The Application of Genetic Algorithms For Density Functional Optimization and Development

by

Matthew Yu-Wei Wang

A dissertation submitted in partial fulfilment of the requirements of the degree of Doctor of Philosophy in Computational Science

Middle Tennessee State University
December, 2018

Dissertation Committee:
Dr. Jing Kong, Chair
Dr. John Wallin
Dr. Anatoliy Volkov

ABSTRACT

This work details the development of the density functional theory (DFT) implementation for nonadditive three-body dispersion using the exchange dipole moment (XDM) and the quantum chemistry functional KP16/B13 to calculate self-consistent field energy of molecular systems through the application of genetic algorithms. In the case of dispersion, current *ab initio* methods are accurate but computationally expensive. The development of three-body dispersion detailed here involves a density functional model approach resulting in comparable performance and advantages in computational efficiency. The KP16/B13 functional is developed to advance the implementation of a single reference functional that addresses nondynamic/strong correlation and for use as a general purpose functional. This work details the optimization of the two models with selected sets of atoms and molecules and benchmarking KP16/B13 with the Minnesota sets involving a variety of chemical properties. Both parts of this work compare results against contemporary methods demonstrating improved performance for some properties and comparable results in others. The calibration and optimization of the methods detailed above are my main contribution. Software development to achieve this goal resulted in a general purpose genetic algorithm code stack. The software developed also facilitated the organization of results and computation of the methods on computer clusters at MTSU and supercomputers at Oak Ridge National Laboratory. Through multiple iterations, refactoring, and design input from colleagues, advisors, and users, the final software stack is robust and will continue to be leveraged by members of Dr. Kong's group in future research.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

HF: Hatree-Fock.
DFT: Density functional theory.
XDM: Exchange dipole moment.
KP16/B13: Kong/Proynov 2016, Becke 2013 functional.
SCF: Self consistent field procedure.
GA_base: Genetic algorithm base.
VDW: Van-der-Waals interactions, also known as dispersion.
CCSD(T): Coupled-cluster with singles, doubles, and perturbative triple excitations [2].
SAPT: Symmetry-adapted perturbation theory [48].
MP2: Møller-Plesset perturbation theory, second order.
XC: Exchange-correlation.
PBE: Pure functional from 1996 from Perdew, Burke, and Ernzerhof [17].
P86: Functional with non-local correlation from Perdew 86 [17].
M06: Hybrid functional from Truhlar and Zhao[57].
M06-2X: Variation of M06 from Zhao[57].
MAD: Mean absolute deviation.
FCI: Full configuration interaction.
B05: Becke's B05 method [35, 30].
B13: Becke's B13 method [30].
PSTS: Perdew-Staroverov-Tao-Scuseria functional. [35]
B3LYP: Becke three-parameter functional with Lee-Yang-Parr expression for non-local correlation [33, 17].
PMF: Present model functional.
HF-HGPBE: runge-3.5 functional.
B3tLap: B3LYP-like hybrid exchange based on Becke's B88 generalized gradient approximation (GGA) and the tLap (also known as PK06) meta-GGA correlation [42]. **SCAN**: Strongly constrained and appropriately normed meta-GGA functional [46].

1 Introduction

The title of this dissertation aptly sums up the theme of this work: the application of genetic algorithms for density functional optimization and development. Before getting into the main bodys of work, there is an introduction to the concepts presented with some brief background into density functional theory and genetic algorithms. The first two sections are on the development of DFT methods. First are the damping schemes for dispersion and second is the KP16/B13 functional. The last chapter continues research of the KP16/B13 with establishing the optimized KP16/B13 as a general purpose functional through benchmarking a variety of chemical systems. These studies used my organizational framework for my genetic algorithm implementation. The developed software efficiently utilized multiple high performance resources to maximize computation throughput. It is my hope that this work adequately demonstrates the tenets of computational science: advancement of research in a scientific domain, software development and application, mathematical theory, and effective use of high performance computing.

1.1 Brief Outline of Density Functional Theory

The explanation of DFT first begins with quantum chemistry's solution to the Schrödinger equation [29, p. 3]. The Schrödinger equation, proposed by Schrödinger in 1926, describes the discrete energy levels of a quantum system [20, p. 132]. The equation is proposed based on the hyperbolic partial differential wave equation and experiments involving electron's light wave properties[20, p. 132]:

$$\hat{H}\Psi = \hat{E}\Psi \tag{1}$$

$\hat{H}$ is the Hamiltonian operator, Ψ is the wave function, and $\hat{E}$ is the energy. The general equation can be cast into two different forms, the time-dependent and time-independent Schrödinger equations as shown below:

$$\hat{H}\Psi = i\hbar\frac{\partial\Psi}{\partial t} \tag{2}$$

$$\hat{H}\Psi = E\Psi \tag{3}$$

The first equation includes the partial derivative of the wave function with respect to time, denoting it as the time-dependent Schrödinger equation. For the time-independent equation, also known as the stationary Schrödinger equation, energy and/or Ψ no longer changes over time. For the purposes of this work, the focus is the time-independent Schrödinger equation due to the state of the considered atomic or molecular system remaining constant over time. The Hamiltonian operator, $\hat{H}$, in atomic units is of the form

$$\hat{H} = -\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{A=1}^{M}\frac{1}{2M_A}\nabla_A^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}} + \sum_{A=1}^{M}\sum_{B>A}^{M}\frac{Z_A Z_B}{R_{AB}} \tag{4}$$

The terms are as follows: N is the number of electrons, M is the number of nuclei, M_A is the ratio of the mass of nucleus A to the mass of an electron, Z_A is the atomic number of nucleus A, r_{iA} is the distance between nucleus A and electron i, r_{ij} is the distance between electron i and j, and R_{AB} is the distance between nucleus A and nucleus B. The differentiation operators ∇_i^2 and ∇_A^2 are with respect to the coordinates of the ith electron and Ath nucleus, respectively. The first and second terms of the Hamiltonian are the kinetic energy operators for the electrons and nuclei respectively. The third term is the Coulomb attraction between electron i and nucleus A. The fourth term is one Coulomb repulsion term between electrons i and j and the fifth is the repulsion between nucleus A and nucleus B.

For the purposes of the work, the focus is on the reduced electronic Hamiltonian resulting from the Born-Oppenheimer approximation [29, p. 5]. The Born-Oppenheimer approximation states that the kinetic energy and Coulomb repulsion terms of nuclei can be neglected due to their heavy atomic weight and low velocity in comparison to electrons. Repulsion between nuclei is effectively a constant while the kinetic energy is approximately zero[29, p. 5][47, p. 43]. The resulting term is called the electronic Hamiltonian:

$$\hat{H}_{elec} = -\sum_{i=1}^{N} \frac{1}{2}\nabla_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{M} \frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N} \frac{1}{r_{ij}} \tag{5}$$

Outside of simple atoms such as H, He$^+$, and other one-electron systems, the electronic Schrödinger equation cannot be solved analytically for many-electron systems due to the Coulomb pair potential energy term. Therefore, approximate wave functions are created. This begs the question, how does one know the best approximate solution. The variational principle states that any approximate solution to the equation of this eigenvalue form will have higher energy than the exact solution of Schrödinger equation[47, p. 32]. Consequently, the wave function that minimizes the energy is the best approximate trial solution to the Schrödinger equation. One of the first advances to solving the approximate wave function for the electronic Schrödinger equation of many electrons is the Hartree-Fock approximation[47, p. 53]. A principal assumption of the Hartree-Fock approximation is that electrons occupy spatial orbitals and have either spin up or spin down state[29, p. 9]. To resolve the complicated electron-electron repulsion of a many electron system, the wave function, Ψ, is represented in a general form as a product of single electron spin-orbital functions, ϕ_i [29, p. 9]. The spin-orbital function is made up of a spatial (orbital) and a spin component. Each electron experiences repulsion from all other electrons in an average potential field [29, p. 11][47, p. 54] . To satisfy the antisymmetry principle, which states that interchanging any two electrons changes the sign of the wave function, the Slater determinant is used to represent Ψ[47, p. 45-53]. The Slater determinant also satisfies the Pauli Exclusion Principle where two electrons of the same spin cannot occupy the same orbital[47, p. 45-53]. If two rows or columns of the Slater determinant are identical, then the result of the determinant is zero and the Pauli principle is fulfilled.

$$\begin{aligned}
\Psi(\vec{r}_1, \vec{r}_2, \cdots, \vec{r}_N) &= \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\vec{r}_1) & \psi_1(\vec{r}_2) & \cdots & \psi_1(\vec{r}_N) \\ \psi_2(\vec{r}_1) & \psi_2(\vec{r}_2) & \cdots & \psi_2(\vec{r}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_N(\vec{r}_1) & \psi_N(\vec{r}_2) & \cdots & \psi_N(\vec{r}_N) \end{vmatrix} && (6) \\
&= \frac{1}{\sqrt{N!}} \det |\psi_1(\vec{r}_1)\psi_2(\vec{r}_2)\cdots\psi_N(\vec{r}_N)| && (7)
\end{aligned}$$

Here ψ_i is the molecular orbital i and $\vec{r}_j$ is the coordinate of electron j. Molecular orbitals are made up of a linear combination of atomic orbitals, denoted as ϕ, abbreviated as MO LCAO method[47, p. 56].

Two of the main challenges of the Hartree-Fock approximation is to correctly choose functions for ϕ and how to determine whether the calculated energy is the correct energy of the system. For the issue of the calculated energy, the variational principle states that the energy of the approximate solution is greater than or equal to the ground state energy of the atom or molecule[47, p. 32]. Therefore, the wave function, Ψ, that minimizes the energy is the approximate solution to the Schrödinger equation. As for what functions to use for ϕ, two main options are the Slater type orbital[47, p. 56] and the Gaussian type orbital[47, p. 56]. The general Slater type orbital in the spherical coordinate system has the form:

$$\phi(r,\theta,\phi) = N_{\alpha,n} r^{n-1} e^{-\alpha r} Y_l^m(\theta,\phi) \tag{8}$$

$$N_{\alpha,n} = \frac{(2\alpha)^{n+\frac{1}{2}}}{[(2n)!]^{\frac{1}{2}}} \tag{9}$$

For the Slater type orbital, α is the exponential parameter, n is the principle quantum number, $N_{\alpha,n}$ is the normalizing factor, and $Y_l^m(\theta,\phi)$ is the angular part represented by the spherical harmonic function[29, p. 100-101]. Within the spherical harmonic, l is the orbital angular quantum number and m is the magnetic quantum number. The general Gaussian type orbital in Cartesian coordinates reads as:

$$\phi(x,y,z) = g(\vec{r}) = N_\phi x^a y^b z^c e^{-\alpha r^2} \tag{10}$$

$$N_\phi = (\frac{2}{\pi})^{3/4} \frac{2^{(a+b+c)} \alpha^{(2a+2b+2c+3)/4}}{[(2a-1)!!(2b-1)!!(2c-1)!!]^{1/2}} \tag{11}$$

$$l = a+b+c \tag{12}$$

Similar to the Slater type orbital, α is the exponential parameter and N_ϕ is the normalizing factor. However, this form is Cartesian instead of the spherical form of the Slater type orbital given above. With the Cartesian function, the angular quantum number is equal to the sum of a, b, and c which are the exponents for the angular components in the x, y, and z directions respectively.

In 1951, Roothaan[29, p. 94][47, p. 138] developed the Roothaan equations to solve the Hartree-Fock equations by using a set of known basis functions, such as Slater or Gaussian type functions, and converting the problem into a matrix eigenvalue problem. ψ is expressed as a linear combination of spatial basis functions with the intent of finding the energy-minimizing coefficients for the solution of the matrix eigenvalue problem. From the Slater determinant, it is apparent that the Schrödinger equation is a nonlinear problem, requiring an iterative process of solving the eigenvalue problem from the Roothaan equations named the Self Consistent Field procedure[29, p. 12][47, p. 140,146], abbreviated as SCF. An overview of the SCF procedure is as follows:

1. Setup molecule information such as coordinates, atomic numbers, number of electrons, and basis set information.

2. Calculate integral values of the Hamiltonian matrix and additional terms. The full term is called the Fock matrix, **F**

3. Solve the eigenvalue equation

$$\mathbf{FC} = \mathbf{C}\boldsymbol{\epsilon} \tag{13}$$

   to solve for the energies, $\boldsymbol{\epsilon}$, and coefficients of the orbital basis functions, **C**.

4. Determine whether the solution has converged by comparing the energy of the current iteration and that from the previous iteration against a defined threshold. If the energy difference is above the defined threshold, return to step 2 with the new coefficients and repeat the process.

When converting to an eigenvalue problem using a defined basis and the Roothaan equations, the Coulomb interaction becomes a sum of four-centered two-electron integrals:

$$\langle \mu\nu | \lambda\sigma \rangle = \int \phi^*_\mu(1)\phi_\nu(1)r^{-1}_{1,2}\phi^*_\lambda(2)\phi_\sigma(2)d\vec{r}_1 d\vec{r}_2 \tag{14}$$

The above equation uses the physicist's notation of the Coulomb integrals where μ, ν, λ, and ρ are indicies of basis functions, and 1 and 2 represent electrons with $r_{1,2}$ as the distance between them[47, p. 67,154]. Due to the four generally different indicies, the creation of the Fock matrix is an N^4 calculation. Now that wave function theory and Hartree-Fock have been discussed, the transition to density functional theory can be made which is the main focus of this work.

Density functional theory began with a paper by Hohenberg and Kohn in 1964 [29], where the theorems proven in that work are the basis for modern day density functional theories. The first Hohenberg-Kohn theorem states that 'the external potential $V_{ext}(\vec{r})$ is (to within a constant) a unique functional of $\rho(\vec{r})$'[29, p. 33] and vice versa. The second theorem establishes the variational principle applied to the ground state energy as a function of the density. From these two theorems, it can be concluded that a single ground state density, $\rho_0(\vec{r})$, exists for a ground state energy E_0 and is subject to the variational principle. Leveraging these two theorems, Kohn and Sham, in 1965, presented the Kohn-Sham equations to use the electron density, made up of one electron orbitals, as the functional variable of the energy functional while separating interacting and non-interacting terms [29, p. 41]. This separation requires a new Hamiltonian and orbitals denoted as Kohn-Sham and abbreviated as KS:

$$\hat{H}_{KS} = -\frac{1}{2}\sum_{i}^{N}\nabla_i^2 + \sum_{i}^{N} V_S(\vec{r}_i) \tag{15}$$

As the above Hamiltonian shows, it only contains non-interacting terms, meaning no explicit electron-electron interaction. It also introduces a new term for the effective Kohn-Sham local potential $V_S(\vec{r})$. The ground state wave function is also represented as a Slater determinant, but uses what is called Kohn-Sham orbitals obtained by solving the self-consistent eigenvalue problem with the Kohn-Sham Hamiltonian from eq. (15). The result is similar to the Hartree-Fock eigenvalue equation [29, p. 43]:

$$\hat{f}^{KS}\phi_i = \epsilon_i\phi_i \tag{16}$$

where $\hat{f}^{KS}$ is the Kohn-Sham operator corresponding to the Hamiltonian in eq. (15), ϕ_i are the Kohn-Sham orbitals, and ϵ_i are the associated orbital energies. With the separation of interacting and non-interacting terms, the full energy equation is [29,

p. 44]:

$$E[\rho(\vec{r})] = T_S[\rho] + J[\rho] + E_{Ne}[\rho] + E_{XC}[\rho] \tag{17}$$

$T_S[\rho]$ is the non-interacting kinetic energy, $J[\rho]$ is the Coulomb term, $E_{Ne}[\rho]$ is the attraction energy between nuclei and electrons, and $E_{XC}[\rho]$ is the correction term for kinetic energy due to electron correlation and exchange effects. The above equation is very similar the Hartree-Fock formalism except that Kohn-Sham orbitals are used, the effective Kohn-Sham local potential V_S is different from Hartree-Fock, and the term $E_{XC}[\rho]$ is the only unknown since the noninteracting kinetic energy T_S and the electron density are calculated explicitly as functions of the occupied Kohn-Sham orbitals. Applying the variational principle, the resulting equation is [29, p. 45]:

$$\left(-\frac{1}{2}\nabla^2 + V_{eff}(\vec{r}_1)\right)\phi_i = \epsilon_i\phi_i \tag{18}$$

$$V_{eff}(\vec{r}_1) = \left[\int \frac{\rho(\vec{r}_2)}{r_{12}} d\vec{r}_2 + V_{XC}(\vec{r}_1) - \sum_A^M \frac{Z_A}{r_{1A}}\right] \tag{19}$$

where V_{XC} is the functional derivative of the exchange-correlation energy correction term.

$$V_{XC} = \frac{\delta E_{XC}}{\delta \rho} \tag{20}$$

One of the principle challenges of density functional theory is the determination of the term V_{XC} [29, p. 44-50] and part of the focus of this work.

1.2 Genetic Algorithms

Many results of this work use an implementation of a genetic algorithm[21, p. 143] to optimize parameters of the approximation scheme or functional for density functional codes. A genetic algorithm is a random parameter space optimization scheme, similar to Monte Carlo or simulated annealing[21, p. 138], that uses ideas from biological genetic processes to generate new points in the N-dimensional search space. When discussing genetic algorithms, biological terms are used to describe the features. Iterations are called generations, a single parameter set is labeled as an organism, and a set of parameter sets is called a population. As with other parameter optimization schemes, a fitness value or score is minimized or maximized to obtain the optimal result. The characteristics of the fitness function to calculate the fitness score can significantly affect direction of optimization and yield results that deviate from the original goal. Similarly, over fitting is a concern for the genetic algorithm. For complex problems, the fitness function should be carefully tailored to reach the desired results [21, p. 164-173].

The main driving processes of a genetic algorithm are mutation, crossover, replication, and selection [21, p. 143-155]. Mutation is the random modification of the organism. Several modification options can be used such as varying a single parameter, many parameters, or all parameters. The variance can be small shifts, to search the local space, or a completely random value. Crossover is a unique process of genetic algorithms where two previously calculated organisms, labeled as parents, are randomly chosen from the population and two new organisms, labeled as children, are created by breeding the parameters of the parents. As with mutation, there are several methods of breeding that involve interchanging the parameters of the parents. Some examples are shown in Figure 1.

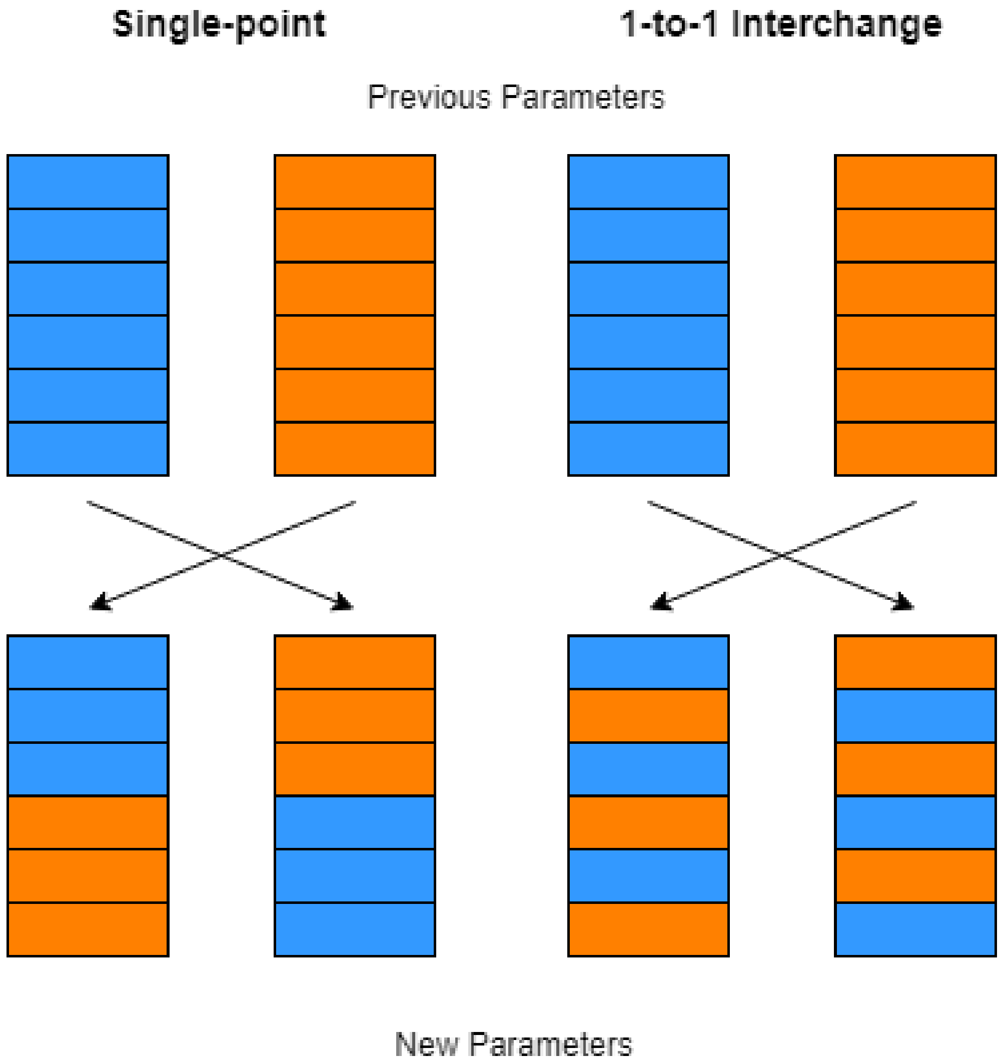

Figure 1: An example of two types of crossover are detailed above. The example on the left is a split cross breeding through the middle of the parameters. The example on the right is a full shuffle of parameters.

These two processes, mutation and crossover, have many different implementations and knowledge of the problem, parameter space, or domain science is helpful in determining the mutation and crossover process to use. While mutation and crossover are influenced by genetics, replication and selection are related to evolution and natural selection[21, p. 143]. Replication is the process of favoring the more fit organisms for reproduction. With this process, these organisms should be more likely to undergo mutation and crossover and influence further populations. Lastly, selection, also known as culling, is the process of removing unfit organisms and creating a new population of the fittest organisms for seeding the next population for investigation [21, p. 144].

2 Software and Codes

2.1 GA_base

The main suite of programs used for this body of work I wrote is labeled as GA_base and involves a series of Bash (Bourne-again shell) and Perl scripts. The basic function of GA_base is to facilitate the general use of genetic algorithms for any problem involving the optimization of a set of parameters. This series of scripts was used in all projects described in this work and the ease of use and effective implementation of genetic algorithms allowed users to quickly and effectively calculate results. I implemented a genetic algorithm with further additions of more sophisticated methods of mutation and crossover available to create a more robust genetic algorithm.

Table 1: Overview of GA_base software

Software	Description
gen_new_pop.pl	Generates a new population and either creates a new population of parameters to be searched, or uses results from previous populations to calculate a new population. Uses three different mutation schemes and three basic crossover scheme. Uses hashes to ensure no repeat parameter calculations and has measures to avoid premature convergence
run_pop.sh	Runs the software to be optimized. Includes several different schemes for running on a single machine, running across a simple Beowulf cluster, or running across a supercomputer with a scheduling system.
job_setup.sh	Sets up the necessary files, data, and input files for actual run of the software.
err_calc.sh	The general error calculation script that calculates error and aggregates it across a single generation. Puts the values in a form that can then be used in the next generation and in the gen_new_pop.pl script.
calc_gen_error.*	Calculates the generation error. This script, implemented by the user, is called by err_calc.sh and extracts the necessary values from output files generated by the software of investigation and calculates the error or fitness. The wild card character implies that the decision for the programming language is left up to the user. In this work, a Bash and Perl version have been implemented.

The intent of GA_base is to be flexible, modular, and allow users to take advantage of the genetic algorithm to do a guided search of high dimensionality space towards optimization involving a specific software. It also organizes input files, output files, calculated data, and intermediate scripts for the ease of the user. Bash and Perl were predominately used for ease of modification and organization of input, output, and other necessary files. Both languages allow native access to regular expressions for parameter replacement and text input files. Regular expressions also provide pattern matching to extract necessary values for calculating errors and fitness. The organization of the parameter sets follows the overall genetic algorithm scheme of separating results by generation with aggregate information present in each.

The computation of results took a variety of forms in this work. Whether the software is a serial, multi-threaded, or a multi-node program, the setup and organization allowed users to take full advantage of the computer systems. For single thread software, multiple executions can be run on single node systems to fully utilize the number of cores. For software already multi-threaded, jobs could be submitted across nodes to take advantage of all available computational power. Overall, the setup allows flexibility in execution for a variety of computer systems.

The genetic algorithm implementation in GA_base has a basic foundation of a genetic algorithm[21] with modifications to avoid repeated parameters and premature convergence. As described in the genetic algorithms section, the basics of a GA are evolutionary processes such as mutation, crossover, replication, and selection. In this implementation, crossover has, by default, a 50% parameter split at the half section. The visual representation is the first example in Figure 1. The crossover rate is determined by the user within the gen_new_pop.pl script. The number of crossovers is determined by the following equation.

$$N = \left[\frac{P}{C_r} e^{-r} \right] \tag{21}$$

For the above equation, N is the final number of crossovers to perform for a given iteration, P is the population size, C_r is the crossover rate, and r is a random value between 0 and 1. The brackets represent rounding to the nearest integer value. The choice for an exponential curve is to allow at least some crossover, regardless of the calculated random value. Various crossover rates allow users to control the probability of selecting more or less crossovers.

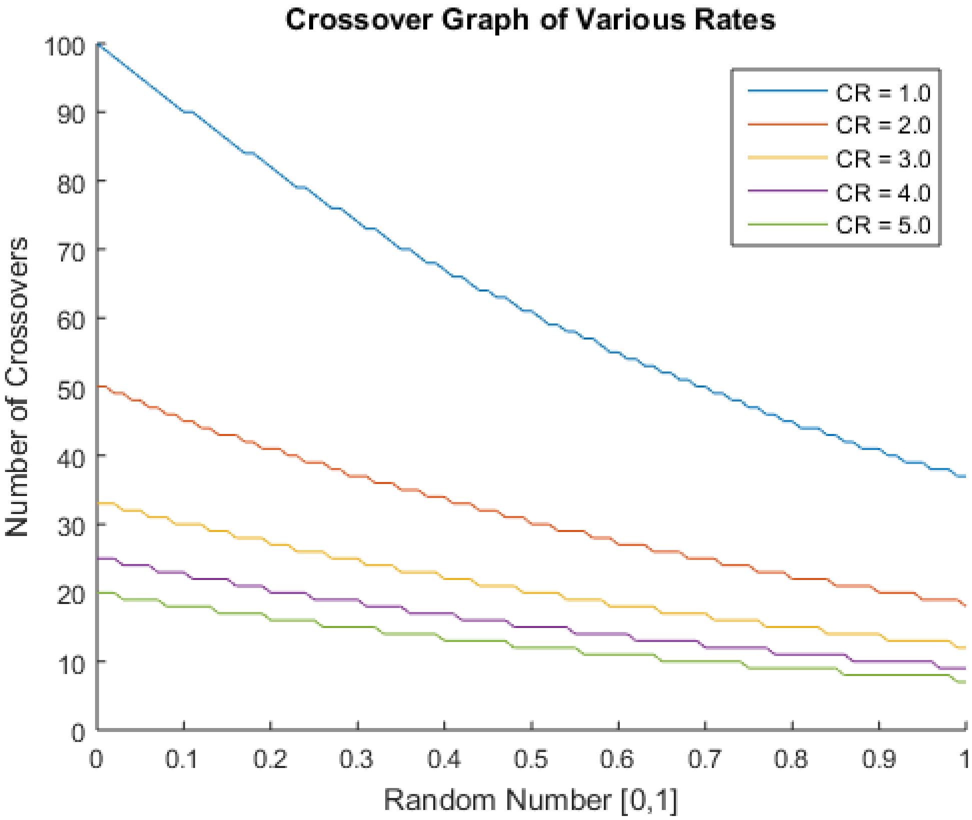

Figure 2: Graph of different user-defined crossover rates detailing the relationship between a random number and the resulting number of crossovers.

Further improvements to this function may include the specification of maximum and minimum number of crossovers with an additional parameter to control the decay. All projects in this work used a crossover rate of 3 to allow a maximum of a third to a minimum of about 20% of the population to undergo crossover. I chose this value in order to ensure a balanced number of crossovers and mutations. In other problems, it may be more beneficial for crossover or mutation to dominate but that is informed by experimentation and domain knowledge. The projects in this work had no prior informed knowledge of the space. Therefore, I implemented three types of crossover with adjustable probabilities to cover a wide range of possibilities. The three types are a one-to-one interchanging of parameters visualized in the right picture of Figure 1, a uniform crossover where the probability that an interchange of parameters is 50%, and a single point crossover visualized in the right picture of Figure 1. To elaborate on the uniform crossover, each parameter has a 50% chance of undergoing crossover. This is a more random set than a one-to-one interchange.

My implementation of mutation comes in three options, total mutation of the parameters to form a new parameter set, a total mutation of a randomly selected subset of parameters, and up to a 5% perturbation of a randomly selected subset of parameters for an existing set. The total mutation of parameters is used to search a totally new area of parameter space by generating a whole new set of parameters. The second mutation method of changing a subset of parameters is to fix some parameters but search other points in space for the other parameters. The last mutation method seeks to search around specific points in space, particularly those with high fitness. In effect this is similar to trying to trace a local minimum or local maximum. The number of perturbations is of the range of one to all parameters undergoing perturbation. The total number of mutations is dependent on the difference between population size and the number of crossovers.

The selection process of choosing a population to undergo crossover and mutation is determined by the top N of all searched parameters and the previous population provided they are not within a user defined distance threshold of the existing top N. Any additional selections are random from the total searched parameter sets. I took such measures to reduce instances of premature convergence that occurred during the research and development process. Additional improvements, such as more sophisticated crossovers, could be added for more user options to tailor the genetic algorithm to the application.

2.2 Density Functional Computational Codes

The main computational code, programmed in C++ and Fortran, in this work is tentatively labeled xTron. This software was developed by Fenglai Liu, a previous postdoctoral researcher Dr. Kong's group, with necessary integral routines used by the density functionals to evaluate Gaussian type orbitals. The code calculates self-consistent field energy for molecular systems as well as additional properties such as odd-electron populations and dipole moments. The calculated energy is conveniently divided into energy components that make up the total energy of a molecule such as nuclear repulsion, effective core potential, exchange-correlation, and kinetic energies. Additionally, xTron supports many contemporary DFT functionals such as B3LYP, B05, PBE, and BLYP to name a few. xTron is still undergoing development towards using general purpose graphical processing units, abbreviated as GPGPU's, and other improvements. For high performance, xTron supports threaded capability through C++'s Thread Building Blocks (TBB) and the Boost library. It will use CPU specifications to take advantage of the maximum number of cores on a given system. No manual input of the number of cores is needed for threaded capability. However, xTron does not support multi-node functionality at the moment.

3 Density-functional Approach to the Three-Body Dispersion Interaction Based on the Exchange Dipole Moment

3.1 Introduction

The first application of GA_base on a density functional project was the optimization of two non-additive three-body dispersion damping models. Damping refers to functions needed to prevent dispersion energy divergence at small inter-nuclear distances and errors from multipole expansion of electrostatic interaction [40, 43]. Dispersion refers to van-der-Waals (VDW) interactions of instantaneous electric dipoles formed from concentrations of electrons within atoms or molecules. Additive dispersion is the summation of pairwise interactions while non-additive has no such summation. In the context of three-body, the non-additive dispersion between three atoms or molecules is given by a single term and cannot be ignored for accurate results as discussed in Kennedy et. al[28]. The models in this work address computational cost and accuracy in DFT methods. "*Ab initio* correlation methods, such as coupled-cluster with singles, doubles, and perturbative triple excitations (CCSD(T)) and symmetry-adapted perturbation theory (SAPT) are capable of accurately estimating the dispersion interaction" [40] but are computationally expensive. "Non-empirical DFT models have also been put forward based on response formalism [19], model exchange hole [8], exchange-correlation (XC) hole [1], and ground-state electron density with reference values for the free atoms [19]." [40]. One such example is the exchange dipole moment (XDM) model of Becke and Johnson [8, 6, 26], "a nonempirical density functional model of dispersion, in which the dipole moment of a model exchange hole centered on one system induces an instantaneous dipole moment in a weakly interacting neighboring system" [40]. Most DFT models focus on two-body dispersion because it makes up the majority of dispersion interaction [40]. It is unclear how much the non-additive three-body term contributes the total dispersion [40]. For stabilization energy, Wen et. al estimates it to be as large as 10% in some molecular crystals[54] while von Lilienfield and Tkatchenko report about "46% of the energy difference between folded and elongated polyalanine decamers[34]. The three-body dispersion contribution was also shown to be crucial to the third virial coefficient.[34, 55, 25, 11]"[40]. These reported estimated energy contributions indicate that "current DFT methods tend to overestimate the two-body dispersion interaction partly due to the omission of the three-body contribution that is often of the opposite sign" [40]. The implementation of the XDM model includes the non-additive three-body dispersion component with two proposed damping functions. Performance of the models was measured by isolating the non-additive three-body dispersion component for benzene trimers from Sherrill et. al. [28] and from SAPT results from Szalewicz et. al. for He_3 and Ar_3 trimers [12, 11, 9]. Isolation involved taking the difference between Møller-Plesset perturbation theory (MP2) and CCSD(T) energies [28]. Parameters for the damping functions were optimized using GA_base and my implementation of genetic algorithms. The results were published in *The Journal of Chemical Physics* [40].

3.2 Background

3.2.1 Møller-Plesset Perturbation Theory

This section will briefly cover background on MP2, CCSD(T), and SAPT with descriptions of dispersion in the context of these methods. The first method to discuss is MP2, a wave function method from perturbation theory that approximately and directly describes intermolecular forces such as electrostatics, induction, and dispersion[36, 51]. A brief description is given here starting with the Hamiltonian defined as:

$$\hat{H} = \hat{H}_0 + V = \sum_{i}^{n} \hat{F}_i + V \tag{22}$$

where the Fock operators, $\hat{F}_i$, are assumed nonperturbative [52]. The resulting electron correlation is given as:

$$E_{MP2} = -\sum_{i<j}^{n_{occ}} \sum_{a<b}^{n_{vir}} \frac{|\,\langle ij|ab\rangle - \langle ij|ba\rangle\,|}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} \tag{23}$$

where i and j are indicies over the occupied orbitals, ϕ_i and ϕ_j, a and b are indicies over the virtual orbitals, ϕ_a and ϕ_b, and ϵ are the orbital energies of a specified index [52]. MP2 overestimates the energy from dispersion interaction specifically from the coefficients[51] or neglect of the three-body terms[28]. Many modern variations have been developed to correct or improve MP2 such as DF-MP2/RI-MP2 (density fitting/resolution of identity), LMP2, TRIM-MP2 (local correlation), and combinations such as DF-LMP2, RI-TRIM [51, 36].

3.2.2 Symmetry-Adapted Perturbation Theory

SAPT stands for symmetry-adapted perturbation theory and begins with isolated monomers or unperturbed molecules. The interaction energy is treated as small perturbations by Coulombic intermonomer interactions [48]. To begin, the Schrödinger's equations for isolated monomers A and B are given as [48]:

$$H_X \Phi_X = E_X \Phi_X, \qquad X = A \text{ or } B \tag{24}$$

where H_X is the Hamiltonian, Φ_X is the wave function, and E_X is the energy for monomer X. “Next, the monomers are placed in the dimer configuration and all electrons and nuclei of monomer A then interact with those of monomer B according to Coulomb's law."[48] The sum of these Coulomb interaction terms is denoted by V, the intermolecular interaction operator [48]. The Hamiltonian with V for the dimer is then:

$$H = H_A + H_B + V = H_0 + V \tag{25}$$

The unperturbed part of the problem with the Hamiltonian H_0 has the solution as the product of isolated monomers for the wave function: $\Psi^{(0)} = \Phi_A \Phi_B$ and the eigenenergy as $E^{(0)} = E_A + E_B$. The operator V uses the standard Raleigh-Schrödinger (RS) perturbation theory expressed as a sum of perturbation corrections [48]:

$$E_{int} = E^{(1)}_{RS} + E^{(2)}_{RS} + \cdots \tag{26}$$

This is the most basic formalism of SAPT with improved versions and methods proposed for convergence, performance, and applications to categories of problems, such as many-electron systems. In the context of dispersion, SAPT is able to describe monomer properties through their direct relation to SAPT interaction energies and naturally decomposes them into physically interpretable components. The main components include electrostatic, induction, dispersion, and exchange energy contributions. SAPT, in the context of this work and dispersion, allows the direct computation of three-body nonaddtive energies to the effect that a three-body SAPT(DFT) has been developed recently [48] by Podeszwa and Szalewicz[38].

3.2.3 CCSD(T)

CCSD(T), considered the 'gold standard' of reproducing experimental results [56], is a wave function method from coupled-cluster theory where the wave function is written as [2]:

$$\Psi_{CC} = (1 + \hat{T} + \frac{\hat{T}^2}{2} + \frac{\hat{T}^3}{3!} + \cdots)\Phi_0 \tag{27}$$

where Φ_0 is the HF wave function and $\hat{T}$ is the cluster operator. $\hat{T}$ is the sum of excitation cluster operators [2]:

$$\begin{aligned}
\hat{T} &= \hat{T}_1 + \hat{T}_2 + \cdots + \hat{T}_N && (28)\\
\hat{T}_n &= (n!)^{-2} \sum_{\substack{i,j,\ldots\\ a,b,\ldots}} t_{ij\ldots}^{ab\ldots} \hat{c}_a^\dagger \hat{c}_b^\dagger \cdots \hat{c}_j^\dagger \hat{c}_i^\dagger && (29)\\
\hat{T}_1\Phi_0 &= \sum_{i,a} t_i^a \Phi_i^a && (30)\\
\hat{T}_2\Phi_0 &= \sum_{i>j,a>b} t_{ij}^{ab} \Phi_{ij}^{ab} && (31)\\
\hat{T}_3\Phi_0 &= \sum_{i>j>k,a>b>c} t_{ijk}^{abc} \Phi_{ijk}^{abc} && (32)\\
\hat{T}_4\Phi_0 &= \sum_{i>j>k>l,a>b>c>d} t_{ijkl}^{abcd} \Phi_{ijkl}^{abcd} && (33)
\end{aligned}$$

Here the operator $\hat{T}_n$ expands to single Φ_i^a, double Φ_{ijk}^{abc}, triple Φ_{ijk}^{abc}, etc. excitations [2]. From the description of CCSD(T) and the expansion of higher order excitations between bodies, it is observed that CCSD(T) treats correlation very accurately [13] and thus describes dispersion effects well, as compared to MP2 and SAPT, though at a significant computational cost.

3.3 Theory

The dispersion energy can be represented as a many body series expansion [49]:

$$E_{disp} = \frac{1}{2}\sum_{A,B} E_{(2)}(\vec{R_A}, \vec{R_B}) + \frac{1}{6}\sum_{A,B,C} E_{(3)}(\vec{R_A}, \vec{R_B}, \vec{R_C}) + ... \tag{34}$$

with groups of atoms (A, B, C, ...), positions ($\vec{R}_A, \vec{R}_B, \vec{R}_C, ...$), and energies ($E_{(2)}, E_{(2)}, ..., E_{(N)}$). The two-body term is given as:

$$E_{(2)} = -\frac{C_6^{AB}}{R_{AB}^6} - \frac{C_8^{AB}}{R_{AB}^8} - \frac{C_{10}^{AB}}{R_{AB}^{10}} - ... \tag{35}$$

for spherical neutral atoms with non-overlapping densities [40]. The term R_{AB} is the distance from atoms A to atoms B and the dispersion coefficients, C_n^{AB} are given from the XDM model of Becke and Johnson [26]. Similarly, the non-additive three-body interaction can be expressed as:

$$E_{(3)}(\vec{R_A}, \vec{R_B}, \vec{R_C}) = C_9^{ABC} \frac{(3\cos(\phi_A)3\cos(\phi_B)3\cos(\phi_C) + 1)}{R_{AB}^3 R_{AC}^3 R_{BC}^3} \tag{36}$$

where ϕ_A, ϕ_B, ϕ_C are the internal angles of the triangle ABC and C_9^{ABC} is given by the Axilrod-Teller formula [28, 40]. As observed from the above equations, at small internuclear distances the resulting dispersion energy increases dramatically with erroneous values. This case occurs in molecules, necessitating damping factors (f) [40]. Damping functions are also needed to correct for errors from the multipole expansion of electrostatic interaction [43]. The dispersion energy term is then rewritten as:

$$E_{disp} = \frac{1}{2}\sum_{A,B} E_{(2)}(\vec{R_A}, \vec{R_B}; f_{AB}(R_{AB})) + \frac{1}{6}\sum_{A,B,C} E_{(3)}(\vec{R_A}, \vec{R_B}, \vec{R_C}; f_{ABC}(\overline{R_{ABC}})) + ... \tag{37}$$

with the two-body function, f_{AB}, and non-additive three-body dispersion interactions, f_{ABC} and $\overline{R_{ABC}}$ is a geometrically averaged effective distance to be specified [40]. Before the discussion of the two damping schemes, it should be mentioned that the C_9^{ABC} was also a major focus of the publication but was not part of my contribution to this work. The modified term from Eq. (2.16) in Tang's paper [49] is given as:

$$C_9^{ABC} = \frac{3}{2}\alpha_A\alpha_B\alpha_C\frac{\eta_A\eta_B\eta_C(\eta_A+\eta_B+\eta_C)}{(\eta_A+\eta_B)(\eta_B+\eta_C)(\eta_C+\eta_A)} \tag{38}$$

The term η_i are constants of an average atomic excitation energy[40]. The approximate form is given by Tang [49] as a function of the two-body interaction coefficients:

$$\eta_A = \frac{4}{3}\frac{C_6^{AA}}{\alpha_A^2} \tag{39}$$

In this work, a modified η_A value is obtained by linking the XDM model and the three-body dispersion model of Tang and Karpus to produce[40]:

$$\eta_A = \frac{2}{3}\frac{\langle d_X^2\rangle_A}{\alpha_A} \tag{40}$$

The result is "a nonempirical estimate of nonadditive three-body interaction contribution to the dispersion energy within the XDM model."[40] Returning to damping, two damping functions for the three-body term are suggested beginning with Scheme A which is similar to the triple product equation in Ref. [43] with a different numerator [40] and given by:

$$E_{(3)}(\vec{R}_A,\vec{R}_B,\vec{R}_C;f_{ABC}) = C_9^{ABC}\frac{(3\cos(\phi_A)3\cos(\phi_B)3\cos(\phi_C)+1)}{(\overline{R}_{vdW,AB}^3+R_{AB}^3)(\overline{R}_{vdW,BC}^3+R_{BC}^3)(\overline{R}_{vdW,CA}^3+R_{CA}^3)} \tag{41}$$

"$R_{vdW,AB}$ is a sum of effective atomic VDW radii of atoms A and B related to the critical interatomic distance R_c, AB as defined in the two-body XDM model. It is the distance where the three components of the two-body dispersion energy become roughly equal in magnitude[26], which is when [40]:"

$$
\begin{aligned}
R_{c,ij} &= \frac{1}{3}\Big[\Big(\frac{C_{8,ij}}{C_{6,ij}}\Big)^{1/2} + \Big(\frac{C_{10,ij}}{C_{6,ij}}\Big)^{1/4} + \Big(\frac{C_{10,ij}}{C_{8,ij}}\Big)^{1/2}\Big] && (42)\\
\overline{R}_{vdW,ij} &= \alpha_1 R_{c,ij} + \alpha_2 && (43)
\end{aligned}
$$

α_1 and α_2 are the two adjustable damping parameters optimized for using my genetic algorithm implementation. The second damping function named Scheme B uses a geometric-mean of the XDM effective VDW radii [40],

$$
\begin{aligned}
E_{(3)}(\vec{R_A}, \vec{R_B}, \vec{R_C}; f_{ABC}(\overline{R}_{ABC})) &= C_9^{ABC}\frac{(3\cos(\phi_A)3\cos(\phi_B)3\cos(\phi_C)+1)}{(\overline{R}^9_{vdW,ABC} + R^3_{AB}R^3_{AC}R^3_{BC})} && (44)\\
\overline{R}_{vdW,ABC} &= \sqrt[3]{R_{vdW,AB}R_{vdW,BC}R_{vdW,CA}} && (45)
\end{aligned}
$$

The main difference between the two are the denominators where Scheme A uses $R_{c,ij}$, the critical interatomic distance and, as shown above, Scheme B relates the three interatomic distances in an effective mean radius, $\overline{R}_{vdW,ABC}$ [40].

3.4 Computational Details

Development and execution of the VDW code and the GA_base was done on the two machines used by the group labeled as kong-srv and kong-wk to denote the server and the workstation respectively. Each energy calculation took on the order of 10 milliseconds with the calculation of all 62 benzene trimers at approximately 2-3 seconds. The calculation of the non-additive three-body dispersion involved a program written by members in the group labeled as 'VDW'. Calculation of the full energy was done using Q-Chem [44]. Languages used were C++ for the 'VDW' program and Perl scripts to execute and run the genetic algorithm. Execution of the genetic algorithm was further improved by threading each parameter set under investigation into it's own separate thread. Therefore, each generation of the genetic algorithm was approximately 2-6 seconds of execution time with a population size of 16 to take full advantage of the 16 core cpu. Mentioned early, the weights associated with each set of trimers (benzene, He_3, and Ar_3) also involved some discussion and experimentation to obtain the desired results.

3.5 Results and Discussion

The present model is an extension of the XDM algorithm from Ref [31]. The fully analytic form is detailed in the theory section. An unpruned grid (194,590) consisting of 194 radial and 590 angular points provided the desired accuracy within 1 kcal/mol for the numerical integration in all studied cases [40]. The dispersion energy is calculated post-self-consistent field (SCF). "The electron density and the non-dispersion part of the DFT energy are calculated with the exchange-correlation(XC) DFT scheme P86(X) + PBE(C) as recommended in Ref. [27][40]." The investigation of the C_9^{ABC} involved homonuclear trimers and noble atom triplets while the damping involved 62 symmetry-unique benzene trimers. Four benzene trimers were removed as extreme outliers due to occasional poor agreement, resulting in these four outliers dominating the error and optimization [40]. For the homonuclear trimers and noble atom triplets, the SAPT values from Szalewicz et. al.[12, 9, 10] for He_3 and Ar_3 are used for their accurate estimation of dispersion. As detailed in the background section of this work and applied by Sherrill et. al.[28], the non-additive three-body intermolecular dispersion interaction can be estimated by taking the difference between CCSD(T) and MP2 total three-body energy contributions [40]. This is done "under the assumption that the two-body and additive three-body interactions in CCSDT(T) are mostly recovered by the MP2 energy[40]." Figure 3 illustrates the intermolecular interactions only of E(CCSD(T))-E(MP2) deduced from reference [28].

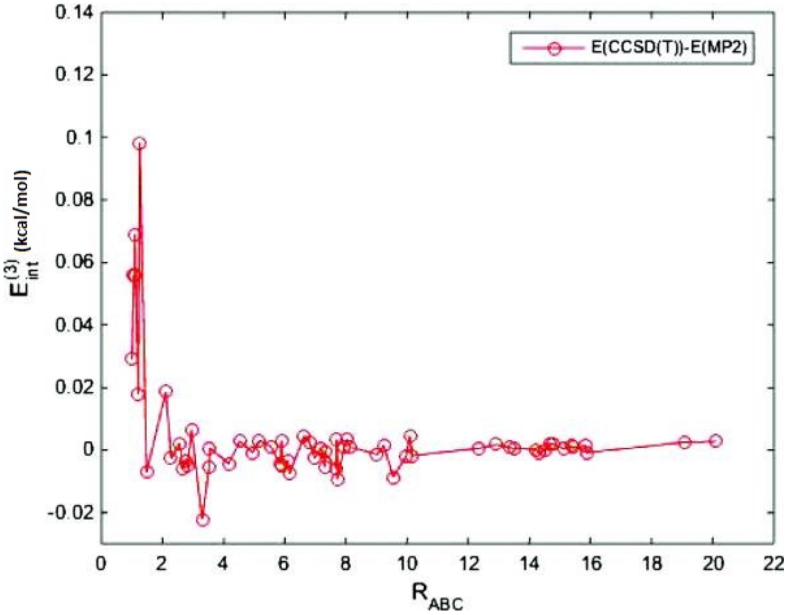

Figure 3: E(CCSD(T))-E(MP2) for a series of benzene trimers [28] as a function of $R_{ABC} \equiv R^3_{AB}R^3_{AC}R^3_{BC}$ intermolecular distance normalized by the smallest R_{ABC} value in the series [40] to the three-body dispersion energy, $E^{(3)}_{int}$. Values are available in the appendix.

The X-axis values, $\overline{R}_{ABC} \equiv R^3_{AB}R^3_{AC}R^3_{BC}$, are normalized by the smallest R_{ABC} value in the series. To first establish that E(CCSD(T))-E(MP2) is an accurate estimate of non-additive three-body intermolecular dispersion, we first compare the calculated values to the basic, nondamped $E_{(3)}(\vec{R}_A, \vec{R}_B, \vec{R}_C)$ from equation 41. The C_9 value is the newly proposed term, briefly covered in the theory section. The accuracy and results of the non-damped three-body dispersion interaction and the C_9 was verified and detailed in Table III in the original paper. The comparisons were between highly accurate benchmarks of noble atom triplets[32, 50, 40]. In order to correctly measure the nondamped non-additive three-body dispersion, the 7 most distant symmetry-unique timers, measured by R_{ABC}, were chosen due to the negligible amount of damping for well-separated trimers[40]. The sum of these 7 trimers is 0.00120 kcal/mol and the sum of the latter is 0.00125 kcal/mol, with a mean absolute deviation (MAD) on these trimers of 0.00005 kcal/mol.

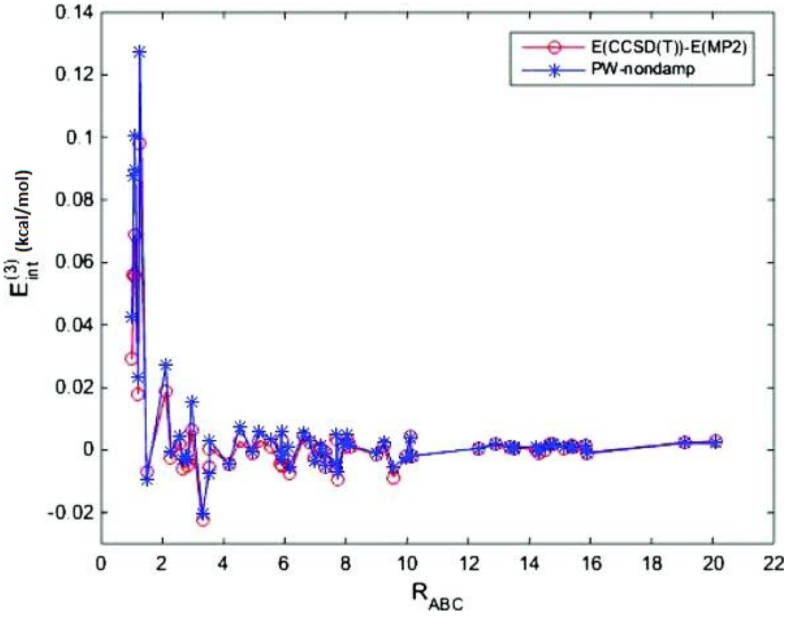

Figure 4: E(CCSD(T))-E(MP2) versus nondamped present model under investigation. Function of R_{ABC} and $E_{int}^{(3)}$

The observed results are within chemical accuracy (1 kcal/mol). This shows the energy difference, E(CCSD(T))-E(MP2), is, mostly, due to the three-body dispersion, $E_{(3)}$, [40]. As Figure 3 and 4 shows, the energy dispersion and the models agreement deteriorates as the trimers become closer together. This observation highlights the need for damping [40].

This work presents and tests two damping schemes, denoted as Scheme A and Scheme B, detailed in the theory section. The initial damping parameter values were first determined by the original XDM values of the two parameters from the damping formula for two-body dispersion, $\alpha_1 = 0.80$ and $\alpha_2 = 1.49$ angstroms, as recommended in Ref. [27] for the same basis set (aug-cc-PVDZ) and density functional (P8E(X) + PBE(C)) for the benzene trimers [40]. Figure 5 illustrates the performance of the model with each of the two damping schemes using the original damping parameters optimized for two-body interactions.

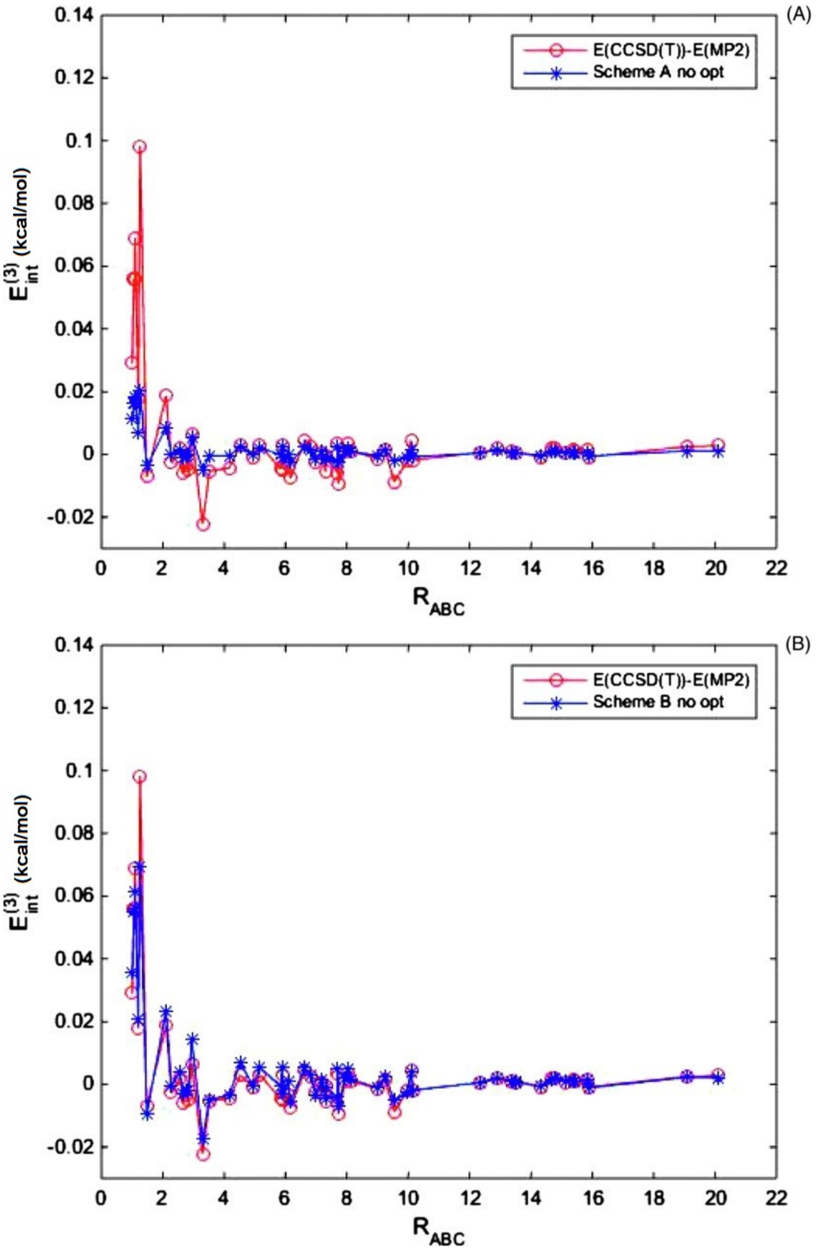

Figure 5: E(CCSD(T))-E(MP2) versus the values calculated with the present method for a series of benzene trimers. The figures are Scheme A and Scheme B respectively. Both use the original XDM two-body damping parameters meaning they are not the optimized parameters.

For the benzene trimers observed in Figure 5, Scheme B performs better than Scheme A. “However, the original damping parameters yield less accurate results for the He_3 and Ar_3 trimers when compared to the fitted analytical potential of

Szalewicz et. al. (see Table 2)[40].” Also for smaller values of R_{ABC}, the deterioration is more apparent and, with the damping scheme, performs worse than without any damping functions as displayed in Figure 4. This discrepancy highlights the need for optimization of each damping scheme for all three data sets: benzene, He_3, and Ar_3 trimers. This shows the need for a combined optimization of our three-body damping parameters using the three different data sets altogether. The genetic algorithm used a weighted sum of the MAD for each trimer set as the fitness value and minimized this. The relative weight was used to equalize three-body dispersion energy values to similar magnitudes[40]. Results and optimized parameters can be observed in Table 2. “Concerning first the benzene trimers, the MAD with optimized Scheme B (0.00205 kcal/mol) remains about the same as in the case of using the original damping parameters a1 and a2 (0.0021 kcal/mol). Scheme A benefits more from the optimization here with MAD reduced from 0.006 kcal/mol to 0.0019 kcal/mol [40].”

As Table 2 shows, Scheme A performs slightly better than Scheme B overall, but both damping schemes have reasonable accuracy overall after optimization [40]. Observing the total errors for each of the three sets between optimized damping and no damping highlights the necessity of damping, specifically from the noble-gas trimers. Figure 6 illustrates this comparison between no damping and optimized damping parameters.

Table 2: Mean Absolute Deviations (MAD) of Intermolecular Dispersion Energy: MAD (kcal/mol) “of our results for the non-additive 3-body intermolecular dispersion energy for the set of 62 benzene trimers, 203 He_3 trimers, and 203 Ar_3 trimers (benchmark data from Refs. [28, 12, 9]) The mean unsigned relative error is given in square brackets. For the series of benzene trimers, the deviations are estimated with respect to the energy difference E(CCSD(T))-E(MP2) reported in Ref. [28]. For the series of He_3 and Ar_3 trimers, the deviations are with respect to the triple-dipole part of the damped dispersion component of the SAPT non-additive three-body interaction potential fitted to FCI and CCSDT(Q), respectively (the 111 term in Eqs. (15)-(18) in Ref. [10]) [40].”

Scheme	$a_1(\mathring{A})$	$a_2(\mathring{A})$	$(C_6H_6)_3$	He_3	Ar_3
No damping	...	...	0.00372 [0.5690]	3.849×10^{-6} [0.3927]	5×10^{-5} [0.0946]
A(noopt)	0.80	1.49	0.0060 [3.9747]	9.606×10^{-7} [0.2300]	1.16×10^{-4} [1.2742]
B(noopt)	0.80	1.49	0.0021 [0.5948]	2.591×10^{-6} [0.3810]	1.16×10^{-4} [0.0725]
A(opt)	0.08438	1.938	0.0019 [0.5687]	9.748×10^{-7} [0.2108]	1.464×10^{-5} [0.1520]
B(opt)	0.2525	2.927	0.002053 [0.5770]	1.432×10^{-6} [0.3547]	1.531×10^{-5} [0.0779]

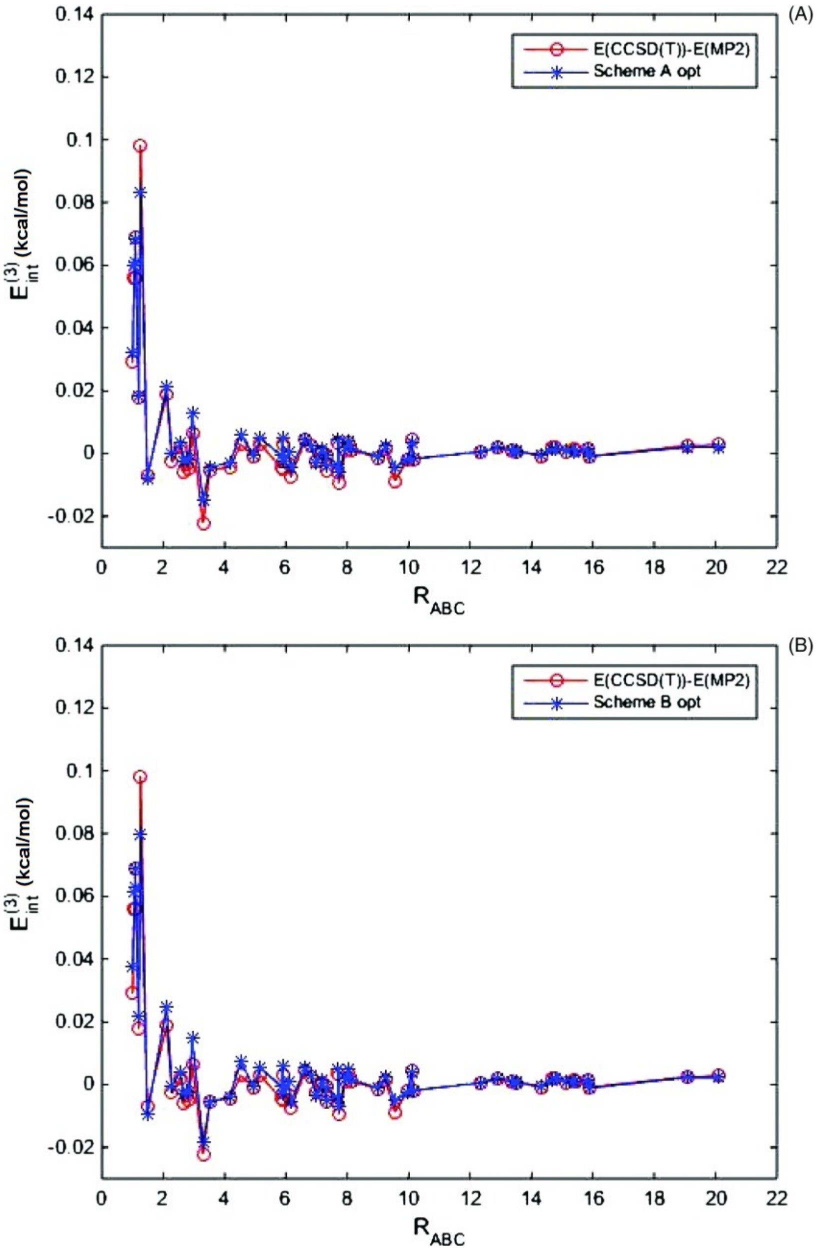

Figure 6: Comparisons after optimizing the damping schemes displaying Scheme A and Scheme B respectively.

"Listed in Table 2 are also mean unsigned relative errors for each scheme. The main contribution to the relative errors is the small interaction energies as a result from the angle factors in the three-body formalism (Eq. (41)) even at relatively close distances [40]."

3.6 Conclusion

This work presents two damping functions for non-additive three-body dispersion from VDW interactions. The method is an extension of the DFT XDM version for two-body interaction. The first part of the original paper is the investigation of the dispersion coefficient, C_9, with encouraging results. The C_9 is an accurate estimate for noble atom triplets compared to values in literature [43] and for benzene trimers by Sherrill and Co. [28]. The results are necessary for the three-body dispersion term, $E_{(3)}$, but only briefly detailed for this work due to my focus and contribution on the damping schemes. For damping, two new schemes were purposed, benchmarked, and optimized against a set of benzene, He_3, and Ar_3 trimers. The data for benzene is from Sherrill and Co. [28] while He_3 and Ar_3 are from Szalewicz and Co. [12, 9]. Optimization involved using my genetic algorithm implementation. After optimization, both schemes show comparable results with Scheme A slightly outperforming Scheme B. Future work could include further improvement of non-additive three-body interactions through computation of the third virial coefficients [40, 43, 28, 12].

4 Density Functional Model for Nondynamic and Strong Correlation

4.1 Introduction

xTron calibration first involved the optimization of functional values within the KP16 functional developed by Kong and Proynov [30]. The KP16/1B13 functional is a single-term density functional model that handles nondynamic/strong correlation and demonstrates the viability of such a functional instead of using multiple configurations. Nondynamic correlation refers to the issue of using a single wave function configuration as insufficient to represent degenerate electron states. The category of methods that use multiple configurations to handle nondynamic correlation are labeled as multiconfigurational methods. Some examples of strongly correlated systems include bond dissociation limits, transition metals, and diradicals [30]. One approach to addressing nondynamic correlation is multiconfigurations, the method where the wave function is a linear combination of configuration states. The mix of near-degenerate configurations address nondynamic correlation for multiconfiguration methods [52]. However, multiconfiguration DFT dissociates a bond but double counts some of the correlation [24, 45, 22]. Other noteworthy methods that address nondynamic correlation are density matrix functional theory (DMFT), random phase approximation (RPA), and noncolinear DFT methods for strongly correlated radicals. For this work, KP16/B13 is compared to other functionals of similar type such as hybrid-GGA's. Notable single determinate methods that incorporate nondynamic correlation include B05[5], B13[4] by Becke, and PSTS by Perdew et. al [37]. These methods compensate with a local correction for the delocalization of the exact exchange [30]. For standard systems, these single determinate methods perform similarly on standard thermodynamic benchmarks compared to mainstream methods, but much better on strongly correlated systems [39, 30]. Other functionals included for comparison are the nonempirical PBE and the parameterized hybrid functionals B3LYP and M06-2X. In this work, the analytical method KP16/B13 was proposed to calculate the population of effectively unpaired electrons for characterizing nondynamic correlation with a single-determinate representation [39, 30].

4.2 Theory and Experiments

The KP16 functional starts with full Kohn-Sham exact exchange, E_x^{ex} and uses the adiabatic connection method to approximate the correlation energy and include nondynamic correlation as an estimation to the exchange energy. The exact exchange here refers to Hartree-Fock (HF) which calculates the exchange exactly but, on its own, poorly dissociates bonds and yields large errors for fractional spin[40]. The resulting exchange-correlation term is as follows:

$$E_{xc} = E_x^{ex} + \int \frac{1+e^{bz}}{1-e^{bz}} \left[1 - \frac{2}{bz}\ln\left(\frac{1+e^{bz}}{2}\right)\right] \tilde{u}_c^{nd} d^3r \tag{46}$$

$$\tilde{u}_c^{nd} = (u^{opp}_{statC} + c u^{par}_{statC}) \left(1 + \frac{1}{2}\sqrt{\frac{\alpha}{\pi}} e^{-\alpha/z^2}\left(\left(\frac{D_\alpha}{\rho_\alpha^{5/3}}\right)^{1/3} + \left(\frac{D_\beta}{\rho_\beta^{5/3}}\right)^{1/3}\right)\right) \tag{47}$$

E_{xc} is the exchange correlation energy and $\tilde{u}_c^{nd}$ is the dynamic correlation potential energy density which adds the correction of nondynamic correlation. The parameters under consideration for optimization were α, b, and c, where α and b are empirical parameters with c as a scaling component to the parallel-spin part of the nondynamic correlation. Other terms of note include u^{opp}_{statC}, the nondynamic opposite-spin component, and u^{par}_{statC}, the parallel-spin component. The functional was first optimized using the quantum chemistry code Q-Chem [44], while the finishing touches on the integral codes were completed in xTron. While the goal of KP16 is to address the nondynamic correlation energy, as a general purpose DFT functional, KP16 should perform well for equilibrium molecules and dissociation. Therefore, the optimization of the functional involved three sets of molecules and atoms with the goal of reducing the overall mean absolute deviation of atomization energy values to less than 1 kcal/-mol. The value 1 kcal/mol is standard here for experimental accuracy. Atomization energy refers to the difference between the energy of the molecule and the sum of the isolated atom energies of the constituent parts of the molecule. An example for H_2O is shown below:

$$AtomizationEnergy(\mathrm{H_2O}) = E(\mathrm{H_2O}) - (2E(\mathrm{H}) + E(\mathrm{O})) \tag{48}$$

The sets used to calibrate the parameters were polyatomic molecules, diatomic molecules, and fractional spin atoms. Polyatomics and diatomics are the standard benchmark sets for atomization. Fractional spin atoms are included to optimize for fractional spin error which dominates at the dissociation limit and is also important to many systems near equilibrium [30, 14]. A limited set of molecules was chosen to reduce over-fitting. The full list is available in the Appendix. With the set of fractional spin atoms, the error is calculated by taking the difference between the ground state energy of the atom and the same energy but with fractional spin occupancy corresponding to its homonuclear dissociation limit [30]. The fitness function is a weighted sum of the mean absolute deviations of the included sets. Optimization of the parameters was done through the genetic algorithm and GA_base detailed previously to minimize the fitness function. All calculations used the self-consistent field (SCF) procedure for energy, G3Large for the basis sets, and a large unpruned grid of 192 by 590 points.

4.3 Computational Details

The majority of the computation for this work was done on Darter, the now decommissioned, supercomputer at Oak Ridge National Labs (ORNL). Babbage, the MTSU Computational Science (COMS) cluster, was also used for development and computation, but once the procedure, genetic algorithm, and GA_base were debugged and tested, the code base was migrated to Darter. Each compute node had 64 cores. The software used for atomic and molecular calculation was Q-Chem [44]. An acknowledgment must be made for the software Eden, developed by Scott Simmerman at ORNL, that ran single-threaded jobs over a multi-core compute node. This allowed computation at full efficiency due to the single-threaded nature of Q-Chem, the necessity to use all available resources, and the large number of atomic and molecular calculations. Computation time was limited by the largest molecules due to the necessity for all molecules to be computed for a particular parameter set (α, b, c) before a fitness value could be calculated. The molecules that took the most of amount of time were butane, pyrol, and pyridine. Atomic makeup of these molecules is listed in the Appendix. This resulted in approximately 6 hours of compute time per parameter set and a total of approximately 200,000 hours of compute time.

4.4 Results and Discussion

The result of the mean absolute deviations (MAD) are detailed in Table 3.

Table 3: MAD of Fractional-Spin Error and Atomization Energy (kcal/mol) for the Assessment Set

	Fractional-Spin	Atomization Energy	Average
HF	179.21	107.37	143.29
PMF(2)	8.94	6.30	7.62
PMF(3)	8.14	3.64	5.89
RI-B05	55.95	2.32	29.14
B13	8.78	3.90	6.34
PSTS	50.97	5.18	28.08
PBE	81.50	12.74	47.12
B3LYP	57.56	2.99	30.28
M06-2X	92.17	1.98	47.08

PMF(2) stands for present model functional with two parameters and PMF(3) is the three parameter model. PMF(2) was optimized using uniform grid search while PMF(3), now labeled as KP16/B13, was optimized using my implementation of the genetic algorithm. The final parameter values for the two functionals are $b = 1.2$, $\alpha = 0.037$ for PMF(2) and $b = 1.355$, $\alpha = 0.038$, and $c = 1.128$ for PMF(3). As the Table 3 shows, PMF(2) and PMF(3) significantly improve the error of equilibrium and fractional spin atoms and molecules compared to the mainstream methods listed.

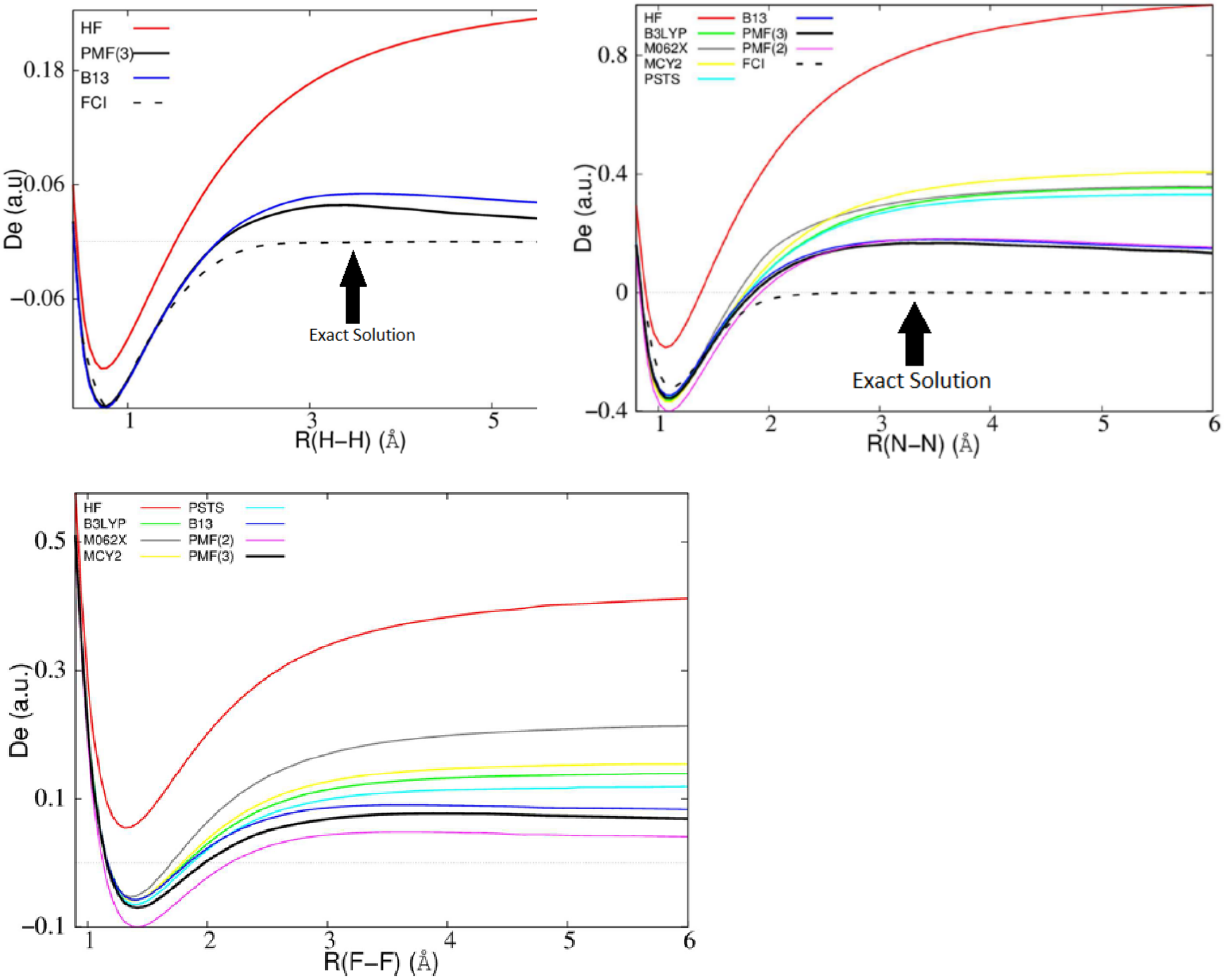

Figure 7: Dissociation curves of H_2, N_2, and F_2. De is the dissociation energy. The Full Configuration Interaction (FCI) curve represents the expected behavior where the difference should approach 0 as the interatomic distance grows[40].

To assess the performance of PMF(2) and PMF(3) for strong correlation, the dissociation curves of H_2, F_2, and N_2 using PMF(2) and PMF(3) are compared against mainstream methods. The 6-31G* basis set is used for the calculations with these plots. As the figures show, the newly optimized functionals, PMF(2) and PMF(3), and B13 perform similarly and are significantly better than the other hybrid functionals, B3LYP and M06-2X [30]. This difference is most readily shown in the N_2 case.

However, the curve of PMF(2) and PMF(3) still exhibits the behavior of becoming positive and then approaching the 0 line. Further improvements to the functional are currently under investigation. However, the model with 3 parameters under consideration for optimization performs well for nondynamic/strong correlation, atomization energies, and singlet-triplet energy splitting without relying on error cancellation. Additionally, the models presented in this work and B13 provide evidence of the feasibility of single-determinant DFT functionals for describing nondynamic and strong correlation[30].

4.5 Conclusion

In summary, the KP16/B13 was developed to include nondynamic correlation and correct fractional spin error as a single determinant-based functional. The genetic algorithm was successful in optimizing the functional to comparable accuracy of contemporary single determinant-based hybrid functionals. The results are favorable for general and strongly correlated systems from the observed polyatomic, diatomic, and fractional spin error values. The results here contrast contemporary DFT methods which rely on error cancellation or heavy parameterization and, together with B13, provide additional evidence for single-determinant Kohn-Sham DFT functionals that address nondynamic correlation.

5 xTron Performance on Chemical Properties Using Minnesota Sets

5.1 Introduction

As detailed in the previous section, the KP16/B13 functional was optimized using standard benchmark sets and a fractional-spin set. Additionally, it was designed for general purpose calculation with the intention of handling systems ranging from dynamic to nondynamic correlation. The purpose of this work is to extend the benchmarks with the Minnesota sets[58] and determine whether KP16/B13 can handle weak and normal correlation as well as a variety of chemical properties. The Minnesota sets refers to the Minnesota 2015 (MN15) database[58] of molecules. This set is compiled by Zhao and Truhlar[58] for the purposes of benchmarking the M06 and M06-2X functionals. It contains a total of 29 subsets. However, only 17 subsets are investigated in this work. The remaining sets include corrections for VDW interactions, which KP16/B13 does not contain yet, solids, and other geometries that are currently beyond the capability of KP16/B13. Excited states are noteworthy, but the focus of KP16/B13 has currently been ground state energy. The selected sets include molecules exhibiting properties of thermochemistry, kinetics, noncovalent interactions, transition metal bonding, atom excitation energies, and molecular excitation energies[57] to name a few. An exhaustive list is detailed in Table 4.

5.2 Computational Details

The benchmarking of KP16/B13 in this section follows from the optimization of the functional detailed in the previous chapter. Computation of the molecules did not involve multiple iterations. As such, the total run time was minimal compared to the original optimization. Initial calculations were done by Dwayne John with Dr. Jing

Kong continuing with other functionals and adjusting specifications for optimal results. The results computed here used the GA_base framework for ease of adding and organizing molecular sets. GA_base also provides flexibility and simplicity in changing job specifications, such as initial functional and convergence criterion, for each molecule. Computation involving xTron was done on Roughshod, the temporarily resurrected (now fully restored) Babbage cluster.

5.3 Methods and Results

For a comprehensive comparison, the results of KP16 functional are compared to a variety of functionals. Hartree-Fock (HF) is included because B13 and KP16/B13 include full HF exchange. B3LYP is included as a popular hybrid GGA (generalized gradient approximation) with a few parameters. M06 is included for representation of a multi-parameter, extensively parameterized DFT method. PSTS is included as an example of a local hybrid functional that also includes nondynamic correlation. B05 is included as the predecessor for B13 and KP16/B13. The recent SCAN exchange/-correlation is included as an example of nonempirical pure meta-GGA. Another recent rung-3.5 functional HF-HGPBE is also included as new practical alternative to approaching nondynamic correlation. Lastly, B3tLap is included with B3LYP-like hybrid exchange and the meta-GGA correlation tLAP.

Table 4: Overview of Minnesota Sets [58]

Set Name	Description
SR-MGM-BE9	Single-reference main-group metal bond energies, KOH → $CH_3F + Cl$
SR-TM-BE17	Single-reference transition-metal bond energies, $MnF_2 \rightarrow Mn + 2F$
MR-MGM-BE4	Multi-reference main group metal bond energies, $MgS \rightarrow Mg + S$
MR-MGN-BE1	Multi-reference main-group non-metal bond energies, $SO_2 \rightarrow Mg + S$
MR-TM-BE13	Multi-reference transition-metal bond energies, $TiCl \rightarrow Ti + Cl$
MR-TMD-BE2	Multi-reference transition-metal dimer dissociations, $Cr_2 \rightarrow 2Cr$
IP23	Ionization potentials, $O_2 \rightarrow O_2 + e$
4pIsoE4	4p isomerization energies, $CH_3CH_2CBr_3 \rightarrow CH_2BrCHBrCH_2Br$
2pIsoE4	2p isomerization energies, $C_4H_2O \rightarrow C_4H_7OH$
IsoL6	Isomerization energies of large molecules, $(C_6H_4NH_2)_2 \rightarrow (C_6H_5NH)_2$
EA13	Electron affinities, $SH + e \rightarrow SH^-$
PA8	Proton affinities, $H_2O + H^+ \rightarrow H_3^+O$
πTC13	Thermochemistry of π-systems
HTBH38	Hydrogen transfer barrier heights, $CH_3 + H_2 \rightarrow CH_4 + H$
NHTBH38	Non-hydrogen transfer barrier heights, $CH_3 + FCl \rightarrow CH_3F + Cl$
HC7	Hydrocarbon chemistry
DC9	Difficult systems

All results used the basis set G3LargeXP [15]. The numerical grid is an atom-centered un-pruned ultra-fine grid with 128 radial and 302 angular points per shell within Becke's relative weights integration scheme [3]. All calculations are done with the self-consistent field (SCF) procedure with direct inversion of the iterative subspace (DIIS [41]) and/or geometric direct minimization (GDM [53]) for convergence determination. Computation with B05, B13, KP16/B13 and PSTS was done using the in-house code xTron. Results from other functionals not available in xTron were computed using the Q-Chem [44] and Gaussian09 [23] programs.

The individual errors associated with each molecular system is the atomization error computed by taking the difference between the energy of the molecule and the sum of the energies of the molecule's constituent atoms. The summary values of the sets are the mean absolute deviations (MAD) of the errors of the molecules. A few entries in those datasets were omitted due to failures of the SCF due to non-convergent oscillatory behavior. All Cr containing compounds are omitted because the SCF procedure did not converge for the Cr atom with B05 and KP16/B13. Elements beyond the 3rd row of the periodic table in the set IP23 are not included due to the unavailability of B13 parameters.

Table 5: Overview of Minnesota Sets MAD in kcal/mol

	HF	SCAN	B3LYP	B3TLYP	M06	PSTS	HF-HGPBE
SR-MGM-BE9	37.892	3.257	6.279	3.023	5.792	7.927	20.73
SR-TM-BE17	41.384	8.205	11.266	8.595	12.375	9.433	19.75
MR-MGM-BE4	55.313	5.958	6.053	6.991	4.431	5.528	46.653
MR-MGN-BE17	115.886	5.8	5.299	5.262	4.694	8.23	30.539
MR-TM-BE13	132.63	18.136	7.928	88.62	9.33	11.32	79.372
MR-TMD-BE2	356.293	7.235	39.8	36.697	36.358	12.94	44.694
IP23	24.631	5.381	4.397	6.419	3.48	3.715	14.863
4pIsoE4	6.389	3.16	4.024	4.607	2.041	2.394	7.499
2pIsoE4	3.9	2.582	4.614	5.334	1.776	3.308	4.121
IsoL6	3.57	1.123	2.626	7708.8	2.145	3.471	4.4
EA13	36.248	3.038	2.258	5.009	1.759	3.044	17.377
PA8	3.296	1.306	1.308	14.565	2.177	3.432	15.717
TC13	10.166	7.164	5.732	8.554	4.285	8.484	23.132
HTBH38-08	15.118	7.458	4.378	2.729	2.431	4.508	9.818
NHTBH38-08	13.672	7.49	4.563	3.031	2.6	6.211	14.999
HC7-11	15.114	6.756	16.312	19.88	2.289	9.768	42.841
DC9-12	261.223	13.519	17.931	17.077	10.634	24.406	75.425

Table 6: Overview of Minnesota Sets MAD in kcal/mol continued

	B05	B13	KP16/B13
SR-MGM-BE9	3.846	4.507	4.416
SR-TM-BE17	8.02	7.105	7.798
MR-MGM-BE4	6.421	20.624	9.78
MR-MGN-BE17	4.44	4.836	5.92
MR-TM-BE13	75.687	16.143	10.779
MR-TMD-BE2	14.62	26.31	12.641
IP23	3.174	4.18	3.732
4pIsoE4	16.422	5.435	3.852

Analysis of the results begins with significant multireference character systems, denoted by the MR prefix of the set name. B13 and KP16/B13 are both designed to handle nondynamic correlation, so the expectation is good performance for these MR sets. However, the results are mixed. B13 outperforms KP16/B13 for main-group nonmetals, labeled as MR-MGN-BE17, while KP16/B13 does better for the other three sets. B3LYP performs the best for binding energies of some transition-metal compounds (MR-TM-BE13 set). PSTS, the local hybrid method, consistently reports good results, and SCAN, the pure meta-GGA functional, performs well except for the MR-TM-BE13 set.

Of the four multireference sets, the most difficult set is MR-TM-BE2 due to the inclusion of Cr_2. Cr_2 is well-known as a difficult system exhibiting both static and dynamic correlation [44]. Due to the convergence failure for Cr_2 by the B13 and KP16/B13 functionals, the results of the set MR-TM-BE2 are not reported in the aggregate table but in the separate Table 7.

Table 7: Binding energy (kcal/mol) predictions of Cr_2 and V_2 with various methods. Values are in kcal/mol. 'Ref' stands for the reference value for comparison available from the Minnesota Set[58].

	Ref	HF	SCAN	B3LYP	B3tLap	M06	PSTS	HF-HGPBE
Cr_2	36.02	-602.50	-35.60	-45.42	-89.79	-47.42	-25.64	-117.07
V_2	64.20	-292.09	56.97	24.40	27.50	27.84	51.26	19.51

Table 8: Binding energy predictions continued. The * denotes an SCF convergence failure.

	Ref	B05	B13	KP16/B13
Cr_2	36.02	*	43.52	*
V_2	64.20	49.58	90.51	51.56

As the values in Table 7 show, the B13 functional performs the best by a significant margin, possibly due to the element-wise parameterization. As for V_2, SCAN, KP16/B13, and PSTS perform well. M06 performs similarly to B3LYP, possibly demonstrating the limit of multivariable parameterizations. A possible explanation for low performance of B13 and KP16/B13 is the underestimation of the nondynamic correlation of intermediate strength. This is illustrated by the dissociation curves of covalent homonuclear diatomics in the previous work introducing the KP16/B13 functional [30].

As for the rest of the sets, B13 and KP16/B13 have larger errors with single-reference transition-metal bond energies (SR-TM-BE17), hydrocarbons (HC7), and difficult cases (DC9). Reaction barriers result in decent predictions from B13 and KP16 which are significantly better than B3LYP. B05, B13, and KP16/13 all perform better than M06 for hydrogen transfers with B05 the best of the three and B13 in second. The B05 results are consistent with previous reaction barrier benchmarks from Dickson and Becke [18] and Liu et al.[35]. However, KP16/B13 performs the best for nonhydrogen transfers among all methods tested. For SR-TM-BE17, the mainstream functionals, M06 and B3LYP, do not perform as well as the rest (B05, PSTS, B13, KP16/B13, SCAN, and B3tLap). DC9, in particular, has poor results for all methods evaluated here with M06 giving the smallest error. For the hydrocarbons (HC7), SCAN and PSTS perform well where intermediate-range between atoms and VDW weak interactions are significant properties [18]. The SCAN positive result for van der Waals has been discussed previously by Sun et. al [46]. For the rest of the sets, B13 and KP16/B13 perform similarly to each other, reasonably close to B3LYP, and mostly under-performing against B05. As a whole, B05 performs best for four sets while M06 has six sets. Recent works have shown B05 to yield accurate results for dipole moments [16] and charge-transfer complexes [7] using the variational extension. The fact that B05 contains 100% Hartree-Fock exchange with four parameters further shows possible improvement of DFT functionals at fundamental levels.

5.4 Conclusion

To summarize, the recently developed functionals for strong/nondynamic correlation, B13 and KP16/B13, have been benchmarked with a subset of molecular systems from the Minnesota 2015 dataset [58]. They were compared against the mainstream functionals B3LYP and M06. Also included are less popular functionals, such as B05, PSTS, and more recently developed SCAN[46]. The results show that B05, B13, and KP16/B13, functionals with 100% HF exchange, recover the majority of correlation where it is significant as indicated by their relatively low error as compared to the large errors from HF. B05, B13, and KP16/B13 also perform well for reaction barriers

(HTBH38 and NHTBH38) but do not definitively outperform for multireference systems. Overall, B05, B13, and KP16/B13 are competitive to B3LYP, and B05 is even competitive to M06 for most sets despite containing 100% HF exchange. This work and the previous work involving KP16/B13 demonstrates the possibility of handling correlation of all strengths with only three empirical parameters and without relying on excessive parameterization. Future projects may include comparison against multiconfiguration methods, for both accuracy and computational time. There are also parameters within KP16/B13 that are available for optimization, such as a parallel and an opposite correlation coefficient, to further improve the functional.

6 Summary and Conclusion

The body of work presented here demonstrates the success of a genetic algorithm implementation, GA_base, I implemented for the purposes of developing new DFT models. To that end, two damping-dispersion schemes and the general purpose functional KP16/B13 were proposed, optimized, and tested. The three-body damping dispersion schemes successfully calculated three-body dispersion with damping of molecules in close proximity. The KP16/B13 functional was also developed and optimized for general purpose molecules but maintains accuracy for nondynamic correlation and fractional properties. Extension of the previous work to benchmarking for a wide variety of chemical sets showed promising results, particularly when considering a limited training set and the parameterization of only 3 values. These details present possible future improvements through optimization of KP16/B13 and computational chemical functionals.

7 Publications

E. Proynov, F. Liu, Z. Gan, M. Wang, J. Kong. “Density-functional approach to the three-body dispersion interaction based on the exchange dipole moment.” In *J. Chem. Phys.*, 143.8:084125 (August 2015).

C. M. Klinger, L. Paoli, R. J. Newby, M. Y.-W. Wang, H. D. Carroll, J. D. Leblond, C. J. Howe, J. B. Dacks, C. Bowler, A. B. Cahoon, R. G. Dorrell, E. Richardson. “Plastid Transcript Editing across Dinoflagellate Lineages Shows Lineage-Specific Application but Conserved Trends.” In: *Genome Biology And Evolution*, 10.4 (2018), p. 10191038.

8 References

[1] J. G. Angyan. “On the exchange-hole model of London dispersion forces”. In: *Journal of Chemical Physics* 127.2 (2007), p. 024108. ISSN: 00219606.

[2] R. J. Bartlett and M. Musial. “Coupled-cluster theory in quantum chemistry”. In: *Reviews of Modern Physics* 79 (2007).

[3] A. D. Becke. “A multicenter numerical integration scheme for polyatomic molecules”. In: *The Journal of Chemical Physics* 88.4 (1988), pp. 2547–2553.

[4] A. D. Becke. “Density functionals for static, dynamical, and strong correlation”. In: *The Journal Of Chemical Physics* 138.7 (2013), p. 074109. ISSN: 1089-7690.

[5] A. D. Becke. “Real-space post-HartreeFock correlation model.” In: *Journal of Chemical Physics* 122.6 (2005), N.PAG. ISSN: 00219606.

[6] A. D. Becke and E. R. Johnson. “Exchange-hole dipole moment and the dispersion interaction: High-order dispersion coefficients.” In: *Journal of Chemical Physics* 124.1 (2006), p. 014104.

[7] A.D. Becke, S.G. Dale, and E.R. Johnson. “Communication: Correct charge transfer in CT complexes from the Becke’05 density functional”. In: *Journal of Chemical Physics* 148.21 (2018). ISSN: 00219606.

[8] A.D. Becke and E.R. Johnson. “A density-functional model of the dispersion interaction”. In: *Journal of Chemical Physics* 123.15 (2005). ISSN: 00219606.

[9] W. Cencek, K. Patkowski, and K. Szalewicz. “Full-configuration-interaction calculation of three-body nonadditive contribution to helium interaction potential”. In: *The Journal of Chemical Physics* 131 (2009).

[10] W. Cencek, M. Jeriorska, O. Akin-Ojo, and K. Szalewicz. “Three-body Contribution to the helium interaction potential”. In: *The Journal of Chemical Physics A* 111 (2007).

[11] W. Cencek, K. Szalewicz, G. Garberoglio, A.H. Harvey, and M.O. McLinden. “Three-body nonadditive potential for argon with estimated uncertainties and third virial coefficient”. In: *Journal of Physical Chemistry A* 117.32 (2013), pp. 7542–7552. ISSN: 10895639.

[12] W. Cencek, G. Garberoglio, A. H. Harvey, M. O. McLinden, and K. Szalewicz. “Three-body nonadditive potential for argon with estimated uncertainties and third virial coefficient”. In: *The Journal of Chemical Physics A* 117 (2013).

[13] G. Chalasinski and M. M. Szczesniak. “Origins of structure and energetics of van der Waals clusters from ab initio calculations”. In: *Chemical Reviews* 7 (1994), p. 1723. ISSN: 0009-2665.

[14] A. J. Cohen, M. Paula, and Y. Weitao. “Insights into Current Limitations of Density Functional Theory”. In: *Science* 5890 (2008), p. 792. ISSN: 00368075.

[15] L. A. Curtiss, P. C. Redfern, and K. Raghavachari. "Gaussian-4 theory". In: *Journal of Chemical Physics* 126.8 (2007), p. 084108. ISSN: 00219606.

[16] S. G. Dale, E. R. Johnson, and A. D. Becke. "Interrogating the Becke'05 density functional for non-locality information". In: *The Journal Of Chemical Physics* 147.15 (2017), p. 154103. ISSN: 1089-7690.

[17] *Density Functional (DFT) methods*. `http://gaussian.com/dft/`. Last updated on: 29 June 2018, Accessed: 2018-11-08.

[18] R. M. Dickson and A. D. Becke. "Reaction barrier heights from an exact-exchange-based density-functional correlation model". In: *Journal of Chemical Physics* 11 (2005), p. 111101. ISSN: 0021-9606.

[19] M. Dion, H. Rydberg, E. Schroder, D. C. Langreth, and B. I. Lundqvist. "van der Waals density functional for general geometries". In: *Physical Review Letters* 92.24 (2004), p. 246401. ISSN: 0031-9007.

[20] R. M. Eisberg and R. Resnick. *Quantum Physics of Atoms, Molecules, Solids, Nuclei, and Particles*. New York : Wiley, c1985., 1985. ISBN: 047187373X.

[21] A. P. Engelbrecht. *Computational Intelligence : an Introduction*. Chichester, England ; Hoboken, NJ : John Wiley & Sons, c2007., 2007. ISBN: 0470035617.

[22] M. Filatov and S. Shaik. "A spin-restricted ensemble-referenced Kohn-Sham method and its application to diradicaloid situations". In: *Chemical Physics Letters* 5-6 (1999), p. 429. ISSN: 0009-2614.

[23] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, . Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox. *Gaussian09 Revision E.01*. Gaussian Inc. Wallingford CT 2009.

[24] P. Gori-Giorgi, J. Toulouse, and A. Savin. "A short-range correlation energy density functional with multi-determinantal reference". In: *Theoretical Chemistry Accounts* 114.4-5 (2005), pp. 305 –308. ISSN: 1432881X.

[25] B. Jger, R. Hellmann, E. Bich, and E. Vogel. "Ab initio virial equation of state for argon using a new nonadditive three-body potential". In: *The Journal Of Chemical Physics* 135.8 (2011), p. 084308. ISSN: 1089-7690.

[26] E. R. Johnson and A. D. Becke. “A post-Hartree-Fock model of intermolecular interactions: Inclusion of high-order corrections”. In: *The Journal Of Chemical Physics* 124 (2006).

[27] F. O. Kannemann and A. D. Becke. “van der Waals interactions in density-functional theory: Intermolecular complexes”. In: *Journal of Chemical Theory and Computation* 6 (2010).

[28] M.R. Kennedy, A.R. McDonald, A.E. DePrince III, M.S. Marshall, C.D. Sherrill, and R. Podeszwa. “Resolving the three-body contribution to the lattice energy of crystalline benzene: Benchmark results from coupled-cluster theory”. In: *The Journal of Chemical Physics* 140 (2010).

[29] W. Koch and M. C. Holthausen. *A Chemist’s Guide to Density Functional Theory.* Weinheim ; Chichester : Wiley-VCH, c2000., 2000. ISBN: 3527299181.

[30] J. Kong and E. Proynov. “Density Functional Model for Nondynamic and Strong Correlation”. In: *Journal of Chemical Theory and Computation* 12 (2016).

[31] J. Kong, Z. Gan, E. Proynov, M. Freindorf, and T. R. Furlani. “Efficient computation of the dispersion interaction with density functional theory”. In: *Physical Review A* 79 (2009).

[32] A. Kumar and AJ. Thakkar. “Dipole oscillator strength distributions with improved high-energy behavior: dipole sum rules and dispersion coefficients for Ne, Ar, Kr, and Xe revisited”. In: *The Journal of Chemical Physics* 132 (2010).

[33] C. Lee, W. Yang, and R. G. Parr. “Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density”. In: *Physical Review B* 37 (2 1988), pp. 785–789.

[34] O. A. von Lilienfeld and A. Tkatchenko. “Two- and three-body interatomic dispersion energy contributions to binding in molecules and solids”. In: *Journal of Chemical Physics* 132.23 (2010). ISSN: 00219606.

[35] F. Liu, E. Proynov, J. Yu, T. R Furlani, and J. Kong. “Comparison of the performance of exact-exchange-based density functional methods”. In: *The Journal Of Chemical Physics* 137.11 (2012), p. 114104. ISSN: 1089-7690.

[36] R. T. McGibbon, A. G. Taube, A. G. Donchev, K. Siva, F. Hernndez, C. Hargus, K. Law, J. L. Klepeis, and D. E. Shaw. “Improving the accuracy of Mller-Plesset perturbation theory with neural networks”. In: *The Journal Of Chemical Physics* 147.16 (2017), p. 161725. ISSN: 1089-7690.

[37] J. P. Perdew, V. N. Staroverov, J. Tao, and G. E. Scuseria. “Density functional with full exact exchange, balanced nonlocality of correlation, and constraint satisfaction”. In: *Physical Review A* 78 (5 2008), p. 052513. DOI: 10.1103/PhysRevA.78.052513.

[38] R. Podeszwa and K. Szalewicz. “Three-body symmetry-adapted perturbation theory based on Kohn-Sham description of the monomers”. In: *The Journal of Chemical Physics* 126.19 (2007), p. 194101. DOI: 10.1063/1.2733648.

[39] E. Proynov, F. Liu, and J. Kong. “Analyzing effects of strong electron correlation within Kohn-Sham density-functional theory”. In: *Physical Review A* 3A (2013), p. 032510. ISSN: 1050-2947.

[40] E. Proynov, F. Liu, Z. Gan, M. Wang, and J. Kong. “Density-functional approach to the three-body dispersion interaction based on the exchange dipole moment”. In: *The Journal Of Chemical Physics* 143.8 (2015), p. 084125. ISSN: 1089-7690.

[41] P. Pulay. “Convergence acceleration of iterative sequences. the case of scf iteration”. In: *Chemical Physics Letters* 73.2 (1980), pp. 393 –398. ISSN: 0009-2614.

[42] *Q-Chem 4.3 User's Manual: Density Functional Theory*. `http://www.q-chem.com/qchem-website/manual/qchem43_manual/sect-DFT.html`. Accessed: 2018-11-08.

[43] A. Otero-de-la Roza and E.R. Johnson. “Many-body dispersion interactions from the exchange-hole dipole moment model”. In: *The Journal Of Chemical Physics* 138 (2013).

[44] Y. Shao et al. *Advances in molecular quantum chemistry contained in the Q-Chem 4 program package*. 2014.

[45] K. Sharkas, J. Toulouse, and A. Savin. “Double-hybrid density-functional theory made rigorous”. In: *Journal of Chemical Physics* 134.6 (2011). ISSN: 00219606.

[46] J. Sun, R. C. Remsing, Y. Zhang, Z. Sun, A. Ruzsinszky, H. Peng, Z. Yang, A. Paul, U. Waghmare, X. Wu, M. L. Klein, and J. P. Perdew. “SCAN: An Efficient Density Functional Yielding Accurate Structures and Energies of Diversely-Bonded Materials.” In: *ArXiv e-prints* (Nov. 2015). eprint: `1511.01089`.

[47] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. First. Dover Publications, Inc., 1996.

[48] K. Szalewicz. “Symmetry-adapted perturbation theory of intermolecular forces”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 2.2 (), pp. 254–272. DOI: `10.1002/wcms.86`.

[49] K. T. Tang. “Dynamic polarizabilities and van der Waals coefficients”. In: *Physical Review* 177 (1969).

[50] L. Y. Tang, Z. C. Yan, T. Y. Shi, J. F. Babb, and J. Mitroy. “The long-range non-additive three-body dispersion interactions for the rare gases, alkali, and alkaline-earth atoms”. In: *The Journal of Chemical Physics* 136 (2012).

[51] A. Tkatchenko, M. Scheffler, R. A. Distasio Jr., and M. Head-Gordon. “Dispersion-corrected Mller-Plesset second-order perturbation theory”. In: *Journal of Chemical Physics* 131.9 (2009). ISSN: 00219606.

[52] T. Tsuneda. *Density Functional Theory in Quantum Chemistry*. Springer Japan c2014., 2014. ISBN: 9784431548249.

[53] T. V. Voorhis and M. Head-Gordon. "A geometric approach to direct minimization". In: *Molecular Physics* 100.11 (2002), pp. 1713–1721.

[54] S. Wen, K. Nanda, Y. Huang, and G.J.O. Beran. "Practical quantum mechanics-based fragment methods for predicting molecular crystal properties". In: *Physical Chemistry Chemical Physics* 21 (2012), pp. 7578–7590. ISSN: 14639076.

[55] S. Wen, K. Nanda, Y. Huang, and G.J.O. Beran. "Practical quantum mechanics-based fragment methods for predicting molecular crystal properties". In: *Physical Chemistry Chemical Physics* 14.21 (2012), pp. 7578–7590. ISSN: 14639076.

[56] Z. Yan and D. G. Truhlar. "Comparative DFT study of van der Waals complexes: Rare-gas dimers, alkaline-earth dimers, zinc dimers, and zinc-rare-gas dimers". In: *Journal of Physical Chemistry A* 15 (2006), p. 5121. ISSN: 1089-5639.

[57] Y. Zhao and D. G. Truhlar. "The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals". In: *Theoretical Chemistry Accounts* 1-3 (2008), p. 215. ISSN: 1432-881X.

[58] Zhao, Y. and Truhlar, D. "The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals". In: *Theoretical Chemistry Accounts* 120 (2008).

9 Appendix 1 - Software Codes

Listing 1: Genetic Algorithm code: gen_new_pop.pl

```
#!/bin/perl
###########################################################################################
# Created by Matthew Wang
# This script generates a new population. It requires 3 or 4 parameters, two of which
#  are input parameters and 1 to specify the output. The 4 parameter is optional and
#  specifies the previous population and it's fitness/error. This is then used to do
#  the crossover step of a genetic algorithm.
#
# The genetic algorithm implemented here contains 3 version of crossover and 3 versions
#  of mutation. Rates for each can be adjusted in the code. Future improvements may
#  make these options more accessible. There also contains a check so no duplicate
#  parameter sets are run.
#
# Cmd line args:
#  1. Population size
#  2. Parameter file
#  3. File to put the new population into
#  (4. Previous population file)
# Output:
#  New population inside the (3.)
###########################################################################################

use strict;
use warnings;

my ($pop_size, $param_file, @para_min, @para_max, $pop_prev_file, $pop_new_file, @para_type);
my $gen_flag = 0; # 0 for completely new population, 1 for using a pre-existing population

my $num_args = $#ARGV + 1;
if($num_args == 3)
{
   print "You␣used␣3␣arguments,␣creating␣a␣new␣population\n";
   $gen_flag = 0;
   $pop_size = int($ARGV[0]);
   $param_file = $ARGV[1];
   $pop_new_file = $ARGV[2];
   chomp($pop_new_file);
}
elsif($num_args == 4)
{
   print "You␣used␣4␣arguments,␣creating␣a␣new␣population␣using␣previous␣population\n";
   $gen_flag = 1;
   $pop_size = int($ARGV[0]);
   $param_file = $ARGV[1];
   $pop_new_file = $ARGV[2];
   $pop_prev_file = $ARGV[3];
   chomp($pop_prev_file);
}
else
{
   die "\nERROR:␣Requires␣either␣3␣or␣4␣parameters:\n1.␣Initial␣population␣size
2.␣Parameter␣file\n3.␣File␣to␣store␣the␣new␣population
(4.␣Previous␣population␣file␣to␣use␣for␣seeding)\n";
}
my $num_params = 0;

# Open the parameters file and read in the parameters into array
open PARAM_FILE, "<$param_file" or die "Can't␣open␣parameter␣file␣$param_file\n";
my $line = <PARAM_FILE>;
chomp($line);
$num_params = int($line);

for(my $i=0; $i<$num_params; $i++)
{
   $line = <PARAM_FILE>;
   my @params = split("␣", $line);
   # set the parameter bounds from the parameter input file
   $para_min[$i] = $params[0];
   $para_max[$i] = $params[1];
   if($params[2] =~ m/d/)
   {
      $para_type[$i] = $params[2];
   }
   elsif($params[2] =~ m/f/)
   {
      $para_type[$i] = $params[2];
   }
   else
   {
      die "Invalid␣parameter␣type\n";
   }

}
close(PARAM_FILE);


my $num_new = 1;
my $new_param_str = "";
# the parameters for the previous population and hash for the previous population as a string
#  for easy searching
my (@prev_pop_params, %prev_params);
# opening the file for the new population
open NEW_POP, ">$pop_new_file" or die "Can't␣not␣open␣new␣population␣file␣$pop_new_file\n";

# If statement for using the previous population file for crossover
if($gen_flag == 1 and -e $pop_prev_file)
{
   open POP_FILE, "<$pop_prev_file" or die
      "Can't␣open␣previous␣population␣file:␣$pop_prev_file\n";

   my ($line, $i);
   my $total_pop = 0;
   # iterate over the file and store contents for crossover and duplicate checking
   foreach $line (<POP_FILE>)
   {
      chomp($line);
      if($line !~ /^#/)     # for comments
      {
         my @param_line = split("␣", $line);
         my $tmp_num_params = @param_line;
         # Error checking, +3 for gen number, iteration number, and error at least
         if($tmp_num_params < $num_params+3)
         {
            die "Possible␣error␣in␣$pop_prev_file.␣Less␣parameters␣than␣number",
               "␣of␣parameters␣in␣$param_file:␣$num_params!\n";
         }
         if($total_pop < $pop_size)
         {
            # save the top population for crossover
            for(my $j=0; $j<$num_params; $j++)      # last column is error
            {
               # first and second column are gen number and iter number
               $prev_pop_params[$total_pop][$j] = $param_line[$j+2];
            }
         }
         # create a parameter line and add it to hash for easy searching
         my $param_line = join("␣", @param_line[2 .. $num_params+1]);
         if( exists $prev_params{$param_line} )
         {
            print "Possible␣error␣in␣$pop_prev_file,␣parameter",
               "␣set␣$param_line␣already␣exists!\n";
         }
         else
         {
            $prev_params{$param_line} = 0;
         }
         $total_pop = $total_pop + 1;
      }
   }

   close(POP_FILE);

   # Check to see if the total previous population is greater than 1 and
   #  the required population size is greater than 1. Cannot do crossover otherwise.
   if($total_pop > 1 && $pop_size > 1)
   {
      # prev_pop_params now has the previous population. Compute the next generation
      # calculte the number of crossovers to be done
      my $numCross = &calcCrossNum($pop_size);
      print "Number␣of␣crossovers:␣$numCross\n";
      for(my $i=0; $i < $numCross; $i++)
      {
         # choose 2 random sets from population
         my $index1 = int($pop_size*rand());
         my $index2 = int($pop_size*rand());
         # check for duplication
         while($index1 == $index2)
         {
            $index2 = int($pop_size*rand());
         }
         # calculate which dimension to cross
         my $cross_dim = int(($num_params-1)*rand()) + 1;
         my $j;

         $new_param_str = "$prev_pop_params[$index1][0]";
         # do the crossover for first child
         for($j=1; $j<$cross_dim; $j++)
         {
            $new_param_str = "$new_param_str␣$prev_pop_params[$index1][$j]";
         }
         for($j=$cross_dim; $j<$num_params; $j++)
         {
            $new_param_str = "$new_param_str␣$prev_pop_params[$index2][$j]";
         }
         # figure out if first child already exists in searched space
         if(! exists $prev_params{$new_param_str})
         {
            print NEW_POP "$new_param_str\n";
            $prev_params{$new_param_str} = 0;
            $num_new++;
         }
         $new_param_str = "$prev_pop_params[$index2][0]";
         # do the crossover for second child
         for($j=1; $j<$cross_dim; $j++)
         {
            $new_param_str = "$new_param_str␣$prev_pop_params[$index2][$j]";
         }
         for($j=$cross_dim; $j<$num_params; $j++)
         {
            $new_param_str = "$new_param_str␣$prev_pop_params[$index1][$j]";
         }
         # figure out if second child already exists in searched space
         if(! exists $prev_params{$new_param_str})
         {
            print NEW_POP "$new_param_str\n";
            $prev_params{$new_param_str} = 0;
            $num_new++;
         }
      }

      # slight perturbation code:
      my $numPerturb = int(rand()*$pop_size/3.0) + 1;
      for(my $i=0; $i < $numPerturb; $i++)
      {
         #my $popPerturb = int(rand()*$pop_size);
         my $popPerturb = $i;
         my @temp_params;
         for(my $j=0; $j < $num_params; $j++)  # copy over the parameters
         {
            $temp_params[$j] = $prev_pop_params[$popPerturb][$j];
         }

         my $numParaShift = int(rand()*$num_params/2.0)+1; # shift from 1 to half of the parameters
         for(my $j=0; $j < $numParaShift; $j++)
         {
            my $paraShift = int($num_params*rand());
            # shift between -0.5 to 0.5 times range/10.0 around parameter
            my $shift_val = ($para_max[$paraShift]-$para_min[$paraShift])/10.0*(rand() - 0.5);
            $shift_val = int(sprintf("%.5f", $shift_val));
            if($temp_params[$paraShift] + $shift_val < $para_max[$paraShift] and
               $temp_params[$paraShift] + $shift_val > $para_min[$paraShift])
            {
               $temp_params[$paraShift] = $temp_params[$paraShift] + $shift_val;
            }
         }
         $new_param_str = "$temp_params[0]";
         for(my $j=1; $j < $num_params; $j++)
         {
            $new_param_str = "$new_param_str␣$temp_params[$j]";
         }
         if(! exists $prev_params{$new_param_str})
         {
            print NEW_POP "$new_param_str\n";
            $prev_params{$new_param_str} = 0;
            $num_new++;
         }
      }
   }
}
elsif($gen_flag == 1 and (! -e $pop_prev_file))
{
   print "$pop_prev_file␣does␣not␣exist!␣Cannot␣use␣previous␣population.\n";
}

while($num_new <= $pop_size)
{
   my $val = ($para_max[0]-$para_min[0])*rand() + $para_min[0];
   $new_param_str = sprintf("%.5f", $val);
   # calculate a new organism
   for(my $j=1; $j<$num_params; $j++)
   {
      my $val = ($para_max[$j]-$para_min[$j])*rand() + $para_min[$j];
      if($para_type[$j] =~ m/d/)
      {
         $val = int($val);
      }
      if($para_type[$j] =~ m/f/)
      {
         $val = sprintf("%.5f", $val);
      }
      $new_param_str = "$new_param_str␣$val";
   }
   # figure out if randomly generated parameter set is a duplicate
   #  (low chance i know, just paranoid)
   if(! exists $prev_params{$new_param_str})
   {
      print NEW_POP "$new_param_str\n";
      $prev_params{$new_param_str} = 0;
      $num_new++;
   }
}


close(NEW_POP);
print "\nDone␣creating␣new␣population.\n";

# subroutine to calculate how many crossovers to do
# can change cross_rate for more or less crossovers
# smaller is more crossovers, larger is less
# follows an exponential curve rather than linear
sub calcCrossNum
{
   my $pop_size = shift;
   my $cross_rate = 3.0;   # 2.0 - comes out roughly a 4th

   my $random = rand();
   my $n = int($pop_size/$cross_rate*exp(-$random));
   return $n;
}
```

Listing 2: Run population code: run_pop.sh

```
#!/bin/bash
# Created by Matthew Wang
#############################################################
# - This is the main driver file for the Genetic algorithm
#   for running jobs using input parameters
# - This script does:
#  1. Set up necesary directories
#  2. Goes through each line of the population, grabs
#     the parameters and calls another script
#     (placeholder called job_setup.sh)
#     to set up input files
#  3. Constructs a job commands txt file for execution
#     Ex. exe test.in test.out
#  4. Runs the jobs/commands. Commands in run_script.sh to
#     be executed in current env or in commands to be
#     used in other scripts.
# - Ex: ./run_pop.sh pop_init.txt
#
#
# - job_setup.sh is a placeholder for a script that must take at minimum
#  1. A file with the parameter sets to use for that run
#  2. The path to the current generation for program use purposes
#  3. The run script which is the script which will run the commands
#
#  Any further arguments needed can be added to the line that calls
#  run script and modified in the run script. These 3 arguments are the
#  3 I've deemed necessary so far.
#
#  The setup script will then populate the run script with any necessary
#  set up or env variables for the executable. It will also do any set up
#  necessary for the job into the iteration path directory. All executable
#  calls will go into the $run_script .
#############################################################

# Command line argument check
if [[ $# -ne 1 ]]
then
   echo -e "run_pop.sh␣requires␣input␣argument␣for␣population␣of␣parameters!"
   echo -e "Ex:␣./run_pop.sh␣pop.txt\n"
   exit
fi

# The file with the population parameters
pop_file=$1

# Config file
config_file="config.txt"
if [[ ! -e $config_file ]]
then
   echo "In␣run_pop.sh:␣$config_file␣does␣not␣exist!"
   exit
fi
```

```
# Home path
ga_path=$(grep "ga_path" $config_file | awk '{print $2}')
opt_scheme=$(grep "opt_scheme" $config_file | awk '{print $2}')
home_path="$ga_path/$opt_scheme"
if [[ -z $ga_path || -z $opt_scheme || -z $home_path ]]
then
   echo "ERROR: home_path, ga_path, or opt_scheme is not set in run_pop.sh!"
   exit
fi

# Where inputs and outputs will go
io_path="$home_path/IO_files"

# The current generation number
gen_num=1
# This is where the inputs and outputs of the new generation will go
gen_path="$io_path/gen$gen_num"
while [ -d $gen_path ]
do
   gen_num=$(($gen_num+1))
   gen_path="$io_path/gen$gen_num"
done
echo "Generation $gen_num"

# Job scripts path, can make job path to be separate from the generation directory
#  This directory should contain scripts and files necessary for the whole job.
job_path="$gen_path/job_files"

#set up directories
mkdir -p $gen_path
mkdir -p $job_path

# copy the generation parameter information into the folder
cp $pop_file $gen_path/population.txt

# Create run script
run_script="$job_path/run_script.sh"
#job_commands="$job_path/job_commands.txt"
echo -e "#!/bin/bash\n\ndate\n" > $run_script

# call script that creates the input files and sets up necessary folders for the specific job
$home_path/job_setup.sh $pop_file $gen_path $run_script

echo -e "echo \"Done with generation $gen_num\"" >> $run_script

echo -e "\ndate\n" >> $run_script

# calculate difference
date1=`date "+%d %H %M %S"`

num_sets=$(wc -l $pop_file | awk '{print $1}')
for (( i=1; i<=$num_sets; i++ ))
do
       cat $gen_path/scripts_configs/config$i/commands >> $gen_path/scripts_configs/commands
done

####################################
# One of the below must be chosen!
####################################
## This is used to run on all roughshod nodes. Specific to roughshod or some other beowulf cluster
#perl multi-node-run.pl $gen_path/scripts_configs

## For SGE job submission
#./submit_jobs.sh $gen_path/scripts_configs $ga_path/header.sge

## The cmds below run the run script in the current environment
#chmod 755 $run_script
#echo "Running"
#$run_script

date2=`date "+%d %H %M %S"`
timeDiff=`echo "$date1 $date2" | awk '{printf "%f", (\
$5-$1)*24.0 + $6-$2 + \
($7-$3)/60.0 + ($8-$4)/(60.0*60.0)}'`
echo "$date1"
echo "$date2"
echo "Time job took to complete:  $timeDiff hrs"
echo "Finished with generation $gen_num"
```

Listing 3: Calculates the error of a population: err_calc.sh

```
#!/bin/bash
# Can either take a specific generation number in io_path to calculate the error
#  for or will iterate through all the generations in IO_files
# Will call calc_gen_error.sh or user specified script to create gen_errors.txt
#  in the specified generation folder. It will then add these to the all_pop
#  file which will store all the errors. Sort is called to keep lowest errors at
#  the top. THE ONLY STIPULATION IS THAT CALC_GEN_ERROR WILL CREATE A GEN_ERRORS.TXT
##############################################

config_file="config.txt"
if [[ ! -e $config_file ]]
then
   echo "In␣run_pop.sh:␣$config_file␣does␣not␣exist!"
   exit
fi

# Home path
ga_path=$(grep "ga_path" $config_file | awk '{print $2}')
opt_scheme=$(grep "opt_scheme" $config_file | awk '{print $2}')
home_path="$ga_path/$opt_scheme"
if [[ -z $ga_path || -z $opt_scheme || -z $home_path ]]
then
   echo "ERROR:␣home_path,␣ga_path,␣or␣opt_scheme␣is␣not␣set␣in␣err_calc.sh!"
   exit
fi

# Where inputs and outputs are
io_path="$home_path/IO_files"

# The population files
all_pop="$home_path/pop_all.txt"
all_pop_sorted="$home_path/pop_all_sorted.txt"

if [[ "$#" -eq 1 ]]
then
   gen="$io_path/gen$1"
   $home_path/calc_gen_error.sh $gen
else
   for i in $io_path/gen*
   do
      if [ ! -e $i/gen_errors.txt ]
      then
         echo "Calculating␣error␣for␣$i"
         # Script to calculate the error for a specific generation below
         # Can be shell or perl script or anything else
         #$home_path/calc_gen_error.sh $i
         perl $home_path/calc_gen_error.pl $i
         cat $i/gen_errors.txt $all_pop > $home_path/temp
         mv $home_path/temp $all_pop
#     else
#        echo -e "gen_errors.txt alread exists in $i.
#Please remove comments if you would like to recalculate error\n"
      fi
   done

   # sort by the last column
   num_cols=$(head -n 1 $all_pop | awk '{printf NF}')
   sort -g -k$num_cols $all_pop > $home_path/temp
   cat $home_path/pop_label.txt $home_path/temp > $all_pop_sorted
   rm $home_path/temp
#  echo -e "Finished calculating error\n"
fi
```

10 Appendix 2 - Supplementary Material for Non-additive Three-body Dispersion

Table 9: The table reports non-additive three-body intermolecular dispersion energy (in kcal/mol) with the normalized distances for each benzene trimer for the different methods discussed previously. [a] Results from Sherrill et. al. [28] from E(CCSD(T))-E(MP2) to isolate the non-additive three-body dispersion energy. The four underlined trimers are the outliers omitted from the optimization procedure. [b]Nondamped version of the present method. [c] Scheme A with original XDM parameter values. d Scheme B with original XDM parameter values. Last two columns are the optimized parameter models.

Trimer	R_{ABC}	E(CCSD(T))- E(MP2)[a]	$E_{int}^{(3)}$ nodamp[b]	Scheme A no-opt[c]	Scheme B no-opt[d]	Scheme A Opt	Scheme B Opt
0001	1.000000	0.029143	0.042472	0.011302	0.035454	0.030921	0.037711
0014	1.05966	0.056103	0.087645	0.01646	0.054907	0.055691	0.060746
0011	1.08245	0.056212	0.089655	0.016473	0.055847	0.056811	0.062041
0012	1.08517	0.068995	0.100703	0.018191	0.061375	0.063291	0.068318
0031	1.19685	0.01786	0.02352	0.007096	0.020753	0.017755	0.021723
0009	1.25707	0.098341	0.127462	0.020073	0.069255	0.076045	0.078306
0043	1.47483	-0.00714	-0.00967	-0.00373	-0.00954	-0.007802	-0.009598
0038	2.10629	0.018711	0.02715	0.008331	0.023495	0.020531	0.024673
0055	2.28084	-0.0027	-0.00049	0.000168	-0.00045	-0.000183	-0.000469
0051	2.55225	0.002067	0.00428	0.001523	0.003922	0.003330	0.004025
0081	2.67368	-0.00587	-0.00314	-0.00076	-0.002990	-0.002235	-0.003040
0086	2.77877	-0.00358	-0.00203	-0.00041	-0.001930	-0.001439	-0.001965
0094	2.77878	-0.00358	-0.00203	-0.00041	-0.001930	-0.001439	-0.001965
0049	2.82250	-0.00513	-0.00289	-0.00012	-0.002160	-0.001587	-0.002329
0057	2.8225	-0.00512	-0.00284	-0.00011	-0.002120	-0.001555	-0.002286
0073	2.94338	0.006415	0.015545	0.005551	0.014427	0.012252	0.014835
0045	3.29523	-0.02227	-0.02047	-0.00486	-0.01731	-0.014124	-0.018165
0052	3.52073	-0.00534	-0.00731	-0.00045	-0.00483	-0.003800	-0.005302
0079	3.54715	0.000625	0.003038	0.001028	0.002717	0.002280	0.002775
0080	4.16483	-0.00429	-0.00467	-0.00035	-0.00345	-0.002568	-0.003745
0103	4.52322	0.002849	0.007594	0.002464	0.006972	0.005725	0.007121
0164	4.94626	-0.0008	-0.00064	-0.00024	-0.00064	-0.000506	-0.000638
0109	5.16552	0.003006	0.005796	0.00217	0.005433	0.004594	0.005540
0118	5.55052	0.001201	0.003617	0.001856	0.003566	0.003147	0.003585
0139	5.83854	-0.00461	-0.00068	-0.00026	-0.00066	-0.000551	-0.000665
0121	5.89480	0.003097	0.005797	0.00267	0.005666	0.004873	0.005711
0140	5.93042	-0.00475	-0.00319	-0.00109	-0.00308	-0.002458	-0.003112
0145	6.09161	-0.00353	0.000915	0.000233	0.000853	0.000635	0.000866
0177	6.15088	-0.00757	-0.0054	-0.00229	-0.00533	-0.004440	-0.005361
0115	6.63216	0.004465	0.005545	0.002291	0.005298	0.004467	0.005366
0116	6.81351	0.002356	0.002939	0.001537	0.002902	0.002594	0.002920
0163	6.97849	-0.00258	-0.00348	-0.00127	-0.00339	-0.002742	-0.003421
0186	7.18777	0.000583	0.001629	0.001012	0.001625	0.001512	0.001628
0141	7.30137	-0.00549	-0.00509	-0.00139	-0.00469	-0.003657	-0.004792
0235	7.30551	-0.0006	-0.00088	-0.00039	-0.00087	-0.000723	-0.000876
0230	7.68114	-0.00579	-0.00415	-0.00184	-0.00411	-0.003440	-0.004124
0183	7.69937	0.003407	0.004922	0.002261	0.004816	0.004121	0.004851
0175	7.72338	-0.00935	-0.007	-0.00263	-0.00681	-0.005561	-0.006871
0208	7.94617	0.001031	0.002205	0.001095	0.002191	0.001874	0.002197
0184	8.03238	0.003476	0.004899	0.002155	0.004776	0.004036	0.004814

0194	8.13891	0.000949	0.001662	0.000986	0.001659	0.001522	0.001664
0181	8.13891	0.000949	0.001643	0.000977	0.001639	0.001506	0.001645
0249	9.02144	-0.00134	-0.00125	-0.00047	-0.00123	-0.000986	-0.001236
0210	9.23894	0.001406	0.00266	0.001258	0.002634	0.002230	0.002644
0229	9.54984	-0.00874	-0.00527	-0.00212	-0.00516	-0.004252	-0.005197
0236	9.97271	-0.00182	-0.00242	-0.001	-0.00238	-0.001965	-0.002394
0206	10.08848	0.004491	0.003807	0.00166	0.003725	0.003120	0.003750
0207	10.08848	0.004491	0.003807	0.00166	0.003725	0.003120	0.003750
0247	10.17288	-0.00176	-0.00192	-0.00074	-0.00189	-0.001533	-0.001899
0285	12.35459	0.000584	0.000589	0.000431	0.000589	0.000560	0.0005892
0283	12.89706	0.0018	0.002184	0.001312	0.00217	0.001976	0.002175
0263	13.37153	0.000783	0.001143	0.000617	0.001137	0.000998	0.001139
0297	13.48999	0.000576	0.000624	0.00037	0.000622	0.000559	0.000623
0296	13.51446	0.000695	0.000862	0.000467	0.000858	0.000749	0.000860
0333	14.19470	0.000215	0.001129	0.000646	0.001125	0.000992	0.001127
0319	14.31483	-0.00083	-0.00064	-0.00029	-0.00064	-0.000534	-0.000640
0334	14.51048	-2.2E-05	0.000891	0.000559	0.000889	0.000812	0.000890
0261	14.65563	0.001869	0.001683	0.000787	0.001662	0.001401	0.001669
0260	14.78262	0.001965	0.001979	0.000929	0.001957	0.001654	0.001965
0355	15.11490	0.000432	0.000961	0.000513	0.000958	0.000829	0.000959
0298	15.40517	0.001241	0.000754	0.000369	0.00075	0.000633	0.000752
0295	15.43960	0.000786	0.000871	0.000418	0.000866	0.000729	0.000868
0259	15.83553	0.00142	0.001495	0.000696	0.001478	0.001242	0.001484
0320	15.87463	-0.00081	-0.0011	-0.0005	-0.00109	-0.000913	-0.001093
0331	19.10939	0.002531	0.002297	0.001112	0.002263	0.001923	0.002272
0332	20.08028	0.002772	0.002244	0.001087	0.002202	0.001888	0.0022125

11 Appendix 3 - Supplementary Material DFT Model for Nondynamic Correlation

The diatomic molecules are listed as: H_2, N_2, F_2, O_2, S_2, P_2, Cl_2, HF, CO, NO, PN, CN, NH, CS, CH, OH, HCl, SiO, LiF, MgS, ClF, ClO, Li_2, LiH, SO, Si_2.
Polytomic molecules are listed as: HCN, H_2O, H_2S, CO_2, NH_3, PH_3, N_2O, H_2O_2, SiH_4, CH_4, C_2H_2, C_2H_4, C_2H_6, H_2CO, CH_3OH, C_6H_6, C_4H_6_buta, C_4H_5N_pyrol, BF_3, CF_4, CHF_3, C_5H_5N_pyrid , CH_2OH, $AlCl_3$, BCl_3, C_2Cl_4, C_2H_4O, CCH, CCl_4, CH_2a, CH_2b, CH_3Cl, CH_3CN, CH_3CO, CH_3SH, CH_3NH_2, CH_3NO_2, HCO, N_2H_4, Si_2H_6, SiH_2a, SiH_2b, SO_2.
Fractional spin atoms are listed as: C_frac, Cl_frac F_frac, H_frac, N_frac, O_frac, S_frac, Si_frac.
Geometries of the above molecules and atoms can be furnished upon request.