variates are used to reduce the variance for the MC integration of $f(x)$, the control variate estimate of μ_f becomes

$$\tilde{\mu}_f = \frac{1}{N} \sum_{n=1}^N f(x_n) - \vec{\alpha} \cdot (\vec{g}(x_n) - \vec{\mu}_g) \Big|_{x \sim \rho(x)}, \quad (4.73)$$

where $\vec{\alpha}$ is now a vector of adjustable parameters, $\vec{g}(x_n)$ is the vector of the control variate functions evaluated at x_n , and $\vec{\mu}_g$ is the vector of the exact integrals of the control variate functions. The variance of the control variate estimate of μ_f , assuming that it has been minimized with respect to the adjustable parameters, is

$$\sigma_{cvf}^2 = \sigma_f^2 - \vec{\alpha} \cdot \mathbf{K}_{f,\vec{g}}. \quad (4.74)$$

where $\mathbf{K}_{f,\vec{g}}$ is the vector of the covariances between $f(x)$ and $\vec{g}(x)$, and the values of the adjustable parameters which minimize the variance are found by solving

$$\mathbf{K}_{\vec{g},\vec{g}} \vec{\alpha} = \mathbf{K}_{f,\vec{g}}, \quad (4.75)$$

where $\mathbf{K}_{\vec{g},\vec{g}}$ is the covariance matrix of the control variates.

As stated, the only condition for $g(x)$ to be used as a control variate for the MC integration of $f(x)$ is that μ_g must be known. For $g(x)$ to be an effective control variate, as demonstrated by Eq. (4.72), $g(x)$ and $f(x)$ should be either strongly correlated or strongly anti-correlated. Additionally, it is desirable for the evaluation of $g(x)$ to not be overly burdensome, to prevent the compression of the variance from being overshadowed by additional computational costs.

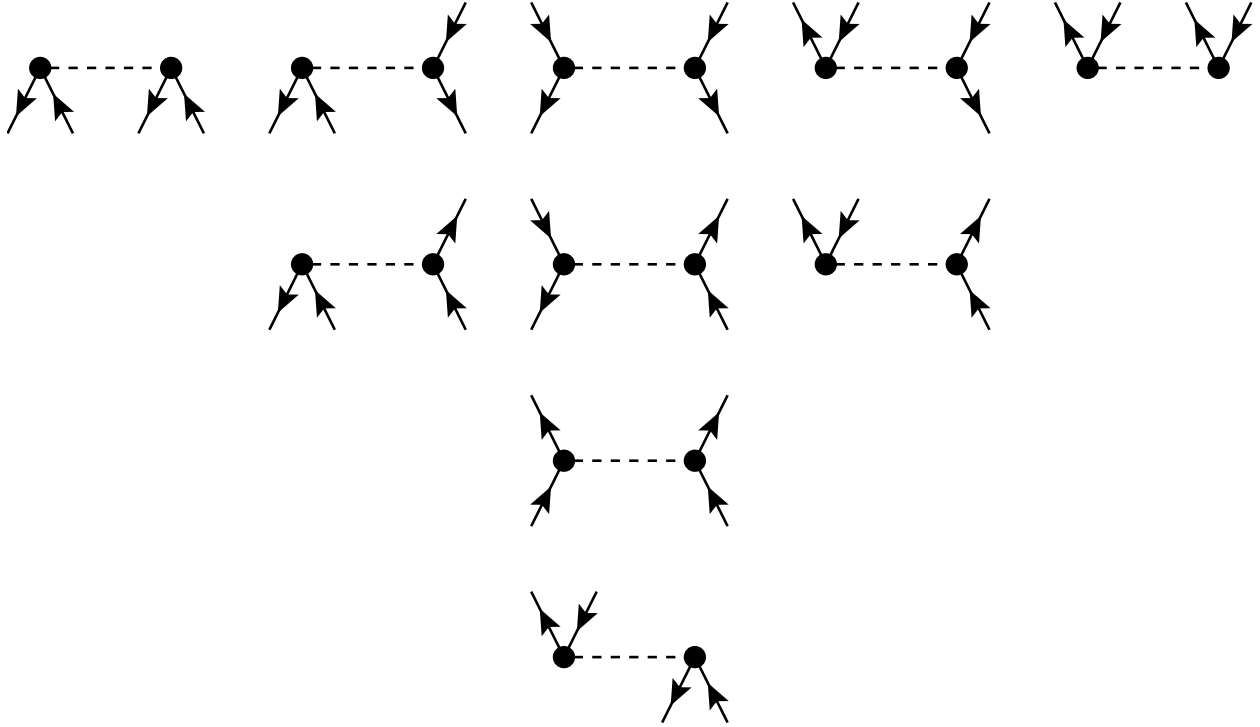


Figure 4.5: All possible two-particle interaction vertices that may be present in Goldstone diagrams.

4.4.2 Control Variates for MC-MP Methods

For the MC-MP family of methods, many useful control variates may be produced in a straightforward manner. The control variates for a particular rank of perturbation theory are generated by multiplying the integrand of the relevant energy correction by additional factors of the inner-electronic distances. This action, if carefully applied, will be shown to cause the entire integral to collapse to zero regardless of any other factors. Thus, the criterion to know the exact integrals of the proposed control variate functions is satisfied. By virtue of being minimally modified versions of the integrand of the correlation energy, these control variates satisfy the need to be strongly correlated with the target integral. For the same reason, the additional cost to compute the control variates will be shown to be small as they are already evaluated during the course of an MC-MP simulation.

As noted, care must be taken when multiplying the integrand of the energy corrections by additional factors of the inner-electronic distances to produce integrals which evaluate to zero. Diagrammatic arguments establish a scheme to create such control variates. First, the action of multiplying a single interaction vertex by a factor of its associated inner-electronic distance is established. Figure 4.5 presents all of the possible orientations of the propagator lines with respect to an interaction vertex. The interaction vertex in the

left-most column is interpreted to be

$$\text{---}\text{---}\text{---} = \frac{1}{2} \sum_{i,j,a,b} \langle ij|ab \rangle \{i^\dagger j^\dagger ba\} \quad (4.76)$$

$$= \frac{1}{2} \sum_{i,j,a,b} \iint \frac{\phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_2)\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_2)}{r_{12}} d\mathbf{r}_1 d\mathbf{r}_2 \{i^\dagger j^\dagger ba\}. \quad (4.77)$$

Multiplying the integrand by an additional factor of r_{12} cancels the r_{12} in the denominator. After multiplication by the inner-electronic distance, the integration problem becomes

$$\frac{1}{2} \sum_{i,j,a,b} \iint \phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_2)\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \{i^\dagger j^\dagger ba\} = \frac{1}{2} \sum_{i,j,a,b} \langle i|a \rangle \langle j|b \rangle \{i^\dagger j^\dagger ba\} \quad (4.78)$$

$$= \frac{1}{2} \sum_{i,j,a,b} \delta_{i,a} \delta_{j,b} \{i^\dagger j^\dagger ba\} \quad (4.79)$$

$$= 0. \quad (4.80)$$

Thus, the action of multiplying by a factor of the inner-electronic distance converts the two-electron integral into a product of two overlap integrals. In going from Eq. (4.78) to Eq. (4.79), that MC-MP methods employ canonical HF orbitals has been exploited to evaluate the overlap integrals into Kronecker deltas. Similarly, when going from Eq. (4.79) to Eq. (4.80), that i, j and a, b are in different sets of orbitals has been used to evaluate the Kronecker deltas to be zero.

Using the logic of the preceding derivation, it can be shown that any half-vertex with its pair of propagator lines in a V-like orientation will produce a Kronecker delta between an occupied and a virtual orbital after its integrand is multiplied by a factor of the inter-electronic distance. Therefore, the integral of any vertex with at least one half-vertex in such an orientation is zero when its integrand is multiplied by an additional factor of its associated inner-electronic distance. The first three vertices in the middle column of Figure 4.5 are the only orientations that do not have any pairs of propagator lines in such a configuration, and are not evaluated to zero.

With the action of modifying any orientation of an interaction vertex established, a scheme to determine a subset of the acceptable combinations of the inner-electronic distances to multiply the integrands of the energy corrections at any order to produce control variates whose integral is zero is proposed. The general topology of a Goldstone diagram contributing to the energy is shown in Figure 4.6. The top and bottom vertices, i.e., the ‘external’ vertices, of Goldstone diagrams for the energy at any order always have two half-vertices in a V-like configuration. Therefore, if an integrand contributing to the n^{th} -order energy correction is multiplied by a factor of either r_{12} , $r_{2n-1,2n}$, or both, its integral will become zero. Furthermore, as

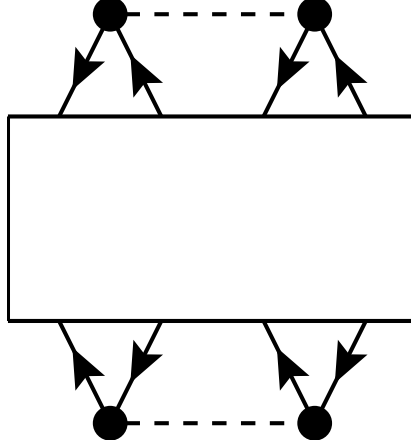


Figure 4.6: The general topology of a MBPT diagram.

long as one of these three factors multiplies the integrand, any combination of the inner-electronic distances associated with the ‘internal’ vertices may also multiply the integrand to create additional control variates. The expansion of

$$(r_{12} + r_{2n-1,2n} + r_{12}r_{2n-1,2n}) \prod_{i=2}^{n-1} (1 + r_{2i-1,2i}) \quad (4.81)$$

is used to enumerate $3 \cdot 2^{n-2}$ factors in which the integrand of the n^{th} correlation energy may be multiplied to create acceptable control variates whose exact integral is zero.

Further control variates are generated by partitioning the integrands for the energy corrections based on the implicit sum over diagrams. The integrands for $E^{(2)}$, $E^{(3)}$, and $E^{(4)}$ are constructed from 2, 12, and 300 diagrams respectively.

For MC-MP2, the sum over diagrams is fully partitioned to produce 6 control variate functions. The integrands for MC-MP3 and MC-MP4 are not fully partitioned on the diagrammatic basis. If they were, 72 and 3600 control variates would be produced. For MC-MP4, this would be particularly problematic as an excessively large covariance matrix would be needed which would hinder the overall calculation speed. Instead, the integrands for MC-MP3 and MC-MP4 are partitioned each into six groups. For MC-MP3, the groups are the consecutive pairs of diagrams in Figure 2.6, i.e., $\{(1, 2), (3, 4), \dots\}$. For MC-MP4, the sum is partitioned based on the diagrammatic topologies introduced in Figure 2.9, with the type D diagrams being broken down into 3 groups. The three groups of type D diagrams are given by Figures J.4, J.5, and J.6. This scheme produces 36 and 72 control variates for MC-MP3 and MC-MP4 respectively.

Table 4.7: The second- through fourth-order correlation energies and their uncertainties in E_h for MC-MP2 through MC-MP4 calculations with and without using control variates.

Order	Molecule	E	$\widehat{E}_{MC}$	σ_{MC}	$\widehat{E}_{CV}$	σ_{CV}	$\sigma_{MC}^2/\sigma_{CV}^2$
2	Ammonia	-0.186635	-0.186514	0.000395	-0.186555	0.000145	7.43
2	Methane	-0.160987	-0.160555	0.000447	-0.160856	0.000140	10.27
2	Water	-0.201627	-0.201510	0.000320	-0.201690	0.000142	5.05
2	Benzene	-0.783937	-0.779126	0.007273	-0.781556	0.001931	14.19
3	Ammonia	-0.013670	-0.014368	0.000888	-0.013816	0.000200	19.71
3	Methane	-0.020237	-0.020320	0.001112	-0.020600	0.000253	19.23
3	Water	-0.007002	-0.006715	0.000540	-0.007035	0.000128	17.78
3	Benzene	-0.033962	0.041644	0.051832	-0.022028	0.011196	21.43
4	Ammonia	-0.005751	-0.007709	0.002278	-0.006621	0.000397	32.92
4	Methane	-0.005900	-0.002779	0.003074	-0.006065	0.000542	32.17
4	Water	-0.005230	-0.006348	0.001243	-0.005531	0.000225	30.59
4	Benzene	-0.042631	-0.129855	0.775777	-0.154275	0.067739	131.16

4.4.3 Performance Analysis

The efficacy of employing the proposed control variate functions to accelerate MC-MP calculations is probed by applying MC-MP2 through MC-MP4 to several small molecules, water ($r_{OH} = 0.9573\text{\AA}$), methane ($r_{CH} = 1.0848\text{\AA}$), ammonia ($r_{NH} = 1.0123\text{\AA}$), and benzene ($r_{CC} = 1.3970\text{\AA}$, $r_{CH} = 1.0869\text{\AA}$). All calculations use the direct-sampling algorithm discussed in Section 4.3, and integrate the imaginary-time coordinates using MC integration. The calculations were performed using the cc-pVDZ basis set and 64 redundant walkers. All uncertainties and energies for each species were obtained from a single MC-MP4 calculation averaging 3481600 MC steps. Calculations used to establish the impact on the time-per-MC-step were performed on a single Blue Waters XE node using all of its processors.

Table 4.7 examines the reduction in the uncertainty as a result of employing the proposed control variates. It is immediately apparent that the uncertainties of the control variate estimates of the energy are lower than the uncontrolled estimates in all cases. Similarly, the control variate estimates of the energy are closer to the exact energy in all cases except one, the third-order correlation energy for methane. However, the single deviation is simply due to the nature of MC integration, in that the MC estimate of a quantity may be, by happenchance, close to the exact value.

The last column of Table 4.7 gives the square of the ratio of the uncertainties. As such, it corresponds to the factor that the number of MC steps would need to be increased by for the uncontrolled estimates to reach the same uncertainty as the control variate estimates. For the examined molecules, the control variate estimates require ~ 5 to ~ 15 fewer MC steps at the MC-MP2 level of theory. As the level of theory increases, the number of steps required by the controlled estimates further decreases. At the MC-MP4 level of theory, the control estimates require, on average, ~ 30 times fewer MC steps for the ten-electron molecules and an

Table 4.8: The time-per-step in seconds for MC-MP2 through MC-MP4 calculations with and without calculating control variates.

Order	Molecule	t_{MC}	t_{CV}	$t_{\text{CV}}/t_{\text{MC}}$
2	Ammonia	0.07670(6)	0.07943(8)	1.035(1)
2	Methane	0.08783(7)	0.09092(5)	1.035(1)
2	Water	0.06624(5)	0.06883(6)	1.039(1)
2	Benzene	0.2999(1)	0.30618(4)	1.0209(5)
3	Ammonia	0.4229(4)	0.6113(7)	1.445(2)
3	Methane	0.4398(7)	0.6185(4)	1.406(2)
3	Water	0.4013(4)	0.5972(6)	1.488(2)
3	Benzene	0.6792(4)	0.8510(4)	1.2530(10)
4	Ammonia	47.190(1)	106.409(6)	2.2549(1)
4	Methane	46.762(1)	106.633(6)	2.2803(1)
4	Water	47.149(1)	106.37(2)	2.2559(3)
4	Benzene	47.009(2)	105.78(3)	2.2501(7)

astonishing ~ 130 steps for benzene.

The additional time required to compute the control variates is now examined. The third and fourth columns of Table 4.8 present the time-per-step for simulations that do and do not calculate the additional control variate functions. The final column gives the ratio of the two preceding columns. From the last column of Table 4.8, it is observed that as the level of theory is increased, the computation of the control variates has a more significant impact on the simulation time. The increasing impact is due to the step in the MC-MP algorithm in which the control variates are computed, the computation of the energy correction.

At the MC-MP2 level of theory, the computation of the energy correction is far from the algorithmic hotspot. As a consequence, the additional work required to compute the control variate functions causes nearly no impact on the time-per-step, and the ratio of the simulation times is near unity.

At the MC-MP3 level of theory, the algorithmic hotspot given a sufficient number of redundant walkers becomes the computation of the energy correction, and in particular, the calculation of the integrand of $E^{(3)}$. Recall from Section 2.5, that in the MC-MP3 algorithm, the entirety of the MC-MP2 algorithm is performed and then the integrand of $E^{(3)}$ is calculated. On this basis, the increase in the time-per-step in going from the standard MC-MP2 algorithm to the standard MC-MP3 algorithm is attributed to the calculation of $E^{(3)}$. Further, the increase is significant enough that nearly the entire time-per-step in an MC-MP3 simulation is spent calculating $E^{(3)}$. Since the calculation of the control variates increases the work performed in this costly step, the impact of calculating them naturally causes a greater increase in the simulation time for MC-MP3. The same analysis holds when transitioning from MC-MP3 to MC-MP4.

Table 4.9 combines the results of Table 4.7 and Table 4.8 to present the speedup achieved by using control variates to estimate the correlation energy. The acceleration provided by employing the proposed

Table 4.9: The improvement in the MC efficiency for MC-MP2 through MC-MP4 calculations provide by utilizing the control variates estimates of the correlation energies as opposed to the standard estimates.

Molecule	MC-MP2	MC-MP3	MC-MP4
Ammonia	7.18	13.63	14.60
Methane	9.92	13.67	14.11
Water	4.86	11.95	13.56
Benzene	13.90	17.11	58.29

control variates for MC-MP2 ranges from 4.88 to 13.9. The degree of acceleration becomes more significant as the level of theory is increased due to the increased variance reduction. The forth-order energy correction of benzene is massively accelerated by a factor of 58.29. The increased variance reduction is, however, partially dampened by increasing simulation times. This result is advantageous when considering the greater computational demands that the higher-order methods demand.

4.4.4 Conclusions

The acceleration of MC-MP methods by the use of control variates has been presented and evaluated. The proposed control variate functions originate from the observation that by slightly modifying the MC-MP integrands by multiplying by appropriate products of the inner-electronic distances produce integrals that are analytically evaluable to zero. Many appropriate products of the inner-electronic distances are enumerated by a generating function establish by considering the general topology of the Goldstone diagrams which define the order-by-order contributions to the correlation energy. Further control variates are created by partitioning the integrands of the correlation energies on a diagrammatic basis. The combination of the two approaches is used to create 6, 36, and 48 control variate functions for MC-MP2, MC-MP3, and MC-MP4, respectively.

The proposed control variates are remarkably successful in accelerating MC-MP calculations. The success is due to the control variates being minimally modified versions of the MC-MP integrands. Thus, they are strongly correlated with the MC-MP integrands and have a low impact on the simulation time. As a consequence, when applied to small molecules, speedups of up to 13.9, 17.11, and 58.29 are observed for MC-MP2, MC-MP3, and MC-MP4 respectively. The speedup for MC-MP4 is especially relevant as it helps to alleviate the difficulties encountered during the establishment of the method and brings the estimates of $E^{(4)}$ to be firmly in line with the exact values.

Control variate functions for only the MC-MP family of methods have been proposed; however, the creation of similar control variate functions for other MC-MB methods is feasible. The techniques used to create the MC-MP control variate functions are readily extensible to MC-GF methods as the action of

modifying a single interaction vertex will remain unchanged. The generating function to produce acceptable products of inner-electronic distances may need to be altered as the diagrams defining the self-energy have a different general topology. The creation of similar control variate functions for the MC explicitly correlated family of methods will be more challenging. The primary reason is that the integrals in the explicitly correlated methods are not the standard two-electron integrals as found in MBPT and MBGF methods, but are significantly more complicated and are two-, three-, and four-electron integrals over a multitude of operators. Converting the integrals in MC explicitly correlated methods into overlap integrals by multiplying the integrands by the inverse of the relevant operators may create useful control variate functions. The exploration of the control variates for these methods is likely to yield similar performance benefits as those observed here, and are therefore critical to extending the range of applicability of these methods.

4.5 Conclusions

In this chapter, four methods to improve the rate of convergence of MC-MB methods have been examined. Each technique provides, at a minimum, a modest improvement to the MC-MB family of methods. However, as the techniques are essentially independent of each other, their combination produces a significant speedup. By applying all of the techniques presented in this chapter, a third-order MC-MB method applied to methane could achieve a speedup of as much as 22800 [5.5 from GPU-acceleration, 226 from the stochastic integration of imaginary time coordinates, 1.34 from direct sampling of the electron coordinates, 13.7 from control variates] when compared to the base implementation. These improvements are instrumental to MC-MB methods in two regards. The first is to reduce the prefactor on cost functions of existing MC-MB methods and thus extend their range of useful applicability. The second is to reduce the prefactor on the cost functions yet again, but with regard to difficulties encountered when exploring higher-order MC-MB methods. For both of these reasons, the production of further novel acceleration techniques should be maintained as one of the core research areas for MC-MB methods.

Chapter 5

Conclusions

The initial goal of pursuing MC implementations of many-body methods was to circumvent the difficulties encountered when preparing algorithms to perform these methods by conventional means. These difficulties include the high operational cost to perform calculations with many-body methods, the requirements for large arrays which necessitates the use of distributed memory, and the complexity of the expressions that define the methods themselves. These difficulties make preparing efficient parallel algorithms to perform many-body methods a challenging endeavor, and thus, limit the range of chemistry that may be explored by many-body methods.

The MC-MB methods have been successful in circumventing these difficulties. All of these methods achieve high parallel efficiency, have low operational costs to perform a single MC step, and have trivial memory requirements. There are currently seven MC-MB methods that have been developed, five of which were presented here. Stochastic MBPT methods include MC-MP2, MC-MP3, MC-MP4, and MC-MP2-F12, while stochastic MBGF methods include MC-GF2, MC-GF3, and MC-GF2-F12. The MC-GF2 and MC-GF3 methods are capable of computing electron binding energies within the frequency-independent diagonal, frequency-independent, or diagonal approximation or without approximation. The second- and third-order methods capture much of the relevant chemistry and thus enable qualitative predictions, while the implementation of fourth-order methods opens the path to perform benchmark quality calculations with the MC-MB family of methods. When available, these methods present favorable cost scaling functions with system size compared to their conventional counterparts, which establishes the long term viability of these methods.

These methods encounter a significant prefactor on their cost functions. However, this challenge is readily combated by employing a range of convergence acceleration techniques. Five acceleration techniques to the MC-MB family of methods were presented here. These include the redundant-walker algorithm, strategies to efficiently perform the integration of the imaginary-time coordinate, an algorithm to directly sample the electronic importance function, the proposal of control variates for the integration of the electronic coordinates, and the application of GPU-accelerators. While some interplay exists between the convergence accelera-

tion techniques, they are essentially decoupled, and thus, the speedup obtain is multiplicative, granting a significant increase in performance.

Toward the future, given their wide range of methods and their computational characteristics, particularly the high parallel efficiency, the MC-MB family of methods will allow the exploration of novel chemistry on current and future supercomputing resources. Additionally, given the relative ease of implementation, the range of methods available will likely continue to grow as well as the techniques used to accelerate the MC-MB family of methods, both of which serve to extend the range of chemistry that may be explored.

Chapter 6

Bibliography

- (1) Szabo, A; Ostlund, N. S., *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*; Dover: 1989.
- (2) Shavitt, I.; Bartlett, R., *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*; Cambridge University Press: 2009.
- (3) March, N.; Young, W.; Sampanthar, S., *The Many-Body Problem in Quantum Mechanics*; Dover: 1995.
- (4) Mattuck, R. D., *A Guide to Feynman Diagrams in the Many-Body Problem*; Dover: 2012.
- (5) Fetter, A.; Walecka, J., *Quantum Theory of Many-Particle Systems*; Dover: 2012.
- (6) Linderberg, J; Öhrn, Y, *Propagators in Quantum Chemistry*; Wiley: 2004.
- (7) Bartlett, R. J.; Musiał, M. *Rev. Mod. Phys.* **2007**, *79*, 291–352.
- (8) Bartlett, R. J. *Ann. Rev. Phys. Chem.* **1981**, *32*, 359–401.
- (9) Corzo, H. H.; Ortiz, J. *Adv. Quantum Chem.* **2017**, *74*, 267.
- (10) Doran, A. E.; Hirata, S. *J. Chem. Theory Comput.* **2016**, *12*, 4821.
- (11) Johnson, C. M.; Doran, A. E.; Zhang, J.; Valeev, E. F.; Hirata, S. *J. Chem. Phys.* **2016**, *145*, 154115.
- (12) Johnson, C. M.; Doran, A. E.; Ten-no, S. L.; Hirata, S. *J. Chem. Phys.* **2018**, *149*, 174112.
- (13) Doran, A. E.; Hirata, S. *J. Chem. Theory Comput.* **2019**.
- (14) Willow, S. Y.; Kim, K. S.; Hirata, S. *J. Chem. Phys.* **2012**, *137*, 204122.
- (15) Willow, S. Y.; Hirata, S. *J. Chem. Phys.* **2014**, *140*, 024111.
- (16) Willow, S. Y.; Kim, K. S.; Hirata, S. *J. Chem. Phys.* **2013**, *138*, 164111.
- (17) Willow, S. Y.; Kim, K. S.; Hirata, S. *Phys. Rev. B* **2014**, *90*, 201110.
- (18) Hermes, M. R.; Hirata, S. *J. Chem. Phys.* **2014**, *141*, 084105.
- (19) Hermes, M. R.; Hirata, S. *J. Chem. Phys.* **2015**, *143*, 129903.

- (20) Neuhauser, D.; Rabani, E.; Baer, R. *J. Chem. Theory Comput.* **2013**, *9*, 24–27.
- (21) Neuhauser, D.; Rabani, E.; Baer, R. *J. Phys. Chem. Lett.* **2013**, *4*, 1172–1176.
- (22) Ge, Q.; Gao, Y.; Baer, R.; Rabani, E.; Neuhauser, D. *J. Phys. Chem. Lett.* **2014**, *5*, 185–189.
- (23) Cytter, Y.; Neuhauser, D.; Baer, R. *J. Chem. Theory Comput.* **2014**, *10*, 4317.
- (24) Neuhauser, D.; Gao, Y.; Arntsen, C.; Karshenas, C.; Rabani, E.; Baer, R. *Phys. Rev. Lett.* **2014**, *113*, 076402.
- (25) Rabani, E.; Baer, R.; Neuhauser, D. *Phys. Rev. B* **2015**, *91*, 235302.
- (26) Takeshita, T. Y.; de Jong, W. A.; Neuhauser, D.; Baer, R.; Rabani, E. *J. Chem. Theory Comput.* **2017**, *13*, 4605–4610.
- (27) Neuhauser, D.; Baer, R.; Zgid, D. *J. Chem. Theory Comput.* **2017**, *13*, 5396–5403.
- (28) Drabold, D. A.; Sankey, O. F. *Phys. Rev. Lett.* **1993**, *70*, 3631–3634.
- (29) Sankey, O. F.; Drabold, D. A.; Gibson, A. *Phys. Rev. B* **1994**, *50*, 1376–1381.
- (30) Iitaka, T.; Nomura, S.; Hirayama, H.; Zhao, X.; Aoyagi, Y.; Sugano, T. *Phys. Rev. E* **1997**, *56*, 1222–1229.
- (31) Baer, R.; Seideman, T.; Ilani, S.; Neuhauser, D. *J. Chem. Phys.* **2004**, *120*, 3387–3396.
- (32) Baer, R.; Rabani, E. *Nano Lett.* **2012**, *12*, 2123–2128.
- (33) Hohenstein, E. G.; Parrish, R. M.; Martínez, T. J. *J. Chem. Phys.* **2012**, *137*, 044103.
- (34) Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. *J. Chem. Phys.* **2012**, *137*, 224106.
- (35) Hohenstein, E. G.; Parrish, R. M.; Sherrill, C. D.; Martínez, T. J. *J. Chem. Phys.* **2012**, *137*, 221101.
- (36) Parrish, R. M.; Hohenstein, E. G.; Schunck, N. F.; Sherrill, C. D.; Martínez, T. J. *Phys. Rev. Lett.* **2013**, *111*, 132505.
- (37) Hohenstein, E. G.; Kokkila, S. I. L.; Parrish, R. M.; Martínez, T. J. *J. Phys. Chem. B* **2013**, *117*, 12972–12978.
- (38) Hohenstein, E. G.; Kokkila, S. I. L.; Parrish, R. M.; Martínez, T. J. *J. Chem. Phys.* **2013**, *138*, 124111.
- (39) Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J.; Sherrill, C. D. *J. Chem. Phys.* **2013**, *138*, 194107.
- (40) Parrish, R. M.; Sherrill, C. D.; Hohenstein, E. G.; Kokkila, S. I. L.; Martínez, T. J. *J. Chem. Phys.* **2014**, *140*, 181102.

- (41) Kokkila Schumacher, S. I. L.; Hohenstein, E. G.; Parrish, R. M.; Wang, L.-P.; Martínez, T. J. *J. Chem. Theory Comput.* **2015**, *11*, 3042–3052.
- (42) Song, C.; Martínez, T. J. *J. Chem. Phys.* **2016**, *144*, 174111.
- (43) Parrish, R. M.; Hohenstein, E. G.; Martínez, T. J. *J. Chem. Phys.* **2016**, *145*, 027101.
- (44) Song, C.; Martínez, T. J. *J. Chem. Phys.* **2017**, *146*, 034104.
- (45) Song, C.; Martínez, T. J. *J. Chem. Phys.* **2017**, *147*, 161723.
- (46) Kvasnička, V.; Laurinc, V.; Biskupič, S. *Mol. Phys.* **1980**, *39*, 143–161.
- (47) Slater, J. C. *Phys. Rev.* **1929**, *34*, 1293–1322.
- (48) Condon, E. U. *Phys. Rev.* **1930**, *36*, 1121–1133.
- (49) Feynman, R. P. *Phys. Rev.* **1949**, *76*, 769–789.
- (50) Goldstone, J. *P. Roy. Soc. A-Math Phys.* **1957**, *239*, 267–279.
- (51) Manne, R. *Int. J. Quantum Chem.* **1977**, *11*, 175–192.
- (52) Simons, J.; Smith, W. D. *J. Chem. Phys.* **1973**, *58*, 4899–4907.
- (53) Purvis, G.; Öhrn, Y. *Chem. Phys. Lett.* **1975**, *33*, 396.
- (54) Jørgensen, P.; Simons, J. *J. Chem. Phys.* **1975**, *63*, 5302.
- (55) Von Niessen, W.; Schirmer, J.; Cederbaum, L. *Comput. Phys. Rep.* **1984**, *1*, 57.
- (56) Schirmer, J.; Cederbaum, L. S.; Walter, O. *Phys. Rev. A* **1983**, *28*, 1237–1259.
- (57) Ortiz, J. V. *J. Chem. Phys.* **1998**, *108*, 1008–1014.
- (58) Ortiz, J. V. *WIREs Comput. Mol. Sci.* **2013**, *3*, 123.
- (59) Hirata, S.; Doran, A. E.; Knowles, P. J.; Ortiz, J. V. *J. Chem. Phys.* **2017**, *147*, 044108.
- (60) Kalos, M.; Whitlock, P., *Monte Carlo Methods*; Wiley: 2008.
- (61) Owen, A. B., *Monte Carlo Theory, Methods and Examples*, 2013.
- (62) Kroese, D.; Taimre, T.; Botev, Z., *Handbook of Monte Carlo Methods*; Wiley: 2013.
- (63) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087.
- (64) Hastings, W. K. *Biometrika* **1970**, *57*, 97.
- (65) Geman, S.; Geman, D. *IEEE T. Pattern Anal.* **1984**, *PAMI-6*, 721–741.
- (66) Duane, S.; Kennedy, A.; Pendleton, B. J.; Roweth, D. *Phys. Lett. B* **1987**, *195*, 216 –222.

- (67) Matsumoto, M.; Nishimura, T. *ACM Trans. Model. Comput. Simul.* **1998**, *8*, 3–30.
- (68) Box, G. E. P.; Muller, M. E. *Ann. Math. Statist.* **1958**, *29*, 610–611.
- (69) Flyvbjerg, H.; Petersen, H. G. *J. Chem. Phys.* **1989**, *91*, 461.
- (70) Almlöf, J. *Chem. Phys. Lett.* **1991**, *181*, 319.
- (71) Kronrod, A. S., *Nodes and Weights of Quadrature Formulas: Sixteen-Place Tables*; Consultants Bureau: 1965.
- (72) Obara, S.; Saika, A. *J. Chem. Phys.* **1986**, *84*, 3963–3974.
- (73) Willow, S. Y.; Hermes, M. R.; Kim, K. S.; Hirata, S. *J. Chem. Theory Comput.* **2013**, *9*, 4396.
- (74) Head-Gordon, M.; Pople, J. A.; Frisch, M. J. *Chem. Phys. Lett.* **1988**, *153*, 503.
- (75) Doran, A. E.; Hirata, S. Unpublished.
- (76) Barlett, R. J.; Sekino, H.; Purvis, G. D. *Chem. Phys. Lett.* **1983**, *98*, 66–71.
- (77) Cole, S. J.; Bartlett, R. J. *J. Chem. Phys.* **1987**, *86*, 873–881.
- (78) Laidig, W. D.; Fitzgerald, G.; Bartlett, R. J. *Chem. Phys. Lett.* **1985**, *113*, 151–158.
- (79) Knowles, P.; Somasundram, K.; Handy, N.; Hirao, K. *Chem. Phys. Lett.* **1985**, *113*, 8–12.
- (80) Cremer, D.; He, Z. *J. Phys. Chem.* **1996**, 6173–6188.
- (81) Kucharski, S. A.; Noga, J.; Bartlett, R. J. *J. Chem. Phys.* **1989**, *90*, 7282.
- (82) Kucharski, S. A.; Bartlett, R. J. *Chem. Phys. Lett.* **1995**, *237*, 264–272.
- (83) He, Z.; Cremer, D. *Int. J. Quantum Chem.* **1996**, *59*, 15–29.
- (84) He, Z.; Cremer, D. *Int. J. Quantum Chem.* **1996**, *59*, 31–55.
- (85) He, Z.; Cremer, D. *Int. J. Quantum Chem.* **1996**, *59*, 57–69.
- (86) He, Z.; Cremer, D. *Int. J. Quantum Chem.* **1996**, *59*, 71–95.
- (87) Redmon, L. T.; Purvis, G.; Öhrn, Y. *J. Chem. Phys.* **1975**, *63*, 5011.
- (88) Öhrn, Y.; Born, G. *Adv. Quantum Chem.* **1981**, *13*, 1.
- (89) Toulouse, J.; Assaraf, R.; Umrigar, C. J. *Adv. Quantum Chem.* **2016**, *73*, 285.
- (90) Cociorva, D.; Baumgartner, G.; Lam, C.; Sadayappan, P.; Ramanujam, J.; Nooijen, M.; Bernholdt, D.; Harrison, R. *ACM Sigplan Notices* **2002**, *37*, 177.
- (91) Almlöf, J.; Faegri, K.; Korsell, K. *J. Comp. Chem.* **1982**, *3*, 385.
- (92) Dunning, T. H. *J. Chem. Phys.* **1989**, *90*, 1007.

- (93) Dolgounitcheva, O.; Zakrzewski, V. G.; Ortiz, J. V. *J. Phys. Chem. A* **2000**, *104*, 10032.
- (94) Deleuze, M. S. *J. Chem. Phys.* **2002**, *116*, 7012.
- (95) Huang, D. L.; Dau, P. D.; Liu, H. T.; Wang, L. S. *J. Chem. Phys.* **2014**, *140*, 224315.
- (96) Lichtenberger, D. L.; Rempe, M. E.; Gogosha, S. B. *Chem. Phys. Lett.* **1992**, *198*, 454.
- (97) Lichtenberger, D. L.; Jatcko, M. E.; Nebesny, K. W.; Ray, C. D.; Huffman, D. R.; Lamb, L. D. *Mat. Res. Soc. Symp. Proc.* **1990**, *206*, 673.
- (98) Vogt, L.; Olivares-Amaya, R.; Kermes, S.; Shao, Y.; Amador-Bedolla, C.; Aspuru-Guzik, A. *J. Chem. Phys. A* **2008**, *112*, 2049–2057.
- (99) Yasuda, K. *J. Comp. Chem.* **2008**, *29*, 334–342.
- (100) Ufimtsev, I. S.; Martínez, T. J. *Comput. Sci. Eng.* **2008**, *10*, 26–34.
- (101) Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2008**, *4*, 222–231.
- (102) Ufimtsev, I. S.; Martinez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 1004–1015.
- (103) Ufimtsev, I. S.; Martinez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 2619–2628.
- (104) Olivares-Amaya, R.; Watson, M. A.; Edgar, R. G.; Vogt, L.; Shao, Y.; Aspuru-Guzik, A. *J. Chem. Theory Comput.* **2010**, *6*, 135–144.
- (105) DePrince, A. E.; Hammond, J. R. *J. Chem. Theory Comput.* **2011**, *7*, 1287–1295.
- (106) Ma, W.; Krishnamoorthy, S.; Villa, O.; Kowalski, K. *J. Chem. Theory Comput.* **2011**, *7*, 1316–1327.
- (107) Asadchev, A.; Gordon, M. S. *J. Chem. Theory Comput.* **2012**, *8*, 4166–4176.
- (108) Tomlinson, D. G.; Asadchev, A.; Gordon, M. S. *J. Comp. Chem.* **2016**, *37*, 1274–1282.
- (109) Häser, M.; Almlöf, J. *J. Chem. Phys.* **1992**, *96*, 489–494.
- (110) Takatsuka, A.; Ten-no, S.; Hackbusch, W. *J. Chem. Phys.* **2008**, *129*, 044112.
- (111) Kats, D.; Usvyat, D.; Loibl, S.; Merz, T.; Schütz, M. *J. Chem. Phys.* **2009**, *130*, 127101.
- (112) Helmich-Paris, B.; Visscher, L. *J. Comp. Phys.* **2016**, *321*, 927–931.
- (113) Willow, S. Y.; Zhang, J.; Valeev, E. F.; Hirata, S. *J. Chem. Phys.* **2014**, *140*, 031101.
- (114) IRWIN, J. O. *Biometrika* **1927**, *19*, 225–239.
- (115) PHILIP, H. *Biometrika* **1927**, *19*, 240–244.
- (116) Scavo, T. R.; Thoo, J. B. *Am. Math. Mon.* **1995**, *102*, 417–426.

Appendix A

Calculations for Imaginary-time Integration

Table A.1: The number of steps, variances, and simulation times of the second-order self-energy calculations used to compare the efficacy of the direct-MC integration of the imaginary time coordinate compared to the use of a quadrature rule. Variances are given in E_h^2 .

τ -Integration	Molecule	Formula	Electrons	Steps	Time (s)	σ^2 ^a
MC	Methane	CH ₄	10	1190528.00	1785	0.13
MC	Propane	C ₃ H ₈	26	1125632.00	1783	1.84
MC	Benzene	C ₆ H ₆	42	1090944.00	1786	2.12
MC	<i>n</i> -Nonane	C ₉ H ₂₀	74	510848.00	1749	11.40
MC	<i>n</i> -Decane	C ₁₀ H ₂₂	82	501248.00	1750	17.53
MC	Pentacene	C ₂₂ H ₁₄	146	476288.00	1748	51.83
Quadrature	Methane	CH ₄	10	125408.00	1814	0.06
Quadrature	Propane	C ₃ H ₈	26	56688.00	1807	0.77
Quadrature	Benzene	C ₆ H ₆	42	55472.00	1808	0.84
Quadrature	<i>n</i> -Nonane	C ₉ H ₂₀	74	85648.00	1788	4.42
Quadrature	<i>n</i> -Decane	C ₁₀ H ₂₂	82	84208.00	1788	7.05
Quadrature	Pentacene	C ₂₂ H ₁₄	146	72336.00	1780	15.98

^aVariance of the diagonal element of the second-order self-energy matrix associated with the HOMO.

Table A.2: The same as Table A.1 but for the third-order self-energy.

τ -Integration	Molecule	Formula	Electrons	Steps	Time (s)	σ^2 ^a
MC	Methane	CH ₄	10	19393536	$1.393 \cdot 10^5$	$8.634 \cdot 10^{-1}$
MC	Propane	C ₃ H ₈	26	1577984	$1.243 \cdot 10^4$	$5.536 \cdot 10^1$
MC	Benzene	C ₆ H ₆	42	17617920	$1.400 \cdot 10^5$	$2.773 \cdot 10^2$
MC	Nonane	C ₉ H ₂₀	74	1571840	$1.239 \cdot 10^4$	$5.189 \cdot 10^3$
MC	n-Decane	C ₁₀ H ₂₂	82	1555456	$1.237 \cdot 10^4$	$6.280 \cdot 10^3$
MC	Pentacene	C ₂₂ H ₁₄	146	17075200	$1.400 \cdot 10^5$	$8.417 \cdot 10^4$
Quadrature	Methane	CH ₄	10	40960	$1.295 \cdot 10^5$	$4.442 \cdot 10^{-1}$
Quadrature	Propane	C ₃ H ₈	26	35840	$1.175 \cdot 10^5$	$1.788 \cdot 10^1$
Quadrature	Benzene	C ₆ H ₆	42	38912	$1.293 \cdot 10^5$	$8.336 \cdot 10^1$
Quadrature	Nonane	C ₉ H ₂₀	74	56320	$1.804 \cdot 10^5$	$1.647 \cdot 10^3$
Quadrature	n-Decane	C ₁₀ H ₂₂	82	71680	$2.351 \cdot 10^5$	$3.176 \cdot 10^3$
Quadrature	Pentacene	C ₂₂ H ₁₄	146	56320	$1.879 \cdot 10^5$	$1.173 \cdot 10^4$

^aVariance of the diagonal element of the third-order self-energy matrix associated with the HOMO.

Appendix B

Calculations for System-size Dependence of Cost of MC-MP2

Table B.1: The results of the calculations used to establish the scaling of the variance of the second-order correlation-energy with system size for MC-MP2. Uncertainties are given in E_h . All calculations are 16384 MC steps long, were executed on a single Blue Water XK node, and used 512 redundant walkers.

Molecule	Formula	n_{ele}	cc-pVDZ			aug-cc-pVDZ		
			n_{bas}	σ	Time (s)	n_{bas}	σ	Time (s)
Water	H ₂ O	10	24	$1.597 \cdot 10^{-3}$	838	41	$3.609 \cdot 10^{-3}$	1118
Ethane	C ₂ H ₆	18	58	$4.028 \cdot 10^{-3}$	1413	100	$1.171 \cdot 10^{-2}$	2118
Formic acid	CH ₂ O ₂	24	52	$5.204 \cdot 10^{-3}$	1283	87	$1.187 \cdot 10^{-2}$	1912
Propane	C ₃ H ₈	26	82	$8.020 \cdot 10^{-3}$	1819	141	$2.159 \cdot 10^{-2}$	2843
Acetone	C ₃ H ₆ O	32	86	$9.752 \cdot 10^{-3}$	1853	146	$1.079 \cdot 10^{-2}$	2947
Isopropyl alcohol	C ₃ H ₈ O	34	96	$1.184 \cdot 10^{-2}$	2003	164	$3.183 \cdot 10^{-2}$	3203
Butane	C ₄ H ₁₀	34	106	$1.347 \cdot 10^{-2}$	2219	182	$5.040 \cdot 10^{-2}$	3551
Glycine	C ₂ H ₅ NO ₂	40	95	$1.245 \cdot 10^{-2}$	1979	160	$4.468 \cdot 10^{-2}$	3086
Benzene	C ₆ H ₆	42	114	$1.541 \cdot 10^{-2}$	2276	192	$4.773 \cdot 10^{-2}$	3641
Alanine	C ₃ H ₇ NO ₂	48	119	$1.889 \cdot 10^{-2}$	2386	201	$2.155 \cdot 10^{-2}$	3936
Serine	C ₃ H ₇ NO ₃	56	133	$2.893 \cdot 10^{-2}$	2635	224	$8.040 \cdot 10^{-2}$	4475
Proline	C ₅ H ₉ NO ₂	62	157	$2.899 \cdot 10^{-2}$	2983	265	$7.855 \cdot 10^{-2}$	5005
Threonine	C ₄ H ₉ NO ₃	64	157	$3.325 \cdot 10^{-2}$	2988	265	$7.751 \cdot 10^{-2}$	5038
Valine	C ₅ H ₁₁ NO ₂	64	167	$3.801 \cdot 10^{-2}$	3156	283	$1.138 \cdot 10^{-1}$	5284
Asparagine	C ₄ H ₈ N ₂ O ₃	70	166	$3.366 \cdot 10^{-2}$	3186	279	$4.096 \cdot 10^{-2}$	5253
Salicylic acid	C ₇ H ₆ O ₃	72	170	$3.289 \cdot 10^{-2}$	3218	284	$9.611 \cdot 10^{-2}$	5272
Isoleucine	C ₆ H ₁₃ NO ₂	72	191	$4.080 \cdot 10^{-2}$	3617	324	$1.751 \cdot 10^{-1}$	6018
Leucine	C ₆ H ₁₃ NO ₂	72	191	$3.992 \cdot 10^{-2}$	3617	324	$1.549 \cdot 10^{-1}$	6019
Glutamic acid	C ₅ H ₉ NO ₄	78	185	$4.609 \cdot 10^{-2}$	3486	311	$1.364 \cdot 10^{-1}$	5736
Glutamine	C ₅ H ₁₀ N ₂ O ₃	78	190	$4.309 \cdot 10^{-2}$	3539	320	$1.360 \cdot 10^{-1}$	5891
Histidine	C ₆ H ₉ N ₃ O ₂	82	199	$4.692 \cdot 10^{-2}$	3680	334	$1.514 \cdot 10^{-1}$	6144
Methamphetamine	C ₁₀ H ₁₅ N	82	229	$5.811 \cdot 10^{-2}$	4289	388	$1.929 \cdot 10^{-1}$	7053
Phenylalanine	C ₉ H ₁₁ NO ₂	88	223	$6.457 \cdot 10^{-2}$	4146	375	$1.762 \cdot 10^{-1}$	6829
Nicotine	C ₁₀ H ₁₄ N ₂	88	238	$6.901 \cdot 10^{-2}$	4430	402	$2.070 \cdot 10^{-1}$	7432
Aspirin	C ₉ H ₈ O ₄	94	222	$5.322 \cdot 10^{-2}$	4125	371	$1.422 \cdot 10^{-1}$	6877
Tyrosine	C ₉ H ₁₁ NO ₃	96	237	$6.462 \cdot 10^{-2}$	4413	398	$1.965 \cdot 10^{-1}$	7310
Adrenaline	C ₉ H ₁₃ NO ₃	98	247	$6.159 \cdot 10^{-2}$	4571	416	$1.176 \cdot 10^{-1}$	7725
Caffeine	C ₈ H ₁₀ N ₄ O ₂	102	246	$7.089 \cdot 10^{-2}$	4550	412	$1.026 \cdot 10^0$	7688
Tryptophan	C ₁₁ H ₁₂ N ₂ O ₂	108	270	$7.690 \cdot 10^{-2}$	4918	453	$2.548 \cdot 10^{-1}$	8307
Ibuprofen	C ₁₃ H ₁₈ O ₂	112	300	$1.134 \cdot 10^{-1}$	5548	507	$2.811 \cdot 10^{-1}$	9209
THC	C ₂₁ H ₃₀ O ₂	172	472	$2.705 \cdot 10^{-1}$	9018	799	$6.089 \cdot 10^{-1}$	14698
Taxol	C ₄₇ H ₅₁ NO ₁₄	452	1123	$1.677 \cdot 10^0$	21041			

Table B.2: The results of the benzene calculations used to establish the scaling of the variance of the second-order correlation-energy with basis size using MC-MP2. Uncertainties are given in E_h .

Basis	n_{bas}	σ
cc-pVDZ	114	$1.541 \cdot 10^{-2}$
aug-cc-pVDZ	192	$4.773 \cdot 10^{-2}$
cc-pVTZ	264	$8.120 \cdot 10^{-2}$
aug-cc-pVTZ	414	$8.878 \cdot 10^{-2}$
cc-pVQZ	510	$1.210 \cdot 10^{-1}$
aug-cc-pVQZ	756	$1.347 \cdot 10^{-1}$

Appendix C

Calculations for System-size Dependence of Cost of MC-MP3

Table C.1: The results of the cc-pVDZ calculations used to establish the scaling of the variance of the third-order correlation-energy with system size for MC-MP3. Variances are given in E_h^2 .

Molecule	Formula	Electrons	Steps	σ^2
Acetylene	C ₂ H ₂	14	33554432	$1.184 \cdot 10^0$
Ethylene	C ₂ H ₄	16	33554432	$7.572 \cdot 10^0$
Ethane	C ₂ H ₆	18	33554432	$1.737 \cdot 10^1$
Propane	C ₃ H ₈	26	33554432	$1.411 \cdot 10^2$
Butadiyne	C ₄ H ₂	26	33554432	$2.400 \cdot 10^1$
Butadiene	C ₄ H ₆	30	33554432	$2.693 \cdot 10^2$
Butane	C ₄ H ₁₀	34	33554432	$7.055 \cdot 10^2$
n-Pentane	C ₅ H ₁₂	42	33554432	$2.395 \cdot 10^3$
Hexatriene	C ₆ H ₈	44	33554432	$2.788 \cdot 10^3$
Hexane	C ₆ H ₁₄	50	33554432	$7.225 \cdot 10^3$
n-Heptane	C ₇ H ₁₆	58	33554432	$1.486 \cdot 10^4$
n-Octane	C ₈ H ₁₈	66	33554432	$3.535 \cdot 10^4$
Napthalene	C ₁₀ H ₈	68	33554432	$2.780 \cdot 10^4$
Nonane	C ₉ H ₂₀	74	33554432	$1.737 \cdot 10^5$
n-Decane	C ₁₀ H ₂₂	82	33554432	$1.070 \cdot 10^5$
Anthracene	C ₁₄ H ₁₀	94	33554432	$1.973 \cdot 10^5$
Naphthacene	C ₁₈ H ₁₂	120	33554432	$1.113 \cdot 10^6$
Pentacene	C ₂₂ H ₁₄	146	33554432	$4.368 \cdot 10^6$

Appendix D

Calculations for System-size Dependence of Cost of MC-GF2

Table D.1: The results of the aug-cc-pVDZ calculations used to establish the scaling of the variance of the second-order self-energy with system size for MC-GF2 within the diagonal, frequency-independent and non-diagonal, frequency-independent approximations. Variances are given in eV^2 .

Molecule	Formula	Electrons	Steps	σ_{pp}^2 ^a	σ^2 ^b
Acetylene	C ₂ H ₂	14	6673408	$1.751 \cdot 10^{-1}$	$2.137 \cdot 10^{-1}$
Ethylene	C ₂ H ₄	16	6650880	$3.034 \cdot 10^{-1}$	$2.553 \cdot 10^{-1}$
Ethane	C ₂ H ₆	18	5740544	$9.744 \cdot 10^{-1}$	$1.450 \cdot 10^0$
Butadiyne	C ₄ H ₂	26	5728256	$5.125 \cdot 10^{-1}$	$6.237 \cdot 10^{-1}$
Propane	C ₃ H ₈	26	5671936	$3.007 \cdot 10^0$	$3.697 \cdot 10^0$
Butadiene	C ₄ H ₆	30	5680128	$1.250 \cdot 10^0$	$1.640 \cdot 10^0$
Butane	C ₄ H ₁₀	34	5598208	$2.973 \cdot 10^0$	$2.841 \cdot 10^0$
n-Pentane	C ₅ H ₁₂	42	5531648	$4.408 \cdot 10^0$	$5.197 \cdot 10^0$
Benzene	C ₆ H ₆	42	5657600	$3.087 \cdot 10^0$	$3.758 \cdot 10^0$
Hexatriene	C ₆ H ₈	44	5564416	$3.083 \cdot 10^0$	$3.024 \cdot 10^0$
Hexane	C ₆ H ₁₄	50	6178816	$1.168 \cdot 10^1$	$1.109 \cdot 10^1$
n-Heptane	C ₇ H ₁₆	58	6102016	$1.068 \cdot 10^1$	$1.071 \cdot 10^1$
n-Octane	C ₈ H ₁₈	66	5965824	$1.760 \cdot 10^1$	$1.761 \cdot 10^1$
Napthalene	C ₁₀ H ₈	68	6147072	$9.952 \cdot 10^0$	$1.475 \cdot 10^1$
Nonane	C ₉ H ₂₀	74	5856256	$1.905 \cdot 10^1$	$1.996 \cdot 10^1$
n-Decane	C ₁₀ H ₂₂	82	5769216	$2.744 \cdot 10^1$	$2.781 \cdot 10^1$
Anthracene	C ₁₄ H ₁₀	94	5892096	$2.233 \cdot 10^1$	$3.710 \cdot 10^1$
Naphthacene	C ₁₈ H ₁₂	120	5569536	$3.914 \cdot 10^1$	$8.457 \cdot 10^2$
Pentacene	C ₂₂ H ₁₄	146	5328896	$8.608 \cdot 10^1$	$2.224 \cdot 10^2$

^aVariance of the binding energy of the HOMO within the diagonal, frequency-independent approximation.

^bVariance of the binding energy of the HOMO within the non-diagonal, frequency-independent approximation.

Table D.2: The same as Table D.1, but with the aug-cc-pVTZ basis set.

Molecule	Formula	Electrons	Steps	σ_{pp}^2 ^a	σ^2 ^b
Acetylene	C ₂ H ₂	14	7341056	$5.138 \cdot 10^{-1}$	$1.286 \cdot 10^0$
Ethylene	C ₂ H ₄	16	7264256	$1.122 \cdot 10^0$	$1.157 \cdot 10^0$
Ethane	C ₂ H ₆	18	7151616	$5.521 \cdot 10^0$	$7.248 \cdot 10^0$
Butadiyne	C ₄ H ₂	26	7207936	$1.846 \cdot 10^0$	$2.243 \cdot 10^0$
Propane	C ₃ H ₈	26	6890496	$1.678 \cdot 10^1$	$2.058 \cdot 10^1$
Butadiene	C ₄ H ₆	30	6868992	$5.130 \cdot 10^0$	$5.224 \cdot 10^0$
Butane	C ₄ H ₁₀	34	6649856	$1.554 \cdot 10^1$	$1.273 \cdot 10^1$
n-Pentane	C ₅ H ₁₂	42	6409216	$2.907 \cdot 10^1$	$3.264 \cdot 10^1$
Benzene	C ₆ H ₆	42	5254144	$1.287 \cdot 10^1$	$1.329 \cdot 10^1$
Hexatriene	C ₆ H ₈	44	6598656	$1.272 \cdot 10^1$	$3.455 \cdot 10^1$
Hexane	C ₆ H ₁₄	50	6907904	$5.997 \cdot 10^1$	$9.796 \cdot 10^1$
n-Heptane	C ₇ H ₁₆	58	6601728	$4.356 \cdot 10^1$	$5.068 \cdot 10^1$
n-Octane	C ₈ H ₁₈	66	6304768	$8.527 \cdot 10^1$	$1.084 \cdot 10^2$
Napthalene	C ₁₀ H ₈	68	6832128	$4.965 \cdot 10^1$	$1.110 \cdot 10^2$
Nonane	C ₉ H ₂₀	74	6037504	$8.048 \cdot 10^1$	$8.536 \cdot 10^1$
n-Decane	C ₁₀ H ₂₂	82	5585920	$1.292 \cdot 10^2$	$3.069 \cdot 10^2$
Anthracene	C ₁₄ H ₁₀	94	6159360	$9.555 \cdot 10^1$	$1.685 \cdot 10^2$
Naphthacene	C ₁₈ H ₁₂	120	5321728	$1.765 \cdot 10^2$	$5.063 \cdot 10^2$
Pentacene	C ₂₂ H ₁₄	146	3533824	$2.941 \cdot 10^2$	$2.071 \cdot 10^3$

^aVariance of the binding energy of the HOMO within the diagonal, frequency-independent approximation.

^bVariance of the binding energy of the HOMO within the non-diagonal, frequency-independent approximation.

Appendix E

Calculations for System-size Dependence of Cost of MC-GF3

Table E.1: The results of the aug-cc-pVDZ calculations used to establish the scaling of the variance of the third-order self-energy with system size for MC-GF3 within the diagonal, frequency-independent and non-diagonal, frequency-independent approximations. Variances are given in eV^2 .

Molecule	Formula	Electrons	Steps	σ_{pp}^2 ^a	σ^2 ^b
Methane	CH ₄	10	6745088	$8.679 \cdot 10^{-1}$	$1.477 \cdot 10^0$
Acetylene	C ₂ H ₂	14	6673408	$1.066 \cdot 10^0$	$1.417 \cdot 10^0$
Ethylene	C ₂ H ₄	16	6650880	$4.200 \cdot 10^0$	$4.502 \cdot 10^0$
Ethane	C ₂ H ₆	18	5740544	$1.442 \cdot 10^1$	$2.690 \cdot 10^1$
Butadiyne	C ₄ H ₂	26	5728256	$1.088 \cdot 10^1$	$1.722 \cdot 10^1$
Propane	C ₃ H ₈	26	5671936	$5.724 \cdot 10^1$	$6.691 \cdot 10^1$
Butadiene	C ₄ H ₆	30	5680128	$6.190 \cdot 10^1$	$9.359 \cdot 10^1$
Butane	C ₄ H ₁₀	34	5598208	$1.708 \cdot 10^2$	$2.052 \cdot 10^2$
n-Pentane	C ₅ H ₁₂	42	5531648	$4.223 \cdot 10^2$	$5.330 \cdot 10^2$
Benzene	C ₆ H ₆	42	5657600	$2.570 \cdot 10^2$	$8.888 \cdot 10^3$
Hexatriene	C ₆ H ₈	44	5564416	$4.189 \cdot 10^2$	$1.076 \cdot 10^3$
Hexane	C ₆ H ₁₄	50	6178816	$1.137 \cdot 10^3$	$2.096 \cdot 10^3$
n-Heptane	C ₇ H ₁₆	58	6102016	$1.796 \cdot 10^3$	$3.306 \cdot 10^3$
n-Octane	C ₈ H ₁₈	66	5965824	$3.173 \cdot 10^3$	$1.208 \cdot 10^4$
Napthalene	C ₁₀ H ₈	68	6147072	$2.076 \cdot 10^3$	$7.139 \cdot 10^3$
Nonane	C ₉ H ₂₀	74	5856256	$5.153 \cdot 10^3$	$2.692 \cdot 10^4$
n-Decane	C ₁₀ H ₂₂	82	5769216	$9.502 \cdot 10^3$	$4.667 \cdot 10^4$
Anthracene	C ₁₄ H ₁₀	94	5892096	$9.582 \cdot 10^3$	$7.081 \cdot 10^4$
Naphthacene	C ₁₈ H ₁₂	120	5569536	$2.501 \cdot 10^4$	$3.293 \cdot 10^5$
Pentacene	C ₂₂ H ₁₄	146	5328896	$1.114 \cdot 10^5$	$7.458 \cdot 10^6$

^aVariance of the binding energy of the HOMO within the diagonal, frequency-independent approximation.

^bVariance of the binding energy of the HOMO within the non-diagonal, frequency-independent approximation.

Table E.2: The same as Table E.1, but with the aug-cc-pVTZ basis set.

Molecule	Formula	Electrons	Steps	σ_{pp}^2 ^a	σ^2 ^b
Methane	CH ₄	10	5783552	$8.371 \cdot 10^0$	$2.230 \cdot 10^1$
Acetylene	C ₂ H ₂	14	7341056	$1.061 \cdot 10^1$	$1.406 \cdot 10^1$
Ethylene	C ₂ H ₄	16	7264256	$4.010 \cdot 10^1$	$4.812 \cdot 10^1$
Ethane	C ₂ H ₆	18	7151616	$1.006 \cdot 10^2$	$1.224 \cdot 10^2$
Butadiyne	C ₄ H ₂	26	7207936	$6.601 \cdot 10^1$	$3.517 \cdot 10^2$
Propane	C ₃ H ₈	26	6890496	$1.320 \cdot 10^3$	$2.871 \cdot 10^3$
Butadiene	C ₄ H ₆	30	6868992	$6.291 \cdot 10^2$	$1.055 \cdot 10^3$
Butane	C ₄ H ₁₀	34	6649856	$1.724 \cdot 10^3$	$3.744 \cdot 10^3$
n-Pentane	C ₅ H ₁₂	42	6409216	$7.221 \cdot 10^3$	$1.322 \cdot 10^4$
Benzene	C ₆ H ₆	42	5254144	$2.328 \cdot 10^3$	$6.264 \cdot 10^3$
Hexatriene	C ₆ H ₈	44	6598656	$2.745 \cdot 10^3$	$7.579 \cdot 10^3$
Hexane	C ₆ H ₁₄	50	6907904	$9.220 \cdot 10^3$	$1.126 \cdot 10^5$
n-Heptane	C ₇ H ₁₆	58	6601728	$1.434 \cdot 10^4$	$6.014 \cdot 10^4$
n-Octane	C ₈ H ₁₈	66	6304768	$2.988 \cdot 10^4$	$1.753 \cdot 10^5$
Napthalene	C ₁₀ H ₈	68	6832128	$1.796 \cdot 10^4$	$1.421 \cdot 10^5$
Nonane	C ₉ H ₂₀	74	6037504	$4.470 \cdot 10^4$	$2.403 \cdot 10^5$
n-Decane	C ₁₀ H ₂₂	82	5585920	$5.422 \cdot 10^4$	$6.886 \cdot 10^5$
Anthracene	C ₁₄ H ₁₀	94	6159360	$9.449 \cdot 10^4$	$1.553 \cdot 10^6$
Naphthacene	C ₁₈ H ₁₂	120	5321728	$2.879 \cdot 10^5$	$1.048 \cdot 10^7$
Pentacene	C ₂₂ H ₁₄	146	3533824	$1.147 \cdot 10^6$	$6.014 \cdot 10^7$

^aVariance of the binding energy of the HOMO within the diagonal, frequency-independent approximation.

^bVariance of the binding energy of the HOMO within the non-diagonal, frequency-independent approximation.

Appendix F

Format for the Integrands

The following three sections of the appendix contain the integrands of MC-MP3, MC-MP4, and MC-GF3, respectively. Due to both the large number and complexity of the integrands, they are written in a compact format. First, all arguments to the functions are omitted, as they are readily implied from the Sections on the methods. Additionally, all inner-electronic distances from the integrands are omitted. Thus, the expressions in the following sections are connected to the original integrands by

$$f^{(n:d)} = \frac{\bar{f}^{(n:d)}}{r_{1,2} \cdots r_{2n,2n+1}} \quad (\text{F.1})$$

where $f^{(n:d)}$ is the integrand arising from the d^{th} diagram of an n^{th} -order energy or self-energy.

A more compact form for the traces of the advanced and retarded HF Green's functions is also used and is given by

$$G_{d,o}^{\pm} = G^{\pm} \left(\mathbf{r}_d, \mathbf{r}_o, \sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \tau_i - \sum_{i=0}^{\lfloor (o-1)/2 \rfloor} \tau_i \right). \quad (\text{F.2})$$

Appendix G

Integrands for MC-MP3

$$f^{(3:1)} = 2G_{1,5}^- G_{5,3}^+ G_{3,1}^+ G_{2,6}^- G_{6,4}^+ G_{4,2}^+$$

$$f^{(3:2)} = -G_{1,5}^- G_{5,4}^+ G_{3,1}^+ G_{2,6}^- G_{6,3}^+ G_{4,2}^+$$

$$f^{(3:3)} = 2G_{5,1}^+ G_{1,3}^- G_{3,2}^+ G_{2,6}^- G_{6,4}^+ G_{4,5}^-$$

$$f^{(3:4)} = -4G_{5,1}^+ G_{1,6}^- G_{6,3}^+ G_{3,5}^- G_{2,4}^- G_{4,2}^+$$

$$f^{(3:5)} = 8G_{5,1}^+ G_{1,5}^- G_{2,4}^- G_{4,2}^+ G_{3,6}^- G_{6,3}^+$$

$$f^{(3:6)} = -4G_{5,1}^+ G_{1,3}^- G_{3,2}^+ G_{2,5}^- G_{6,4}^+ G_{4,6}^-$$

$$f^{(3:7)} = 2G_{5,1}^+ G_{3,5}^- G_{1,3}^- G_{6,2}^+ G_{4,6}^- G_{2,4}^-$$

$$f^{(3:8)} = -G_{5,1}^+ G_{4,5}^- G_{1,3}^- G_{6,2}^+ G_{3,6}^- G_{2,4}^-$$

$$f^{(3:9)} = 2G_{5,1}^+ G_{1,6}^- G_{6,4}^+ G_{4,2}^- G_{2,3}^- G_{3,5}^-$$

$$f^{(3:10)} = -4G_{5,1}^+ G_{3,5}^- G_{1,3}^- G_{2,6}^- G_{6,4}^+ G_{4,2}^+$$

$$f^{(3:11)} = 2G_{5,1}^+ G_{1,3}^- G_{3,6}^- G_{6,4}^+ G_{4,2}^- G_{2,5}^-$$

$$f^{(3:12)} = -4G_{5,1}^+ G_{1,5}^- G_{2,3}^- G_{3,6}^- G_{6,4}^+ G_{4,2}^+$$

Appendix H

Integrands for MC-MP4

$$f^{(4:1)} = -2G_{3,1}^+ G_{4,2}^+ G_{1,5}^- G_{2,6}^- G_{5,7}^- G_{6,8}^- G_{7,3}^+ G_{8,4}^+$$

$$f^{(4:2)} = G_{3,2}^+ G_{4,1}^+ G_{1,5}^- G_{2,6}^- G_{5,7}^- G_{6,8}^- G_{7,3}^+ G_{8,4}^+$$

$$f^{(4:3)} = -2G_{1,3}^- G_{2,4}^- G_{5,1}^+ G_{6,2}^+ G_{7,5}^+ G_{8,6}^+ G_{3,7}^- G_{4,8}^-$$

$$f^{(4:4)} = G_{2,3}^- G_{1,4}^- G_{5,1}^+ G_{6,2}^+ G_{7,5}^+ G_{8,6}^+ G_{3,7}^- G_{4,8}^-$$

$$f^{(4:5)} = -16G_{4,1}^+ G_{2,6}^- G_{5,8}^- G_{7,3}^+ G_{1,4}^- G_{6,2}^+ G_{8,5}^+ G_{3,7}^-$$

$$f^{(4:6)} = 8G_{4,1}^+ G_{1,5}^- G_{6,8}^- G_{7,3}^+ G_{2,4}^- G_{5,2}^+ G_{8,6}^+ G_{3,7}^-$$

$$f^{(4:7)} = 8G_{3,1}^+ G_{2,6}^- G_{5,8}^- G_{7,3}^+ G_{1,4}^- G_{6,2}^+ G_{8,5}^+ G_{4,7}^-$$

$$f^{(4:8)} = 8G_{3,2}^+ G_{1,5}^- G_{5,7}^- G_{8,4}^+ G_{2,3}^- G_{6,1}^+ G_{7,6}^+ G_{4,8}^-$$

$$f^{(4:9)} = 8G_{3,1}^+ G_{2,6}^- G_{5,7}^- G_{7,4}^+ G_{1,3}^- G_{6,2}^+ G_{8,5}^+ G_{4,8}^-$$

$$f^{(4:10)} = -4G_{3,1}^+ G_{1,6}^- G_{5,7}^- G_{8,3}^+ G_{2,4}^- G_{6,2}^+ G_{7,5}^+ G_{4,8}^-$$

$$f^{(4:11)} = -4G_{3,1}^+ G_{1,5}^- G_{5,7}^- G_{8,4}^+ G_{2,3}^- G_{6,2}^+ G_{7,6}^+ G_{4,8}^-$$

$$f^{(4:12)} = -4G_{4,1}^+ G_{1,6}^- G_{5,7}^- G_{7,3}^+ G_{2,4}^- G_{6,2}^+ G_{8,5}^+ G_{3,8}^-$$

$$f^{(4:13)} = -4G_{3,1}^+ G_{2,5}^- G_{5,8}^- G_{7,3}^+ G_{1,4}^- G_{6,2}^+ G_{8,6}^+ G_{4,7}^-$$

$$f^{(4:14)} = -4G_{3,1}^+ G_{2,6}^- G_{5,7}^- G_{7,3}^+ G_{1,4}^- G_{6,2}^+ G_{8,5}^+ G_{4,8}^-$$

$$f^{(4:15)} = -4G_{4,2}^+ G_{1,5}^- G_{5,7}^- G_{7,3}^+ G_{2,4}^- G_{6,1}^+ G_{8,6}^+ G_{3,8}^-$$

$$f^{(4:16)} = -4G_{3,1}^+ G_{1,5}^- G_{5,7}^- G_{7,3}^+ G_{2,4}^- G_{6,2}^+ G_{8,6}^+ G_{4,8}^-$$

$$f^{(4:17)} = 2G_{3,1}^+ G_{1,5}^- G_{5,7}^- G_{8,3}^+ G_{2,4}^- G_{6,2}^+ G_{7,6}^+ G_{4,8}^-$$

$$f^{(4:18)} = 2G_{3,1}^+ G_{1,6}^- G_{5,7}^- G_{7,3}^+ G_{2,4}^- G_{6,2}^+ G_{8,5}^+ G_{4,8}^-$$

$$f^{(4:19)} = 2G_{4,1}^+ G_{1,5}^- G_{5,7}^- G_{7,3}^+ G_{2,4}^- G_{6,2}^+ G_{8,6}^+ G_{3,8}^-$$

$$f^{(4:20)} = 2G_{3,1}^+ G_{2,5}^- G_{5,7}^- G_{7,3}^+ G_{1,4}^- G_{6,2}^+ G_{8,6}^+ G_{4,8}^-$$

$$f^{(4:21)} = 8G_{1,3}^- G_{3,1}^+ G_{4,2}^- G_{2,6}^- G_{7,5}^+ G_{5,7}^- G_{6,8}^+ G_{8,4}^-$$

$$f^{(4:22)} = -4G_{1,3}^- G_{3,2}^+ G_{4,1}^+ G_{2,6}^- G_{7,5}^+ G_{5,7}^- G_{6,8}^+ G_{8,4}^-$$

$$f^{(4:23)} = -4G_{1,3}^- G_{3,1}^+ G_{4,2}^- G_{2,6}^- G_{7,5}^+ G_{6,7}^- G_{5,8}^+ G_{8,4}^-$$

$$f^{(4:24)} = 2G_{2,3}^- G_{3,1}^+ G_{4,2}^- G_{1,5}^- G_{7,6}^+ G_{5,7}^- G_{6,8}^+ G_{8,4}^-$$

$$f^{(4:25)} = 8G_{3,1}^+ G_{1,3}^- G_{2,4}^- G_{6,2}^+ G_{5,7}^- G_{7,5}^+ G_{8,6}^+ G_{4,8}^-$$

$$f^{(4:26)} = -4G_{3,1}^+ G_{2,3}^- G_{1,4}^- G_{6,2}^+ G_{5,7}^- G_{7,5}^+ G_{8,6}^+ G_{4,8}^-$$

$$f^{(4:27)} = -4G_{3,1}^+ G_{1,3}^- G_{2,4}^- G_{6,2}^+ G_{5,7}^- G_{7,6}^+ G_{8,5}^+ G_{4,8}^-$$

$$f^{(4:28)} = 2G_{3,2}^+ G_{1,3}^- G_{2,4}^- G_{5,1}^- G_{6,7}^- G_{7,5}^+ G_{8,6}^+ G_{4,8}^-$$

$$f^{(4:29)} = 8G_{3,1}^+ G_{2,6}^- G_{1,5}^- G_{6,2}^+ G_{5,7}^- G_{7,3}^+ G_{8,4}^- G_{4,8}^-$$

$$f^{(4:30)} = -4G_{4,1}^+ G_{2,6}^- G_{1,5}^- G_{6,2}^+ G_{5,7}^- G_{7,3}^+ G_{8,4}^- G_{3,8}^-$$

$$f^{(4:31)} = -4G_{3,1}^+ G_{1,5}^- G_{2,6}^- G_{5,2}^+ G_{6,7}^- G_{7,3}^+ G_{8,4}^- G_{4,8}^-$$

$$f^{(4:32)} = 2G_{3,1}^+ G_{1,5}^- G_{2,6}^- G_{5,2}^+ G_{6,8}^+ G_{8,4}^- G_{7,3}^+ G_{4,7}^-$$

$$\begin{aligned}
f^{(4:33)} &= 8G_{1,3}^- G_{6,2}^+ G_{5,1}^+ G_{2,6}^- G_{7,5}^+ G_{3,7}^- G_{4,8}^- G_{8,4}^+ & f^{(4:55)} &= 2G_{1,3}^- G_{4,5}^- G_{5,7}^- G_{7,1}^+ G_{3,2}^+ G_{6,4}^+ G_{8,6}^- G_{2,8}^+ \\
f^{(4:34)} &= -4G_{2,3}^- G_{5,1}^+ G_{6,2}^+ G_{1,5}^- G_{8,6}^+ G_{4,8}^- G_{3,7}^- G_{7,4}^+ & f^{(4:56)} &= 2G_{2,3}^- G_{3,5}^- G_{5,7}^- G_{7,1}^+ G_{4,2}^+ G_{6,4}^+ G_{8,6}^- G_{1,8}^+ \\
f^{(4:35)} &= -4G_{1,3}^- G_{5,1}^+ G_{6,2}^+ G_{2,5}^- G_{7,6}^+ G_{3,7}^- G_{4,8}^- G_{8,4}^+ & f^{(4:57)} &= -8G_{1,7}^- G_{7,5}^+ G_{5,3}^+ G_{3,1}^- G_{2,4}^- G_{4,2}^+ G_{6,8}^- G_{8,6}^+ \\
f^{(4:36)} &= 2G_{1,3}^- G_{5,1}^+ G_{6,2}^+ G_{2,5}^- G_{8,6}^+ G_{4,8}^- G_{3,7}^- G_{7,4}^+ & f^{(4:58)} &= 4G_{1,7}^- G_{7,5}^+ G_{5,3}^+ G_{3,2}^+ G_{2,4}^- G_{4,1}^+ G_{6,8}^- G_{8,6}^+ \\
f^{(4:37)} &= -2G_{3,1}^+ G_{4,2}^+ G_{5,3}^+ G_{6,4}^+ G_{7,5}^+ G_{8,6}^+ G_{1,7}^- G_{2,8}^- & f^{(4:59)} &= 4G_{1,7}^- G_{7,5}^+ G_{6,3}^+ G_{3,1}^- G_{2,4}^- G_{4,2}^+ G_{5,8}^- G_{8,6}^+ \\
f^{(4:38)} &= G_{3,1}^+ G_{4,2}^+ G_{5,3}^+ G_{6,4}^+ G_{7,5}^+ G_{8,6}^+ G_{2,7}^- G_{1,8}^- & f^{(4:60)} &= -2G_{1,7}^- G_{7,5}^+ G_{6,4}^+ G_{4,2}^+ G_{2,3}^- G_{3,1}^- G_{5,8}^- G_{8,6}^+ \\
f^{(4:39)} &= -2G_{1,3}^- G_{2,4}^- G_{3,5}^- G_{4,6}^- G_{5,7}^- G_{6,8}^- G_{7,1}^+ G_{8,2}^+ & f^{(4:61)} &= -8G_{3,1}^+ G_{1,3}^- G_{2,4}^- G_{4,6}^- G_{7,5}^+ G_{5,7}^- G_{6,8}^- G_{8,2}^+ \\
f^{(4:40)} &= G_{1,3}^- G_{2,4}^- G_{3,5}^- G_{4,6}^- G_{5,7}^- G_{6,8}^- G_{7,2}^+ G_{8,1}^+ & f^{(4:62)} &= 4G_{3,1}^+ G_{2,3}^- G_{1,4}^- G_{4,6}^- G_{7,5}^+ G_{5,7}^- G_{6,8}^- G_{8,2}^+ \\
f^{(4:41)} &= -16G_{2,4}^- G_{3,5}^- G_{6,8}^- G_{7,1}^+ G_{4,2}^+ G_{5,3}^+ G_{8,6}^- G_{1,7}^- & f^{(4:63)} &= 4G_{3,1}^+ G_{1,3}^- G_{2,4}^- G_{4,5}^- G_{7,6}^+ G_{5,7}^- G_{6,8}^- G_{8,2}^+ \\
f^{(4:42)} &= 8G_{1,3}^- G_{4,6}^- G_{5,8}^- G_{7,1}^+ G_{3,2}^+ G_{6,4}^+ G_{8,5}^- G_{2,7}^- & f^{(4:64)} &= -2G_{4,1}^+ G_{2,4}^- G_{1,3}^- G_{3,5}^- G_{7,6}^+ G_{5,7}^- G_{6,8}^- G_{8,2}^+ \\
f^{(4:43)} &= 8G_{2,3}^- G_{3,5}^- G_{6,8}^- G_{7,1}^+ G_{4,2}^+ G_{5,4}^+ G_{8,6}^- G_{1,7}^- & f^{(4:65)} &= -8G_{3,5}^- G_{5,3}^+ G_{6,4}^+ G_{8,6}^- G_{7,1}^+ G_{2,8}^- G_{1,7}^- G_{4,2}^+ \\
f^{(4:44)} &= 8G_{1,3}^- G_{4,5}^- G_{5,7}^- G_{8,2}^+ G_{3,1}^+ G_{6,4}^+ G_{7,6}^- G_{2,8}^- & f^{(4:66)} &= 4G_{4,5}^- G_{5,3}^+ G_{6,4}^+ G_{8,6}^- G_{7,1}^+ G_{2,8}^- G_{1,7}^- G_{3,2}^+ \\
f^{(4:45)} &= 8G_{2,3}^- G_{4,6}^- G_{5,7}^- G_{7,1}^+ G_{3,2}^+ G_{6,4}^+ G_{8,5}^- G_{1,8}^- & f^{(4:67)} &= 4G_{4,6}^- G_{6,4}^+ G_{5,3}^+ G_{8,5}^- G_{7,1}^+ G_{1,8}^- G_{2,7}^- G_{3,2}^+ \\
f^{(4:46)} &= -4G_{1,3}^- G_{3,5}^- G_{6,8}^- G_{7,1}^+ G_{4,2}^+ G_{5,4}^+ G_{8,6}^- G_{2,7}^- & f^{(4:68)} &= -2G_{4,5}^- G_{5,3}^+ G_{6,4}^+ G_{8,6}^- G_{7,2}^+ G_{2,8}^- G_{1,7}^- G_{3,1}^+ \\
f^{(4:47)} &= -4G_{1,4}^- G_{3,5}^- G_{5,7}^- G_{8,1}^+ G_{4,2}^+ G_{6,3}^+ G_{7,6}^- G_{2,8}^- & f^{(4:69)} &= -8G_{6,4}^+ G_{4,6}^- G_{3,5}^- G_{5,7}^- G_{2,8}^- G_{7,1}^+ G_{8,2}^- G_{1,3}^- \\
f^{(4:48)} &= -4G_{1,3}^- G_{4,6}^- G_{5,7}^- G_{7,1}^+ G_{3,2}^+ G_{6,4}^+ G_{8,5}^- G_{2,8}^- & f^{(4:70)} &= 4G_{5,4}^+ G_{3,5}^- G_{4,6}^- G_{6,8}^- G_{1,7}^- G_{8,2}^+ G_{7,1}^+ G_{2,3}^- \\
f^{(4:49)} &= -4G_{1,3}^- G_{3,5}^- G_{5,7}^- G_{8,2}^+ G_{4,1}^+ G_{6,4}^+ G_{7,6}^- G_{2,8}^- & f^{(4:71)} &= 4G_{6,4}^+ G_{4,6}^- G_{3,5}^- G_{5,8}^- G_{1,7}^- G_{8,1}^+ G_{7,2}^+ G_{2,3}^- \\
f^{(4:50)} &= -4G_{2,3}^- G_{3,6}^- G_{5,7}^- G_{7,1}^+ G_{4,2}^+ G_{6,4}^+ G_{8,5}^- G_{1,8}^- & f^{(4:72)} &= -2G_{6,3}^+ G_{4,6}^- G_{3,5}^- G_{5,7}^- G_{1,8}^- G_{7,1}^+ G_{8,2}^- G_{2,4}^- \\
f^{(4:51)} &= -4G_{1,3}^- G_{4,5}^- G_{5,7}^- G_{7,2}^+ G_{3,1}^+ G_{6,4}^+ G_{8,6}^- G_{2,8}^- & f^{(4:73)} &= -2G_{5,1}^+ G_{6,2}^+ G_{3,5}^- G_{4,6}^- G_{1,7}^- G_{2,8}^- G_{7,3}^+ G_{8,4}^+ \\
f^{(4:52)} &= -4G_{1,3}^- G_{3,5}^- G_{5,7}^- G_{7,1}^+ G_{4,2}^+ G_{6,4}^+ G_{8,6}^- G_{2,8}^- & f^{(4:74)} &= G_{5,1}^+ G_{6,2}^+ G_{3,5}^- G_{4,6}^- G_{2,8}^- G_{1,7}^- G_{8,3}^+ G_{7,4}^+ \\
f^{(4:53)} &= 2G_{1,3}^- G_{3,5}^- G_{5,7}^- G_{8,1}^+ G_{4,2}^+ G_{6,4}^+ G_{7,6}^- G_{2,8}^- & f^{(4:75)} &= -2G_{1,5}^- G_{2,6}^- G_{5,3}^+ G_{6,4}^+ G_{7,1}^+ G_{8,2}^- G_{3,7}^- G_{4,8}^- \\
f^{(4:54)} &= 2G_{1,3}^- G_{3,5}^- G_{6,7}^- G_{7,1}^+ G_{4,2}^+ G_{5,4}^+ G_{8,6}^- G_{2,8}^- & f^{(4:76)} &= G_{2,6}^- G_{1,5}^- G_{6,4}^+ G_{5,3}^+ G_{7,1}^+ G_{8,2}^- G_{4,7}^- G_{3,8}^-
\end{aligned}$$

$$f^{(4:77)} = -16G_{2,6}^- G_{5,3}^+ G_{7,1}^+ G_{4,8}^- G_{6,2}^+ G_{3,5}^- G_{1,7}^- G_{8,4}^+$$

$$f^{(4:78)} = 8G_{1,5}^- G_{6,4}^+ G_{8,1}^+ G_{3,7}^- G_{5,2}^+ G_{4,6}^- G_{2,8}^- G_{7,3}^+$$

$$f^{(4:79)} = 8G_{2,6}^- G_{5,3}^+ G_{7,1}^- G_{3,8}^- G_{6,2}^+ G_{4,5}^- G_{1,7}^- G_{8,4}^+$$

$$f^{(4:80)} = 8G_{2,5}^- G_{5,3}^+ G_{7,1}^+ G_{4,8}^- G_{6,2}^+ G_{3,6}^- G_{1,7}^- G_{8,4}^+$$

$$f^{(4:81)} = 8G_{2,6}^- G_{5,4}^+ G_{7,1}^- G_{3,7}^- G_{6,2}^+ G_{4,5}^- G_{1,8}^- G_{8,3}^+$$

$$f^{(4:82)} = -4G_{1,6}^- G_{5,3}^+ G_{8,1}^+ G_{3,7}^- G_{6,2}^+ G_{4,5}^- G_{2,8}^- G_{7,4}^+$$

$$f^{(4:83)} = -4G_{1,5}^- G_{5,3}^+ G_{7,1}^+ G_{4,8}^- G_{6,2}^+ G_{3,6}^- G_{2,7}^- G_{8,4}^+$$

$$f^{(4:84)} = -4G_{1,5}^- G_{6,4}^+ G_{7,1}^+ G_{3,7}^- G_{5,2}^+ G_{4,6}^- G_{2,8}^- G_{8,3}^+$$

$$f^{(4:85)} = -4G_{2,5}^- G_{5,3}^+ G_{7,1}^+ G_{3,8}^- G_{6,2}^+ G_{4,6}^- G_{1,7}^- G_{8,4}^+$$

$$f^{(4:86)} = -4G_{2,6}^- G_{5,3}^+ G_{7,1}^+ G_{3,7}^- G_{6,2}^+ G_{4,5}^- G_{1,8}^- G_{8,4}^+$$

$$f^{(4:87)} = -4G_{2,5}^- G_{5,4}^+ G_{7,1}^+ G_{3,7}^- G_{6,2}^+ G_{4,6}^- G_{1,8}^- G_{8,3}^+$$

$$f^{(4:88)} = -4G_{1,5}^- G_{5,3}^+ G_{7,1}^+ G_{3,7}^- G_{6,2}^+ G_{4,6}^- G_{2,8}^- G_{8,4}^+$$

$$f^{(4:89)} = 2G_{1,5}^- G_{5,3}^+ G_{8,1}^- G_{3,7}^- G_{6,2}^+ G_{4,6}^- G_{2,8}^- G_{7,4}^+$$

$$f^{(4:90)} = 2G_{1,5}^- G_{6,3}^+ G_{7,1}^+ G_{3,7}^- G_{5,2}^+ G_{4,6}^- G_{2,8}^- G_{8,4}^+$$

$$f^{(4:91)} = 2G_{1,5}^- G_{5,3}^+ G_{7,1}^- G_{4,7}^- G_{6,2}^+ G_{3,6}^- G_{2,8}^- G_{8,4}^+$$

$$f^{(4:92)} = 2G_{2,5}^- G_{5,3}^+ G_{7,1}^- G_{3,7}^- G_{6,2}^+ G_{4,6}^- G_{1,8}^- G_{8,4}^+$$

$$f^{(4:93)} = 8G_{5,3}^+ G_{3,5}^- G_{4,6}^- G_{8,4}^+ G_{7,1}^- G_{2,8}^- G_{1,7}^- G_{6,2}^+$$

$$f^{(4:94)} = -4G_{5,3}^+ G_{4,5}^- G_{3,6}^- G_{8,4}^+ G_{7,1}^- G_{2,8}^- G_{1,7}^- G_{6,2}^+$$

$$f^{(4:95)} = -4G_{5,3}^+ G_{3,5}^- G_{4,6}^- G_{7,4}^+ G_{8,1}^- G_{1,7}^- G_{2,8}^- G_{6,2}^+$$

$$f^{(4:96)} = 2G_{5,4}^+ G_{3,5}^- G_{4,6}^- G_{7,3}^+ G_{8,1}^- G_{1,7}^- G_{2,8}^- G_{6,2}^+$$

$$f^{(4:97)} = 8G_{3,5}^- G_{5,3}^+ G_{6,4}^- G_{1,7}^- G_{7,1}^+ G_{8,2}^+ G_{4,8}^- G_{2,6}^-$$

$$f^{(4:98)} = -4G_{3,5}^- G_{5,3}^+ G_{6,4}^- G_{2,7}^- G_{7,1}^+ G_{8,2}^+ G_{4,8}^- G_{1,6}^-$$

$$f^{(4:99)} = -4G_{3,5}^- G_{5,4}^+ G_{6,3}^- G_{1,7}^- G_{7,1}^+ G_{8,2}^+ G_{4,8}^- G_{2,6}^-$$

$$f^{(4:100)} = 2G_{3,6}^- G_{6,4}^+ G_{5,3}^- G_{2,7}^- G_{7,1}^+ G_{8,2}^+ G_{4,8}^- G_{1,5}^-$$

$$f^{(4:101)} = 8G_{5,1}^+ G_{6,2}^- G_{1,5}^- G_{4,6}^- G_{3,7}^- G_{7,3}^+ G_{8,4}^- G_{2,8}^-$$

$$f^{(4:102)} = -4G_{6,2}^+ G_{5,1}^- G_{1,6}^- G_{4,5}^- G_{3,7}^- G_{7,3}^+ G_{8,4}^- G_{2,8}^-$$

$$f^{(4:103)} = -4G_{5,1}^+ G_{6,2}^- G_{1,5}^- G_{3,6}^- G_{4,7}^- G_{7,3}^+ G_{8,4}^- G_{2,8}^-$$

$$f^{(4:104)} = 2G_{5,2}^+ G_{6,1}^- G_{1,5}^- G_{4,6}^- G_{3,7}^- G_{7,4}^+ G_{8,3}^- G_{2,8}^-$$

$$f^{(4:105)} = 8G_{1,5}^- G_{2,6}^- G_{5,1}^+ G_{6,4}^- G_{7,3}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+$$

$$f^{(4:106)} = -4G_{2,6}^- G_{1,5}^- G_{6,1}^+ G_{5,4}^- G_{7,3}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+$$

$$f^{(4:107)} = -4G_{1,5}^- G_{2,6}^- G_{5,1}^+ G_{6,3}^- G_{7,4}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+$$

$$f^{(4:108)} = 2G_{2,5}^- G_{1,6}^- G_{5,1}^+ G_{6,4}^- G_{7,3}^- G_{4,7}^- G_{3,8}^- G_{8,2}^+$$

$$f^{(4:109)} = -8G_{4,2}^+ G_{3,1}^- G_{1,5}^- G_{8,4}^- G_{6,7}^- G_{7,6}^- G_{5,3}^- G_{2,8}^-$$

$$f^{(4:110)} = 4G_{4,1}^+ G_{3,2}^- G_{1,5}^- G_{8,4}^- G_{6,7}^- G_{7,6}^- G_{5,3}^- G_{2,8}^-$$

$$f^{(4:111)} = 4G_{4,2}^+ G_{3,1}^- G_{1,5}^- G_{8,4}^- G_{5,7}^- G_{7,6}^- G_{6,3}^- G_{2,8}^-$$

$$f^{(4:112)} = 4G_{4,2}^+ G_{3,1}^- G_{1,5}^- G_{7,4}^- G_{6,7}^- G_{8,6}^- G_{5,3}^- G_{2,8}^-$$

$$f^{(4:113)} = 4G_{3,1}^+ G_{4,2}^- G_{1,5}^- G_{7,3}^- G_{5,7}^- G_{8,6}^- G_{6,4}^- G_{2,8}^-$$

$$f^{(4:114)} = -2G_{3,1}^+ G_{4,2}^- G_{1,5}^- G_{8,3}^- G_{5,7}^- G_{7,6}^- G_{6,4}^- G_{2,8}^-$$

$$f^{(4:115)} = -2G_{3,1}^+ G_{4,2}^- G_{1,5}^- G_{7,3}^- G_{6,7}^- G_{8,6}^- G_{5,4}^- G_{2,8}^-$$

$$f^{(4:116)} = -2G_{3,2}^+ G_{4,1}^- G_{1,5}^- G_{7,3}^- G_{5,7}^- G_{8,6}^- G_{6,4}^- G_{2,8}^-$$

$$f^{(4:117)} = -8G_{1,4}^- G_{4,1}^+ G_{7,5}^- G_{8,6}^- G_{5,3}^- G_{6,2}^- G_{3,7}^- G_{2,8}^-$$

$$f^{(4:118)} = 4G_{1,4}^- G_{4,2}^+ G_{7,5}^- G_{8,6}^- G_{5,3}^- G_{6,1}^- G_{3,7}^- G_{2,8}^-$$

$$f^{(4:119)} = 4G_{1,3}^- G_{4,1}^+ G_{7,5}^- G_{8,6}^- G_{5,4}^- G_{6,2}^- G_{3,7}^- G_{2,8}^-$$

$$f^{(4:120)} = 4G_{1,4}^- G_{4,1}^+ G_{7,6}^- G_{8,5}^- G_{5,3}^- G_{6,2}^- G_{3,7}^- G_{2,8}^-$$

$$\begin{aligned}
f^{(4:121)} &= 4G_{1,3}^- G_{4,2}^+ G_{7,5}^+ G_{8,6}^+ G_{6,4}^+ G_{5,1}^+ G_{3,7}^- G_{2,8}^- & f^{(4:143)} &= 8G_{3,1}^+ G_{1,3}^- G_{7,5}^+ G_{5,7}^- G_{6,4}^+ G_{2,6}^- G_{4,8}^- G_{8,2}^+ \\
f^{(4:122)} &= -2G_{1,4}^- G_{4,2}^+ G_{7,5}^+ G_{8,6}^+ G_{6,3}^+ G_{5,1}^+ G_{3,7}^- G_{2,8}^- & f^{(4:144)} &= -4G_{4,1}^+ G_{1,3}^- G_{7,5}^+ G_{6,8}^- G_{5,4}^+ G_{2,6}^- G_{3,7}^- G_{8,2}^+ \\
f^{(4:123)} &= -2G_{1,3}^- G_{4,1}^+ G_{7,5}^+ G_{8,6}^+ G_{6,4}^+ G_{5,2}^+ G_{3,7}^- G_{2,8}^- & f^{(4:145)} &= -4G_{4,1}^+ G_{2,4}^- G_{7,5}^+ G_{6,8}^- G_{5,3}^+ G_{1,6}^- G_{3,7}^- G_{8,2}^+ \\
f^{(4:124)} &= -2G_{1,3}^- G_{4,2}^+ G_{7,6}^+ G_{8,5}^+ G_{6,4}^+ G_{5,1}^+ G_{3,7}^- G_{2,8}^- & f^{(4:146)} &= -4G_{3,1}^+ G_{2,4}^- G_{7,6}^+ G_{6,8}^- G_{5,3}^+ G_{1,5}^- G_{4,7}^- G_{8,2}^+ \\
f^{(4:125)} &= -8G_{1,3}^- G_{2,4}^- G_{7,6}^+ G_{6,7}^- G_{3,5}^- G_{5,1}^+ G_{4,8}^- G_{8,2}^+ & f^{(4:147)} &= -4G_{3,1}^+ G_{2,4}^- G_{7,6}^+ G_{5,7}^- G_{6,3}^+ G_{1,5}^- G_{4,8}^- G_{8,2}^+ \\
f^{(4:126)} &= 4G_{2,3}^- G_{1,4}^- G_{7,6}^+ G_{6,7}^- G_{3,5}^- G_{5,1}^+ G_{4,8}^- G_{8,2}^+ & f^{(4:148)} &= -4G_{3,1}^+ G_{1,3}^- G_{7,5}^+ G_{5,8}^- G_{6,4}^+ G_{2,6}^- G_{4,7}^- G_{8,2}^+ \\
f^{(4:127)} &= 4G_{1,3}^- G_{2,4}^- G_{7,5}^+ G_{6,7}^- G_{3,6}^- G_{5,1}^+ G_{4,8}^- G_{8,2}^+ & f^{(4:149)} &= -4G_{3,1}^+ G_{1,3}^- G_{7,5}^+ G_{6,7}^- G_{5,4}^+ G_{2,6}^- G_{4,8}^- G_{8,2}^+ \\
f^{(4:128)} &= 4G_{1,3}^- G_{2,4}^- G_{7,6}^+ G_{6,8}^- G_{3,5}^- G_{5,1}^+ G_{4,7}^- G_{8,2}^+ & f^{(4:150)} &= -4G_{4,1}^+ G_{1,3}^- G_{7,5}^+ G_{5,7}^- G_{6,4}^+ G_{2,6}^- G_{3,8}^- G_{8,2}^+ \\
f^{(4:129)} &= 4G_{2,4}^- G_{1,3}^- G_{7,5}^+ G_{6,8}^- G_{4,6}^- G_{5,1}^+ G_{3,7}^- G_{8,2}^+ & f^{(4:151)} &= -4G_{3,1}^+ G_{2,3}^- G_{7,5}^+ G_{5,7}^- G_{6,4}^+ G_{1,6}^- G_{4,8}^- G_{8,2}^+ \\
f^{(4:130)} &= -2G_{2,4}^- G_{1,3}^- G_{7,6}^+ G_{6,8}^- G_{4,5}^- G_{5,1}^+ G_{3,7}^- G_{8,2}^+ & f^{(4:152)} &= 2G_{4,1}^+ G_{2,3}^- G_{7,6}^+ G_{5,8}^- G_{6,4}^+ G_{1,5}^- G_{3,7}^- G_{8,2}^+ \\
f^{(4:131)} &= -2G_{1,4}^- G_{2,3}^- G_{7,5}^+ G_{6,8}^- G_{4,6}^- G_{5,1}^+ G_{3,7}^- G_{8,2}^+ & f^{(4:153)} &= 2G_{4,1}^+ G_{1,3}^- G_{7,5}^+ G_{5,8}^- G_{6,4}^+ G_{2,6}^- G_{3,7}^- G_{8,2}^+ \\
f^{(4:132)} &= -2G_{2,3}^- G_{1,4}^- G_{7,5}^+ G_{6,7}^- G_{3,6}^- G_{5,1}^+ G_{4,8}^- G_{8,2}^+ & f^{(4:154)} &= 2G_{3,1}^+ G_{1,4}^- G_{7,5}^+ G_{6,7}^- G_{5,3}^+ G_{2,6}^- G_{4,8}^- G_{8,2}^+ \\
f^{(4:133)} &= -8G_{4,1}^+ G_{1,4}^- G_{5,7}^- G_{6,8}^- G_{3,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:155)} &= 2G_{4,1}^+ G_{2,4}^- G_{7,6}^+ G_{6,8}^- G_{5,3}^+ G_{1,5}^- G_{3,7}^- G_{8,2}^+ \\
f^{(4:134)} &= 4G_{4,1}^+ G_{2,4}^- G_{5,7}^- G_{6,8}^- G_{3,5}^- G_{1,6}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:156)} &= 2G_{3,1}^+ G_{2,3}^- G_{7,6}^+ G_{5,7}^- G_{6,4}^+ G_{1,5}^- G_{4,8}^- G_{8,2}^+ \\
f^{(4:135)} &= 4G_{3,1}^+ G_{1,4}^- G_{5,7}^- G_{6,8}^- G_{4,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:157)} &= 8G_{4,1}^+ G_{1,4}^- G_{8,6}^- G_{5,7}^- G_{3,5}^- G_{6,2}^+ G_{7,3}^+ G_{2,8}^- \\
f^{(4:136)} &= 4G_{4,1}^+ G_{1,4}^- G_{6,7}^- G_{5,8}^- G_{3,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:158)} &= 8G_{4,2}^+ G_{1,3}^- G_{7,6}^- G_{6,7}^- G_{3,5}^- G_{5,1}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:137)} &= 4G_{3,1}^+ G_{2,4}^- G_{5,7}^- G_{6,8}^- G_{4,6}^- G_{1,5}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:159)} &= 8G_{3,1}^+ G_{1,3}^- G_{7,5}^- G_{5,7}^- G_{4,6}^- G_{6,2}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:138)} &= -2G_{4,1}^+ G_{2,4}^- G_{5,7}^- G_{6,8}^- G_{3,6}^- G_{1,5}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:160)} &= -4G_{3,1}^+ G_{1,4}^- G_{8,6}^+ G_{5,7}^- G_{4,5}^- G_{6,2}^+ G_{7,3}^+ G_{2,8}^- \\
f^{(4:139)} &= -2G_{3,1}^+ G_{1,4}^- G_{6,7}^- G_{5,8}^- G_{4,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:161)} &= -4G_{4,2}^+ G_{1,4}^- G_{8,6}^+ G_{5,7}^- G_{3,5}^- G_{6,1}^+ G_{7,3}^+ G_{2,8}^- \\
f^{(4:140)} &= -2G_{3,1}^+ G_{2,4}^- G_{6,7}^- G_{5,8}^- G_{4,6}^- G_{1,5}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:162)} &= -4G_{4,2}^+ G_{1,3}^- G_{8,6}^+ G_{6,7}^- G_{3,5}^- G_{5,1}^+ G_{7,4}^+ G_{2,8}^- \\
f^{(4:141)} &= 8G_{4,1}^+ G_{1,4}^- G_{7,5}^- G_{6,8}^- G_{5,3}^+ G_{2,6}^- G_{3,7}^- G_{8,2}^+ & f^{(4:163)} &= -4G_{4,2}^+ G_{1,3}^- G_{7,5}^- G_{6,7}^- G_{3,6}^- G_{5,1}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:142)} &= 8G_{3,1}^+ G_{2,4}^- G_{7,6}^- G_{6,7}^- G_{5,3}^+ G_{1,5}^- G_{4,8}^- G_{8,2}^+ & f^{(4:164)} &= -4G_{4,1}^+ G_{1,4}^- G_{8,5}^+ G_{5,7}^- G_{3,6}^- G_{6,2}^+ G_{7,3}^+ G_{2,8}^-
\end{aligned}$$

$$\begin{aligned}
f^{(4:165)} &= -4G_{3,1}^+ G_{1,3}^- G_{7,5}^+ G_{6,7}^- G_{4,6}^- G_{5,2}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:166)} &= -4G_{4,1}^+ G_{1,3}^- G_{7,5}^+ G_{5,7}^- G_{3,6}^- G_{6,2}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:167)} &= -4G_{3,2}^+ G_{1,3}^- G_{7,5}^+ G_{5,7}^- G_{4,6}^- G_{6,1}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:168)} &= 2G_{3,2}^+ G_{1,4}^- G_{8,5}^+ G_{6,7}^- G_{4,6}^- G_{5,1}^+ G_{7,3}^+ G_{2,8}^- \\
f^{(4:169)} &= 2G_{3,1}^+ G_{1,4}^- G_{8,5}^+ G_{5,7}^- G_{4,6}^- G_{6,2}^+ G_{7,3}^+ G_{2,8}^- \\
f^{(4:170)} &= 2G_{4,1}^+ G_{1,3}^- G_{7,6}^+ G_{5,7}^- G_{3,5}^- G_{6,2}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:171)} &= 2G_{4,2}^+ G_{1,4}^- G_{8,6}^+ G_{6,7}^- G_{3,5}^- G_{5,1}^+ G_{7,3}^+ G_{2,8}^- \\
f^{(4:172)} &= 2G_{3,2}^+ G_{1,3}^- G_{7,5}^+ G_{6,7}^- G_{4,6}^- G_{5,1}^+ G_{8,4}^+ G_{2,8}^- \\
f^{(4:173)} &= -8G_{4,1}^+ G_{5,4}^+ G_{8,6}^+ G_{3,7}^- G_{7,3}^+ G_{1,5}^- G_{6,2}^+ G_{2,8}^- \\
f^{(4:174)} &= -8G_{4,2}^+ G_{6,4}^+ G_{8,6}^+ G_{3,7}^- G_{7,3}^+ G_{1,5}^- G_{5,1}^+ G_{2,8}^- \\
f^{(4:175)} &= -8G_{4,2}^+ G_{6,3}^+ G_{7,6}^+ G_{3,7}^- G_{8,4}^+ G_{1,5}^- G_{5,1}^+ G_{2,8}^- \\
f^{(4:176)} &= 4G_{4,1}^+ G_{5,3}^+ G_{8,6}^+ G_{3,7}^- G_{7,4}^+ G_{1,5}^- G_{6,2}^+ G_{2,8}^- \\
f^{(4:177)} &= 4G_{4,2}^+ G_{6,4}^+ G_{8,5}^+ G_{3,7}^- G_{7,3}^+ G_{1,6}^- G_{5,1}^+ G_{2,8}^- \\
f^{(4:178)} &= 4G_{3,2}^+ G_{6,4}^+ G_{8,6}^+ G_{4,7}^- G_{7,3}^+ G_{1,5}^- G_{5,1}^+ G_{2,8}^- \\
f^{(4:179)} &= 4G_{4,2}^+ G_{5,3}^+ G_{7,6}^+ G_{3,7}^- G_{8,4}^+ G_{1,5}^- G_{6,1}^+ G_{2,8}^- \\
f^{(4:180)} &= 4G_{4,1}^+ G_{6,4}^+ G_{8,6}^+ G_{3,7}^- G_{7,3}^+ G_{1,5}^- G_{5,2}^+ G_{2,8}^- \\
f^{(4:181)} &= 4G_{3,1}^+ G_{5,3}^+ G_{7,6}^+ G_{4,7}^- G_{8,4}^+ G_{1,5}^- G_{6,2}^+ G_{2,8}^- \\
f^{(4:182)} &= 4G_{4,1}^+ G_{5,3}^+ G_{7,5}^+ G_{3,7}^- G_{8,4}^+ G_{1,6}^- G_{6,2}^+ G_{2,8}^- \\
f^{(4:183)} &= 4G_{3,2}^+ G_{6,3}^+ G_{7,6}^+ G_{4,7}^- G_{8,4}^+ G_{1,5}^- G_{5,1}^+ G_{2,8}^- \\
f^{(4:184)} &= -2G_{4,2}^+ G_{5,3}^+ G_{8,6}^+ G_{3,7}^- G_{7,4}^+ G_{1,5}^- G_{6,1}^+ G_{2,8}^- \\
f^{(4:185)} &= -2G_{3,1}^+ G_{6,4}^+ G_{8,6}^+ G_{4,7}^- G_{7,3}^+ G_{1,5}^- G_{5,2}^+ G_{2,8}^- \\
f^{(4:186)} &= -2G_{4,1}^+ G_{5,3}^+ G_{7,6}^+ G_{3,7}^- G_{8,4}^+ G_{1,5}^- G_{6,2}^+ G_{2,8}^- \\
f^{(4:187)} &= -2G_{4,2}^+ G_{6,4}^+ G_{7,5}^+ G_{3,7}^- G_{8,3}^+ G_{1,6}^- G_{5,1}^+ G_{2,8}^- \\
f^{(4:188)} &= -2G_{3,1}^+ G_{5,3}^+ G_{7,5}^+ G_{4,7}^- G_{8,4}^+ G_{1,6}^- G_{6,2}^+ G_{2,8}^- \\
f^{(4:189)} &= -8G_{1,4}^- G_{4,5}^- G_{6,8}^- G_{3,7}^- G_{7,3}^+ G_{2,6}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:190)} &= -8G_{2,4}^- G_{4,6}^- G_{6,8}^- G_{3,7}^- G_{7,3}^+ G_{1,5}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:191)} &= -8G_{2,4}^- G_{3,6}^- G_{6,7}^- G_{4,8}^- G_{7,3}^+ G_{1,5}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:192)} &= 4G_{1,3}^- G_{4,5}^- G_{6,8}^- G_{3,7}^- G_{7,4}^+ G_{2,6}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:193)} &= 4G_{2,4}^- G_{4,6}^- G_{5,8}^- G_{3,7}^- G_{7,3}^+ G_{1,5}^- G_{6,1}^+ G_{8,2}^+ \\
f^{(4:194)} &= 4G_{2,3}^- G_{4,6}^- G_{6,8}^- G_{3,7}^- G_{7,4}^+ G_{1,5}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:195)} &= 4G_{2,4}^- G_{3,5}^- G_{6,7}^- G_{4,8}^- G_{7,3}^+ G_{1,6}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:196)} &= 4G_{1,4}^- G_{4,5}^- G_{5,8}^- G_{3,7}^- G_{7,3}^+ G_{2,6}^- G_{6,1}^+ G_{8,2}^+ \\
f^{(4:197)} &= 4G_{1,3}^- G_{3,5}^- G_{6,7}^- G_{4,8}^- G_{7,4}^+ G_{2,6}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:198)} &= 4G_{1,4}^- G_{3,5}^- G_{5,7}^- G_{4,8}^- G_{7,3}^+ G_{2,6}^- G_{6,1}^+ G_{8,2}^+ \\
f^{(4:199)} &= 4G_{2,3}^- G_{3,6}^- G_{6,7}^- G_{4,8}^- G_{7,4}^+ G_{1,5}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:200)} &= -2G_{2,4}^- G_{3,5}^- G_{6,8}^- G_{4,7}^- G_{7,3}^+ G_{1,6}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:201)} &= -2G_{1,3}^- G_{4,6}^- G_{6,8}^- G_{3,7}^- G_{7,4}^+ G_{2,5}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:202)} &= -2G_{1,4}^- G_{3,5}^- G_{6,7}^- G_{4,8}^- G_{7,3}^+ G_{2,6}^- G_{5,1}^+ G_{8,2}^+ \\
f^{(4:203)} &= -2G_{2,4}^- G_{4,6}^- G_{5,7}^- G_{3,8}^- G_{7,3}^+ G_{1,5}^- G_{6,1}^+ G_{8,2}^+ \\
f^{(4:204)} &= -2G_{1,3}^- G_{3,5}^- G_{5,7}^- G_{4,8}^- G_{7,4}^+ G_{2,6}^- G_{6,1}^+ G_{8,2}^+ \\
f^{(4:205)} &= 8G_{4,1}^+ G_{5,4}^+ G_{6,8}^- G_{1,5}^- G_{2,6}^- G_{3,7}^+ G_{7,3}^+ G_{8,2}^+ \\
f^{(4:206)} &= -4G_{4,1}^+ G_{5,3}^+ G_{6,8}^- G_{1,5}^- G_{2,6}^- G_{3,7}^+ G_{7,4}^+ G_{8,2}^+ \\
f^{(4:207)} &= -4G_{4,1}^+ G_{6,4}^+ G_{5,8}^- G_{2,6}^- G_{1,5}^- G_{3,7}^+ G_{7,3}^+ G_{8,2}^+ \\
f^{(4:208)} &= -4G_{3,1}^+ G_{5,3}^+ G_{6,7}^- G_{1,5}^- G_{2,6}^- G_{4,8}^- G_{7,4}^+ G_{8,2}^+
\end{aligned}$$

$$\begin{aligned}
f^{(4:209)} &= -4G_{3,1}^+ G_{6,4}^+ G_{5,7}^- G_{2,6}^- G_{1,5}^- G_{4,8}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:231)} &= -4G_{3,1}^+ G_{4,6}^- G_{5,7}^- G_{1,5}^- G_{6,2}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- \\
f^{(4:210)} &= 2G_{3,1}^+ G_{6,3}^+ G_{5,7}^- G_{2,6}^- G_{1,5}^- G_{4,8}^- G_{7,4}^+ G_{8,2}^+ & f^{(4:232)} &= -4G_{4,1}^+ G_{3,5}^- G_{5,7}^- G_{1,6}^- G_{6,2}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- \\
f^{(4:211)} &= 2G_{3,1}^+ G_{6,4}^+ G_{5,8}^- G_{2,6}^- G_{1,5}^- G_{4,7}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:233)} &= 2G_{3,1}^+ G_{4,5}^- G_{5,7}^- G_{1,6}^- G_{6,2}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- \\
f^{(4:212)} &= 2G_{3,1}^+ G_{5,4}^+ G_{6,7}^- G_{1,5}^- G_{2,6}^- G_{4,8}^- G_{7,3}^+ G_{8,2}^+ & f^{(4:234)} &= 2G_{4,1}^+ G_{3,6}^- G_{5,7}^- G_{1,5}^- G_{6,2}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- \\
f^{(4:213)} &= 8G_{2,4}^- G_{6,3}^+ G_{7,6}^+ G_{1,5}^- G_{5,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:235)} &= 2G_{3,2}^+ G_{4,6}^- G_{5,7}^- G_{1,5}^- G_{6,1}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- \\
f^{(4:214)} &= -4G_{2,4}^- G_{6,3}^+ G_{7,5}^- G_{1,6}^- G_{5,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:236)} &= -4G_{3,2}^+ G_{4,6}^- G_{6,7}^- G_{1,5}^- G_{5,1}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- \\
f^{(4:215)} &= -4G_{1,3}^- G_{6,4}^+ G_{7,5}^- G_{2,6}^- G_{5,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:237)} &= -8G_{4,2}^+ G_{7,5}^- G_{3,6}^- G_{6,3}^+ G_{5,1}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:216)} &= -4G_{1,4}^- G_{5,3}^+ G_{7,5}^- G_{2,6}^- G_{6,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:238)} &= 4G_{4,1}^+ G_{7,6}^+ G_{3,5}^- G_{5,3}^+ G_{6,2}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:217)} &= -4G_{2,3}^- G_{6,4}^+ G_{7,6}^- G_{1,5}^- G_{5,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:239)} &= 4G_{3,2}^+ G_{7,5}^- G_{4,6}^- G_{6,3}^+ G_{5,1}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:218)} &= 2G_{1,3}^- G_{6,4}^+ G_{7,5}^- G_{2,6}^- G_{5,1}^+ G_{4,7}^- G_{3,8}^- G_{8,2}^+ & f^{(4:240)} &= 4G_{4,2}^+ G_{7,5}^- G_{3,6}^- G_{5,3}^+ G_{6,1}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:219)} &= 2G_{2,3}^- G_{6,4}^+ G_{7,5}^- G_{1,6}^- G_{5,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:241)} &= 4G_{3,1}^+ G_{7,5}^- G_{4,6}^- G_{5,3}^+ G_{6,2}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:220)} &= 2G_{1,3}^- G_{6,4}^+ G_{7,6}^- G_{2,5}^- G_{5,1}^- G_{3,7}^- G_{4,8}^- G_{8,2}^+ & f^{(4:242)} &= -2G_{3,1}^+ G_{7,6}^+ G_{4,5}^- G_{5,3}^+ G_{6,2}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:221)} &= 8G_{1,4}^- G_{4,5}^- G_{8,6}^+ G_{5,1}^+ G_{6,2}^- G_{3,7}^- G_{7,3}^- G_{2,8}^- & f^{(4:243)} &= -2G_{4,1}^+ G_{7,5}^- G_{3,6}^- G_{5,3}^+ G_{6,2}^+ G_{8,4}^- G_{1,7}^- G_{2,8}^- \\
f^{(4:222)} &= -4G_{1,3}^- G_{4,5}^- G_{8,6}^+ G_{5,1}^+ G_{6,2}^- G_{3,7}^- G_{7,4}^- G_{2,8}^- & f^{(4:244)} &= -2G_{3,1}^+ G_{7,5}^- G_{4,6}^- G_{5,3}^+ G_{6,2}^+ G_{8,4}^- G_{2,7}^- G_{1,8}^- \\
f^{(4:223)} &= -4G_{1,4}^- G_{4,6}^- G_{8,5}^+ G_{6,2}^- G_{5,1}^- G_{3,7}^- G_{7,3}^- G_{2,8}^- & f^{(4:245)} &= -8G_{2,4}^- G_{5,7}^- G_{3,6}^- G_{6,3}^+ G_{1,5}^- G_{4,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:224)} &= -4G_{1,3}^- G_{4,6}^- G_{7,5}^- G_{6,2}^- G_{5,1}^- G_{3,7}^- G_{8,4}^- G_{2,8}^- & f^{(4:246)} &= 4G_{1,4}^- G_{6,7}^- G_{3,5}^- G_{5,3}^+ G_{2,6}^- G_{4,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:225)} &= -4G_{1,3}^- G_{3,5}^- G_{7,6}^- G_{5,1}^+ G_{6,2}^- G_{4,7}^- G_{8,4}^- G_{2,8}^- & f^{(4:247)} &= 4G_{2,4}^- G_{5,7}^- G_{4,6}^- G_{6,3}^+ G_{1,5}^- G_{3,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:226)} &= 2G_{1,3}^- G_{4,6}^- G_{8,5}^+ G_{6,2}^- G_{5,1}^+ G_{3,7}^- G_{7,4}^- G_{2,8}^- & f^{(4:248)} &= 4G_{2,4}^- G_{6,7}^- G_{3,6}^- G_{5,3}^+ G_{1,5}^- G_{4,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:227)} &= 2G_{1,3}^- G_{4,5}^- G_{7,6}^- G_{5,1}^+ G_{6,2}^- G_{3,7}^- G_{8,4}^- G_{2,8}^- & f^{(4:249)} &= 4G_{1,3}^- G_{5,7}^- G_{3,5}^- G_{6,4}^- G_{2,6}^- G_{4,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:228)} &= 2G_{1,3}^- G_{3,6}^- G_{7,5}^- G_{6,2}^- G_{5,1}^- G_{4,7}^- G_{8,4}^- G_{2,8}^- & f^{(4:250)} &= -2G_{1,3}^- G_{6,7}^- G_{3,5}^- G_{5,4}^- G_{2,6}^- G_{4,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:229)} &= 8G_{4,2}^+ G_{3,6}^- G_{6,7}^- G_{1,5}^- G_{5,1}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- & f^{(4:251)} &= -2G_{1,3}^- G_{5,7}^- G_{4,5}^- G_{6,4}^- G_{2,6}^- G_{3,8}^- G_{7,1}^+ G_{8,2}^+ \\
f^{(4:230)} &= -4G_{4,2}^+ G_{3,5}^- G_{6,7}^- G_{1,6}^- G_{5,1}^+ G_{7,3}^+ G_{8,4}^- G_{2,8}^- & f^{(4:252)} &= -2G_{2,3}^- G_{5,7}^- G_{3,5}^- G_{6,4}^- G_{1,6}^- G_{4,8}^- G_{7,1}^+ G_{8,2}^+
\end{aligned}$$

$$\begin{aligned}
f^{(4:253)} &= -8G_{4,1}^+ G_{7,5}^+ G_{5,3}^+ G_{6,4}^+ G_{1,6}^- G_{3,7}^- G_{8,2}^+ G_{2,8}^- \\
f^{(4:254)} &= 4G_{3,1}^+ G_{8,6}^+ G_{6,4}^+ G_{5,3}^+ G_{1,5}^- G_{4,7}^- G_{7,2}^+ G_{2,8}^- \\
f^{(4:255)} &= 4G_{3,1}^+ G_{7,5}^+ G_{5,3}^+ G_{6,4}^+ G_{2,6}^- G_{4,8}^- G_{8,2}^+ G_{1,7}^- \\
f^{(4:256)} &= 4G_{4,1}^+ G_{7,5}^+ G_{5,3}^+ G_{6,4}^+ G_{2,6}^- G_{3,7}^- G_{8,2}^+ G_{1,8}^- \\
f^{(4:257)} &= 4G_{3,1}^+ G_{7,5}^+ G_{5,3}^+ G_{6,4}^+ G_{1,6}^- G_{4,7}^- G_{8,2}^+ G_{2,8}^- \\
f^{(4:258)} &= -2G_{4,1}^+ G_{7,5}^+ G_{5,3}^+ G_{6,4}^+ G_{2,6}^- G_{3,8}^- G_{8,2}^+ G_{1,7}^- \\
f^{(4:259)} &= -2G_{4,1}^+ G_{8,6}^+ G_{6,4}^+ G_{5,3}^+ G_{1,5}^- G_{3,7}^- G_{7,2}^+ G_{2,8}^- \\
f^{(4:260)} &= -2G_{3,1}^+ G_{7,5}^+ G_{5,3}^+ G_{6,4}^+ G_{2,6}^- G_{4,7}^- G_{8,2}^+ G_{1,8}^- \\
f^{(4:261)} &= -8G_{1,4}^- G_{5,7}^- G_{3,5}^- G_{4,6}^- G_{6,1}^+ G_{7,3}^+ G_{8,2}^+ G_{2,8}^- \\
f^{(4:262)} &= 4G_{1,3}^- G_{6,7}^- G_{4,6}^- G_{3,5}^- G_{5,1}^+ G_{8,4}^+ G_{7,2}^+ G_{2,8}^- \\
f^{(4:263)} &= 4G_{1,3}^- G_{5,7}^- G_{3,5}^- G_{4,6}^- G_{6,2}^+ G_{8,4}^+ G_{7,1}^+ G_{2,8}^- \\
f^{(4:264)} &= 4G_{2,4}^- G_{5,7}^- G_{3,5}^- G_{4,6}^- G_{6,1}^+ G_{7,3}^+ G_{8,2}^+ G_{1,8}^- \\
f^{(4:265)} &= 4G_{1,3}^- G_{5,7}^- G_{3,5}^- G_{4,6}^- G_{6,1}^+ G_{7,4}^+ G_{8,2}^+ G_{2,8}^- \\
f^{(4:266)} &= -2G_{2,3}^- G_{6,8}^- G_{4,6}^- G_{3,5}^- G_{5,1}^+ G_{7,4}^+ G_{8,2}^+ G_{1,7}^- \\
f^{(4:267)} &= -2G_{1,3}^- G_{5,7}^- G_{3,5}^- G_{4,6}^- G_{6,1}^+ G_{8,4}^+ G_{7,2}^+ G_{2,8}^- \\
f^{(4:268)} &= -2G_{2,4}^- G_{6,7}^- G_{4,6}^- G_{3,5}^- G_{5,1}^+ G_{7,3}^+ G_{8,2}^+ G_{1,8}^- \\
f^{(4:269)} &= 8G_{3,1}^+ G_{6,8}^- G_{5,4}^+ G_{4,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ G_{1,7}^- \\
f^{(4:270)} &= 8G_{4,1}^+ G_{6,7}^- G_{5,3}^+ G_{3,5}^- G_{1,6}^- G_{7,4}^+ G_{8,2}^+ G_{2,8}^- \\
f^{(4:271)} &= 8G_{4,1}^+ G_{5,7}^- G_{6,4}^+ G_{3,5}^- G_{1,6}^- G_{7,3}^+ G_{8,2}^+ G_{2,8}^- \\
f^{(4:272)} &= -4G_{4,1}^+ G_{6,7}^- G_{5,3}^+ G_{3,5}^- G_{1,6}^- G_{8,4}^+ G_{7,2}^+ G_{2,8}^- \\
f^{(4:273)} &= -4G_{3,1}^+ G_{6,8}^- G_{5,3}^+ G_{4,5}^- G_{2,6}^- G_{7,4}^+ G_{8,2}^+ G_{1,7}^- \\
f^{(4:274)} &= -4G_{4,2}^+ G_{5,7}^- G_{6,3}^+ G_{3,5}^- G_{1,6}^- G_{8,4}^+ G_{7,1}^+ G_{2,8}^- \\
f^{(4:275)} &= -4G_{3,1}^+ G_{6,7}^- G_{5,3}^+ G_{4,6}^- G_{1,5}^- G_{8,4}^+ G_{7,2}^+ G_{2,8}^- \\
f^{(4:276)} &= -4G_{3,1}^+ G_{5,7}^- G_{6,3}^+ G_{4,6}^- G_{1,5}^- G_{7,4}^+ G_{8,2}^+ G_{2,8}^- \\
f^{(4:277)} &= -4G_{3,1}^+ G_{5,7}^- G_{6,4}^+ G_{4,5}^- G_{1,6}^- G_{7,3}^+ G_{8,2}^+ G_{2,8}^- \\
f^{(4:278)} &= -4G_{4,1}^+ G_{5,7}^- G_{6,4}^+ G_{3,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ G_{1,8}^- \\
f^{(4:279)} &= -4G_{3,2}^+ G_{5,7}^- G_{6,4}^+ G_{4,6}^- G_{1,5}^- G_{7,3}^+ G_{8,1}^+ G_{2,8}^- \\
f^{(4:280)} &= 2G_{3,1}^+ G_{5,7}^- G_{6,3}^+ G_{4,6}^- G_{1,5}^- G_{8,4}^+ G_{7,2}^+ G_{2,8}^- \\
f^{(4:281)} &= 2G_{4,1}^+ G_{5,7}^- G_{6,3}^+ G_{3,5}^- G_{1,6}^- G_{8,4}^+ G_{7,2}^+ G_{2,8}^- \\
f^{(4:282)} &= 2G_{3,1}^+ G_{6,8}^- G_{5,3}^+ G_{4,6}^- G_{2,5}^- G_{7,4}^+ G_{8,2}^+ G_{1,7}^- \\
f^{(4:283)} &= 2G_{4,1}^+ G_{6,7}^- G_{5,4}^+ G_{3,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ G_{1,8}^- \\
f^{(4:284)} &= 2G_{3,1}^+ G_{5,7}^- G_{6,4}^+ G_{4,5}^- G_{2,6}^- G_{7,3}^+ G_{8,2}^+ G_{1,8}^- \\
f^{(4:285)} &= 8G_{1,3}^- G_{8,6}^+ G_{5,4}^+ G_{4,5}^- G_{6,2}^- G_{3,7}^+ G_{7,1}^+ G_{2,8}^- \\
f^{(4:286)} &= 8G_{1,4}^- G_{7,6}^+ G_{5,3}^+ G_{3,5}^- G_{6,1}^+ G_{4,7}^- G_{8,2}^+ G_{2,8}^- \\
f^{(4:287)} &= 8G_{1,3}^- G_{7,6}^+ G_{6,4}^+ G_{3,5}^- G_{5,1}^+ G_{4,7}^- G_{8,2}^+ G_{2,8}^- \\
f^{(4:288)} &= -4G_{1,3}^- G_{8,5}^+ G_{6,4}^+ G_{4,6}^- G_{5,1}^+ G_{3,7}^- G_{7,2}^+ G_{2,8}^- \\
f^{(4:289)} &= -4G_{2,3}^- G_{7,5}^+ G_{6,4}^+ G_{3,6}^- G_{5,1}^+ G_{4,8}^- G_{8,2}^+ G_{1,7}^- \\
f^{(4:290)} &= -4G_{2,4}^- G_{7,5}^+ G_{5,3}^+ G_{3,6}^- G_{6,1}^+ G_{4,8}^- G_{8,2}^+ G_{1,7}^- \\
f^{(4:291)} &= -4G_{1,3}^- G_{8,6}^+ G_{6,4}^+ G_{3,5}^- G_{5,1}^+ G_{4,7}^- G_{7,2}^+ G_{2,8}^- \\
f^{(4:292)} &= -4G_{1,3}^- G_{7,5}^+ G_{6,4}^+ G_{3,6}^- G_{5,1}^+ G_{4,7}^- G_{8,2}^+ G_{2,8}^- \\
f^{(4:293)} &= -4G_{1,3}^- G_{7,5}^+ G_{5,4}^+ G_{4,6}^- G_{6,1}^+ G_{3,7}^- G_{8,2}^+ G_{2,8}^- \\
f^{(4:294)} &= -4G_{2,4}^- G_{7,5}^+ G_{5,3}^+ G_{4,6}^- G_{6,1}^+ G_{3,7}^- G_{8,2}^+ G_{1,8}^- \\
f^{(4:295)} &= -4G_{2,3}^- G_{7,5}^+ G_{6,4}^+ G_{4,6}^- G_{5,1}^+ G_{3,7}^- G_{8,2}^+ G_{1,8}^- \\
f^{(4:296)} &= 2G_{1,4}^- G_{8,6}^+ G_{5,3}^+ G_{4,5}^- G_{6,1}^+ G_{3,7}^- G_{7,2}^+ G_{2,8}^-
\end{aligned}$$

$$f^{(4:297)} = 2G_{1,3}^- G_{8,6}^+ G_{6,4}^+ G_{4,5}^- G_{5,1}^+ G_{3,7}^- G_{7,2}^+ G_{2,8}^-$$

$$f^{(4:299)} = 2G_{2,4}^- G_{7,5}^+ G_{6,3}^+ G_{4,6}^- G_{5,1}^+ G_{3,7}^- G_{8,2}^+ G_{1,8}^-$$

$$f^{(4:298)} = 2G_{2,3}^- G_{7,6}^+ G_{6,4}^+ G_{3,5}^- G_{5,1}^+ G_{4,8}^- G_{8,2}^+ G_{1,7}^-$$

$$f^{(4:300)} = 2G_{2,4}^- G_{7,5}^+ G_{5,3}^+ G_{3,6}^- G_{6,1}^+ G_{4,7}^- G_{8,2}^+ G_{1,8}^-$$

Appendix I

Integrands for MC-GF3

$$\bar{f}_{pq}^{(3:1)} = -G_{2,6}^+ G_{4,5}^+ G_{3,2}^- G_{5,1}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:17)} = -2G_{1,5}^+ G_{2,3}^+ G_{4,6}^+ G_{5,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:2)} = G_{2,4}^+ G_{3,6}^+ G_{4,5}^+ G_{5,1}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:18)} = -2G_{1,4}^+ G_{2,6}^+ G_{4,5}^+ G_{5,3}^- G_{6,2}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:3)} = -G_{2,5}^+ G_{3,6}^+ G_{4,2}^- G_{5,1}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:19)} = -2G_{1,3}^+ G_{2,6}^+ G_{4,5}^+ G_{5,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:4)} = -G_{2,6}^+ G_{3,5}^+ G_{4,1}^- G_{5,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:20)} = 4G_{1,3}^+ G_{2,6}^+ G_{4,5}^+ G_{5,4}^- G_{6,2}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:5)} = 2G_{2,6}^+ G_{3,5}^+ G_{4,2}^- G_{5,1}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:21)} = -G_{2,3}^+ G_{4,6}^+ G_{3,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:6)} = -2G_{2,4}^+ G_{3,5}^+ G_{4,6}^+ G_{5,1}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:22)} = G_{2,6}^+ G_{3,1}^- G_{4,2}^- G_{5,4}^- G_{6,3}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:7)} = 2G_{2,5}^+ G_{4,6}^+ G_{3,2}^- G_{5,1}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:23)} = -G_{2,4}^+ G_{4,6}^+ G_{3,1}^- G_{5,2}^- G_{6,3}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:8)} = 2G_{2,6}^+ G_{3,5}^+ G_{4,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:24)} = -G_{2,4}^+ G_{4,6}^+ G_{3,2}^- G_{5,3}^- G_{6,1}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:9)} = 2G_{2,6}^+ G_{4,5}^+ G_{3,1}^- G_{5,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:25)} = -2G_{2,6}^+ G_{3,1}^- G_{4,2}^- G_{5,3}^- G_{6,4}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:10)} = -4G_{2,6}^+ G_{4,5}^+ G_{3,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_1) e^{-\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:26)} = 2G_{2,4}^+ G_{4,6}^+ G_{3,1}^- G_{5,3}^- G_{6,2}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:11)} = G_{1,5}^+ G_{2,3}^+ G_{4,6}^+ G_{5,4}^- G_{6,2}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:27)} = 2G_{2,3}^+ G_{4,6}^+ G_{3,1}^- G_{5,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:12)} = -G_{1,5}^+ G_{2,6}^+ G_{4,2}^- G_{5,4}^- G_{6,3}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:28)} = 2G_{2,4}^+ G_{3,6}^+ G_{4,2}^- G_{5,3}^- G_{6,1}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:13)} = G_{1,5}^+ G_{2,4}^+ G_{4,6}^+ G_{5,2}^- G_{6,3}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:29)} = 2G_{2,4}^+ G_{4,6}^+ G_{3,2}^- G_{5,1}^- G_{6,3}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:14)} = G_{1,4}^+ G_{2,5}^+ G_{4,6}^+ G_{5,3}^- G_{6,2}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:30)} = -4G_{2,4}^+ G_{3,6}^+ G_{4,2}^- G_{5,1}^- G_{6,3}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_1) e^{-\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:15)} = -2G_{1,5}^+ G_{2,4}^+ G_{4,6}^+ G_{5,3}^- G_{6,2}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:31)} = G_{1,3}^+ G_{2,6}^+ G_{4,5}^+ G_{3,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_5) e^{\omega(\tau_1+\tau_2)}$$

$$\bar{f}_{pq}^{(3:16)} = 2G_{1,5}^+ G_{2,6}^+ G_{4,2}^- G_{5,3}^- G_{6,4}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_3) e^{\omega\tau_2}$$

$$\bar{f}_{pq}^{(3:32)} = -G_{1,3}^+ G_{2,4}^+ G_{3,6}^+ G_{4,5}^- G_{6,2}^- \phi_p^*(\mathbf{r}_1) \phi_q(\mathbf{r}_5) e^{\omega(\tau_1+\tau_2)}$$

$$\begin{aligned}
\bar{f}_{pq}^{(3:33)} &= G_{1,3}^+ G_{2,5}^+ G_{3,6}^+ G_{4,2}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:55)} &= -2G_{1,5}^+ G_{2,4}^+ G_{4,6}^+ G_{3,1}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} \\
\bar{f}_{pq}^{(3:34)} &= G_{1,6}^+ G_{2,3}^+ G_{3,5}^+ G_{4,2}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:56)} &= 2G_{1,5}^+ G_{2,6}^+ G_{3,1}^- G_{4,2}^- G_{6,4}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} \\
\bar{f}_{pq}^{(3:35)} &= 2G_{1,3}^+ G_{2,4}^+ G_{3,5}^+ G_{4,6}^- G_{6,2}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:57)} &= -2G_{1,5}^+ G_{2,4}^+ G_{3,6}^+ G_{4,2}^- G_{6,1}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} \\
\bar{f}_{pq}^{(3:36)} &= -2G_{1,3}^+ G_{2,6}^+ G_{3,5}^+ G_{4,2}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:58)} &= -2G_{1,4}^+ G_{2,6}^+ G_{4,5}^+ G_{3,1}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} \\
\bar{f}_{pq}^{(3:37)} &= -2G_{1,3}^+ G_{2,5}^+ G_{4,6}^+ G_{3,2}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:59)} &= -2G_{1,4}^+ G_{2,6}^+ G_{3,5}^+ G_{4,2}^- G_{6,1}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} \\
\bar{f}_{pq}^{(3:38)} &= -2G_{1,6}^+ G_{2,4}^+ G_{3,5}^+ G_{4,2}^- G_{6,3}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:60)} &= 4G_{1,4}^+ G_{2,6}^+ G_{3,5}^+ G_{4,1}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} \\
\bar{f}_{pq}^{(3:39)} &= -2G_{1,5}^+ G_{2,4}^+ G_{4,6}^+ G_{3,2}^- G_{6,3}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:61)} &= 2G_{2,6}^+ G_{3,5}^+ G_{4,2}^- G_{5,4}^- G_{6,3}^- \phi_p^*(r_1) \phi_q(r_1) \\
\bar{f}_{pq}^{(3:40)} &= 4G_{1,5}^+ G_{2,4}^+ G_{3,6}^+ G_{4,2}^- G_{6,3}^- \phi_p^*(r_1) \phi_q(r_5) e^{\omega(\tau_1+\tau_2)} & \bar{f}_{pq}^{(3:62)} &= -2G_{2,4}^+ G_{3,6}^+ G_{4,5}^+ G_{5,3}^- G_{6,2}^- \phi_p^*(r_1) \phi_q(r_1) \\
\bar{f}_{pq}^{(3:41)} &= -G_{1,4}^+ G_{2,6}^+ G_{4,2}^- G_{5,1}^- G_{6,3}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:63)} &= -4G_{2,6}^+ G_{3,5}^+ G_{4,2}^- G_{5,3}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_1) \\
\bar{f}_{pq}^{(3:42)} &= G_{1,3}^+ G_{2,4}^+ G_{4,6}^+ G_{5,2}^- G_{6,1}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:64)} &= 4G_{2,4}^+ G_{3,5}^+ G_{4,6}^+ G_{5,3}^- G_{6,2}^- \phi_p^*(r_1) \phi_q(r_1) \\
\bar{f}_{pq}^{(3:43)} &= -G_{1,3}^+ G_{2,6}^+ G_{4,2}^- G_{5,4}^- G_{6,1}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:65)} &= -2G_{1,4}^+ G_{2,6}^+ G_{4,5}^+ G_{5,2}^- G_{6,1}^- \phi_p^*(r_3) \phi_q(r_3) \\
\bar{f}_{pq}^{(3:44)} &= -G_{1,6}^+ G_{2,3}^+ G_{4,2}^- G_{5,1}^- G_{6,4}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:66)} &= 2G_{1,6}^+ G_{2,5}^+ G_{4,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_3) \\
\bar{f}_{pq}^{(3:45)} &= 2G_{1,3}^+ G_{2,6}^+ G_{4,2}^- G_{5,1}^- G_{6,4}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:67)} &= -4G_{1,5}^+ G_{2,6}^+ G_{4,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_3) \\
\bar{f}_{pq}^{(3:46)} &= -2G_{1,3}^+ G_{2,4}^+ G_{4,6}^+ G_{5,1}^- G_{6,2}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:68)} &= 4G_{1,4}^+ G_{2,6}^+ G_{4,5}^+ G_{5,1}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_3) \\
\bar{f}_{pq}^{(3:47)} &= 2G_{1,6}^+ G_{2,4}^+ G_{4,2}^- G_{5,1}^- G_{6,3}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:69)} &= 2G_{1,3}^+ G_{2,6}^+ G_{3,2}^- G_{4,1}^- G_{6,4}^- \phi_p^*(r_5) \phi_q(r_5) \\
\bar{f}_{pq}^{(3:48)} &= 2G_{1,3}^+ G_{2,6}^+ G_{4,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:70)} &= -2G_{1,4}^+ G_{2,3}^+ G_{4,6}^+ G_{3,1}^- G_{6,2}^- \phi_p^*(r_5) \phi_q(r_5) \\
\bar{f}_{pq}^{(3:49)} &= 2G_{1,6}^+ G_{2,4}^+ G_{4,1}^- G_{5,3}^- G_{6,2}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:71)} &= -4G_{1,3}^+ G_{2,6}^+ G_{3,1}^- G_{4,2}^- G_{6,4}^- \phi_p^*(r_5) \phi_q(r_5) \\
\bar{f}_{pq}^{(3:50)} &= -4G_{1,4}^+ G_{2,6}^+ G_{4,1}^- G_{5,3}^- G_{6,2}^- \phi_p^*(r_5) \phi_q(r_3) e^{-\omega\tau_1} & \bar{f}_{pq}^{(3:72)} &= 4G_{1,3}^+ G_{2,4}^+ G_{4,6}^+ G_{3,1}^- G_{6,2}^- \phi_p^*(r_5) \phi_q(r_5) \\
\bar{f}_{pq}^{(3:51)} &= G_{1,5}^+ G_{2,4}^+ G_{3,6}^+ G_{4,1}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} & \bar{f}_{pq}^{(3:73)} &= -G_{1,5}^+ G_{3,6}^+ G_{4,2}^- G_{5,3}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_2) \\
\bar{f}_{pq}^{(3:52)} &= -G_{1,6}^+ G_{2,5}^+ G_{3,1}^- G_{4,2}^- G_{6,4}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} & \bar{f}_{pq}^{(3:74)} &= G_{1,3}^+ G_{3,5}^+ G_{4,6}^+ G_{5,4}^- G_{6,2}^- \phi_p^*(r_1) \phi_q(r_2) \\
\bar{f}_{pq}^{(3:53)} &= G_{1,6}^+ G_{2,4}^+ G_{4,5}^+ G_{3,1}^- G_{6,2}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} & \bar{f}_{pq}^{(3:75)} &= 2G_{1,5}^+ G_{4,6}^+ G_{3,2}^- G_{5,3}^- G_{6,4}^- \phi_p^*(r_1) \phi_q(r_2) \\
\bar{f}_{pq}^{(3:54)} &= G_{1,5}^+ G_{2,4}^+ G_{4,6}^+ G_{3,2}^- G_{6,1}^- \phi_p^*(r_3) \phi_q(r_5) e^{\omega\tau_1} & \bar{f}_{pq}^{(3:76)} &= -2G_{1,4}^+ G_{3,5}^+ G_{4,6}^+ G_{5,3}^- G_{6,2}^- \phi_p^*(r_1) \phi_q(r_2)
\end{aligned}$$

$$\bar{f}_{pq}^{(3:77)} = -G_{1,5}^+ G_{2,6}^+ G_{3,1}^- G_{5,2}^- G_{6,4}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_4)$$

$$\bar{f}_{pq}^{(3:81)} = G_{1,3}^+ G_{2,4}^+ G_{4,6}^+ G_{3,2}^- G_{5,1}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_6)$$

$$\bar{f}_{pq}^{(3:78)} = G_{1,6}^+ G_{2,4}^+ G_{3,5}^+ G_{5,1}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_4)$$

$$\bar{f}_{pq}^{(3:82)} = -G_{1,4}^+ G_{2,6}^+ G_{3,1}^- G_{4,2}^- G_{5,3}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_6)$$

$$\bar{f}_{pq}^{(3:79)} = 2G_{1,5}^+ G_{2,6}^+ G_{3,1}^- G_{5,4}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_4)$$

$$\bar{f}_{pq}^{(3:83)} = -2G_{1,3}^+ G_{2,4}^+ G_{3,6}^+ G_{4,2}^- G_{5,1}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_6)$$

$$\bar{f}_{pq}^{(3:80)} = -2G_{1,4}^+ G_{2,6}^+ G_{3,5}^+ G_{5,1}^- G_{6,2}^- \phi_p^*(\mathbf{r}_3) \phi_q(\mathbf{r}_4)$$

$$\bar{f}_{pq}^{(3:84)} = 2G_{1,3}^+ G_{2,6}^+ G_{3,1}^- G_{4,2}^- G_{5,4}^- \phi_p^*(\mathbf{r}_5) \phi_q(\mathbf{r}_6)$$

Appendix J

Diagrams for MC-MP4

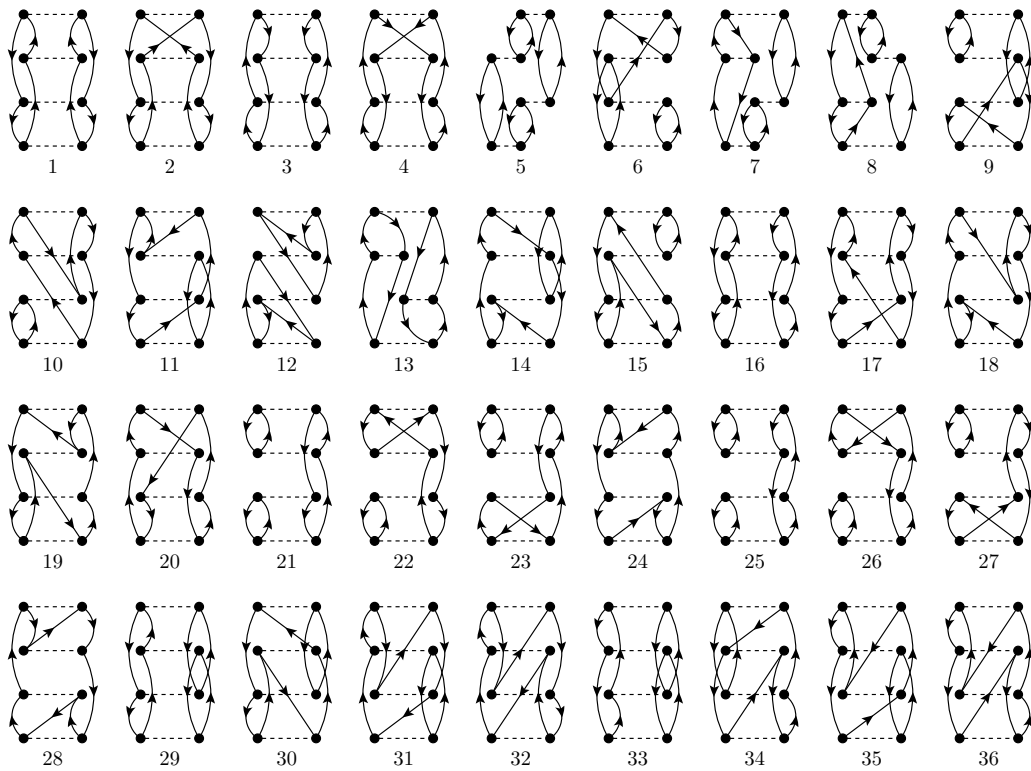


Figure J.1: Thirty-six of the Feynman–Goldstone diagrams defining the fourth-order MPBT correlation energy. The diagrams correspond to the type A diagrams discussed in Section 2.6.

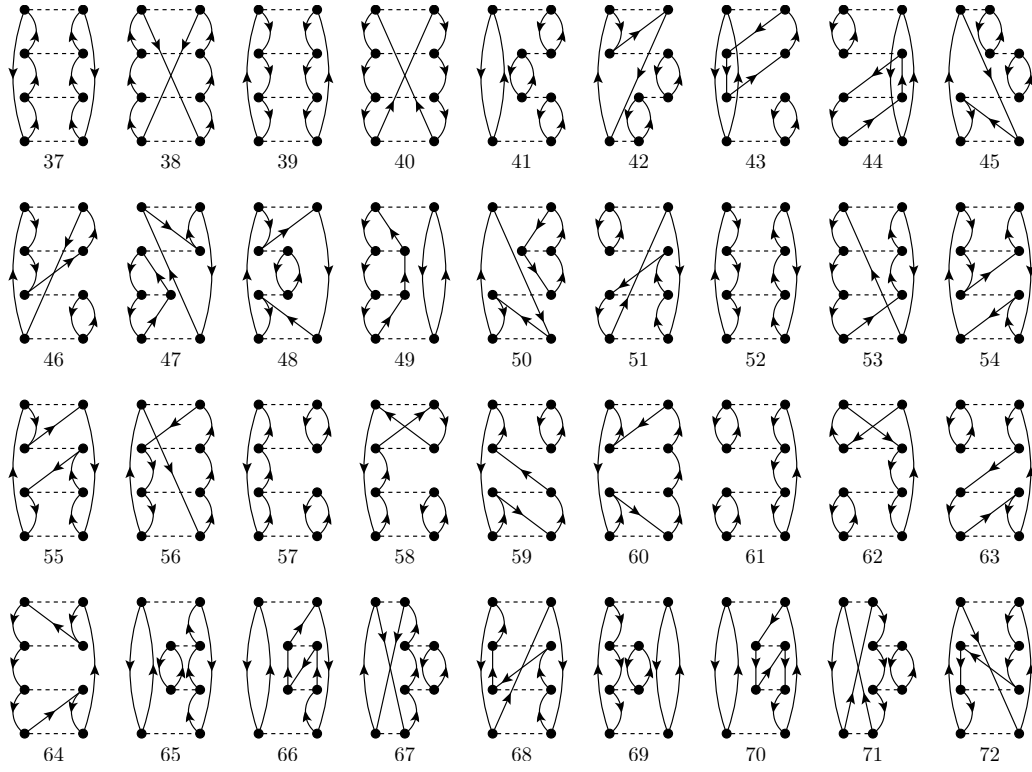


Figure J.2: The same as Figure J.1 but for type B diagrams.

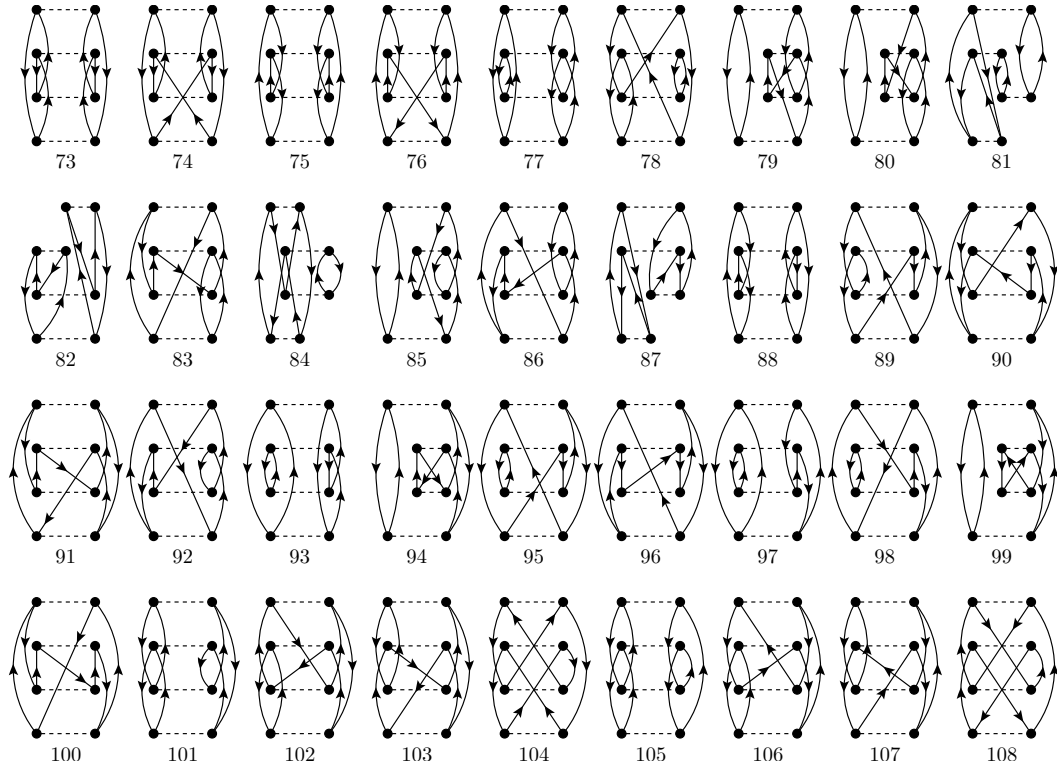


Figure J.3: The same as Figure J.1 but for type C diagrams.

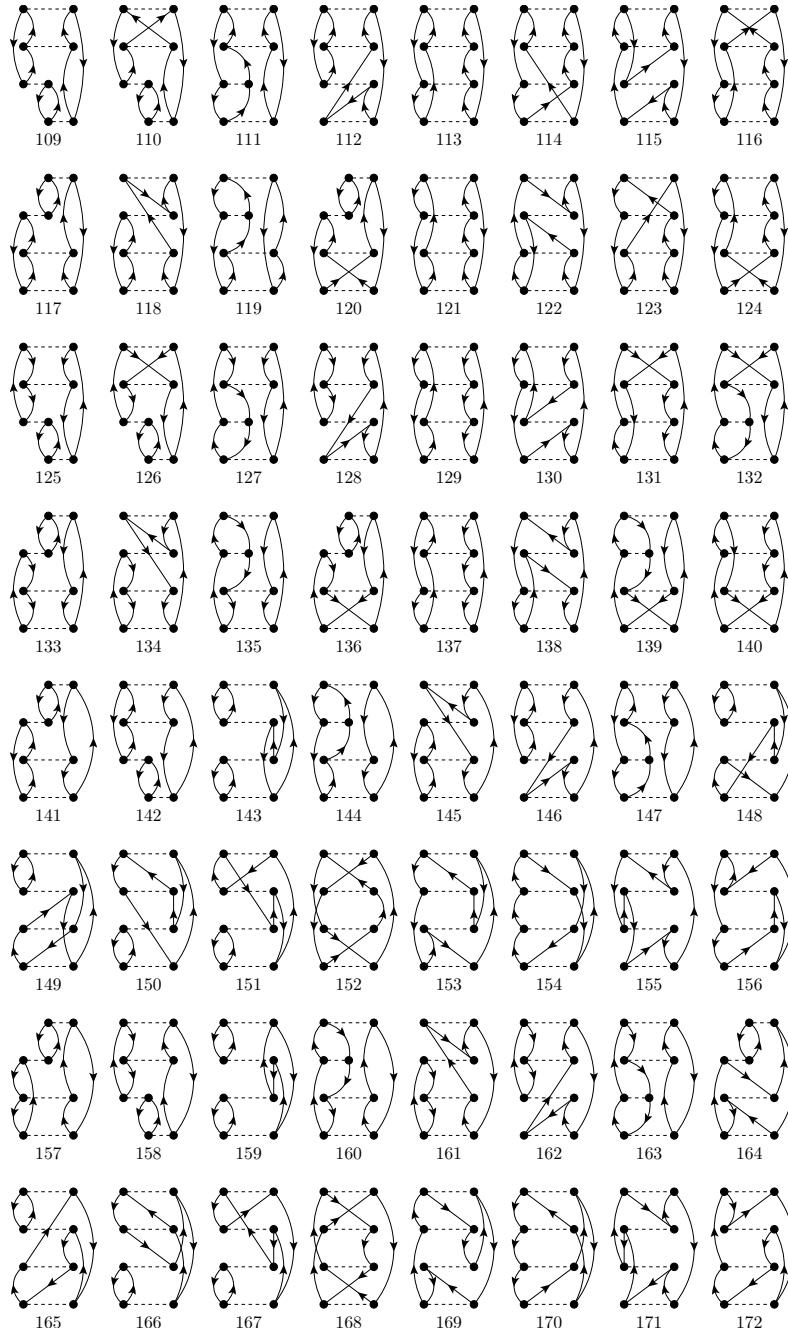


Figure J.4: The same as Figure J.1 but for the first third of the type D diagrams.

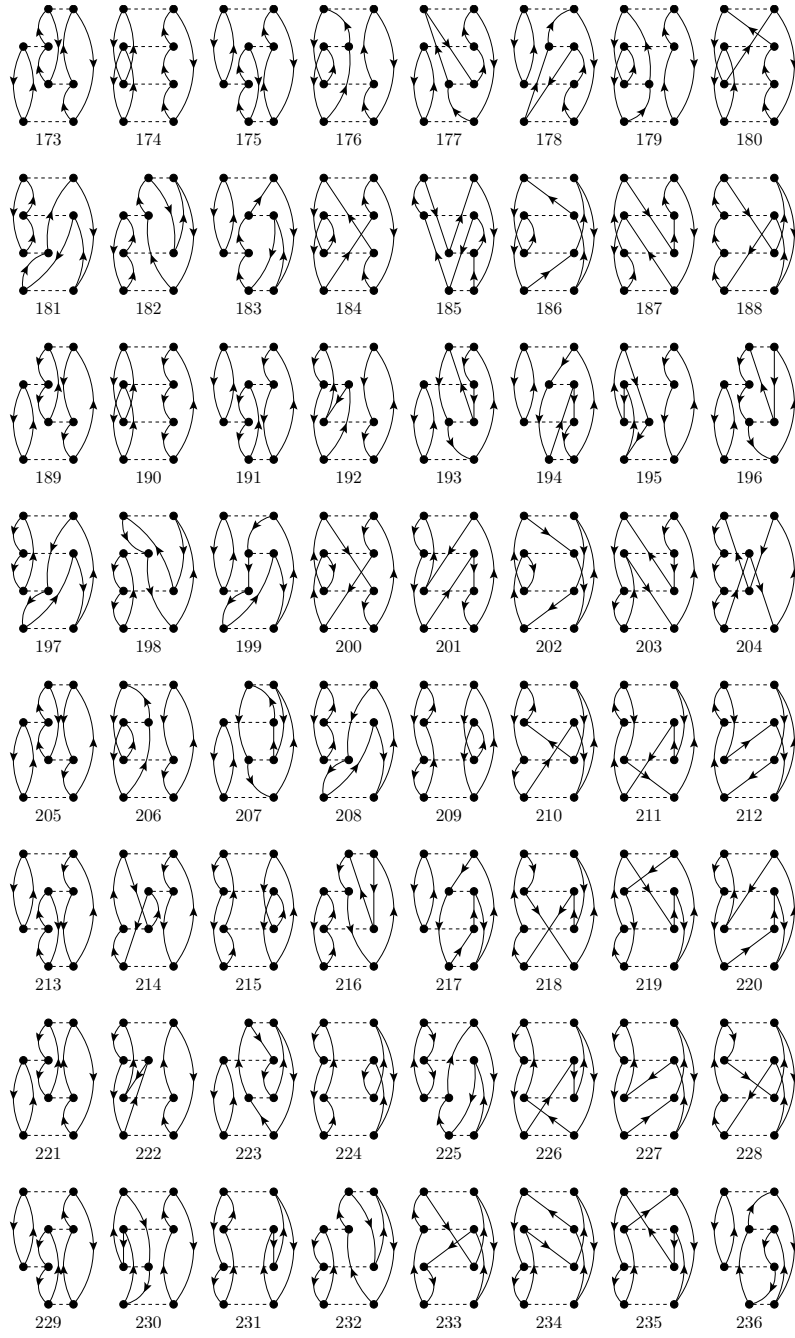


Figure J.5: The same as Figure J.1 but for the second third of the type D diagrams.

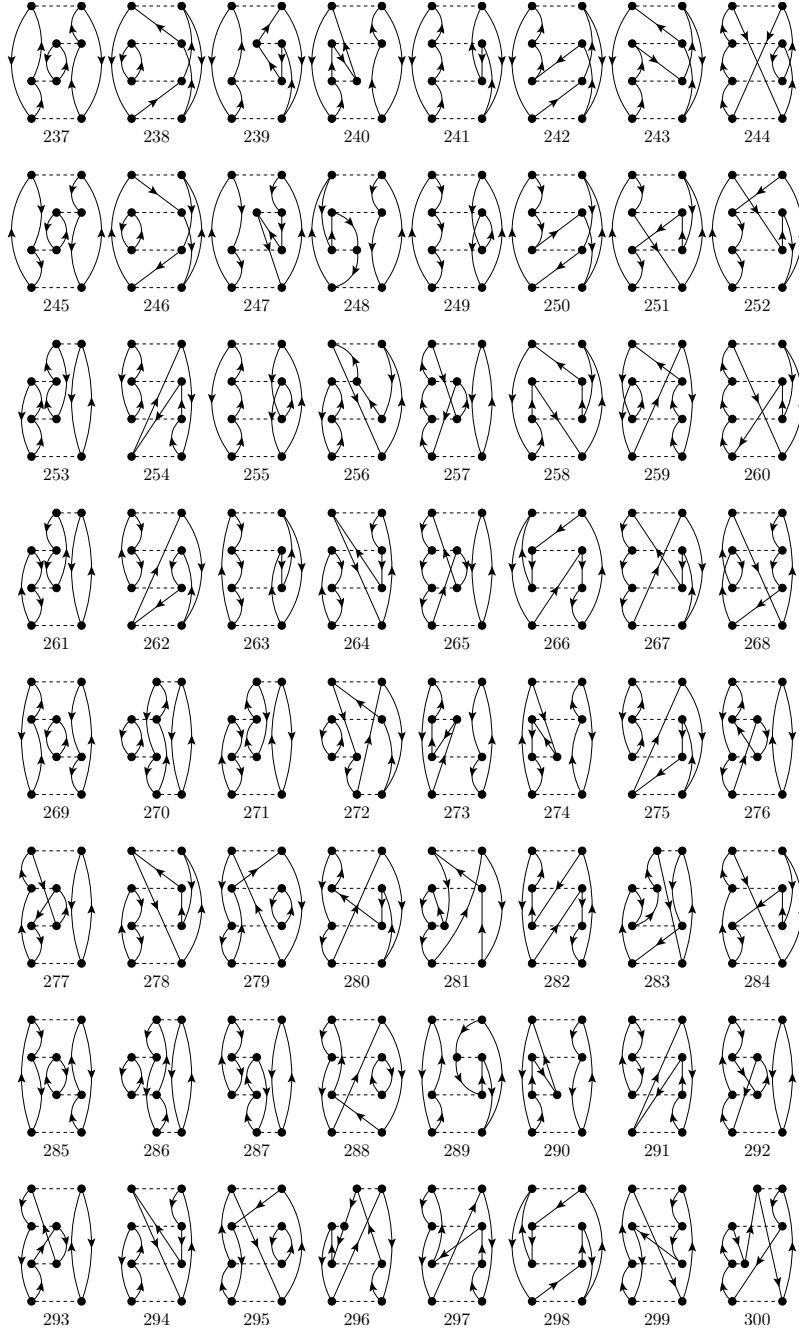


Figure J.6: The same as Figure J.1 but for the last third of the type D diagrams.