

**GLOBAL OPTIMIZATION STRATEGIES FOR MOLECULES  
ENCAPSULATED IN WATER CLUSTERS**

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE  
UNIVERSITY OF HAWAII AT MĀNOA IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

CHEMISTRY

May 2013

By

Morgyn T. Stryker

Thesis Committee:

John D. Head, Chairperson

Kristin Kumashiro

Ho Leung Ng

Keywords: Global Optimization, Basin-Hopping, water clusters

UMI Number: 1523685

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1523685

Published by ProQuest LLC (2013). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 - 1346



## ACKNOWLEDGEMENTS

Thank you to thank my supervisor Prof. John Head for his knowledge, patience, and providing insight with me for this project.

I would also like to thank the Hawaii Open Supercomputing Center (HOSC).

Furthermore, thank you to the committee Prof. Kristin Kumashiro and Prof. Ho Ng for being on my committee and for your time.

## ABSTRACT

A discrete solvent method has been developed for a cluster consisting of a finite number of solute and solvent molecules that is combined with a Basin-Hopping (BH) Monte Carlo algorithm to search for the lowest energy or global minimum structure for the solute-solvent system. The BH algorithm is written in the Python programming language and runs in tandem with the General Atomic and Molecular Electronic Structure System (GAMESS) Quantum Chemistry software. Cycles of the BH algorithm was run using a computationally fast Density Functional Theory (DFT) based effective fragment potential (EFP1) method for the water molecules, and then the final lower energy structures were refined with a more accurate *ab initio* DFT calculation on the complete solute/water cluster. The BH algorithm is applied to the methylammonium ion  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$ , zwitterionic methylcarbamic acid  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_{1-8}$ , and zwitterionic glycine  $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_{7-8}$  clusters.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	i
ABSTRACT.....	ii
LIST OF TABLES .....	v
LIST OF FIGURES .....	vii
LIST OF ACRONYMS AND ABBREVIATIONS .....	ix
1. INTRODUCTION .....	1
1.1 GLOBAL OPTIMIZATION METHODS.....	1
1.2 LOCAL OPTIMIZATION.....	7
1.2.1 SCHRÖDINGER EQUATION .....	8
1.2.2 HARTREE-FOCK THEORY.....	9
1.2.3 DENSITY FUNCTIONAL THEORY .....	12
1.2.4 EFP METHOD .....	15
1.3 PROGRAMMING LANGUAGE .....	18
1.4 OUTLINE OF THESIS.....	19
2. BASIN-HOPPING GLOBAL OPTIMIZATION STRATEGY .....	20
2.1 BASIN-HOPPING .....	20
2.2 UNIFORM RANDOM STEPS .....	23
2.3 GEOMETRICAL EQUIVALENCE TESTING .....	27
2.4 ASSESSING ALGORITHM EFFICIENCY .....	28

2.5 TESTING THE PERFORMANCE OF THE BH ALGORITHM .....	29
3. GLOBAL OPTIMIZATION INVESTIGATION OF SYSTEMS INVOLVED IN THE FORMATION OF GLYCINE .....	31
3.1 INTRODUCTION.....	31
3.2 METHYLAMMONIUM CATION.....	34
3.3 METHYLCARBAMIC ACID .....	41
3.4 GLYCINE .....	60
3.5 CONCLUSION .....	66
APPENDIX.....	68
A. SOURCE CODE FOR THE BH ALGORITHM .....	68
B. ENERGY DATA FOR OPTIMIZED STRUCTURES .....	103
C. COORDINATE DATA FOR OPTIMIZED STRUCTURES .....	106
REFERENCES .....	114

## LIST OF TABLES

<b>Table 1.</b> Comparison of the lowest energies (hartree) obtained from calculations of $\text{H}_2\text{O}$ and $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{0-6}$ with the results of Masamura <sup>a</sup> .	37
<b>Table 2.</b> Energies and enthalpies of solvation obtained from calculations of $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$ .	38
<b>Table 3.</b> Relative energies (kcal/mol) obtained for the 4 lowest energy conformers from calculations for $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{3-6}$ using B3LYP/6-311++G( <i>d,p</i> ) <sup>a</sup>	40
<b>Table 4.</b> Bond lengths for lowest energy values obtained from calculations for $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{0-6}$	40
<b>Table 5.</b> Global minimum geometry data for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})$ structure resulting from the unfragmented B3LYP/6-31+G( <i>d</i> ) level of theory.	44
<b>Table 6.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_2$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.	44
<b>Table 7.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_2$ clusters.	45
<b>Table 8.</b> Relative energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_2$ after a full geometry optimizing using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.	46
<b>Table 9.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_3$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.	48
<b>Table 10.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_3$ clusters.	49
<b>Table 11.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_4$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.	49
<b>Table 12.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_4$ clusters.	50
<b>Table 13.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_5$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.	51
<b>Table 14.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_5$ clusters.	52



<b>Table 15.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_6$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.....	52
<b>Table 16.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_6$ clusters.....	53
<b>Table 17.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.....	54
<b>Table 18.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$ clusters.....	55
<b>Table 19.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$ locally optimized using B3LYP/6-31+G* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G** and MP2/6-311++G** levels of theory.....	56
<b>Table 20.</b> Structural data for the 4 lowest energy $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$ clusters.....	56
<b>Table 21.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$ and $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_7$ using B3LYP/6-311++G**.....	62
<b>Table 22.</b> Structural data for the 4 lowest energy $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_7$ clusters.....	63
<b>Table 23.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$ and $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_8$ using B3LYP/6-311++G**.....	64
<b>Table 24.</b> Structural data for the 4 lowest energy $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_8$ clusters.....	64
<b>Table 25.</b> Energy values obtained after ZPE correction for $\text{CH}_3\text{NH}_3^+$ .....	103
<b>Table 26.</b> Energy values obtained after ZPE correction for methylcarbamic acid.....	104
<b>Table 27.</b> Energy values obtained after ZPE correction for glycine.....	105

## LIST OF FIGURES

<b>Figure 1.</b> 3-Dimensional potential energy surface (PES) based on x and y coordinate displacement.....	2
<b>Figure 2.</b> 2-Dimensional representation of the potential energy surface, where the catchment regions are shown with dashed lines. R represents the configuration of the structure or cluster. ....	6
<b>Figure 3.</b> Flow chart steps following basin-hopping algorithm.....	21
<b>Figure 4.</b> Density within sphere using Marsaglia method (A), generates a mean step of unit length, however not uniformly populated within the sphere with radius 2 units. The density within a sphere using uniform random Cartesian points (B), where a conditional statement is applied for points greater than the radius of 1.31 units to obtain a mean step of unit length. ....	25
<b>Figure 5.</b> Distance populations for Figure 4 using a uniform distance distribution of points (4A) and a uniform distribution of points within a sphere (4B). ....	26
<b>Figure 6.</b> Clusters with 4 waters using only a EFP1 (1), (1) followed by full <i>ab initio</i> unfragmented calculation (2), and full <i>ab initio</i> unfragmented calculation only (3).....	30
<b>Figure 7.</b> Lowest energy structures obtained with $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$ water molecules after ZPE correction. The label xA correspond to the number of water molecules with A representing the GM obtained. ....	36
<b>Figure 8.</b> Lowest energy structures obtained with $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{3-6}$ water molecules after ZPE correction.....	39
<b>Figure 9.</b> Initial zwitterionic methylcarbamic acid structures submitted for global optimization. Atom 1 is N, 2 and 3 are C, and 4 and 5 are O. ....	42
<b>Figure 10.</b> The side and Newman projection of the lowest energy structure for the zwitterionic form of methylcarbamic acid with one water molecule. ....	43
<b>Figure 11.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with two water molecules.....	44
<b>Figure 12.</b> Electrostatic potentials of the two lowest energy structures with two water molecules $\text{CH}_3\text{NH}_2\text{CO}_2(\text{H}_2\text{O})_2$ . (See the electronic copy for the color version.) ....	46
<b>Figure 13.</b> Side and Newman projections of the optimized geometries of the low energy structures zM-2B and zM-2A using B3LYP/6-311++G** and MP2/6-311++G**, respectively. ....	47

<b>Figure 14.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with three water molecules. ....	48
<b>Figure 15.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with four water molecules. ....	50
<b>Figure 16.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with five water molecules. ....	51
<b>Figure 17.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with six water molecules. ....	53
<b>Figure 18.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with seven water molecules. ....	54
<b>Figure 19.</b> The four lowest energy structures for the zwitterionic form of methylcarbamic acid with eight water molecules. ....	56
<b>Figure 20.</b> Distance comparisons for the average values of the 4 lowest energy structures with the error bars representing the standard deviation. ....	57
<b>Figure 21.</b> Angle comparisons for the average values of the 4 lowest energy structures with the error bars representing the standard deviation. ....	58
<b>Figure 22.</b> Direct comparison of N-C(3) distance and OCO angle. ....	59
<b>Figure 23.</b> Initial zwitterionic glycine starting structure submitted for global optimization. Atoms 2 is N, 1 and 3 are C, and 4 and 5 are O. ....	61
<b>Figure 24.</b> The four lowest energy structures for the zwitterionic form of glycine with seven water molecules. ....	63
<b>Figure 25.</b> The four lowest energy structures for the zwitterionic form of glycine with eight water molecules. ....	65

## LIST OF ACRONYMS AND ABBREVIATIONS

**AR** – Active Region

**B3LYP** – Beck 3-term functional; Lee, Yang, Parr exchange

**BH** – Basin Hopping

**BS** – Basis Set

**CPU** – Central Processing Unit

**DMA** – Distributed Multipolar Analysis

**DFT** – Density Functional Theory

**EA** – Evolutionary Algorithm

**EFP** – Effective Fragment Potential

**GA** – Genetic Algorithm

**GAMESS** – General Atomic and Molecular Electronic Structure System

**GGA** – Generalize-Gradient Approximation

**GO** – Global Optimization

**HF** – Hartree-Fock

**HK** – Hohenberg and Kohn

**ISM** – Interstellar Medium

**KS** – Kohn and Sham

**LMOs** – Localized Molecular Orbitals

**LDA** – Local Density Approximation

**LSDA** – Local-Spin-Density Approximation

**MC** – Monte Carlo

**MP2** – Moller-Plesset 2<sup>nd</sup> order perturbation theory

**MT19937** – Mersenne Twister

**NP** – Nondeterministic Polynomial-Time

**OOP** – Object-Oriented Programming

**PES** – Potential Energy Surface

**QM** – Quantum Mechanics

**RHF** – Restricted Hartree-Fock

**SA** – Simulated Annealing

**SR** – Spectator Region

**ZPE** – Zero Point Energy

# 1. INTRODUCTION

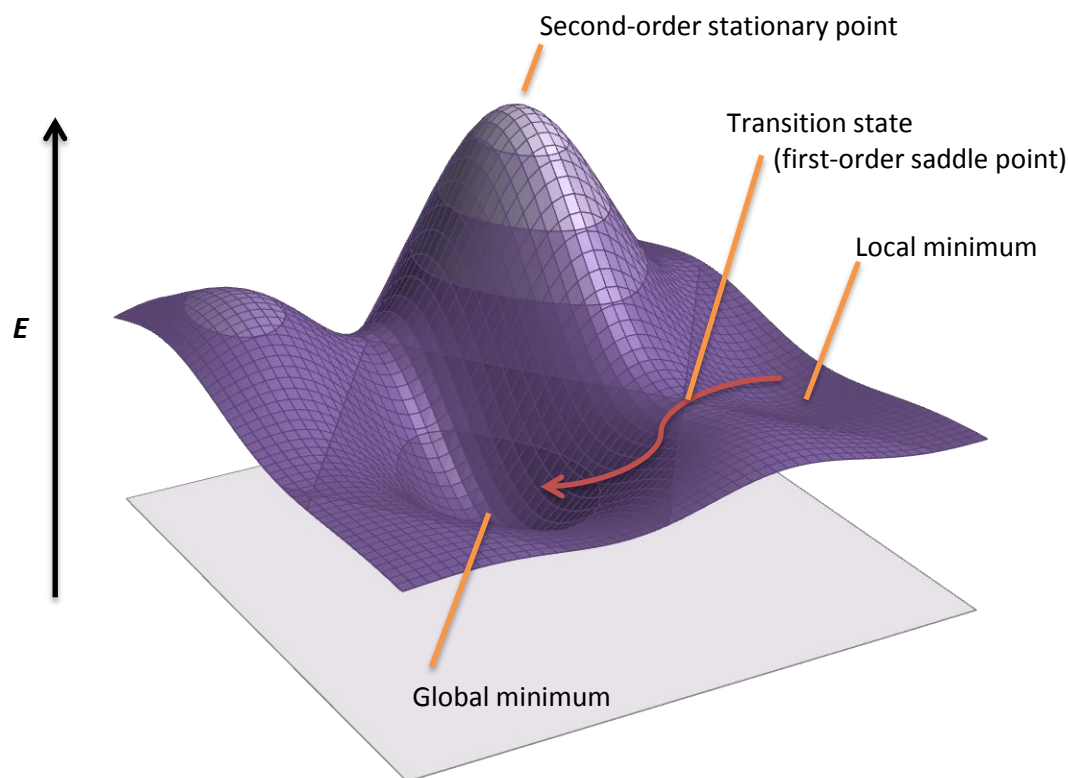
## 1.1 GLOBAL OPTIMIZATION METHODS

Global optimization is a valuable tool with many applications in chemistry. For instance one would often like to be able to computationally predict the possible stable structures for a given molecule or chemical system of interest. In this thesis, the utility of using a global optimization strategy for identifying the lowest energy conformer for a specific molecular structural type is described. In addition to obtaining the lowest energy conformer, the global optimization method should also provide structural information on the closest related low energy conformers which also play a role in the chemistry of the system.

The structural geometry of a nonlinear molecule containing  $N$  nuclei can be represented by  $3N-6$  independent nuclear coordinates ( $q_1, q_2, \dots q_{3N-6}$ ) or  $3N$  Cartesian coordinates, where the number 6 accounts for the translational and rotational motion of the molecule [1]. The energy of the molecule  $E$  is a function of these coordinates, where the function generates a potential energy surface (PES) or hypersurface for the molecule.

$$E_{PES} = E(q_1, q_2, \dots q_{3N-6}) \quad (1-1)$$

It is impossible to visualize the entire PES, except for simple cases, based on one-coordinate or two-coordinate system variations. Figure 1 illustrates an example of a PES where only the x and y Cartesian coordinates in a molecule are altered. The various stationary points on the PES are of interest in chemistry as they represent the stable structures, reactants, products, and transition states. The local minima on the energy surface are of interest in this thesis, where they correspond to the stable or equilibrium structures for a molecule. Using global optimization techniques one can locate different local minima, or unique stable molecular conformations, on the PES, with the lowest energy minimum or extremum being the global minimum (GM).



**Figure 1.** 3-Dimensional potential energy surface (PES) based on x and y coordinate displacement.

The saddle point on the energy barrier separating two minima is also of interest because it represents the transition state structure and is known as the first-order saddle point or transition state. The minima and saddle points are both stationary points on the potential energy surface with the first derivative of the energy, with respect to all of the nuclear coordinates, equal to zero. One can distinguish the different types of stationary points by running frequency calculations on the locally optimized geometries. If  $3N-6$  real frequencies and zero imaginary frequencies are calculated for a non-linear molecule, the geometry corresponds to a local minimum. If one imaginary frequency is calculated, the geometry would represent a transition state between two local minimum. The generation of two imaginary frequencies would then correspond to a second order stationary point.

Using a global optimization strategy does not guarantee that one has found a global minimum, where the lowest energy structure is generally taken as the global minimum until a

lower energy structure is found. Performing several trial searches with different starting geometries that maintain the same lowest energy structure can increase confidence that one has found the global minimum. However, there is no simple test that can classify a structure as the global minimum. There are several parameters which can be varied when performing a global optimization, such as the number of iterations, temperature, coordinate displacement, and boundary conditions. These parameters are optimized during trial runs to improve finding the global minimum with minimal computational resource expenditure.

The stable structures generated from global optimization techniques aid in the prediction and interpretation of structural features found in crystals, nanosystems, or biomolecules and is a subject of combinatorial optimization. The major difficulty with this strategy is the exponential growth of the number of stationary points on the PES as the size of the chemical system increases, known as a nondeterministic polynomial-time (NP)-hard problem [2]. A classic example of the NP-hard problem, sometimes referred to as NP-complete, is the traveling salesman problem. The traveling salesman must travel to  $N$  cities and can take several different routes to visit every location. The objective is to travel to every city once and then return to the starting point while minimizing the cost of traveling to all the locations. With an ever increasing list of cities, an exhaustive or systematic search is no longer feasible and various optimization strategies have to be tried as no method for an exact solution has been found.

In 1953, Metropolis *et al.* [3] introduced a modified Monte Carlo method for investigating the properties of any substance. The Monte Carlo method consists of integration over a random sampling of points instead of some regular array of points. The configurations were chosen with an acceptance probability based on a Boltzmann distribution (Equation 1-2),

$$p = e^{-(\Delta E)/k_B T} \quad (1-2)$$

where  $\Delta E$  is the change in energy,  $k_B$  is the Boltzmann constant, and  $T$  is the temperature. In a  $3N$ -dimensional space, one would place the  $N$  particles in any configuration and then move each particle in the  $k$ th step according to the following formulas for the Cartesian coordinates:



$$\begin{aligned}
 x_{k+1} &= x_k + \alpha \delta_1 \\
 y_{k+1} &= y_k + \alpha \delta_2 \\
 z_{k+1} &= z_k + \alpha \delta_3
 \end{aligned}
 \tag{1-3}$$

where one can change all or some of the Cartesian coordinates. The maximum allowable displacement is  $\alpha$ , and  $\delta_j$ , for  $j = 1, 2$ , or  $3$ , is a uniform random number between  $-1$  and  $1$ . The energy corresponding to the particles at this new position would then be calculated and automatically accepted if  $\Delta E < 0$ . If  $\Delta E > 0$ , then the new geometry is accepted or rejected based on Equation 1-2. When using the Metropolis method in Equation 1-2, a uniform random number between  $0$  and  $1$  is chosen by the computer and if this number is less than the acceptance probability  $p$ , then the new position is accepted. If the chosen number is higher than the acceptance probability, the new position is rejected and the particle is returned to the old position and another permutation is tested. This process is repeated until the desired number of steps has been reached, or some convergence condition has been met.

Global optimization strategies are grouped into two areas [4]: systematic and stochastic, where the stochastic methods are further divided into non-genetic and genetic algorithms.

In the systematic method, one makes a series of specified changes to the atomic structure of the chemical system. This method is commonly used to find the lowest energy for long chain organic molecules [5] or energy minimization of the side chains in proteins. All the rotatable bonds, or dihedral angles, in the molecule are first identified. Then a series of energy calculations are performed where each bond is systematically rotated through  $360^\circ$  using fixed increments, while the remaining bond lengths and bond angles remain fixed during the energy calculations. Alternatively, the different structures may be subjected to local optimization with selected dihedral angles held fixed. While this method enables a systematic analysis of the molecule, the number of geometries that are generated leads to a combinatorial explosion [1] as shown in equation 1-4:

$$\text{number of conformers} = \prod_{i=1}^N \frac{360}{\theta_i}
 \tag{1-4}$$

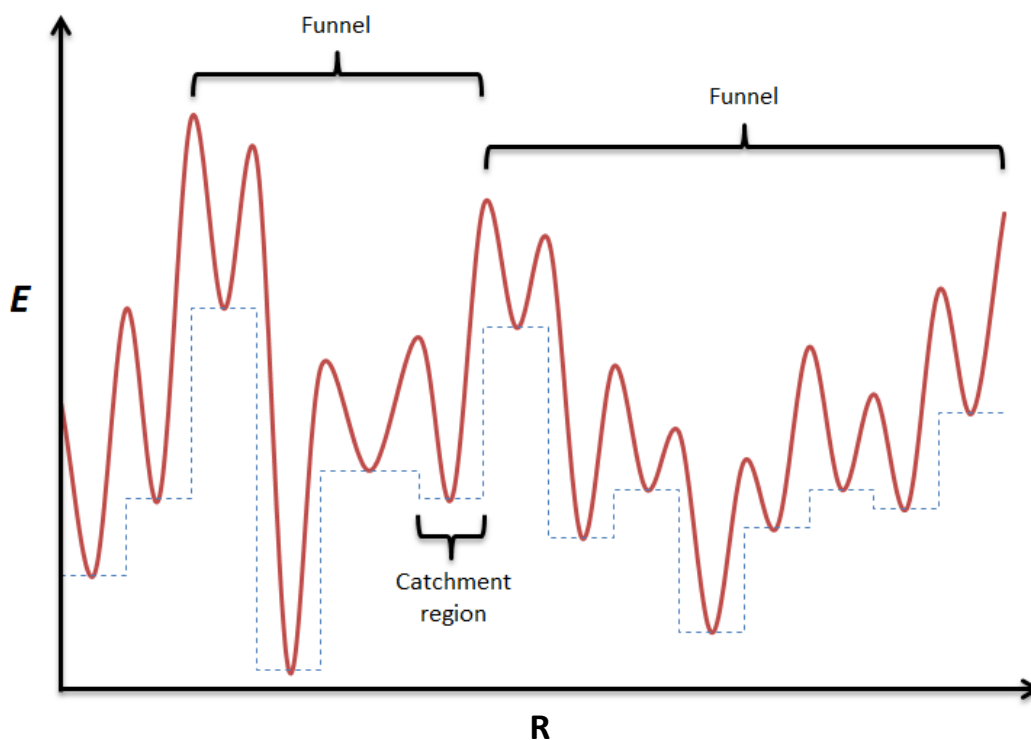
where  $\theta_i$  is the dihedral angle increment chosen for bond  $i$  and  $N$  is the total number of bonds. An example would be the use of  $4$  bonds with a dihedral increment of  $15^\circ$  or  $30^\circ$  leading to

the generation of 331,776 or 20,736 structures, respectively. If you increased the number of bonds to 10 with 15° increments, you would have  $6 \times 10^{12}$  structures.

Two main areas in stochastic non-genetic algorithms are basin hopping and simulated annealing. Simulated annealing (SA) [6] is analogous to the annealing process used in laboratory experiments, where the substance is heated above its critical temperature and subjected to a slow lowering of the temperature to achieve the thermodynamically stable structure. In experiments, the procedure used to decrease the temperature depends on the properties of the specific substance. If this process is not performed properly during experimentation, the substance may not be at thermodynamic equilibrium and result in a substance with many defects and only metastable structures. This method embraces the free energy of the system at high temperatures and tries to trace this free energy as the temperature is lowered, where the free energy global minimum and the global minimum of the system must match at 0 K.

Utilizing the SA algorithm, one starts at a specified high temperature  $T$  and examines the energy of different structures of the system which evolve using Equation 1-3 over a certain number of steps at each sufficiently slow logarithmic decrease of  $T$  until the minimum  $T$  desired is reached. Structures are accepted or rejected based on Equation 1-2 at the corresponding  $T$ . In practice, a fast or non-logarithmic variation in  $T$  is performed [7], which can cause the optimization to get stuck in one catchment region or funnel (Figure 2) and results in a metastable local minimum structure being found.

A catchment region is the area surrounding each local minimum where a local geometry optimization of a molecule will converge to this local minimum as can be shown in Figure 2. The dashed lines represent the transformed energy that is obtained by performing a local optimization. A PES may consist of several funnels, where each funnel represents a specific type of cluster property and contains a number of catchments regions with the same property [8].



**Figure 2.** 2-Dimensional representation of the potential energy surface, where the catchment regions are shown with dashed lines.  $R$  represents the configuration of the structure or cluster.

Basin hopping (BH) [9] also known as Monte-Carlo minimization [10], which is the focus of this thesis, performs a Monte Carlo method to scan the potential energy surface at fixed temperatures and perform a local optimization with each new coordinate displacement given by Equation 1-3. Equation 1-2 is used to decide if the higher energy of the new local minimum is acceptable. The step size and temperature are adjusted to obtain a probability of approximately 50% [9] that a move is accepted. BH is similar to simulated annealing except that it removes the energy barriers separating the local minima [4]. The energy barriers are removed by performing a local geometry optimization (LO) at each step given by equation 1-3 [11], where one or more of the atoms are displaced depending on the algorithm. Figure 2 demonstrates the transformed PES with dashed lines. Using a temperature that is too low would make the system unable to overcome the energy barrier and remain trapped in the local minimum or a funnel, therefore careful consideration should be taken when selecting a temperature to enable hopping between different funnels and enable scanning of the full PES.

BH typically uses larger step sizes than SA to ensure that a “hop” between different local minima is obtained.

The genetic algorithm (GA) is a global optimization strategy inspired by the Darwinian evolution process, where phenotype genetic operations are performed directly on the atoms or particles in the system of interest instead of internal coordinate displacements followed by a local optimization. These operations are comprised of particle permutations, particle displacement, piece rotation, piece reflection, shrinkage, and cut and “mating”, where a plane is used to cut two parent structures and merge a fragment from each parent together [12]. If the number of atoms from the two pieces does not equal the number of atoms from each parent, the child structure is discarded and another cut is performed. The mutation and crossover operations generate diversity to the structures, thereby allowing the structure to avoid being trapped in a local minimum.

Most global optimization (GO) methods involve two main phases: local optimization (LO) and global optimization that finds the extremum of the locally optimized structures. The most commonly used GO methods are GAs and BH. Many variations of these methods have been tested [8]. One problem with optimization is that you may wind up in a catchment region or funnel that does not contain the global minimum and you are not be able to “hop” to another catchment region or funnel that contains the global minimum. Another would be that only part of the PES has been searched. As stated previously, there is no mathematical algorithm to test for these cases. Performing multiple trial searches that generate the same global minimum is the a good to test that the GO has functioned properly.

## 1.2 LOCAL OPTIMIZATION

In order to use a global optimization method to find the most stable structure for a molecule, one needs to be able to calculate the PES  $E(\mathbf{R})$  for the molecule, where  $E(\mathbf{R})$  represents the energy at a specific molecular configuration  $\mathbf{R}$ . A polyatomic chemical system can be described by a wave function that depends on the geometrical parameters of that system and the resulting energy of the system can be obtained. There are four main theoretical approaches to calculating the PES: *ab initio*, semi-empirical, density-functional theory, and

molecular-mechanics methods. First principle, or *ab initio*, methods attempt to solve the Schrödinger equation exactly using the true Hamiltonian to obtain the molecular energy and is computationally intensive. Semi-empirical methods use a simpler Hamiltonian which reduces computational cost and use empirical parameters that are adjusted based on experimental data. Density-functional theory (DFT) has intermediate computational expense and relies on the fact that the molecular energy is uniquely dependent on the electron probability density  $\rho$  and consequently does not calculate the wave function directly. Molecular-mechanics (MM) calculations are much faster, however it is not a quantum-mechanical method as it does not use a Hamiltonian operator or wave function. The energy is expressed as terms of force constants for bond bending and stretching as well as other parameters, which are obtained from experimental and theoretical data, and is used for large systems.

The main focus of this thesis is the development of a BH algorithm that scans the PES using DFT for local optimization. The main chemical system of interest is composed of methylcarbamic acid zwitterion surrounded by water molecules. Due to the hydrogen bonding interaction between the solute and solvent, discrete treatment of the water molecules are applied using the previously developed effective fragment potential (EFP1) [13] method in the GAMESS program [14].

### 1.2.1 SCHRÖDINGER EQUATION

The energies of an atom or a molecule are theoretically found using the time-independent Schrödinger equation:

$$\hat{H}\psi(q_i, q_\alpha) = E\psi(q_i, q_\alpha) \quad (1-5)$$

with  $q_i$  and  $q_\alpha$  representing the electronic and nuclear coordinates, respectively. The nuclei and electrons are taken as point masses, while neglecting spin-orbit and other relativistic interactions. This generates the molecular Hamiltonian operator:

$$\hat{H} = -\frac{\hbar^2}{2} \sum_{\alpha} \frac{1}{m_{\alpha}} \nabla_{\alpha}^2 - \frac{\hbar^2}{2m_e} \sum_i \nabla_i^2 + \sum_{\alpha} \sum_{\beta > \alpha} \frac{Z_{\alpha} Z_{\beta} e'^2}{r_{\alpha\beta}} - \sum_{\alpha} \sum_i \frac{Z_{\alpha} e'^2}{r_{i\alpha}} + \sum_i \sum_{j > i} \frac{e'^2}{r_{ij}} \quad (1-6)$$

where  $\alpha$  and  $\beta$  label the nuclei and  $i$  and  $j$  label the electrons. Nuclei are much heavier than electrons,  $m_\alpha \gg m_e$ , and therefore the electrons move faster, enabling separation of electronic and nuclear motions with the Born-Oppenheimer approximation. The heavier nuclei mean that the electrons will adjust almost instantaneously as the positions of the nuclei change and the Schrödinger equation for electronic motion becomes:

$$(\hat{H}_{el} + V_{NN})\psi_{el} = U\psi_{el} \quad (1-7)$$

where  $U$  is the energy,  $\hat{H}_{el}$  is the electronic Hamiltonian, and  $V_{NN}$  is the nuclear-repulsion term. The electronic nonrelativistic Hamiltonian, in atomic units, then becomes the following.

$$\hat{H}_{el} = -\frac{1}{2} \sum_i \nabla_i^2 - \sum_\alpha \sum_i \frac{Z_\alpha}{r_{i\alpha}} + \sum_i \sum_{j>i} \frac{1}{r_{ij}} \quad (1-8)$$

On the RHS the first, second, and third terms in equation 1-8 corresponds to the electron kinetic energy, nuclear-electron attraction, and electron-electron repulsion, respectively. The nuclear-repulsive term is:

$$V_{NN} = \sum_\alpha \sum_{\beta>\alpha} \frac{Z_\alpha Z_\beta e'^2}{r_{\alpha\beta}} \quad (1-9)$$

The nuclear repulsive term is independent of the electronic coordinates and is a constant that does not affect the wave functions, and is therefore neglected when calculating the electronic energy.

$$\hat{H}_{el}\psi = E_{el}\psi \quad (1-10)$$

The  $U$  then equals the sum of the electrostatic repulsion between the nuclei and the electronic energy.  $U$  is a function of the nuclear coordinates and defines the PES.

$$E_{PES} = U = E_{el} + V_{NN} \quad (1-11)$$

### 1.2.2 HARTREE-FOCK THEORY

In the molecular Hartree-Fock theory the  $N$ -electron wave function is approximated by a single Slater determinant. The normalized Slater determinant is an antisymmetrized product containing  $N$  electrons in spin orbitals  $\chi_1, \chi_2, \dots, \chi_N$ ,

$$\Psi = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(1) & \cdots & \chi_N(1) \\ \vdots & \ddots & \vdots \\ \chi_1(N) & \cdots & \chi_N(N) \end{vmatrix} \quad (1-12)$$

where each spin orbital is a product of a spatial and spin function as shown below.

$$\chi_{2i-1} = \psi_i(\mathbf{r}_1)\alpha(1) = 1s \quad \text{and} \quad \chi_{2i} = \psi_i(\mathbf{r}_1)\beta(1) = \overline{1s} \quad (1-13)$$

The spin functions  $\alpha$  and  $\beta$  correspond to spin up and spin down, respectively, where the bar over the spatial part signifies a spin down arrangement.

The spatial molecular spin orbital is written as a linear combination of atomic orbitals (LCAO).

$$\psi_i = \sum_{\mu=1}^K c_{\mu i} \phi_{\mu} \quad (1-14)$$

Where  $\phi_{\mu}$  corresponds to either an atomic orbital and/or a more general basis function, and  $c_{\mu i}$  is an expansion coefficient. Using the variational method, one can solve for the expected energy using a time-independent Hamiltonian operator. The expected energy of the system is obtained by evaluating the following:

$$\frac{\int \Psi^* \hat{H} \Psi d\tau}{\int \Psi^* \Psi d\tau} \geq E_0 \quad (1-15)$$

The optimum expansion coefficient  $c_{\mu i}$  minimizes the variational energy  $E_0$ , where  $E_0$  is the lowest-energy eigenvalue for the corresponding Hamiltonian. The coefficients are determined such that the lowest energy wavefunction is obtained, where the best set is found when the energy is at a minimum.

$$\frac{\partial E}{\partial c_{vi}} = 0 \quad \begin{matrix} v = 1, \dots K \\ i = 1, \dots N/2 \end{matrix} \quad (1-16)$$

The ground state for the majority of molecules has a closed-shell configuration with an even number of  $N$  electrons and  $N/2$  occupied spatial orbitals. The kinetic and potential energy of each electron-nuclear interaction in a molecular orbital contributes an energy of  $H_{ii}^{core}$ , and two electrons in the orbital contribute an energy of  $2H_{ii}^{core}$ . Using closed-shell orbitals of  $N/2$ , the total energy is a summation over each molecular orbital.

$$\sum_{i=1}^{N/2} 2H_{ii}^{core} = 2 \sum_{i=1}^{N/2} \int \chi_i(1) \left( -\frac{1}{2} \nabla_i^2 - \sum_{A=1}^M \frac{Z_A}{r_{iA}} \right) \chi_i(1) d\tau_1 \quad (1-17)$$

The electron-electron electrostatic repulsion of one orbital interacts four ways with the electrons of a second orbital, thus yielding the Coulomb interaction of  $4J_{ij}$ . The exchange interaction, which is an expression of the Pauli principle has only 2 variations,  $2K_{ij}$ . The Coulomb and exchange equations are shown below.

$$J_{ij} = \int |\chi_i(1)|^2 \left( \frac{1}{r_{12}} \right) |\chi_j(2)|^2 d\tau_1 d\tau_2 \quad (1-18)$$

$$K_{ij} = \int \chi_i(1)\chi_j(2) \left( \frac{1}{r_{12}} \right) \chi_i(2)\chi_j(1) d\tau_1 d\tau_2 \quad (1-19)$$

Inclusion of the Coulomb interaction between each pair of electrons in the same orbital is included, where observation of  $J_{ii} = K_{ii}$  reduces the total Hartree-Fock energy of a closed-shell system as follows:

$$E_{HF} = 2 \sum_{i=1}^{N/2} H_{ii}^{core} + \sum_{i=1}^{N/2} \sum_{j=1}^{N/2} (2J_{ij} - K_{ij}) \quad (1-20)$$

The Fock operator  $\hat{f}_i(1)$  is a one-electron Hamiltonian for that electron in a multiple electron system.

$$\hat{f}_i(1) = \hat{H}^{core}(1) + \sum_{j=1}^{N/2} \{2\hat{J}_j(1) - \hat{K}_j(1)\} \quad (1-21)$$

The Hartree-Fock equation in standard eigenvalue form becomes:

$$\hat{f}_i\chi_i = \varepsilon_i\chi_i \quad (1-22)$$

Self-consistent field (SCF) is used to solve this equation, where a set of trial solutions  $\chi_i$  are used to calculate the Coulomb and exchange operators. The Hartree-Fock equations are solved and generate a new set of solutions  $\chi_i$  that will be used for the next iteration. In this process, one electron will affect the solutions for the other electrons in the system, and the SCF method will continue with lower energies at each step until no variation is observed, leading to solutions that are self-consistent.

In 1951, Roothaan [15] and Hall [16] proposed the derivation of the Hartree-Fock energy using matrix form enabling a linear algebra solution, thus having application to any



geometrical systems. Substitution with orbital expansion converts the Hartree-Fock equation into:

$$\hat{f}_i(1) \sum_{\mu=1}^K c_{\mu i} \phi_{\mu}(1) = \varepsilon_i \sum_{\mu=1}^K c_{\mu i} \phi_{\mu}(1) \quad (1-23)$$

where  $\varepsilon_i$  corresponds to the orbitals energies. The Fock matrix is a  $K \times K$  matrix that can be derived as the following,

$$F_{\mu\nu} = \int \phi_{\mu}(1) \hat{f}_i(1) \phi_{\nu}(1) dv_1 \quad (1-24)$$

where the Roothan-Hall equations are then written in matrix form:

$$\mathbf{FC} = \mathbf{SCE} \quad (1-25)$$

$\mathbf{C}$  is a  $K \times K$  matrix containing the coefficients,  $\mathbf{S}$  is the overlap matrix,  $\mathbf{H}$  is the one-electron Hartree matrix,  $\mathbf{F}$  is the Fock matrix, and  $\mathbf{E}$  is a diagonal  $K \times K$  matrix where the values correspond to orbital energies.

The procedure for performing the SCF calculation is as follows:

1. Construct  $\mathbf{S}$  and  $\mathbf{H}$ , where you assume  $\mathbf{F} = \mathbf{H}$  originally. Two-electron integrals are calculated for the non-direct HF methods.
2. Solve the Fock matrix to obtain the molecular orbital coefficients.
3. Construct the charge-density bond-order matrix.
4. Construct the new  $\mathbf{F}$  matrix with step 3 using Equation 1-24. Direct HF methods recalculate the two-electron integrals with every  $\mathbf{F}$  construction.
5. Continue with step 2 until the molecular coefficients converge.

The local minimum of the  $N$ -particle system has the first derivatives of the total energy  $E$  over the  $3N$  internal coordinates equal to zero using Equation 1-16.

### 1.2.3 DENSITY FUNCTIONAL THEORY

Density functional theory (DFT) is an alternative approach to working with the wavefunction in the HF equation, which calculates the molecular electronic energy from the electron density. In 1964, Hohenberg and Kohn [17] proved that a universal functional of the density,

$F[\rho(\mathbf{r})]$ , independent of the potential  $v(\mathbf{r})$ , exists such that the energy at the minimum value corresponds to the correct ground-state energy.

$$E(\rho) = \int v(\mathbf{r})\rho(\mathbf{r})d^3\mathbf{r} + F[\rho(\mathbf{r})] \quad (1-26)$$

Kohn and Sham [18] (KS) suggested that  $F[\rho]$  should be approximated by the kinetic energy, electron-electron Coulombic energy, and the contributions from exchange and correlation. The universal functional  $F[\rho]$  is valid for any number of particles and any external potential, where one can separate the classical Coulomb energy to obtain:

$$F[\rho] = \frac{1}{2} \int \int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}} d^3\mathbf{r}_1 d^3\mathbf{r}_2 + G[\rho] \quad (1-27)$$

The ground-state energy can then be written as follows.

$$E(\rho) = \int v(\mathbf{r})\rho(\mathbf{r})d^3\mathbf{r} + \frac{1}{2} \int \int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{r_{12}} d^3\mathbf{r}_1 d^3\mathbf{r}_2 + G[\rho] \quad (1-28)$$

The universal functional  $G[\rho]$  can be written as

$$G[\rho] = T_s[\rho] + E_{xc}[\rho] \quad (1-29)$$

where  $T_s[\rho]$  is the kinetic energy of a system of non-interacting electrons and  $E_{xc}[\rho]$  is the exchange-correlation energy of an interacting system.

The Kohn-Sham density of the system is written as the sum of the square of a set of one-electron orthonormal orbitals:

$$\rho(\mathbf{r}) = \sum_{i=1}^N |\psi_i(\mathbf{r})|^2 \quad (1-30)$$

Using this expression for density, the one-electron Kohn-Sham equations for  $M$  nuclei become:

$$\left\{ -\frac{\nabla_1^2}{2} - \left( \sum_{A=1}^M \frac{Z_A}{r_{1A}} \right) + \int \frac{\rho(\mathbf{r}_2)}{r_{12}} d\mathbf{r}_2 + V_{xc}[\mathbf{r}_1] \right\} \psi_i(\mathbf{r}_1) = \varepsilon_i \psi_i(\mathbf{r}_1) \quad (1-31)$$

where  $\varepsilon_i$  are the orbital energies and  $V_{xc}$  is the exchange-correlation functional and is related to the exchange-correlation energy as follows.

$$V_{xc}[\mathbf{r}] = \frac{\delta E_{xc}[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})} \quad (1-32)$$

The total electronic energy can then be calculated using Equation 1-28, where the exchange-correlation functional is crucial to improving the results.

Kohn and Sham [18] (KS) used a procedure known as the local density approximation (LDA) to show that, based on sufficiently slow varying density, the exchange correlation can be:

$$E_{xc}[\rho] \approx E_{xc}^{LDA}[\rho] = \int \rho \varepsilon_{xc}(\rho) d^3\mathbf{r} \quad (1-33)$$

$\varepsilon_{xc}(\rho)$  is the exchange and correlation energy per electron of a uniform electron gas of density  $\rho$ . Results are improved over LDA using local-spin-density approximation (LSDA) for open-shell molecules and molecules with geometries close to dissociation, where LDA is analogous to the RHF method. The LSDA allows electrons with opposite spins to have different spatial KS orbitals.

Various methods have been developed to correct the LSDA for the variation of electron density with position known as the generalized-gradient approximation (GGA). The GGA exchange-correlation functional depends on local spin densities and their gradients. Becke's [19] 1988 exchange functional (B88) is one of the commonly used GGA exchange functionals:

$$E_x^{B88} = E_x^{LSDA} - b \sum_{\sigma=\alpha,\beta} \int \frac{(\rho^\sigma)^{4/3} \chi_\sigma^2}{1 + 6b \chi_\sigma \ln[\chi_\sigma + (\chi_\sigma^2 + 1)]} d^3\mathbf{r} \quad (1-34)$$

where  $b$  is an empirical parameter of 0.0042 atomic units and  $\chi_\sigma \equiv |\nabla \rho^\sigma|/(\rho^\sigma)^{4/3}$ .

Hybrid functionals have been developed to correct for deficiency in GGA functionals. The Becke-Lee-Yang-Parr (B3LYP) is a hybrid GGA three-parameter functional of the Becke [19-23] exchange functional and the Lee-Yang-Parr [24] correlation functional:

$$E_{xc}^{B3LYP} = (1 - a_0)E_x^{LSDA} + a_0E_x^{HF} + a_x\Delta E_x^{B88} + a_cE_c^{LYP} + (1 - a_c)E_c^{VWN} \quad (1-35)$$

where  $a_0$ ,  $a_x$ , and  $a_c$  are semi-empirical coefficients obtained by fitting with values of 0.2, 0.72, and 0.81, respectively. The exchange energy determined by the HF method is  $E_x^{HF}$ .  $E_c^{LYP}$  is the correlation energy functional from Lee, Yang, and Parr [19].  $E_c^{VWN}$  is the standard local correlation functional from Vosko, Wilk, and Nusair (VWN) [25].

To solve the above Kohn-Sham equations a self-consistent approach, similar to that shown previously is used. An initial guess of the density is inputted into equation 1-28 and a set of orbitals are derived. These orbitals lead to an improved value for the density and is then used in the second iteration and so forth until the desired convergence is found.

#### 1.2.4 EFP METHOD

Many QM calculations assume the gas phase has similar properties regardless of solvent and ignore the solvent effects. However, the majority of chemical processes occur in a solvent and modeling this interaction is important to get an accurate representation of the system. The interaction of different solvents can change the charge distribution from that as the gas phase, and solvent effects should not always be neglected. When choosing a method to model the solvent behavior, one must consider the types of interactions in the solute-solvent system. The solvent molecules may be an essential component interacting directly with the solute, the solvent may not directly interact with the solute but still affect the behavior of the system, or by acting as a bulk medium [1]. Modeling the effects of explicit solvent molecules in *ab initio* calculations is computationally intensive and may not be especially important if the solvent does not directly interact with the solute, and thus various methods have been developed to reduce computational cost to model solvent systems. A non-explicit approach to modeling the solvent system is the purpose of a continuum solvent model [26] [27]. However, this thesis is focused on explicit interactions between the solute and solvent.

One discrete method used to model the bulk of the solvent system is known as an effective fragment potential (EFP) method, where the potentials are used to replace solvent molecules in an *ab initio* calculation. The original motivation behind using the EFP method was to develop a computationally efficient model of solvent effects on chemical reactions using a non-empirical alternative to force-field based QM/MM [13],[28-32]. This fragment substitution allows one to model larger systems reducing computational time during BH simulations. One may also combine continuum methods with the EFP method.

The EFP method was first implemented to model the cluster properties of water molecules [13], where the original method is known as EFP1. The EFP uses a rigid-body

approximation, where the internal coordinates of the fragments are fixed while the positions are fully optimized with the *active region (AR)* [33] or QM region. EFPs are used to replace the *spectator region (SR)*, or water molecules, in *ab initio* electronic structure calculations in order to investigate properties of the *AR*, where the Hamiltonian of the total system is shown below and referred to as a fragmented calculation.

$$H_{TOT} = H_{AR} + H_{SR} \quad (1-36)$$

The *AR* is treated with a full *ab initio* Hamiltonian and contains the solute and any solvent molecule(s) that directly participate in bond-making or bond-breaking processes and is treated with the desired *ab initio* wave function. The *spectator* solvent molecules or fragments are treated via three one-electron terms in the full Hamiltonian corresponding to a component in the total interaction: electrostatic (*elec*), polarization (*pol*), and exchange repulsion/charge transfer (*rep*) [33].

$$H_{SR} = \sum_k^K V_k^{elec} + \sum_l^L V_l^{pol} + \sum_m^M V_m^{rep} \quad (1-37)$$

For each of the following terms (Equations 1-38 to 1-41) in the effective fragment interaction, *s* represents a coordinate in the QM part in Equation 1-36. The  $\mu$ th solvent molecule is expanded over the *K*, *L*, and *M* expansion points of 5, 5, and 2, respectively, for the water EFP1.

A distributed multipolar analysis (DMA) [34] of the molecular density is used with *K*=5 points for the water molecule, where the expansion is performed through octopole moments,

$$\begin{aligned} V_k^{elec}(\mu, s) = & \frac{q_k(\mu)q_s}{r_{sk}} - \sum_{\alpha}^{x,y,z} \mu_{\alpha}^k(\mu)F_{\alpha}(\mathbf{r}_{sk}) - \frac{1}{3} \sum_{\alpha,\beta}^{x,y,z} \Theta_{\alpha\beta}^k(\mu)F_{\alpha\beta}(\mathbf{r}_{sk}) \\ & - \frac{1}{15} \sum_{\alpha,\beta,\gamma}^{x,y,z} \Omega_{\alpha\beta\gamma}^k(\mu)F_{\alpha\beta\gamma}(\mathbf{r}_{sk}) \end{aligned} \quad (1-38)$$

Where *q*,  $\mu$ ,  $\Theta$ , and  $\Omega$  are the charge, dipole, quadrupole, and octopole, respectively.  $F_{\alpha}$ ,  $F_{\alpha\beta}$ , and  $F_{\alpha\beta\gamma}$  are the AR electric field, field gradient, and field Hessian, respectively.

The DMA does not account for overlapping charge densities between two molecules because it is a point charge model. Therefore, modification needs to be performed at short distances to account for overlapping electron densities of the molecules. Equation 1-38 can be multiplied by a damping function or distance-dependent cutoff function:

$$V_k^{el}(\mu, s) \rightarrow \left[1 - c_k(\mu)e^{-\alpha_k(\mu)r_{sk}^2}\right] V_k^{el}(\mu, s) \quad (1-39)$$

The polarization interaction is treated with a self-consistent perturbation model utilizing localized molecular orbitals (LMOs), where water has five LMOs. These LMOs consist of the oxygen inner shell, two oxygen lone pairs, and two oxygen-hydrogen bonds. The localized orbital dipole polarizabilities,  $\alpha_{\alpha\beta}^l$ , are taken from finite-field perturbed HF calculations.

$$V_l^{pol}(\mu, s) = - \sum_{\alpha}^{x,y,z} F_{\alpha}(\mathbf{r}_l) \alpha_{\alpha\beta}^l(\mu) \langle F_{\beta}(\mathbf{r}_l) \rangle \quad (1-40)$$

The third term in Equation 1-37 is the exchange repulsion/charge transfer interaction. This interaction is modeled by one-electron Gaussian functions at the fragment atom centers and the center of mass.

$$V_m^{rep}(\mu, s) = - \sum_j^J \beta_{m,j}(\mu) e^{-\alpha_{m,j}(\mu)r_{m,s}^2} \quad (1-41)$$

There are two models, EFP1 and EFP2, in the General Atomic and Molecular Electronic Structure System (GAMESS) program [14]. The EFP1 model is generally limited to systems involving water molecules with RHF-based or DFT-based potentials that are built into the GAMESS program. The EFP2 potentials are constructed using the GAMESS program and applicable to any species. Both models have frozen internal geometries that are allowed to translate and rotate. The EFP1 Gaussian functions are determined by optimizing coefficients and exponents using 192 points on the water dimer PES. The EFP1 water molecule has a frozen internal geometry of  $\angle\text{HOH}=106.7^\circ$  and an O-H bond length of 0.9438 Å.

The equations above have been coded for the analytic gradients for the entire system (active + spectator/fragment), which allows one to perform vibrational analyses using finite

differences of the gradients. The solute is treated with the desired “wave” function as shown previously, and the remaining fragments are represented by the EFP method. There are no exchange repulsion/charge transfer terms in the nuclear fragment interaction, and so results may vary when performing a full *ab initio* calculation as shown previously.

The EFP has been found to underestimate the interaction of the smaller water clusters ( $n \leq 10$ ) and overestimate the interaction for larger clusters ( $n \geq 14$ ) [35]. This method has been applied to water clusters [33], [35], a glycine-water, and formaldehyde-water solute-solvent system with similar results to full *ab initio* calculations [13],[36-37].

A combined three-layer solvation model was developed, where the first layer was optimized glycine in the neutral(N) and zwitterion(Z) form, the second layer consisted of optimized  $Z(H_2O)_8$  and  $N(H_2O)_8$ , the third layer embedded the clusters in a continuum as EFP waters referred as EFP+Onsager model [38]. This method was utilized to achieve a proper bulk simulation.

The main focus of this research is to model the environment effects of a chemical system, where the chemical system is composed of a selected solvent in a water solute. The modeling of a system such as this is done with a discrete method that is combined with a Basin-Hopping Monte Carlo simulation [1].

### 1.3 PROGRAMMING LANGUAGE

When performing high performance numerical computing involving floating point operations, FORTRAN is the most efficient procedural programming language. However, the syntax allows for various unsafe constructs that one can find themselves with subtle errors in the code [39]. For high-level testing, it is often used as a benchmark for comparison with other codes. The object-oriented feature of C++ has allowed it to become the desired programming language to use for science and engineering applications. Object-oriented programming (OOP) languages can take more time to learn than a procedural language, however OOP languages structures the program into logical units and simplifies many tasks in the code. In some cases, unexpected dependences among variables arises

leading C++ unable to optimize the code as efficiently as FORTRAN. There are advanced C++ language features leading to faster speeds surpassing or becoming equivalent to FORTRAN programs. Therefore FORTRAN or C++ is a good choice for generating numerical floating point calculations quickly.

If an algorithm has a low computational cost, one can use Python. Python is ideal because it is powerful, very clean, simple, and has compact syntax [40]. This programming language is easier to learn than C++ and can be used in combination with other C++ and FORTRAN programs or subroutines. The actual Monte Carlo algorithm which generates different molecular geometries and analyses the energy of the optimized molecular structure is computationally inexpensive and so is written in the Python programming language. The Python module then runs in tandem with the General Atomic and Molecular Electronic Structure System (GAMESS) Quantum Chemistry software [14], where the GAMESS software performs the numerically intensive molecular energy calculations using FORTRAN.

## 1.4 OUTLINE OF THESIS

The remainder of this thesis is organized as follows. Chapter 2 will describe in detail the Monte Carlo (MC) Basin-Hopping approach used in this present study and the principles behind the program. The BH program is then applied to find the GM and lowest energy structures for selected solute-solvent or solute-(H<sub>2</sub>O)<sub>n</sub> systems. This application will model experimental results obtained by simulating glycine and glycine precursors in the interstellar medium (ISM). Chapter 3 will focus on the results of this study using the methylammonium-water (CH<sub>3</sub>NH<sub>3</sub><sup>+</sup>(H<sub>2</sub>O)<sub>1-6</sub>) system for 1-6 discrete water molecules, zwitterionic methylcarbamic acid-water system for 1-8 discrete water molecules (CH<sub>3</sub>NH<sub>2</sub><sup>+</sup>COO<sup>-</sup>(H<sub>2</sub>O)<sub>1-8</sub>), and zwitterionic glycine-water system for 7-8 discrete water molecules ((NH<sub>3</sub><sup>+</sup>CH<sub>2</sub>COO<sup>-</sup>(H<sub>2</sub>O)<sub>7-8</sub>).

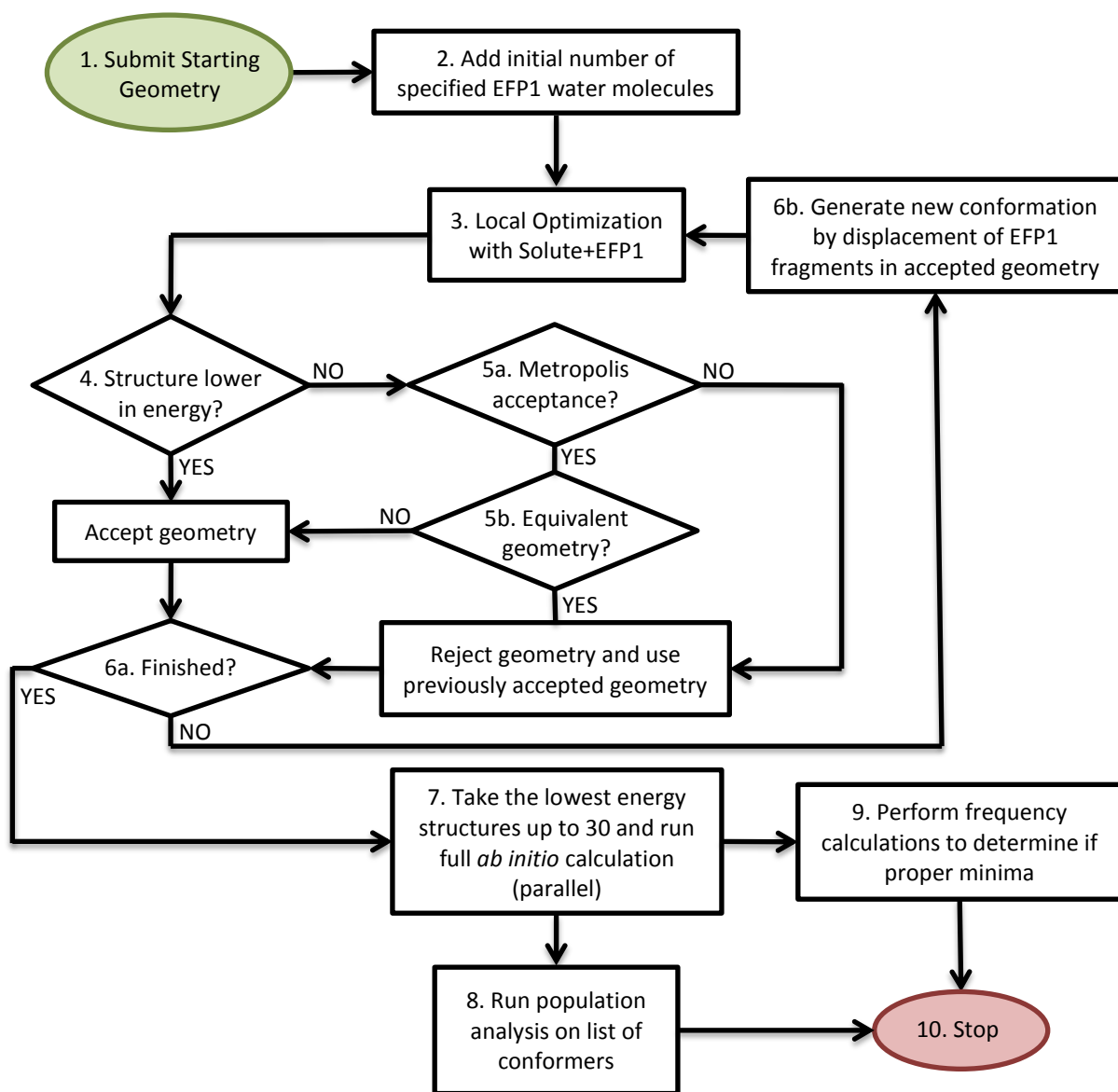


## 2. BASIN-HOPPING GLOBAL OPTIMIZATION STRATEGY

### 2.1 BASIN-HOPPING

The purpose of this thesis is to use the Basin-Hopping (BH) approach to model a solute surrounded by a cluster of water molecules. The approach allows free movement of the solute molecule during the local geometry optimization, while displacing only the surrounding EFP1 water molecules in the BH steps. The solute coordinates are taken from the previous, successful geometry step and not displaced. The first fragmented local geometry optimizations were performed using EFP1 water molecules, while a second non-fragmented local geometry optimization was performed to reduce computational cost. The BH method created for this thesis is similar to the original BH algorithm [9] and uses the Monte Carlo algorithm with the Metropolis method to scan for the global minimum, while using standard local geometry optimization methods with the GAMESS program [14].

Since the major computational effort in the Monte Carlo part of the BH method mainly involves making coordinate displacements, it is not computationally intensive and therefore can be written in Python. Figure 3 shows a flow chart of the main steps in the BH algorithm and more detail on these individual steps are listed below. Prior to running the program, several parameters need to be considered and selected for each system: the number of steps, temperature, average water displacement, shell size or boundary condition, and number of waters. These parameters are adjusted during trial runs in order to achieve the desired acceptance criteria of 50%, where one will want to consider system size when selecting the number of iterations. A large number of steps will waste computational resources, however it will increase confidence that the global minimum has been obtained. Different runs using different starting geometries are performed to verify that the program always produces the same global minimum. If different runs produce different lowest energy structures then changes should be made in the parameters for the system.



**Figure 3.** Flow chart steps following basin-hopping algorithm.

On overview of the BH algorithm is given in Figure 3 and the following discusses the various steps in more detail:

1. A solute molecule is selected and submitted to the program. The number of iterations, temperature, average water displacement, boundary condition, and number of waters desired are fed into the program with the starting solute molecule.

2. The EFP1 solvent water(s) are then added in a random manner around the solute used.
3. The system is then locally optimized with the EFP1 waters, which are also referred to as fragmented waters. Throughout this thesis, local optimization with EFP1 waters is referred to as a fragmented optimization. The GAMESS program was run using 4 CPUs during this step.
4. If the geometry optimization converges without error, the internal coordinates and energies are collected. If the program is after the first iteration, the new energy is compared with the previous energy.
  - a. If the energy is lower, the new system is automatically accepted and added to the list of low energy structures.
5. If the structure is higher in energy, the Metropolis method is run using Equation 1-2.
  - a. If the structure does not meet the Boltzmann probability, it is rejected and the last accepted low energy structure is used as the starting point in the next iteration.
  - b. If the higher energy structure meets the Boltzmann probability, it is tested for geometry equivalence.
    - i. If the structure displays geometrical equivalence, it is rejected, and the last accepted low energy structure is used as the starting point in the next iteration.
    - ii. If the structure displays a geometrical difference, the new structure is accepted and added to the list of low energy structures.
6. The loop continues and is compared to the preselected number of iterations.
  - a. If the loop completes the iterations, the program moves to step 7.
  - b. If the number of iterations has not been reached, each fragmented water is displaced by an average uniform distance specified previously using Equation 1-3. Steps 2 through 5 are then repeated.
7. The 30 lowest energy structures, or all of the lowest energy structures when 30 are not obtained, are then taken and an unfragmented local geometry optimization is performed. The GAMESS program was run using 8 CPUs. Local optimization was performed in parallel on the Hawaii Open Supercomputer.

8. A population analysis of the different structures is performed. This step aids in understanding the PES and can help visualize the random walk performed on the PES, as well as monitor the effectiveness of the program. Each distinct lowest energy structure can also be used to replace step 7.
9. The optimized structures from 7 are then subjected to frequency calculations to confirm that they are proper local minima.
10. The lowest energy structure obtained in step 7 is the GM and the BH program is now complete!

## 2.2 UNIFORM RANDOM STEPS

This algorithm depends on the use of a random number generator, with which one uses a seed value to initiate the set of random numbers to be used throughout the program. Different seed values will produce a different set of random numbers, thereby producing different walks over the PES. The seed is found by taking the system call time and multiplying by 257 and recorded. Any value can be used to multiply the time. However, it should be an odd number since an even number will always produce an even seed and then restrict the random numbers generated in each trial run.

The uniform random values are obtained using the Mersenne Twister (MT19937) method [41]. This pseudo-random number generator has a period of  $2^{19937} - 1$  and is implemented in C. It is threadsafe because it executes in a single Python step. A pseudo-random uniform generator is one in which the points generated are “maximally avoiding” each other [42]. The seed value is recorded with the output generated from the program to allow one to test the algorithm for reproducibility since different computer systems may lead to variations in the structures obtained.

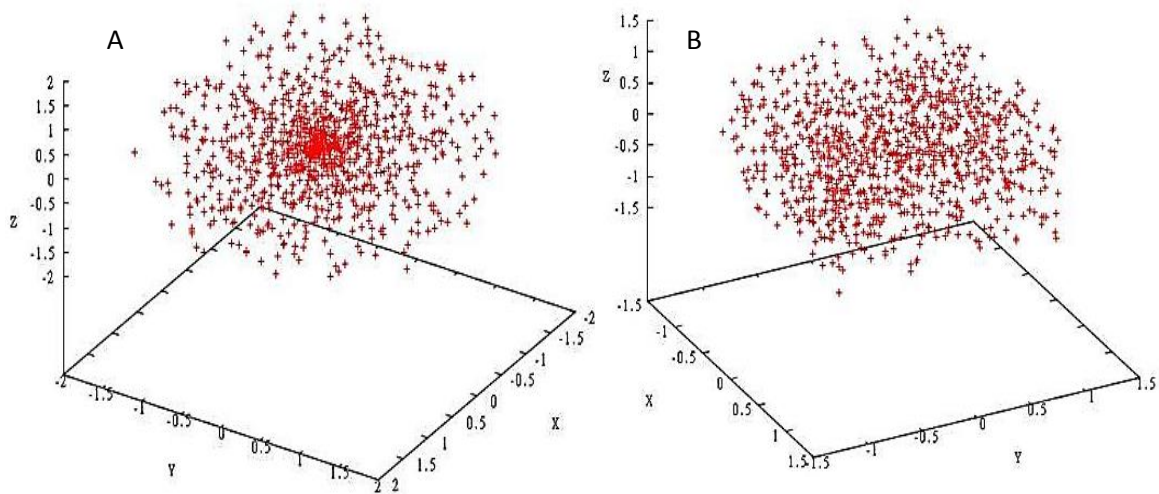
The random numbers are then used in the equations by Marsaglia [43] to generate a random uniform distribution on a unit sphere for each water displacement (Equations 2-1):

$$\begin{aligned}x &= 2x_1\sqrt{1 - x_1^2 - x_2^2} \\y &= 2x_2\sqrt{1 - x_1^2 - x_2^2} \\z &= 1 - 2(x_1^2 + x_2^2)\end{aligned}\tag{2-1}$$

where  $x_1$  and  $x_2$  are uniform random numbers between -1 and 1, inclusive. One can then use this uniform distribution on a sphere to generate an average step of unit length, where one can then multiply this number based on the allowable displacement  $\alpha$  as shown in Equation 2-2.

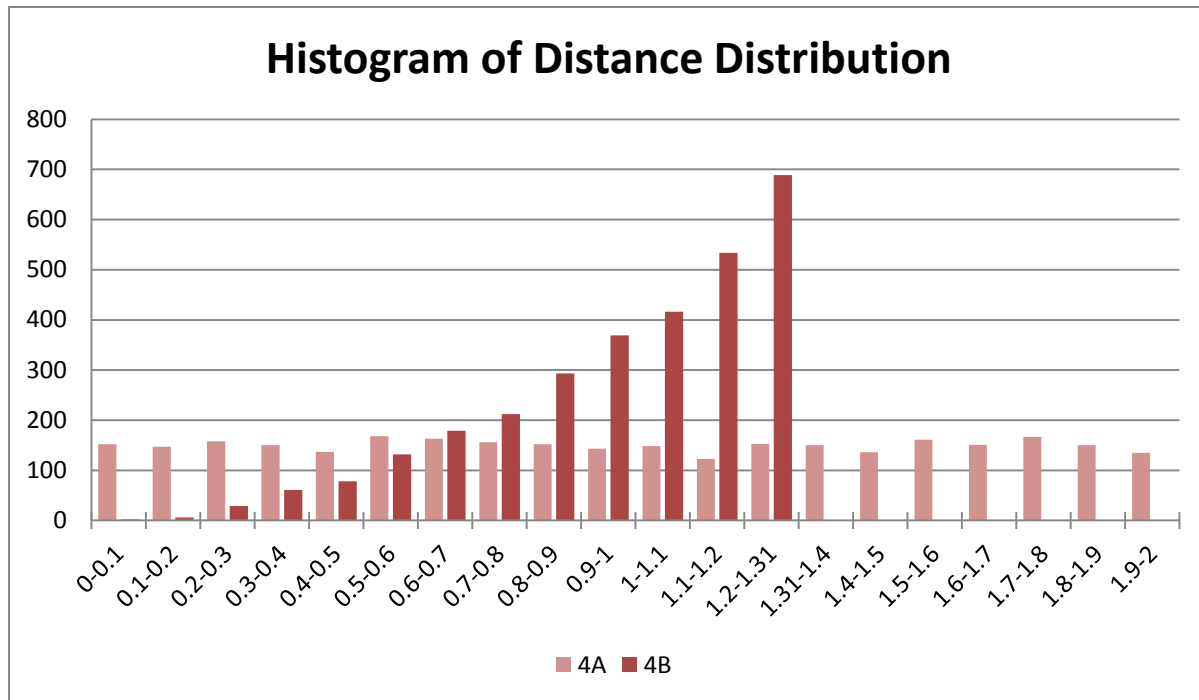
$$\begin{aligned}x_{k+1} &= x_k + \alpha x \\y_{k+1} &= y_k + \alpha y \\z_{k+1} &= z_k + \alpha z\end{aligned}\tag{2-2}$$

Saunders calls this the “kick” method [44]. This application is useful because it allows one to generate the same displacement for each water molecule and also allows a random uniform displacement for all water molecules. Using this method, all waters will have a uniform distance displacement from the starting point or previous point of the atom, and the average distance will be half the radius of the sphere. To generate an average  $\alpha$  of unit length, the radius of the sphere should be 2 units. However, this results in a non-uniform distribution over the space of the entire sphere, but uniform within the distance from the center of the sphere. Figure 4 and Figure 5 below show the distribution of points obtained by the method used in the thesis (A) and the distribution of points obtained by using a uniform population within a sphere (B).



**Figure 4.** Density within sphere using Marsaglia method (A), generates a mean step of unit length, however not uniformly populated within the sphere with radius 2 units. The density within a sphere using uniform random Cartesian points (B), where a conditional statement is applied for points greater than the radius of 1.31 units to obtain a mean step of unit length.

When one uses the uniform distribution of points in a sphere (Figure 4B), one finds that the distribution of the point distances from the origin is not uniform and is heavily populated at larger distances for the uniform sphere as shown in Figure 5. To generate an average distance of 1 unit for a uniform population within a sphere, the radius of the sphere must be set to 1.31 units. This cutoff is due to the non-uniform distance distribution generated.



**Figure 5.** Distance populations for Figure 4 using a uniform distance distribution of points (4A) and a uniform distribution of points within a sphere (4B). Generated with 3000 points using Python.

The Marsaglia method used allows one to get an average step length that is half the radius of the sphere, and allows for a greater displacement of the water molecules. Many applications require the use of uniform distribution within a sphere, whereas the BH algorithm is more concerned with generating a uniform distance displacement.

During several trial runs, it was found that displacing the EFP1 water based on the oxygen coordinates was most effective. Each water molecule is displaced to the new coordinates from the previous coordinates of the oxygen atom using Equations 2-1 and 2-2 with  $\alpha$  being a uniform random number from the parameters used during run initialization. The hydrogen coordinates are then displaced in a random uniform manner using Equations 2-1 and 2-2 around the oxygen atom, using  $\alpha = 0.9468 \text{ \AA}$ . This allows for free rotation of the water molecule where the lone-pair and hydrogen atom interactions with the solute can be modeled uniformly.

A boundary condition must be applied which prevents the EFP1 from getting too far from the solute molecule. The BH program will fail without use of a boundary condition, and it is implemented as a sphere. The farthest atom from the origin (initial sphere distance) is added to the boundary parameter selected during program initiation, to maintain the proximity of the EFP1 water molecules. The origin is an atom closest to the center of the solute and ideally unique. Once the water travels beyond this distance, it is reflected back towards the origin by multiplying the coordinate by the initial sphere distance divided by the current distance.

The oxygen coordinates on the EFP1 water molecule is restricted to at least a 2 Å distance from any atom position in the system, except the corresponding EFP1 hydrogen coordinates. If the water fragment is within this distance, a new O atom displacement or step is selected. This distance restriction allows a water molecule to travel into any cavities that may be present in the solute. Once the number of EFP1 waters meet the minimum distance criteria, the local fragmented geometry optimization is then performed. The solute coordinates are held fixed during the random displacement of the EFP1 coordinates, however they are free to adjust to a new equilibrium structure during optimization.

If the new structure is lower in energy than the previous energy, it is automatically accepted and the next step is performed. If the energy is higher than the previous energy, the Metropolis method [3] is performed using the temperature parameter. Since many similar structures can generate a range of energies, one should also perform equivalence testing after acceptance with the Metropolis method. The step between the new structure and the previously accepted structures continues until the program completes.

### 2.3 GEOMETRICAL EQUIVALENCE TESTING

Gehrke and Reuter [11] included equivalence testing for each new geometric isomer/conformer obtained in their GO method. When a structure passes the Metropolis method, it is tested to the previous structure using Equation 2-3 when the chemical system contains more than three atoms. The equivalence  $\Delta eq$  value used can be changed for the system under study:



$$\Delta eq < \frac{\sum_i (d_{A_i} - d_{B_i})^2}{\sum_i d_{A_i}^2 + d_{B_i}^2} \quad (2-3)$$

The distance of atom  $i$  in structure  $A$  from the origin is  $d_{A_i}$  and structure  $B$  is  $d_{B_i}$ . An atom that is closest to the center of the structure and ideally unique is corrected to the origin for use with Equation 2-3, where the origin is determined based on the farthest maximum atom distance in the solute molecule. In this research,  $\Delta eq$  is set to 0.0001 and was the same result used by Gehrke and Reuter. Upon testing, this value was optimal for the systems used. Since there are multiple atoms of the same type, one cannot use Equation 2-3 explicitly. Due to potential rearrangement of atoms leading to the same structure, each distinct atom is added to an atom array and then sorted based on the distances. The  $i$ th value in the new geometry array  $A$  was tested to the  $i$ th value in the previously accepted geometry array  $B$  using Equation 2-3.

Calculating the distance from a unique atom closest to the center of the solute provides better results, as opposed to using the center of the solute or solute-solvent system. Due to discrepancies observed with each increase in the number of waters, a selected dihedral angle from the solute molecule may be a necessary addition with this equation and must be greater than 7 degrees. The addition of a dihedral comparison increased the performance in the number of new structures obtained for the glycine zwitterion. Different systems may require different formulas to test for equivalence.

## 2.4 ASSESSING ALGORITHM EFFICIENCY

Gehrke and Reuter [11] developed several formulas for assessing the efficiency of a BH algorithm (Equations 2-4 through 2-6). The number of unsuccessful moves where the local optimization of the newly generated structure relaxes back to its previous geometry can be compared using Equation 2-4. The fraction of unsuccessful moves from the locally optimized structures failing the Metropolis method is found with Equation 2-5. The overall efficiency is then determined using Equation 2-6.

$$\alpha_{unsucc} = \frac{N_{unsucc}}{N} \quad (2-4)$$

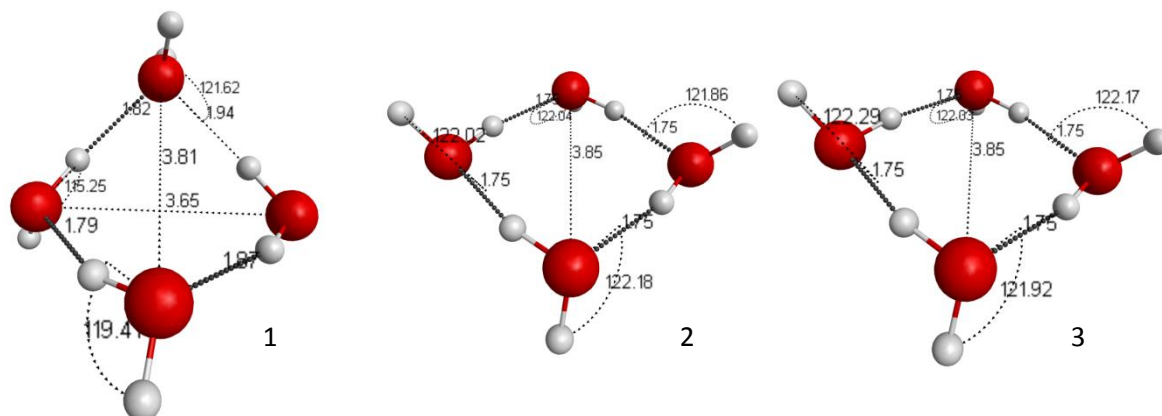
$$\alpha_{high\ E} = \frac{N_{high\ E}}{N} \quad (2-5)$$

$$\alpha_{succ} = 1 - \alpha_{unsucc} - \alpha_{high\ E} \quad (2-6)$$

Wales [9] reports that an efficient BH algorithm should have approximately 50% probability [7] that a move is accepted, where a low acceptance probability may indicate the system is trapped in a local minimum catchment region on the PES surrounded by several higher energy local minimum that will fail the Metropolis method acceptance. Acceptance probability will vary if one elects to solely rely on the energy obtained, as it doesn't test for structures relaxing back into the previous geometry. Adjusting the Metropolis algorithm to use a higher temperature will increase the probability in accepting a higher energy structure, while reducing the temperature will reduce the probability of accepting a higher energy structure. Increasing the average step size will also increase the acceptance probability, however care should be taken when electing to do this.

## 2.5 TESTING THE PERFORMANCE OF THE BH ALGORITHM

The BH program was first tested by finding the GM for some water clusters. Water cluster calculations were performed using B3LYP/6-31++G\*\* with 2-4 EFP1 waters and the geometries were compared with full *ab initio* calculations from the Cambridge Cluster Database [45]. The use of 2 waters models very closely to the full *ab initio* calculation, while 3 waters have slight variation. Figure 6 shows a cluster with 4 waters using only an EFP1(1), the EFP1(1) then followed by full unfragmented calculation (2), and full unfragmented calculation only (3). It should be noted, that unfragmented calculation corresponds to replacing the EFP1 coordinates with explicit O and H atoms and performing the full QM energy calculation. Figure 6 shows there is a distinct variation in the positioning of the water coordinates in the EFP1 calculation alone, however upon subjecting to optimization using the full *ab initio* results in a similar structure to the one obtained using only the full QM calculation.



**Figure 6.** Clusters with 4 waters using only a EFP1 (1), (1) followed by full *ab initio* unfragmented calculation (2), and full *ab initio* unfragmented calculation only (3).

This agrees with the literature which developed the EFP method [33] and confirms the utility of EFPs in the BH algorithm developed for this thesis. Please refer to chapter 1 for the discussion regarding EFP. Ideal results are obtained when a fragmented local optimization followed by a full unfragmented local optimization is lower in computational cost than performing a full unfragmented calculation only. However, performing the local optimization with EFP1 waters does not always generate ideal results. The effectiveness of the BH algorithm with EFP1 was then tested using  $\text{NH}_4(\text{H}_2\text{O})_{5-6}$  waters, where the results were in correlation with their low energy structures obtained by Jiang *et al.* [46] and Douady *et al.* [47].

Geometry optimization with a tight convergence threshold during a fragmented calculation led to a rearrangement of the water molecules during an unfragmented calculation and thereby increased computational cost. In smaller systems this effect is negligible, however in larger systems one must adjust the local geometry convergence threshold to 0.003 during the fragmented calculations. Figure 6 shows a minimal modification of the water cluster structure when going from the EFP1 calculation to the full *ab initio* calculation. Larger differences between EFP1 and full *ab initio* optimized geometries become more apparent for larger water clusters.

### 3. GLOBAL OPTIMIZATION INVESTIGATION OF SYSTEMS INVOLVED IN THE FORMATION OF GLYCINE

#### 3.1 INTRODUCTION

The formation of amino acids is of considerable interest since it can provide insight into the development of prebiotic molecules. The simplest amino acid is glycine, where considerable interest exists in modeling the formation of glycine in the interstellar medium (ISM).

Understanding the chemical properties of glycine lends useful insight into the formation of the molecule. Glycine (gly) exists in its zwitterionic form at the isoelectric point with pH of 6.0, in the crystal and solution form, whereas the neutral species predominates in the gas phase [48-50]. Determining the number of water molecules required to achieve the zwitterion and the number of water molecules to fill the solvation shell of glycine are desirable to accurately model the glycine system and find correlations with experimental results. A computational model that demonstrates the zwitterionic form that is lower in energy than the neutral form would accurately model this experimental determination.

Previous studies have been inconclusive with the number of water molecules necessary to stabilize the zwitterionic over the neutral form of glycine [36],[51-63]. It is found that a 1:1 complex is not enough to stabilise the zwitterionic structure using higher level calculations [55], and no transformation from the zwitterion to the neutral form was found using B3LYP/6-31G\* with normal analysis. However, the CP-corrected optimization transformed the zwitterion to the neutral species. Bachrach [56] reports that two water molecules create the stable zwitterion from the neutral species, where seven molecules reach the isoelectric point with the neutral species, using the B3LYP/6-31G\* level of theory. However, the zwitterion with two water molecules was higher in energy than the neutral form, demonstrating that the system is not an accurate model for the former. An average of 4.4 molecules in the first solvation shell using *ab initio* molecular dynamics methods was determined, with 3 waters around the ammonium group and 2 waters asymmetrically solvated to the two oxygen atoms [63]. It is roughly consistent that the spontaneous intramolecular conversion of the neutral to the zwitterionic form is calculated with the first

solvation shell pentahydrated using DFT by Balta *et al* [54]. Aikens and Gordon [36] found that seven and eight waters are needed to stabilize the zwitterion.

In 2005, Holtom *et al.* [64] investigated the formation of glycine ( $\text{NH}_2\text{CH}_2\text{COOH}$ ) and its isomer methylcarbamic acid ( $\text{CH}_3\text{NHCOOH}$ ). Methylamine-carbon dioxide  $\text{CH}_3\text{NH}_2:\text{CO}_2$  ice mixture deposits were irradiating in an ultrahigh vacuum at 10K to simulate the interstellar medium (ISM) via neighboring radical recombination. Later, in 2009, Bossa *et al.* [65] showed in a similar experiment that methylammonium methylcarbamate  $[\text{CH}_3\text{NH}_3^+][\text{CH}_3\text{NHCOO}^-]$  acts as a glycine salt precursor during their vacuum ultraviolet (VUV) photolysis experiment at 10K. The proposed reaction mechanism begins with the rupture of the carboxyl C-N bond, generating the carbon dioxide anion radical ( $\text{CO}_2^-$ ) and the methyl amino radical ( $\text{CH}_3\text{NH}$ ), which then recombines to form glycinate. If methylcarbamic acid demonstrates a zwitterionic nature, one would expect cleavage of the C-N bond in the methylammonium methylcarbamate, thereby allowing the formation of glycinate in the above work. In the paper by Bossa *et al.*, they observed the formation of methylamine and methylcarbamic acid induced by ultraviolet photons from methylammonium methylcarbamate. In the gas phase, Bossa *et al* [66] found that the neutral form of methylcarbamic acid decomposes to  $\text{CO}_2$  and  $\text{CH}_3\text{NH}_2$ .

The above summary of research on the glycine structure led to the development of the BH algorithm described in chapter 2 to search for the lowest energy optimized structural isomers of a particular compound. In this chapter, I report the results obtained using the BH algorithm on solutes consisting of methylammonium cation, methylcarbamic acid (mca) zwitterion, and glycine (gly) zwitterion surrounded by water molecules. Proper implementation of a GO algorithm increases efficiency in calculations and eliminates biased structural guessing. In developing the BH algorithm presented in chapter 2, many variations in parameters have been tested. The results obtained from the BH algorithm demonstrate that different parameter values will generate different results, and must be tuned to generate a computational efficient unbiased search for the global minimum (GM).

It is the main focus of this thesis to find low energy structures using the BH method described in chapter 2. This global optimization (GO) method allows one to find unbiased structures that may not be obtained based on chemical intuition alone. It is also of interest to identify the low energy structures of the zwitterionic methylcarbamic acid and determine how many waters are needed such that the low energy structures of methylcarbamic acid and glycine switch.

Previous studies by Kayi, Kaiser, and Head [67-69] have modelled the methylcarbamic acid and glycine solvation using, in part, results obtained by Aikens and Gordon [36]. The gly structures were obtained by initially using the low energy structures obtained by Aikens and Gordon and either modifying the positions of the water molecules or by addition of one or two water molecules and performing a new geometry optimization. Prior to the Kayi *et al.* work, there was an absence of mca structures surrounded by water molecules in the literature. The mca structures were generated by randomly placing a water molecule around the neutral mca structure and then optimizing. The next set of mca structures were then produced randomly by placing another water at a location around mca where some new hydrogen bonds might be formed and optimized again. A similar approach was taken for the mca zwitterion. Using chemical intuition based on literature results and experimentation, this strategy can quickly generate several low energy structures which can also be found with a GO method, however the implementation of a GO method should systematically find structures lower in energy. The implementation of a reliable GO method should scan the PES more thoroughly and may explore several structures not anticipated using chemical intuition alone. It is the purpose of this research to confirm that the structures obtained by Kayi *et al.* [69] are the lowest energy conformers. It is also possible to generate a new global minimum structure for each water cluster thereby obtaining a more reliable estimate for the relative stability of the zwitterionic forms of methylcarbamic acid and glycine.

The work of Kayi *et al.* [69] demonstrates that neutral mca is appreciably more stable than the zwitterionic mca. However, Khanna and Moore [70] proposing that the stable structure of carbamic acid is associated with the zwitterionic form ( $\text{NH}_3^+\text{COO}^-$ ). It was elected to focus on identifying the GM for the mca zwitterion with 1-8 waters and gly zwitterion with

7-8 waters. Previous work [36], [56] showed that if there are 1-6 water molecules the neutral gly species is more stable than the zwitterion. In contrast, Kayi *et al* [69] confirmed that the gly zwitterion to be more stable with 9 and 10 waters than the neutral gly at the MP2/6-311++G(d,p) [71-75] level of theory, where there was an increasing overlap in the energies for the zwitterionic form of mca and the zwitterionic form of gly at 7 and 8 water molecules.

In the present work, an incremental number of water molecules are placed around the desired solute molecule to model a water cluster similar to that occurring in the ISM. The following are the main questions this work hopes to address. (1) Will the developed BH program find a new global minimum not previously reported? (2) How many water molecules are needed to stabilize the zwitterionic form of gly over the zwitterionic form of mca? (3) How does the geometry change with the incremental addition of water molecules? (4) Will the computational results provide any new insights into how mca acts as a precursor to gly formation?

### 3.2 METHYLAMMONIUM CATION

It is of particular interest to determine if the amine group ( $-\text{NH}_3^+$ ) in gly forms 3 hydrogen bonds with water. Therefore, the solvation of  $\text{CH}_3\text{NH}_3^+$  is of importance in understanding the interaction with the amine functional group in glycine [76] [77]. Methylammonium ion ( $\text{CH}_3\text{NH}_3^+$ ) has a  $\text{pK}_a$  of  $2.3 \times 10^{-11}$  and involves similar hydrophobic, hydrogen-bonding, and ionic interactions associated with glycine. Using a Monte Carlo simulation with the TIP4P potential for water and analogous potentials for the ions, Alagona *et al.* [76] found that the  $-\text{NH}_3^+$  group has an average coordination number of 3.5. This coordination number corresponds to one water coordinating with each N-H hydrogen and a bridging interaction between these water molecules. Alagona *et al.* [76] also found that waters near the  $-\text{CH}_3$  group of  $\text{CH}_3\text{NH}_3^+$  behave more like the waters around a polar group than the waters around the  $-\text{CH}_3$  group of  $\text{CH}_3\text{COO}^-$ , where those waters behave more like the bulk water. This observation is due to the redistribution of cationic charge near the protonated amine resulting in a large partial positive charge.

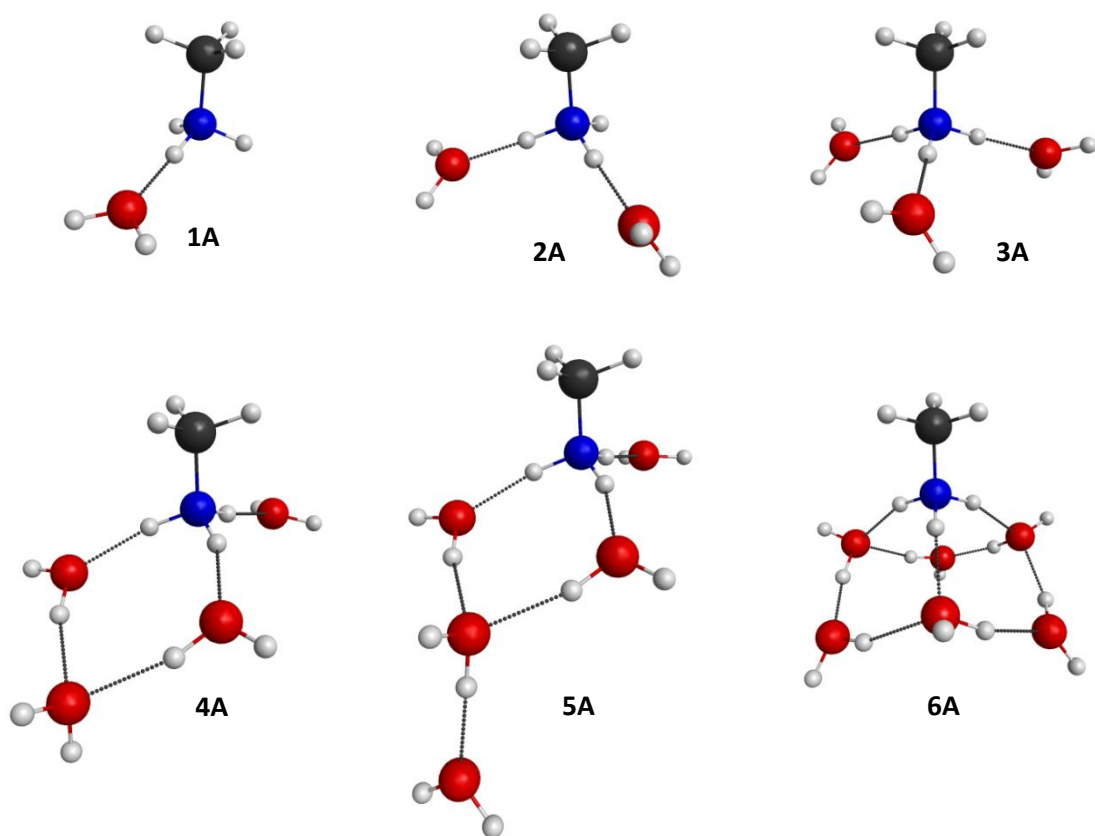
Using the Gaussian 94 program Masamura performed an *ab initio* study of  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$  using MP2/6-31G(d) and RHF/4-31G levels of theory [77], where a slight positive variation in enthalpies of solvation is reported from the compared experimental results [78]. In this paper, Masamura found the C-N bond length of  $\text{CH}_3\text{NH}_3^+$  to be 0.044 Å longer than  $\text{CH}_3\text{NH}_2$  at the MP2/6-31G(d) level of theory and suggested the elongation of the bond was due to resonance of  $\text{CH}_3^+\cdots\text{NH}_3$  (3-1).



The lowest energy structures obtained by the BH algorithm for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$  are shown in Figure 7. For 4-6 water molecules the BH algorithm finds several lower energy structures than those obtained by Masamura [77] for 4-6 water molecules. The corresponding structure energies obtained are given in Table 1. In Masamura's paper, the optimized structures showed no bridging between the water molecules. Masamura incremented the number of water molecules by replacing one of the lone water molecules which was hydrogen bonded to the  $-\text{NH}_3^+$  group with a water dimer for the clusters with 4-6 waters. The bridging interaction is shown below to reduce the energy and corresponds to the extra 0.5 in the coordination number in the work by Alagona *et al.* [76].

**BH Parameters:** Due to the simplicity of the structure, the BH algorithm was performed using one simulation at 100 steps for 1-4 waters and 3 simulations at 200 steps for 5 and 6 waters with a water displacement of 1.0 Å, a boundary condition of 5.0 Å, and a temperature of 298 K. The GAMESS EFP1 optimized structures are performed using B3LYP/6-311++G(*d,p*) [17-24] energy calculation with a gradient convergence tolerance of 0.0001. The full unfragmented calculation is performed with the same basis set and a gradient convergence tolerance of 0.0001. Frequency calculations are performed on the lowest energy structures to determine if the structure corresponds to a proper minimum and calculate the ZPE correction.





**Figure 7.** Lowest energy structures obtained with  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$  water molecules after ZPE correction. The label xA correspond to the number of water molecules with A representing the GM obtained.

Throughout this thesis, the notation “energy method/basis set” is used: hence B3LYP/6-311++G(*d,p*) means that B3LYP DFT energy calculations described in chapter 1 were performed using the standard 6-311++G(*d,p*) [73], [75] basis set, where ++ and *d,p* are the diffuse and polarization functions, respectively. The structures obtained for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$  are labeled with the number of waters and relative energy ordering, where 6A corresponds to the GM with 6 waters and 6B corresponds to next lowest energy structure.

The lowest energy structures for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$  are shown in Figure 7. The energies of the optimized structures are given in Table 1 and compared with the results of Masamura. In Figure 7, the lowest energy structures with 4-6 waters show the water bridging that is expected due to the coordination number of 3.5. Please refer to the Appendix for a complete

list of the atom coordinates and ZPE corrected energies generated and used throughout this chapter.

**Table 1.** Comparison of the lowest energies (hartree) obtained from calculations of H<sub>2</sub>O and CH<sub>3</sub>NH<sub>3</sub><sup>+</sup>(H<sub>2</sub>O)<sub>0-6</sub> with the results of Masamura<sup>a</sup> without ZPE corrections.

<i>n</i>	4-31G <sup>a</sup>	MP2/6-31G( <i>d</i> ) <sup>a</sup>	B3LYP/6-311++G( <i>d,p</i> )
H <sub>2</sub> O	-75.88251	-76.19685	-76.42219
0	-95.44076	-95.86820	-96.18430
1	-171.38915	-172.10043	-172.63746
2	-247.33112	-248.32756	-249.08582
3	-323.26797	-324.55069	-325.53060
4	-399.20279	-400.76946	-401.97492
5	-475.13648	-476.98727	-478.41732
6	-551.06916	-553.20421	-554.86123

<sup>a</sup>Ref. [77]

Masamura [77] calculated the change in energy ( $\Delta E_{n-1,n}$ ) using the following equation [77]:

$$\Delta E_{n-1,n} = E[\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_n] - E[\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{n-1}] - E(\text{H}_2\text{O}) \quad (3-2)$$

These  $\Delta E_{n-1,n}$  results are shown in Table 2 below along with the results generated by the present study. The  $\Delta E_{n-1,n}^{\text{ZPE}}$  value in Table 2 includes the ZPE correction. The enthalpy changes ( $-\Delta H_{n-1,n}^{298K}$ ) are also shown for the CH<sub>3</sub>NH<sub>3</sub><sup>+</sup>(H<sub>2</sub>O)<sub>1-6</sub> clusters, which are calculated using Equation 3-3. The enthalpy is calculated from the electronic energy and contributions from the vibrational, rotational, and translational partition functions [79].

$$\Delta H_{n-1,n}^{298K} = H^{298K}[\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_n] - H^{298K}[\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{n-1}] - H^{298K}(\text{H}_2\text{O}) \quad (3-3)$$

Table 2 demonstrates that the calculations by Masamura overestimate the enthalpy of solvation, whereas having the lower energy cluster and using the B3LYP/6-311++G\*\* level of theory results in better agreement with the experimental values. The zero point energy (ZPE) corrections are also shown in Table 2.

**Table 2.** Energies and enthalpies of solvation obtained from calculations of  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$ .

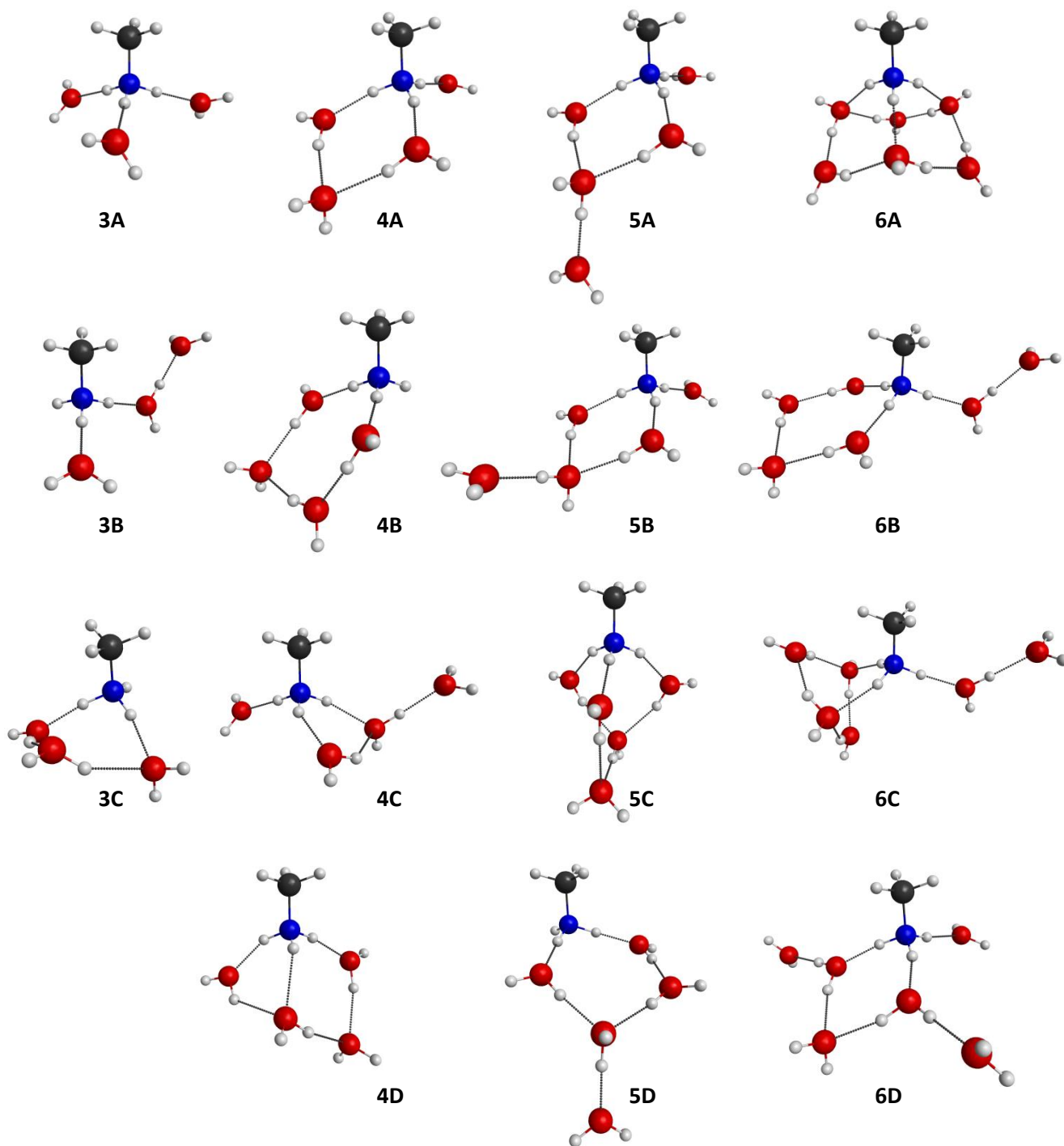
$n$	4-31G <sup>a</sup>		MP2/6-31G(d) <sup>a</sup>		B3LYP/6-311++G(d,p) <sup>c</sup>			Experimental <sub>b</sub>
	$-\Delta E_{n-1,n}$ (kcal/mol)	$-\Delta H_{n-1,n}^{298K}$ (kcal/mol)	$-\Delta E_{n-1,n}$ (kcal/mol)	$-\Delta H_{n-1,n}^{298K}$ (kcal/mol)	$-\Delta E_{n-1,n}$ (kcal/mol)	$-\Delta H_{n-1,n}^{298K}$ (kcal/mol)	$-\Delta E_{n-1,n}^{ZPE}$ (kcal/mol)	$-\Delta H_{n-1,n}^o$ (kcal/mol)
1	24.9*	23.1	22.2	20.8	19.4	18.0	17.6	16.8
2	20.9*	19	19	17.4	16.4	14.9	14.7	14.6
3	17.7*	15.9	16.5	14.9	14.2	12.6	12.6	12.3
4	16.4*	14.4	13.8	-	13.9	12.0	11.2	10.3
5	15.7*	13.7	13.2	-	12.7	10.8	10.3	9
6	15.1*	13	12.6	-	13.6 <sup>†</sup>	11.4 <sup>†</sup>	9.2 <sup>†</sup>	8.5
6B	-	-	-	-	10.1	8.4	8.1	-

<sup>a</sup>Ref. [77], <sup>b</sup>Ref. [78], <sup>c</sup>M.Stryker, \*could not reproduce values using data in ref [77], <sup>†</sup>higher energy structures will produce lower  $-\Delta E$  values.

When viewing these results one should consider the structural changes that occur with each additional water molecule. Masamura assumed there were negligible structural changes and did not account for any bridging between the waters. This bridging would explain the energy and enthalpy variations that are observed at the solvation with 4 waters and again at the solvation with 6 waters for the present results shown in Table 2. The solvation of the methylammonium ion with 4 waters is the first instance of a water bridge for the fully solvated amine group and the solvation with 6 waters shows a significant reordering of the water cluster from that around the amine group with 5 waters. When comparing the next highest energy structure with 6 waters (6B), a better agreement in energy and enthalpy solvation is found. In Masamura's 4 water cluster, a single water and water dimer are hydrogen bonded to 2 and 1 N-H hydrogen(s), respectively. This non-bridging structure presumably models more closely to the experimental results than the GM obtained in the present work at 4 waters.

The 3 and 4 lowest energy conformers for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_3$  and  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{4-6}$ , respectively, are shown in Figure 8 with their corresponding relative energies listed in Table 3. The large structural rearrangement for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_6$  producing 3 bridging water molecules is within chemical accuracy from the next lowest energy structure with 6 water

molecules, where chemical accuracy for experimentally measuring energies is considered to be  $\sim 1$  kcal/mol or 4.18 kJ/mol [80].



**Figure 8.** Lowest energy structures obtained with  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{3-6}$  water molecules after ZPE correction.

**Table 3.** Relative energies (kcal/mol) obtained for the 4 lowest energy conformers from calculations for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{3-6}$  using B3LYP/6-311++G(*d,p*)<sup>a</sup>.

ID	$\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_3$	$\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_4$	$\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_5$	$\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_6$
A	0	0	0	0
B	1.2	2.0	0.02	1.1
C	4.3	2.6	1.4	1.6
D	-	2.7	1.5	2.1

<sup>a</sup>M.Stryker

Table 4 shows, upon incremental addition of each water molecule, the C-N bond length reduces slightly until there are 4 water molecules. With 4 or more water molecules, the C-N bond length does not change significantly. If the water distribution is not uniform around the amine, it is shown that bridging of the water molecules will help stabilize the N-H bond lengths. The shortest N-H bond lengths are obtained when there is no hydrogen bonding between a water molecule and the amine hydrogen, where these N-H distances are starred in Table 4.

**Table 4.** Bond lengths for lowest energy values obtained from calculations for  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{0-6}$

ID	N-H (Å)	N-H (Å)	N-H (Å)	N-C (Å)
$\text{CH}_3\text{NH}_3^+$	1.025	1.025	1.025	1.516
1A	1.023*	1.051	1.023*	1.507
2A	1.042	1.043	1.021*	1.500
3A	1.036	1.037	1.037	1.496
3B	1.053 <sup>††</sup>	1.039 <sup>†</sup>	1.021*	1.498
3C	1.034	1.054	1.021*	1.500
4A	1.038	1.034 <sup>†</sup>	1.038	1.493
4B	1.021*	1.053	1.043	1.496
4C	1.036	1.035	1.036	1.493
4D	1.040	1.027	1.038	1.492
5A	1.033 <sup>†</sup>	1.039	1.040	1.492
5B	1.033 <sup>†</sup>	1.039	1.039	1.492
5C	1.037	1.037	1.038	1.491
5D	1.046	1.020*	1.054	1.495
6A	1.039	1.039	1.0385	1.493
6B	1.035	1.043	1.040 <sup>††</sup>	1.491
6C	1.034	1.045 <sup>††</sup>	1.034	1.494
6D	1.045	1.044	1.031 <sup>†</sup>	1.490

\*Indicates hydrogen atom that has no hydrogen bond with water, <sup>†</sup>Indicates hydrogen bond with one water, and <sup>††</sup>indicates hydrogen bonding with a water dimer.

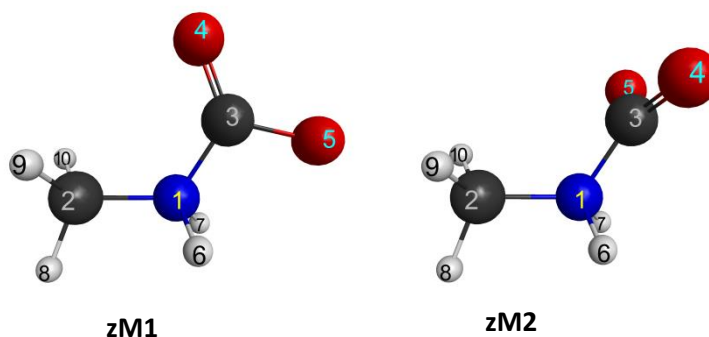
The  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_n$  structures are consistent with the Monte Carlo calculations of Alagona *et al.* [76], where the average coordination number of the amine group is 3.5. The lower energy structures display a bridging between the water molecules and this should also be expected in the hydrated zwitterionic form of glycine.

### 3.3 METHYLCARBAMIC ACID

Holtom *et al.* [64] investigated the formation of gly by irradiating  $\text{CH}_3\text{NH}_2:\text{CO}_2$  in an ultrahigh vacuum at 10K to simulate the interstellar ices, thereby forming the gly zwitterion, anionic gly, and the isomer of anionic gly, i.e.,  $\text{CH}_3\text{NHCO}_2^-$ . Bossa *et al.* [65-66] similarly studied the evolution of interstellar ice analogues through the reactivity between  $\text{H}_2\text{O}:\text{CH}_3\text{NH}_2:\text{CO}_2$  and  $\text{CH}_3\text{NH}_2:\text{CO}_2$  ice mixtures, where methylammonium methylcarbamate [ $\text{CH}_3\text{NH}_3^+[\text{O}_2\text{CNHCH}_3]$ ] and mca was produced. At temperatures above 40K, Bossa *et al.* [66] found that methylammonium methylcarbamate desorbs forming mca, where the gas is unstable and decomposed into  $\text{CH}_3\text{NH}_2$  and  $\text{CO}_2$ . If the neutral form of mca is predominant, this would require proton transfer(s) during decomposition, whereas the zwitterion would not. Reactions of  $\text{CO}_2$  with amines has been studied and found to produce alkylammonium alkylcarbamates [ $\text{NH}_2\text{RR}'[\text{O}_2\text{CNRR}']$ ] [81]. Caplow [82] found that proton transfer to the nitrogen atom in a neutral carbamic acid generating the zwitterion is thermodynamically unfavorable and relatively slow, the formation of a carbamate species is a two-step mechanism with a zwitterion intermediate or a carbamic acid transient species.

Consequently, understanding  $\text{CO}_2$  and  $\text{CH}_3\text{NH}_2$  interactions in water solvent is important when modelling the work by Holtom *et al.* [64] and Bossa *et al.* [65-66]. The uptake of  $\text{CO}_2$  by ionic liquids containing amine groups have also been studied [83-84]. The interaction energies between  $\text{CO}_2$  and amine complexes were studied by Jorgensen *et al.* [85], where they demonstrated that the addition of  $\text{H}_2\text{O}$  and  $\text{CH}_3\text{NH}_2$  solvent molecules increases the  $\text{CO}_2$  and amine interaction energy. Modelling the incremental solvation of the zwitterionic form of methylcarbamic acid will assist in not only the understanding of glycine formation, but aid in the understanding of  $\text{CO}_2$  reactions in amine based ionic liquids.

The zwitterionic form of MCA was explored with the BH method by placing a specified number of water molecules around the two different structures of mca shown in Figure 8. The atom numbering scheme shown in Figure 9 will be used to label structural data in future tables and discussion.

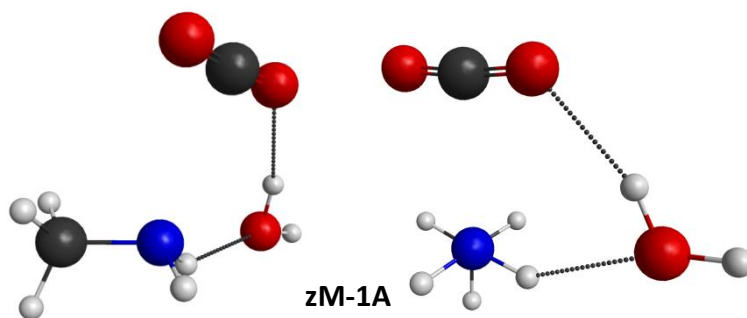


**Figure 9.** Initial zwitterionic methylcarbamic acid structures submitted for global optimization. Atom 1 is N, 2 and 3 are C, and 4 and 5 are O.

Throughout the paper the optimized low energy structures of mca are denoted with a zM-*x*a to zM-*x*d, where *x* is replaced by the number of discrete water molecules in the cluster. The N-C(3) bond was fixed at 1.50 Å during the initial EFP1 calculations to prevent the decomposition into CH<sub>3</sub>NH<sub>2</sub> and CO<sub>2</sub> which occurs otherwise during EFP1 calculations and to allow for the possible loss of the hydrogens from the nitrogen atom. It was expected that the neutral form of mca would be obtained by fixing the N-C (3) bond length since the neutral form has previously been found to be lower in energy [67], [69]. The N-C (3) bond length was then allowed to relax during the non-fragmented geometry optimization calculations. Due to the large system size relative to CH<sub>3</sub>NH<sub>3</sub><sup>+</sup>, the local optimizations were performed using B3LYP/6-31+G(*d*). To verify the effectiveness of using the 6-31+G(*d*) basis set, single-point energy calculations were performed with B3LYP/6-311++G(*d,p*) and MP2/6-311++G(*d,p*) [71-75] followed by ZPE corrections. Single-point energy calculations are performed to obtain the energy at a specific geometrical arrangement without local optimization. In this section, the single-point calculations are performed using the B3LYP/6-31+G(*d*) optimized geometry.

**BH Parameters:** The average displacement of the water molecules is set to 1.5 Å, the temperature is 1000 K, and a boundary condition of 4 Å. Several test runs were performed with the number of trial steps greater than or equal to 400, and in these runs the global minimum was located within 100 and 200 steps for 1-3 and 4-8 waters, respectively. The initial number of steps,  $N$ , are set at 100 and 200 for 1-3 and 4-8 waters, respectively, and are ran with fragmented EFP1 waters. The structures containing EFP1 waters were locally optimized using B3LYP/6-31+G\* [86] with a gradient convergence tolerance of 0.003, while the full unfragmented calculation was performed with the same level of theory with a gradient convergence tolerance of 0.0001. Frequency calculations are performed on the lowest energy structures to determine if the structure corresponds to a proper minimum and calculate the ZPE correction.

**$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_1$ :** Only one stable low energy configuration is found for the addition of one water shown in Figure 10. Refer to Table 5 for the corresponding structural data. In this system, the N-C(3) distance is 2.769 Å which reflects a decomposition of mca into a  $\text{CO}_2$  and  $\text{CH}_3\text{NH}_2$ . The  $\text{CO}_2$ ,  $\text{CH}_3\text{NH}_2$ , and  $\text{H}_2\text{O}$  are held together as a van der Waals complex.



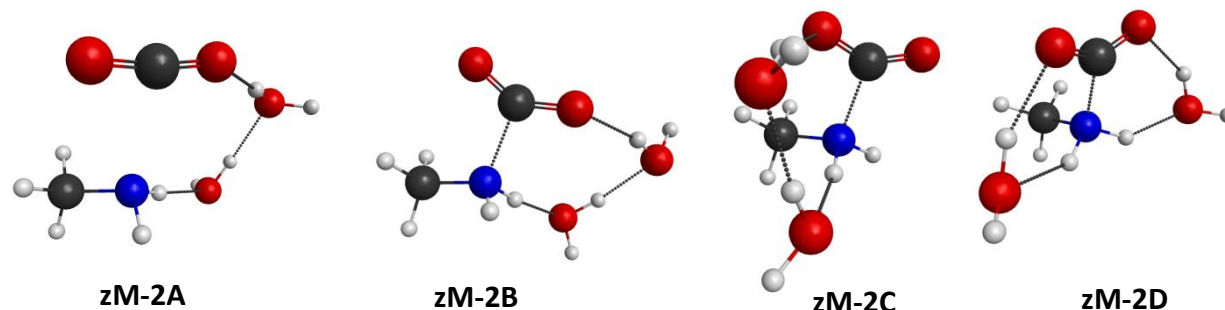
**Figure 10.** The side and Newman projection of the lowest energy structure for the zwitterionic form of methylcarbamic acid with one water molecule.



**Table 5.** Global minimum geometry data for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})$  structure resulting from the unfragmented B3LYP/6-31+G(*d*) level of theory.

ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-1A	1.470	2.769	1.168	1.175	1.019	1.020	108	108	107	173	112	128	1.11

**$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_2$ :** Figure 11 shows the lowest energy optimized structures obtained for the mca zwitterion with two waters, where Table 6 gives their relative energies and Table 7 lists some of the structural data. These structures demonstrate a strong direct correlation between the N-C(3) distance with the energy where the N-C(3) distance decreases as the energy increases. The waters forming an asymmetrical ring with the  $\text{CO}_2$  and  $-\text{NH}_2$  groups increase the N-H bond length when hydrogen bonding with the water shown in Table 7. The N-H hydrogen that lacks hydrogen bonding with water shows a shorter bond length. As Masamura [77] demonstrated with N-C bond length in methylamine and the methylammonium ion, the N-C(2) distance increases when the N-C(3) distance decreases.



**Figure 11.** The four lowest energy structures for the zwitterionic form of methylcarbamate with two water molecules.

**Table 6.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_2$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

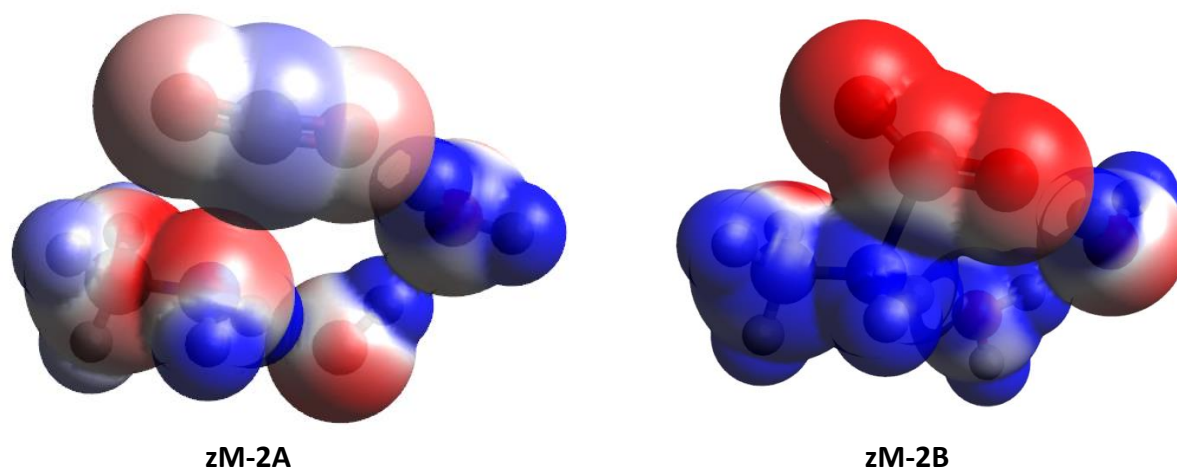
ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-2A	0	0	0
zM-2B	8.3	18.5	56.2
zM-2C	10.4	20.8	57.4
zM-2D	16.9	28.6	58.9

Note there is a large energy difference between the B3LYP and MP2 energies for structures zM-2A and zM-2B in Table 6, most likely due to the difference in calculating the intermolecular/dispersion interactions between carbon dioxide (CO<sub>2</sub>) and methylamine (CH<sub>3</sub>NH<sub>2</sub>) in zM-2A. There is also a strong indication that two waters are not able to stabilize the mca zwitterion, as zM-2A has a N-C(3) distance of 2.692 Å. The longer distance indicates the decomposition of mca into a CH<sub>3</sub>NH<sub>2</sub>, CO<sub>2</sub>, and 2H<sub>2</sub>O van der Waals complex.

**Table 7.** Structural data for the 4 lowest energy CH<sub>3</sub>NH<sub>2</sub><sup>+</sup>CO<sub>2</sub><sup>-</sup>(H<sub>2</sub>O)<sub>2</sub> clusters.

ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-2A	1.470	2.692	1.168	1.176	1.020	1.024	105	116	106	171	109	145	0.99
zM-2B	1.476	1.793	1.221	1.203	1.021	1.037	105	106	107	144	115	179	4.44
zM-2C	1.479	1.772	1.221	1.206	1.037	1.021	107	105	109	143	113	125	4.62
zM-2D	1.487	1.676	1.225	1.225	1.032	1.032	107	107	111	139	112	90	3.94

Table 7 lists the dipole moment for zM-2A at 0.99 Debye. Table 5, using one water, gave a similar dipole moment of 1.11 D for the complex. The energies for zM-2B through zM-2D are very similar with a decreased N-C(3) distances of ~0.9 Å from zM-2A and a much larger dipole moments in the range of 3.9-4.6 D stabilized by the water solvent. The difference in dipole moment is a direct result of the increased polarity in the zM structures with a N-C(3) bond. The results by Kayi *et al.* [69] using the MP2/6-311++G(d,p) level of theory show a relative energy variation of 50.9 kJ/mol between the lowest energy neutral form of mca and the lowest energy mca zwitterion corresponding to structure zM-2D in the present work. The present work indicates that the decomposition into CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub> is more stable than the zwitterionic and neutral form of mca and that two waters are not sufficient to stabilize mca.



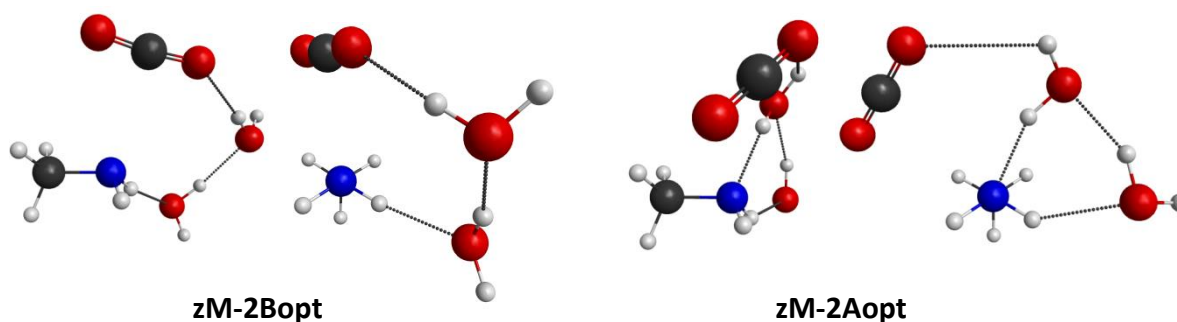
**Figure 12.** Electrostatic potentials of the two lowest energy structures with two water molecules  $\text{CH}_3\text{NH}_2\text{CO}_2(\text{H}_2\text{O})_2$ . (See the electronic copy for the color version.)

Figure 12 shows the electrostatic potential surfaces of zM-2A and zM-2B, scaling from the more electron rich (red) to the electron deficient (blue). It is shown that the charge distribution changes dramatically between the two structures, where the electron rich region of  $\text{CH}_3\text{NH}_2$  on zM-2A is transferred to the  $-\text{CO}_2$  group on zM-2B. This charge distribution correlates with the variation in the dipole moment shown in Table 7.

Performing a further local geometry optimization with the BH coordinates found for zM-2A through zM-2D using the B3LYP/6-311++G(*d,p*) and MP2/6-311++G(*d,p*) levels of theory will lead to geometrical changes for structures zM-2B and zM-2C and zM-2A through zM-2C, respectively. Figure 13 shows the resulting geometries obtained from this optimization, while Table 8 shows the new relative energies.

**Table 8.** Relative energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_2$  after a full geometry optimizing using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-2Aopt	0.6	0
zM-2Bopt	0	0.9
zM-2Copt	0.6	0.3
zM-2Dopt	29.7	60.4



**Figure 13.** Side and Newman projections of the optimized geometries of the low energy structures zM-2B and zM-2A using B3LYP/6-311++G\*\* and MP2/6-311++G\*\*, respectively.

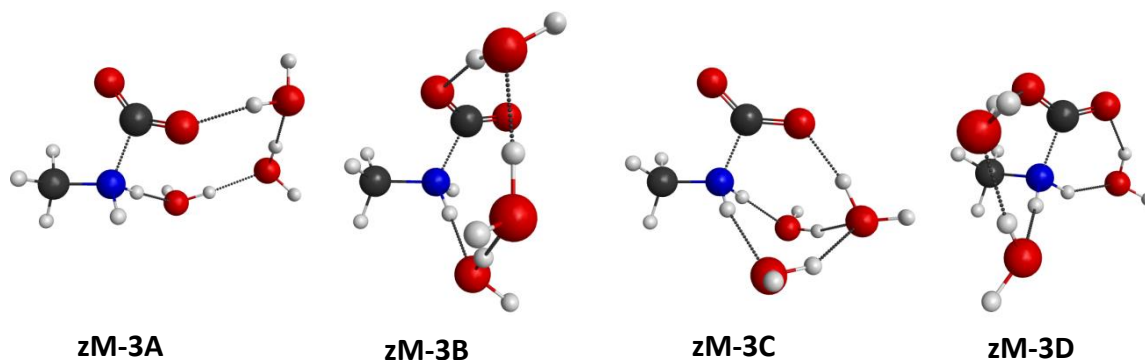
Performing a geometry optimization using the B3LYP/6-311++G(*d,p*) level of theory caused zM-2B and zM-2C to form a structure similar to zM-2A, where zM-2Bopt in Figure 13 is the reordered structure of zM-2B. The only difference in zM-2A and zM-2B is the direction the non-interacting hydrogen atoms face on the water molecules. Optimization using MP2/6-311++G(*d,p*) also generated similar structures for zM-2Aopt, zM-2Bopt, and zM-2Copt, where only zM-2Aopt is shown in Figure 13. This is a clear demonstration of the variation in the B3LYP and MP2 energy methods, where MP2 performs better at modeling the intermolecular interaction. Structures zM-2Aopt to zM-2Copt using the B3LYP/6-311++G(*d,p*) and MP2/6-311++G(*d,p*) levels of theory generate a N-C(3) distance of 2.78 Å and 3.15 Å, respectively. Only zM-2D maintained its original structure upon further optimization using the B3LYP/6-311++G(*d,p*) and MP2/6-311++G(*d,p*) levels of theory. Two waters are not sufficient to stabilize the zwitterionic form of mca. Therefore, it is concluded that the lowest energy zM structures with 2 waters creates a van der Waals complex between CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub>, where this complex is ~10 kJ/mol lower in energy than the neutral mca reported by Kayi *et al.* at the MP2/6-311++G(*d,p*) levels of theory.

**CH<sub>3</sub>NH<sub>2</sub><sup>+</sup>CO<sub>2</sub><sup>-</sup>(H<sub>2</sub>O)<sub>3</sub>:** Figure 14 shows the lowest energy optimized structures obtained for the mca zwitterion with three waters, where Table 9 gives their relative energies and Table 10 lists some of the structural data. The optimized structures with three waters shows a weak correlation between the N-C(3) distance and the energy, where the N-C(3) distance decreases as the energy increases. As observed with two waters, the N-H bond interactions with the

largest number of water molecules will show an increase in the N-H bond length. The lengthening of the two N-H bond is symmetric when both N-H bonds interact with water.

**Table 9.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_3$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-3A	0	0	2.3
zM-3B	0.1	0.7	2.6
zM-3C	2.1	3.2	2.1
zM-3D	3.5	4.0	0



**Figure 14.** The four lowest energy structures for the zwitterionic form of methylcarbamic acid with three water molecules.

Water bridging was shown previously with the  $\text{CH}_3\text{NH}_3^+(\text{H}_2\text{O})_{1-6}$  clusters, and is further demonstrated with this system. The water cage/bridging structure has also been demonstrated by previous work with gly to lower the energy[36],[51-62]. The 4 lowest energy structures in Table 9 have an energy variation less than 4 kJ/mol, which is within chemical accuracy. Multiple structures with energy differences within chemical accuracy and the relative energy reordering observed for MP2 single-point energies suggest that using only the GM may not be as important for describing the system. The structural data in Table 10 shows the 4 lowest energy structures having a N-C(3) distances less than 1.69 Å. Therefore, 3 waters are necessary to stabilize the mca zwitterion.

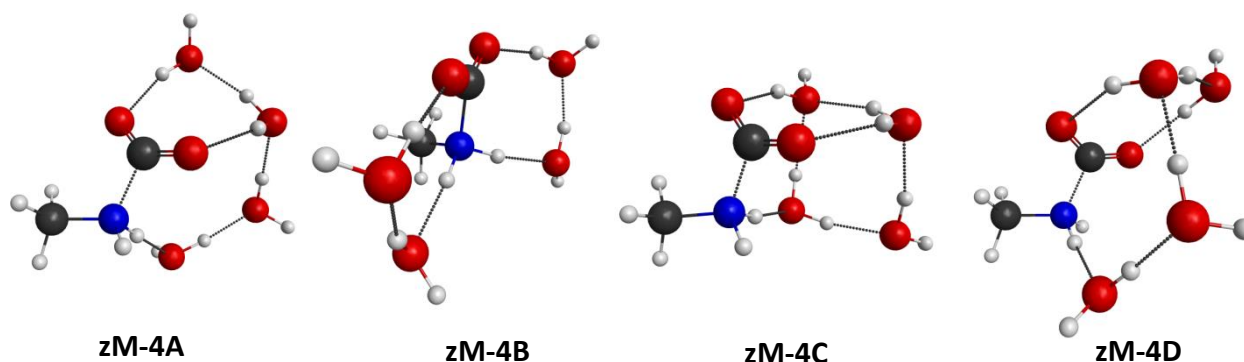
**Table 10.** Structural data for the 4 lowest energy  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_3$  clusters.

ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-3A	1.479	1.688	1.231	1.211	1.021	1.044	106	105	107	141	116	156	3.93
zM-3B	1.481	1.679	1.230	1.214	1.045	1.021	105	106	108	140	115	143	4.10
zM-3C	1.481	1.645	1.213	1.241	1.037	1.037	107	107	105	137	115	179	3.82
zM-3D	1.486	1.635	1.230	1.227	1.041	1.033	107	108	109	137	113	120	4.19

**$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_4$ :** Figure 15 shows the lowest energy optimized structures obtained for the mca zwitterion with four waters, where Table 11 gives their relative energies and Table 12 lists some of the structural data. With four waters, there is no longer an energy correlation with N-C(3) distance. It should be noted that within the lowest 4 structures obtained by Kayi [69], there was a methyl-water interaction. The hydrophobic interaction was not observed in the structures listed, however it did appear within higher energy structures starting at 4 waters. This is an indication that the BH algorithm has found a new lower energy structure and demonstrates that chemical intuition may not always generate a GM on the PES. As previously observed with 2 and 3 waters, the water interaction with the N-H bond causes an increase in the N-H bond length.

**Table 11.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_4$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-4A	0	0	7.6
zM-4B	0.4	0.2	0
zM-4C	1.3	2.6	8.5
zM-4D	1.6	1.4	8.8



**Figure 15.** The four lowest energy structures for the zwitterionic form of methylcarbamic acid with four water molecules.

**Table 12.** Structural data for the 4 lowest energy  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_4$  clusters.

ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-4A	1.483	1.628	1.229	1.227	1.022	1.052	106	105	107	138	116	151	5.49
zM-4B	1.489	1.604	1.233	1.230	1.040	1.041	110	108	108	135	112	118	3.80
zM-4C	1.483	1.624	1.233	1.225	1.022	1.058	107	103	108	138	116	140	3.49
zM-4D	1.484	1.629	1.224	1.232	1.051	1.022	105	106	108	138	115	142	5.67

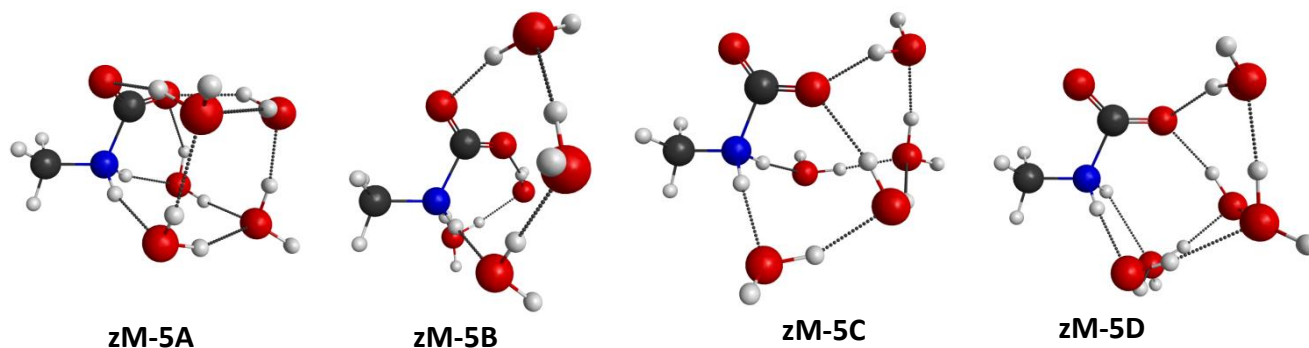
**$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_5$ :** Figure 16 shows the lowest energy optimized structures obtained for the mca zwitterion with five waters, Table 13 gives their relative energies and Table 14 lists some of the structural data. As observed previously, the side of the cluster that has the majority of hydrogen bonding water molecules increases the N-H bond length (Table 14). However, the waters form a cage structure that interacts with both oxygen and nitrogen atoms in the lowest two energy structures, which appears to reduce the N-H variation in bond lengths from that shown with 4 waters. This cage organization should allow for an intermolecular hydrogen transfer resulting in the lower energy neutral mca structure. However, neutral mca was not obtained.

In the work by Kayi *et al* [69], neutral mca did not demonstrate a uniform distribution of water molecules around the solute. The waters clustered around the carboxyl group and no bridging was observed with the N-H hydrogen in the neutral mca, where neutral gly allows a

water bridge. However, the zwitterionic mca shows a bridging by the water molecules stabilizing this form. The BH program developed tries to generate a uniform distribution of water molecules around the solute. Therefore, solutes with preferential asymmetrical distributions of water clusters may not be adequately modeling. Increasing the number of steps in the parameters or changing the level of theory used may produce the neutral form, and it may not due to the nature of the program and the solute-solvent system.

**Table 13.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_5$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-5A	0	1.9	6.4
zM-5B	1.4	0	0
zM-5C	2.7	2.1	3.7
zM-5D	4.1	4.7	9.4



**Figure 16.** The four lowest energy structures for the zwitterionic form of methylcarbamic acid with five water molecules.

In Figure 16, zM-5A shows 3 waters hydrogen bonding to the  $-\text{CO}_2^-$  group and 2 waters hydrogen bonding to  $-\text{NH}_2^+$  group. This is the first instance where the  $-\text{CO}_2^-$  group has more than 2 hydrogen bonds.



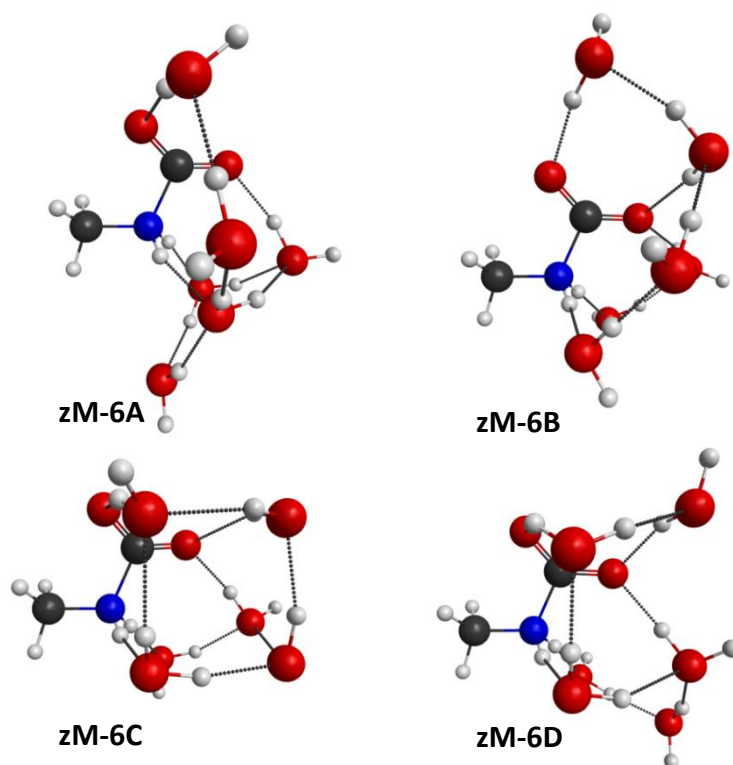
**Table 14.** Structural data for the 4 lowest energy  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_5$  clusters.

ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-5A	1.484	1.585	1.236	1.236	1.049	1.041	104	108	106	135	115	128	2.58
zM-5B	1.486	1.584	1.233	1.234	1.047	1.039	104	110	108	135	115	153	3.24
zM-5C	1.483	1.590	1.218	1.251	1.041	1.043	110	104	106	135	115	163	3.64
zM-5D	1.483	1.597	1.249	1.217	1.054	1.026	106	109	103	136	115	159	3.54

$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_6$ : Figure 17 shows the lowest energy optimized structures obtained for the mca zwitterion with six waters, Table 15 gives their relative energies and Table 16 lists some of the structural data. The clusters show trends similar to those found for 5 waters. In the zM-6A case, the N-C(3) bond distance is longer than the other low energy structures and does not show 3 waters hydrogen bonding to the  $-\text{CO}_2^-$  group (Table 16).

**Table 15.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_6$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-6A	0	2.1	4.8
zM-6B	0.5	0	0
zM-6C	3.6	5.0	6.5
zM-6D	3.7	5.5	5.7



**Figure 17.** The four lowest energy structures for the zwitterionic form of methylcarbamic acid with six water molecules.

**Table 16.** Structural data for the 4 lowest energy  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_6$  clusters.

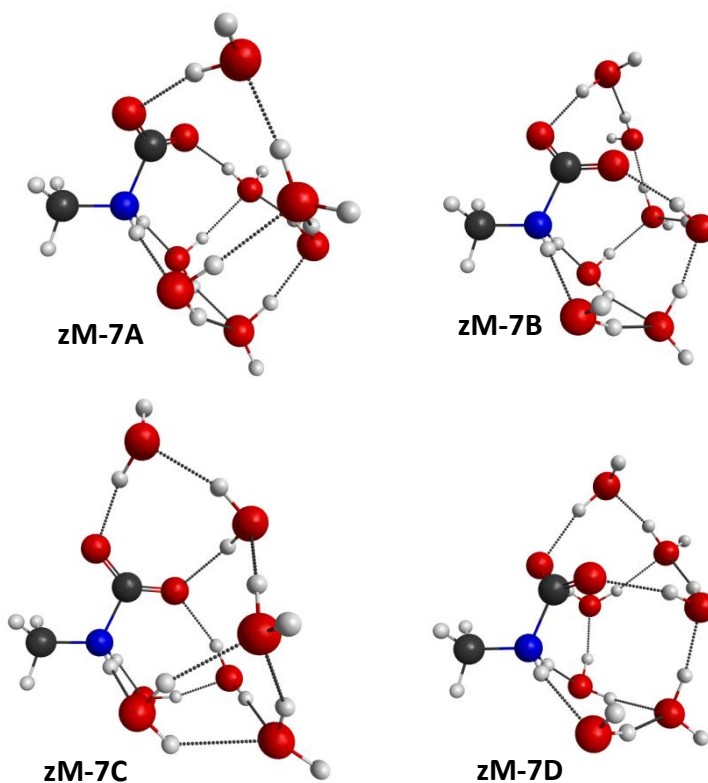
ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-6A	1.484	1.581	1.233	1.237	1.043	1.044	104	111	103	135	116	158	3.58
zM-6B	1.485	1.561	1.230	1.246	1.052	1.042	105	109	107	134	116	168	4.56
zM-6C	1.486	1.563	1.233	1.244	1.062	1.038	102	111	107	133	115	149	4.04
zM-6D	1.484	1.563	1.256	1.221	1.055	1.044	103	109	107	134	116	165	2.82

**$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$ :** Figure 18 shows the lowest energy optimized structures obtained for the mca zwitterion with seven waters, where Table 17 gives their relative energies and Table 18 lists some of the structural data. In the results shown in Table 17, it was found that the single point energy calculation at the B3LYP/6-311++G(*d,p*) level yielded the same energy after ZPE correction as the B3LYP/6-31+G(*d*) level of theory. The lowest energy structure obtained with single point energy calculation at the B3LYP/6-311++G(*d,p*) and MP2/6-

311++G(*d,p*) levels was the same as that obtained using the single point energy calculation at the B3LYP/6-31+G(*d*) level of theory. There continues to be a water cage structure around the hydrophilic  $\text{-NH}_2^+\text{CO}_2^-$  among the low energy structures. When there is an asymmetrical distribution of water molecules on one side of mca an increase in the corresponding N-H bond length is observed. Among the lowest energy structures listed, the waters have a preferential arrangement to form a bridging system.

**Table 17.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-7A	0	0	0
zM-7B	3.3	3.3	4.6
zM-7C	5.2	5.2	3.0
zM-7D	5.3	5.2	4.1



**Figure 18.** The four lowest energy structures for the zwitterionic form of methylcarbamic acid with seven water molecules.

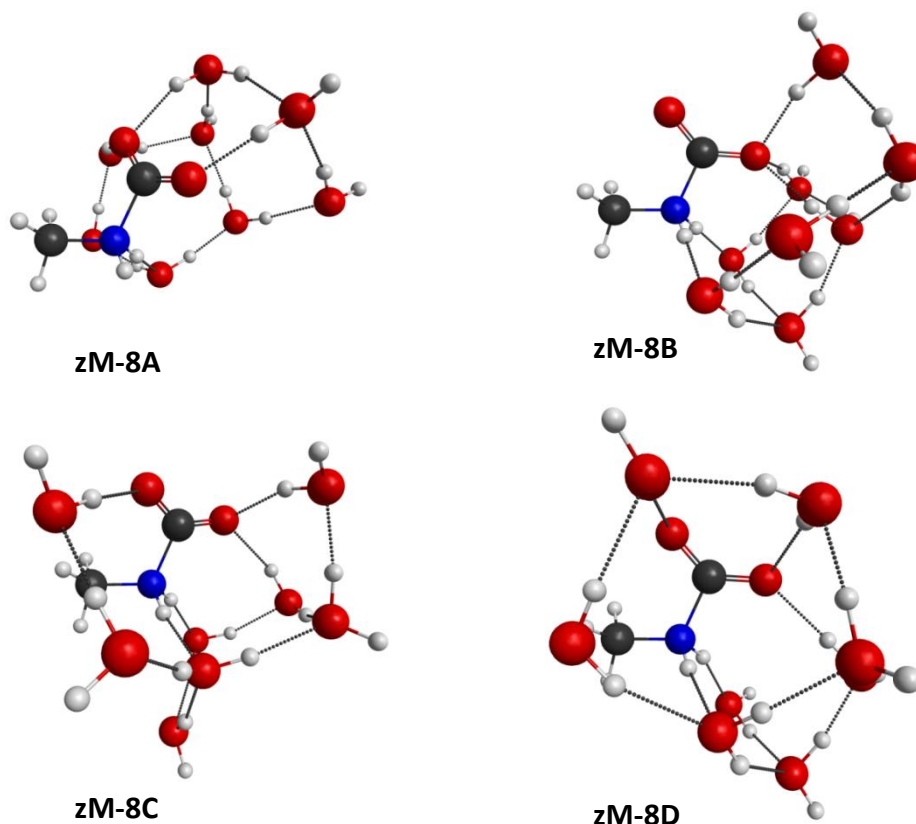
**Table 18.** Structural data for the 4 lowest energy  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$  clusters.

ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-7A	1.487	1.565	1.237	1.237	1.052	1.044	106	110	105	134	114	127	2.00
zM-7B	1.484	1.568	1.236	1.238	1.042	1.052	108	106	104	134	116	157	3.46
zM-7C	1.485	1.562	1.232	1.244	1.051	1.044	105	110	105	134	116	168	2.31
zM-7D	1.487	1.549	1.245	1.236	1.041	1.063	108	106	105	133	115	131	2.18

**$\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$ :** Figure 19 shows the lowest energy optimized structures obtained for the mca zwitterion with eight waters, Table 19 gives their relative energies and Table 20 lists some of the structural data. The lowest energy structure, zM-8A, shows the largest N-H bond length at 1.096 Å with the closest water oxygen atom at 1.546 Å from the amine hydrogen, and is therefore still in the zwitterionic form. This structure demonstrates an asymmetric distribution where 7 water molecules are on one side of the CNC plane in mca, and the 8th water molecule crossing the CNC plane interacts with the oxygen atom on mca. The water cage formed in zM-8A is also preferentially ordered around the  $-\text{CO}_2^-$  group. This ordering causes the relatively large jump in energy from zM-8A to zM-8B than observed in the previous structures with less than 8 water molecules. This demonstrates that a symmetric water cluster is ideal to stabilize the zwitterionic form of mca. If a cage is formed on one side of mca, it appears that conversion to the neutral form may be probable and would generate a lower energy structure. In the results shown in Table 19, it was found that the single point energy calculation for B3LYP/6-311++G(*d,p*) yielded essentially the same energy after ZPE correction as the B3LYP/6-31+G(*d*) level of theory. These results and those shown in Table 17 further validate that, with more water molecules, the smaller B3LYP basis set is adequate for B3LYP calculations.

**Table 19.** Energy comparisons for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$  locally optimized using B3LYP/6-31+G\* level of theory with the corresponding geometries used to perform single-point energy calculations using B3LYP/6-311++G\*\* and MP2/6-311++G\*\* levels of theory.

ID	B3LYP/6-31+G* (kJ/mol)	B3LYP/6-311++G** (kJ/mol)	MP2/6-311++G** (kJ/mol)
zM-8A	0	0	2.9
zM-8B	8.1	8.2	0
zM-8C	10.3	10.3	2.0
zM-8D	13.2	13.2	5.4

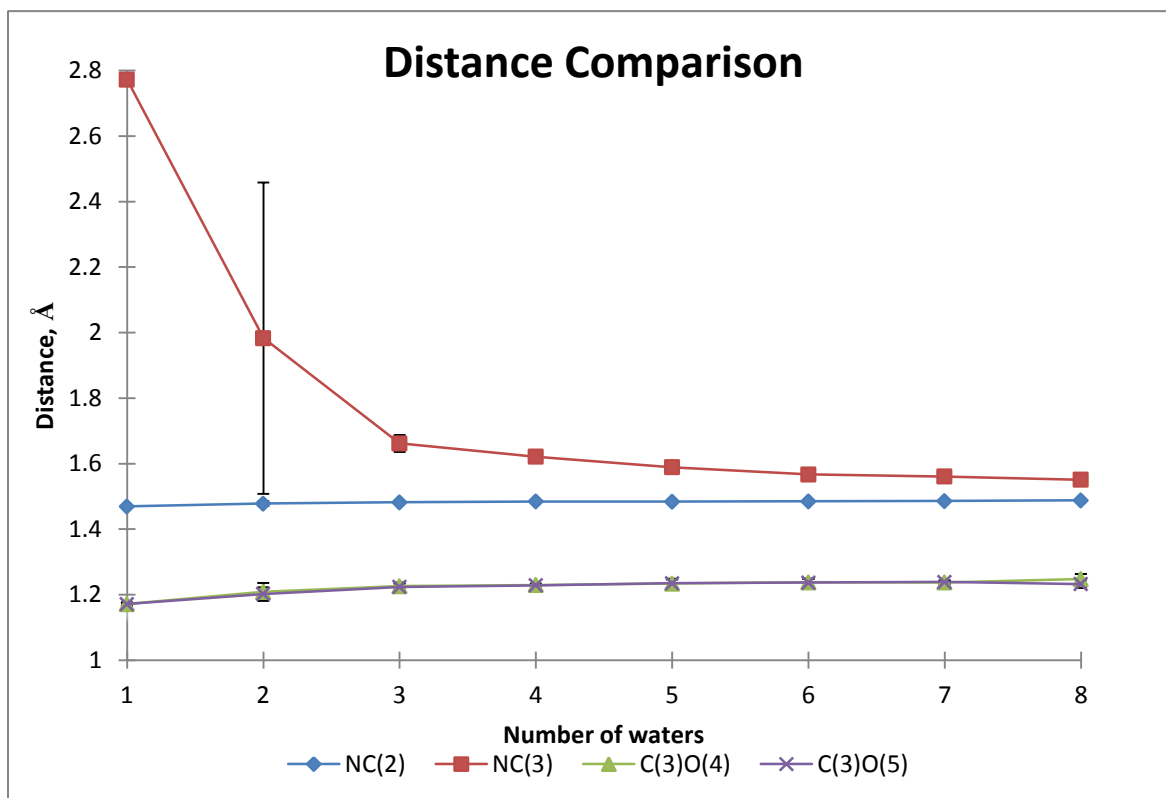


**Figure 19.** The four lowest energy structures for the zwitterionic form of methylcarbamic acid with eight water molecules.

**Table 20.** Structural data for the 4 lowest energy  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$  clusters.

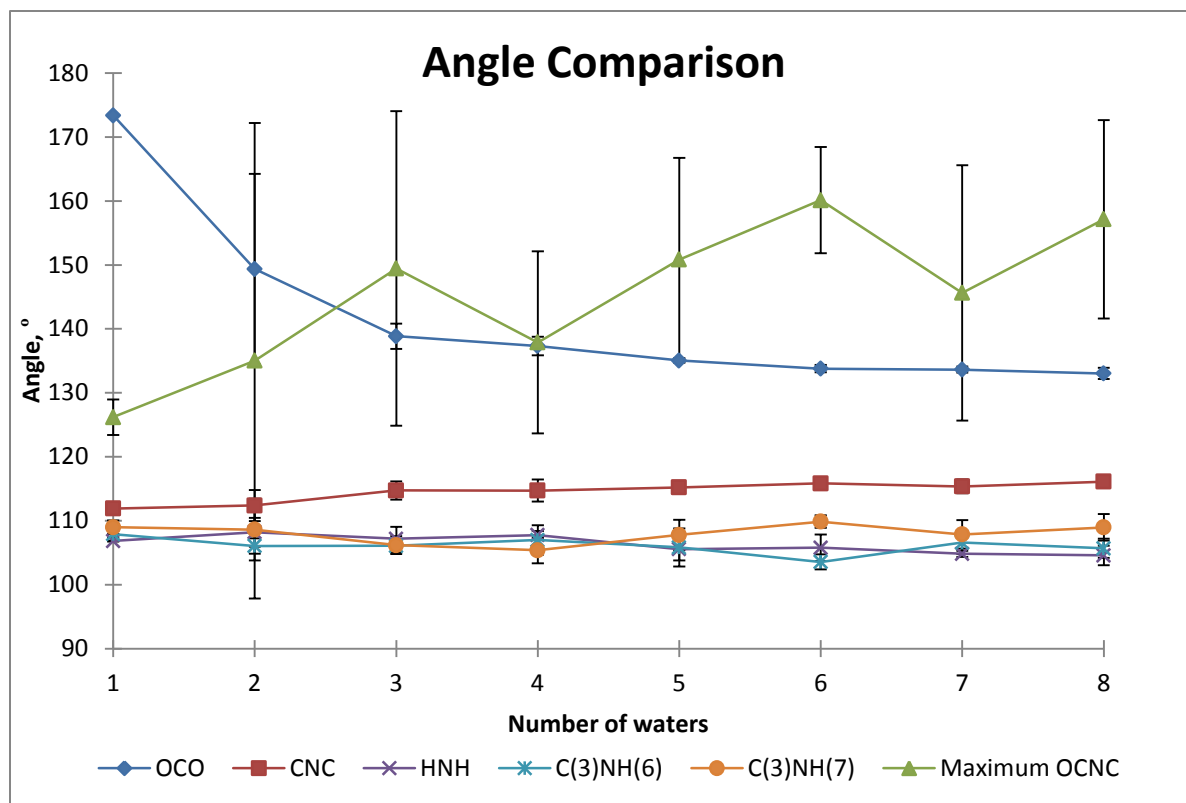
ID	Distance, Å						Angle, °						Dipole Moment, D
	NC <sup>(2)</sup>	NC <sup>(3)</sup>	C <sup>(3)</sup> O <sup>(4)</sup>	C <sup>(3)</sup> O <sup>(5)</sup>	NH <sup>(6)</sup>	NH <sup>(7)</sup>	C <sup>(3)</sup> NH <sup>(6)</sup>	C <sup>(3)</sup> NH <sup>(7)</sup>	HNH	OCO	CNC	MAX OCNC	
zM-8A	1.489	1.548	1.239	1.240	1.023	1.096	106	107	107	134	116	142	2.06
zM-8B	1.485	1.543	1.271	1.215	1.062	1.048	108	108	104	132	115	169	3.82
zM-8C	1.487	1.552	1.236	1.243	1.047	1.054	105	112	103	132	116	145	6.47
zM-8D	1.490	1.561	1.244	1.232	1.046	1.040	105	109	105	134	117	172	3.06

**Discussion of property trends for the mca zwitterion water clusters:** Key properties for  $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_{1-8}$  were taken for comparison and are shown in Figure 20-22. Figure 20 shows the average bond lengths with standard deviation error bars for the four lowest energy structures.



**Figure 20.** Distance comparisons for the average values of the 4 lowest energy structures with the error bars representing the standard deviation.

Figure 20 shows that increasing the number of water molecules in the mca zwitterion water cluster causes a decrease in the N-C(3) distance. The large standard deviation error bar shown with 2 waters reflects the van der Waals complex and mca zwitterion averaging. Figure 20 clearly demonstrates that 3 waters are necessary to stabilize the zwitterionic form of mca. There are also increases in the corresponding N-C(2) bond length and the C-O bond lengths with the incremental addition of water. These findings are expected with the decrease in N-C(3) distance and correlate with the results published by Kayi *et al.* [67].

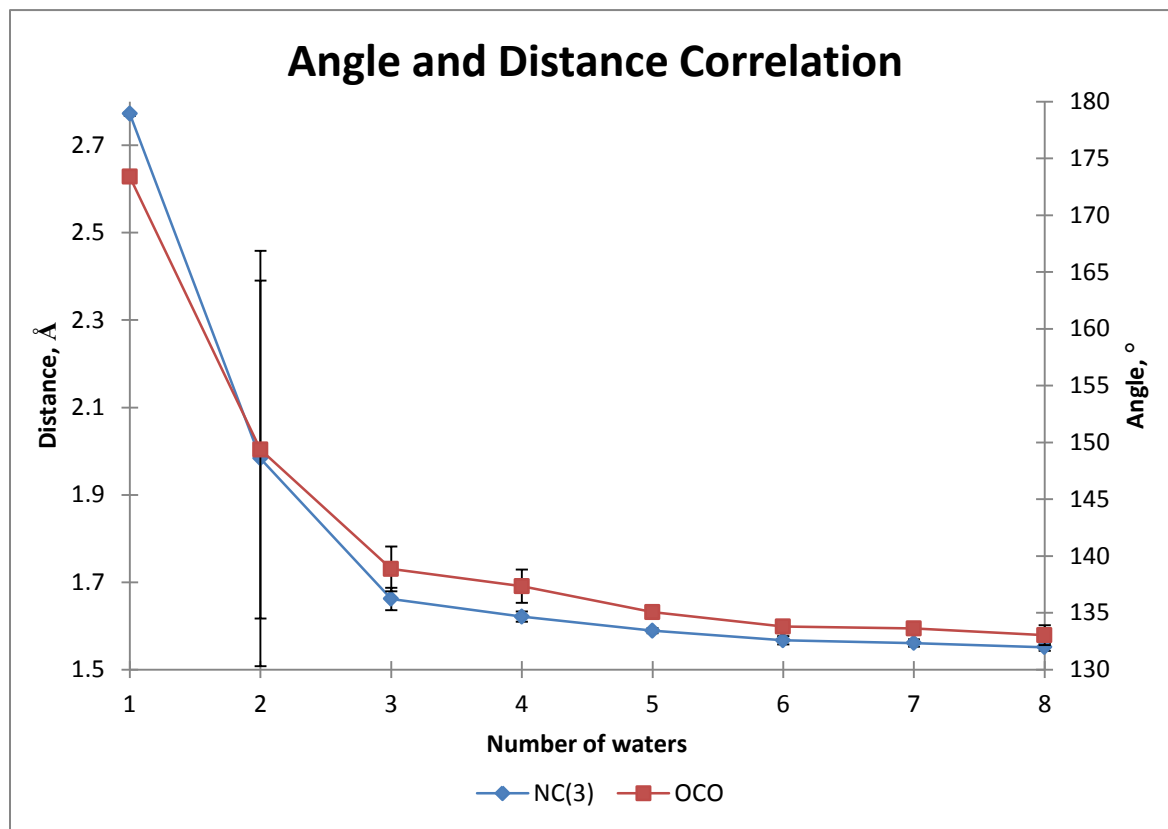


**Figure 21.** Angle comparisons for the average values of the 4 lowest energy structures with the error bars representing the standard deviation.

However, Kayi *et al.* found a slight bump in the values obtained with 2 and 6 waters which may be a consequence of the calculations not having the GM structure available.

Upon increasing the number of water molecules, Figure 21 shows a small increase in the C-N-C angle with an increase in water molecules, a decrease in the O-C-O angle, and a trend towards an O-C-N-C dihedral angle of 180°. The O-C-O angle is consistent with the results published by Kayi *et al.* [67]. However, again there is a slight bump in the values obtained by Kayi with 2 and 6 waters which is not present in this study.

Figure 22 compares the N-C(3) distance and O-C-O angles directly. There is a direct comparison and consistent trend in the average distance and average angle, where the results obtained by Kayi *et al.* for 2 and 6 waters indicates that there were lower energy structures than previously found, suggesting that the present calculations have found new GM structures.



**Figure 22.** Direct comparison of N-C(3) distance and OCO angle.

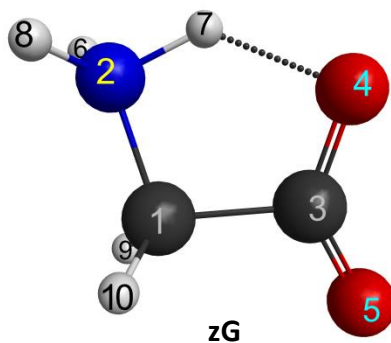
It should be noted that with 2 and 3 waters, a decrease in the N-C(3) distance causes an increase in the structure energy. With 4 or more waters, this trend is no longer observed with this N-C(3) distance and energy. The error bars shown in Figure 22, with two waters, is an indication of the large variations in energy that is directly correlated to the N-C(3) distances observed, where the lowest energy structure did demonstrated a van der Waals complex with CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub>. The lack of neutral mca obtained in the results of the BH method demonstrates that perhaps the zwitterionic form of mca would preferentially decompose to CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub> rather than allow proton transfer to form the neutral species.



### 3.4 GLYCINE

Alagona *et al.* [76] found, using Monte Carlo calculations, the  $\text{--CO}_2^-$  group on acetate had  $\sim 6$  tightly bound waters with the anion while the  $\text{--NH}_3^+$  group had an average coordination number of 3.5. Intuition would lead one to expect that  $\sim 9.5$  waters are necessary to stabilize the zwitterionic form of glycine. However, water bridging will lower this number. Using the PBE1PBE/6-311+G(d,p) level of theory, Bachrach [56] reported that seven waters are needed to reach the isoelectric point of glycine. Aikens and Gordon [36], found that the neutral isomer of glycine is lowest in energy for complexes with up to six water molecules without continuum solvent, and the zwitterionic form of glycine is the global minimum for seven or eight water molecules at the MP2//RHF level of theory. Due to these results, glycine was studied with 7 and 8 waters to determine where the stability of the zwitterion is obtained. A Monte Carlo (MC) simulation with local optimization basin-hopping approach was used using EFP2 water and glycine structures, where computations were determined with geometry optimization using RHF/6-31G(d,p) and MP2/6-31++G(d,p) single-point energies were used to refine the relative energies. Aikens and Gordon noted that several conformers should be considered in sampling due to the large number of low-energy structures.

Glycine was optimized in the zwitterionic form holding the geometry of the  $\text{--NH}_3^+$  group fixed, and then submitted to global optimization. If the conversion to the neutral species occurred via an intramolecular and intermolecular process, this would allow one to conclude that using one starting structure during GO was effective. The NCCO dihedral angle may be monitored to determine the utility in using one starting structure. Therefore, only one structure was used as the starting structure for glycine, which is the zwitterionic form. The N-H bonds on  $\text{NH}_3$  were held fixed during the EFP1 calculations at 1.05 Å and allowed to relax during full unfragmented calculations.



**Figure 23.** Initial zwitterionic glycine starting structure submitted for global optimization. Atoms 2 is N, 1 and 3 are C, and 4 and 5 are O.

**BH Parameters:** The average displacement of the water molecules is set to 1.5 Å, the temperature is 1000 K, and a boundary condition of 4 Å. Several test runs were initially performed with the various trial iterations greater than or equal to 400 to determine the ideal parameters. Using the structure shown in Figure 23, 4 trials with the initial number of steps,  $N$ , set at 200 and 250 iterations for 7 and 8 waters, respectively, and are ran with fragmented EFP1 waters. To model the work of Aikens and Gordon [36] and Kayi *et al.* [69], the EFP1 locally optimized structures were obtained using B3LYP/6-311++G( $d,p$ ) and a gradient convergence tolerance of 0.003. The full unfragmented geometry optimization of the low energy clusters was performed using the same level of theory with a gradient convergence tolerance of 0.0001. Frequency calculations are performed on the lowest energy structures to determine if the structure corresponds to a proper minimum and calculate the ZPE correction.

**$\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_7$ :** Figure 24 shows the lowest energy optimized structures for the gly zwitterion with seven waters, Table 21 gives their relative energies and Table 22 lists some of the structural data. Kayi *et al.* [69] showed that the methylcarbamic acid zwitterion was lower in energy than the glycine zwitterion when there are 1 to 4 waters present, but when there are 5 or more waters the mca zwitterion and gly zwitterion demonstrate overlapping energies for the structures listed. The present calculations find for 7 waters that zwitterionic mca is lower in energy than the zwitterionic form of gly. This could be a result of Kayi *et al.*

using MP2 opposed to B3LYP, where the energies for the two methods have previously been found to reorder [69]. The 10.1 kJ/mol difference between the lowest energy structure of mca and gly is relatively small. To test potential reordering, a single point energy calculation using MP2/6-311++G(*d,p*) was performed on zG-7A, where the zG-7A was found to be lower in energy than zM-7A by 8.8 kJ/mol. However, performing a refined local optimization will give better results.

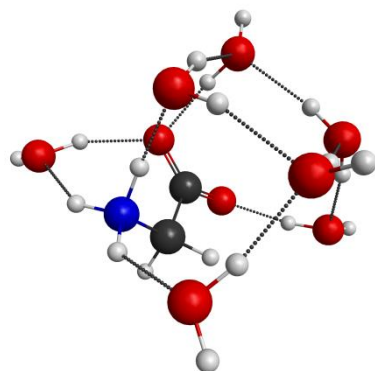
<b>Table 21.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_7$ and $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_7$ using B3LYP/6-311++G**.	
ID	B3LYP/6-311++G** (kJ/mol)
zM-7A	0
zM-7B	3.3
zM-7C	5.2
zM-7D	5.2
zG-7A	10.1
zG-7B	12.4
zG-7C	16.9
zG-7D	17.1

Using the BH method the neutral form of glycine was obtained with 7 waters within 8 kJ/mol and 1 kJ/mol higher in energy than zG-7A and zG-7D, respectively, with intramolecular hydrogen transfer displayed. Intermolecular hydrogen transfer resulting in the neutral form of gly was also obtained at higher energy structures. The resulting energies and coordinates can be found in the Appendix. The lowest energy structures obtained in Figure 24 shows that each hydrogen in the  $-\text{NH}_3^+$  form hydrogen bonds with a water molecule, causing a slight increase in the N-H distance. The zwitterion shows stability with a water bridge between the amine and carboxylic groups with at least three and two hydrogen bonds with water on the carboxylic and amine groups, respectively.

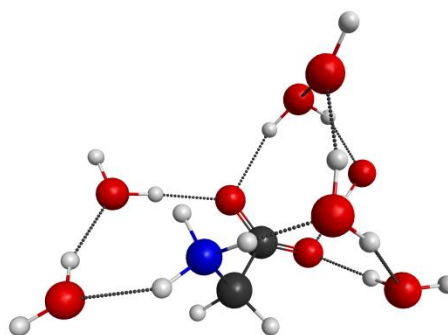
**Table 22.** Structural data for the 4 lowest energy  $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_7$  clusters.

ID	Distance, Å							Angle, °			Dipole Moment, D
	C-N	C-C	C-O	C-O	N-H	N-H	N-H	OCO	NCC	Max NCCO	
zG-7A	1.497	1.546	2.402	2.361	1.028	1.032	1.054	128	113	168	3.251
zG-7B	1.498	1.546	2.377	2.390	1.025	1.026	1.069	127	109	132	3.996
zG-7C	1.498	1.547	2.388	2.383	1.019*	1.046	1.075	127	111	150	2.417
zG-7D	1.502	1.549	2.376	2.388	1.040	1.038	1.033	128	107	99	3.116

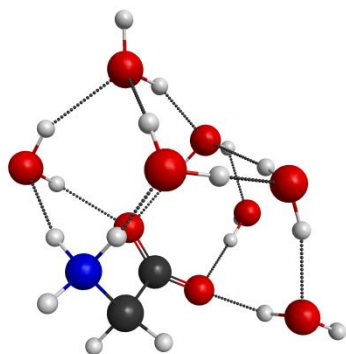
\*Indicates the hydrogens not bonding to water.



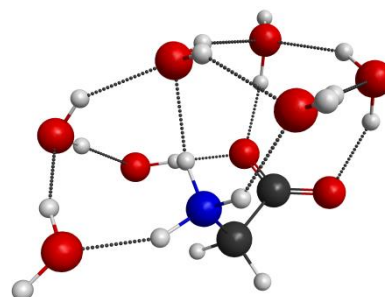
**zG-7A**



**zG-7B**



**zG-7C**



**zG-7D**

**Figure 24.** The four lowest energy structures for the zwitterionic form of glycine with seven water molecules.

$\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_8$ : Figure 25 shows the optimized structures for the gly zwitterion with eight waters, Table 23 gives their relative energies and Table 24 lists some of the structural data. It was found that the lowest energy structures for the zwitterionic forms show that glycine is lower in energy than mca, with the asymmetric solvation of mca mixed within the two lowest energy structures of glycine. This is a clear shift from the results obtained using 7 water molecules. If mca is a precursor to gly, this demonstrates that eight waters would be needed to accurately model this pathway.

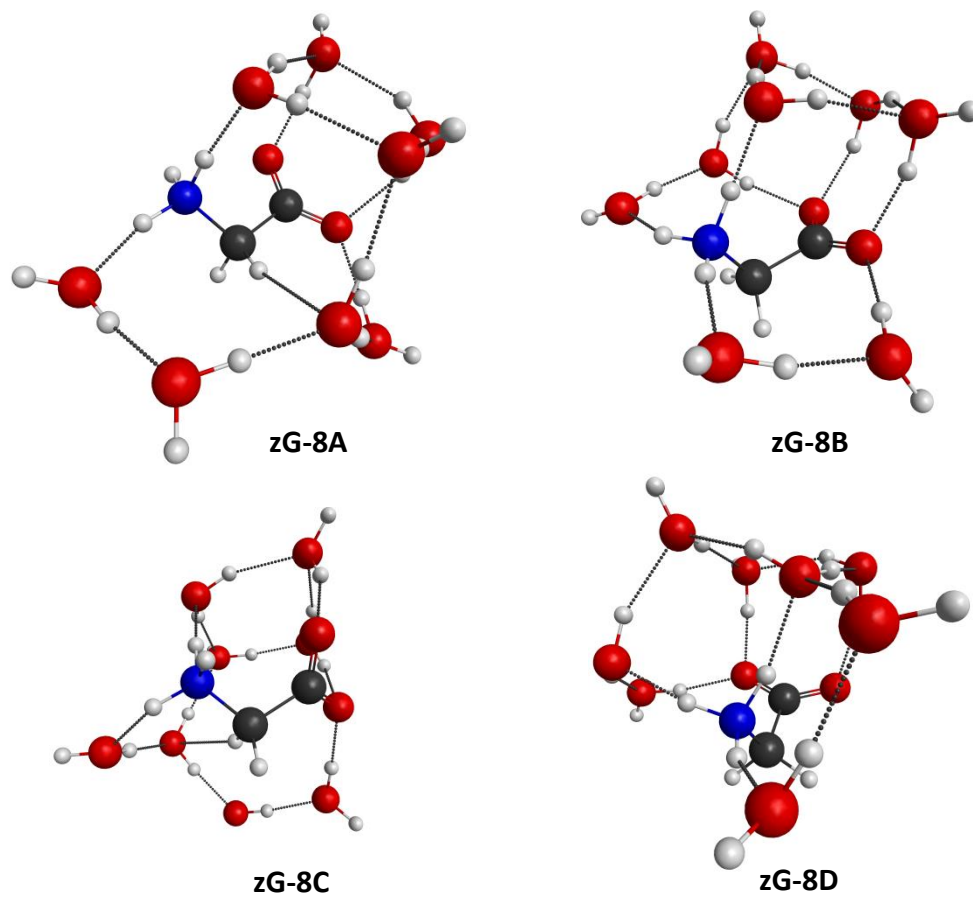
<b>Table 23.</b> Energy comparisons for $\text{CH}_3\text{NH}_2^+\text{CO}_2^-(\text{H}_2\text{O})_8$ and $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_8$ using B3LYP/6-311++G**.	
ID	B3LYP/6-311++G** (kJ/mol)
zG-8A	0
zG-8B	1.5
zG-8C	6.0
zM-8A	7.5
zG-8D	9.4
zM-8B	15.7
zM-8C	17.8
zM-8D	20.8

The lowest energy structures obtained in Figure 25 shows that the  $-\text{NH}_3^+$  hydrogens with more water hydrogen bonding increases the N-H distance. The zwitterion demonstrates stability with a water bridge between the amine and carboxylic groups with at least three and two hydrogen bonds with water on the carboxylic and amine groups, respectively.

**Table 24.** Structural data for the 4 lowest energy  $\text{NH}_3^+\text{CH}_2\text{CO}_2^-(\text{H}_2\text{O})_8$  clusters.

ID	Distance, Å							Angle, °			Dipole Moment, D
	C-N	C-C	C-O	C-O	N-H	N-H	N-H	OCO	NCC	Max NCCO	
zG-8A	1.495	1.544	1.254	1.255	1.053	1.043	1.021*	128	109	154	2.701
zG-8B	1.510	1.544	1.268	1.247	1.052	1.039	1.035	127	110	103	4.196
zG-8C	1.497	1.542	1.252	1.256	1.057	1.019*	1.040	128	110	155	3.065
zG-8D	1.501	1.547	1.275	1.237	1.052	1.041	1.036	127	108	102	2.619

\*Indicates the hydrogens not bonding to water.



**Figure 25.** The four lowest energy structures for the zwitterionic form of glycine with eight water molecules.

### 3.5 CONCLUSION

Many optimized geometries could be obtained at reduced computational cost using the EFP1 water, but there were also some optimized structures that showed an increase in computational cost. The BH method leads to a computationally efficient algorithm, but some systematic improvement of the EFP1 is needed. The EFP1 method consists of the Coulombic interactions, induction/polarization interactions, and charge-transfer/exchange-repulsion terms between the solvent and solute-solvent molecules. The first two terms are determined from *ab initio* calculations with the water monomer, whereas the third is fit to a quantum mechanical dimer potential. The BH method may demonstrate better results by developing and using the EFP2 method.

This BH method generates a pseudo-uniform distribution of water molecules that stabilizes the zwitterionic form of mca. However, an asymmetrical distribution of the solvent will lead towards the possibility to form the lower energy neutral structure despite the presence of this species. It was expected that the neutral form would predominate in the low energy structures in accordance with the results previously obtained by Kayi *et al* [69]. However the lowest energy structures for mca were zwitterions for 3-8 water clusters. Not finding any neutral mca structures from the BH method suggests that perhaps the zwitterionic form of mca would preferentially decompose to generate a van der Waals complex with CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub> rather than allow proton transfer to form the neutral species. Decomposition into CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub> is not observed in 3-8 water clusters, due to the mca zwitterion stability obtained with the water molecules.

The cage formed around the mca zwitterion is more uniform than the bridging waters around neutral mca shown by Kayi *et al* [69]. A pseudo-uniform distribution of water molecules did not convert the zwitterionic form of mca to the neutral species. This observation indicates that a fully solvated zwitterion is stabilized by water molecules, where the gas form will decompose into CO<sub>2</sub> and CH<sub>3</sub>NH<sub>2</sub>.

The results obtained with glycine indicate that the zwitterion is stabilized over the neutral form with 7 and 8 waters, as expected. The presence of higher-energy neutral forms of

glycine at 7 waters demonstrates confidence that both neutral and zwitterion species will be obtained using the zwitterion form as a starting structure. However, the lack of the neutral species being obtained with the methylcarbamic acid zwitterion does not necessarily indicate failure in the BH algorithm. The neutral form of gly will allow a water cage to form between the carboxyl and amine groups, whereas in contrast the neutral mca only allows water-bridging to occur at the carboxyl group.

This BH algorithm has demonstrated utility in finding GMs and low energy structures without bias. A larger number of steps during the GO will increase the confidence that one has found the GM. However, resources often restrict the amount of computational time used. The demonstration of energy variations within chemical accuracy of  $\sim 4$  kJ/mol and the generation of several similar cluster arrangements indicate that identification of a GM is less significant and statistical averaging should be considered in future analysis. Future work should be performed modeling neutral methylcarbamic acid and glycine with discrete water molecules for a more thorough comparison of the work by Kayi *et al* [69]. From these results and the production of alkylammonium alkylcarbamates formed from  $\text{CO}_2$  and amines [81], it is of interest to perform a stability comparison with an incremental number of waters using the following systems:  $[\text{CH}_3\text{NH}_3^+][\text{O}_2\text{CNHCH}_3](\text{H}_2\text{O})_n$  and  $[\text{CH}_3\text{NH}_3^+][\text{O}_2\text{CCH}_2\text{NH}_2](\text{H}_2\text{O})_n$ .

When modeling the formation of gly, it is clear that solvation is a key factor in the relative stability of the chemical species. The number of water molecules needed to stabilize the zwitterionic form of gly over the neutral species of gly leads one to conclude that there will also be a conversion between the stability of the mca zwitterion and the gly zwitterion. Therefore, interstellar ices may stabilize zwitterionic gly despite the lower energy obtained for neutral mca [69].



## APPENDIX

### A. SOURCE CODE FOR THE BH ALGORITHM

An example of time required to run this program using mca with 5 waters is as follows:

- EFP section (1-6) bit took ~150 hours spread over 4 procs ~37 hours real time
- Minimum time: 1 *ab initio* calculation takes ~19 hours. Spread over 4 and 8 CPUs it takes ~4.8 and ~2.4 hours real time, respectively.
- Maximum time: 1 *ab initio* calculation takes ~44 hours. Spread over 4 and 8 CPUs it takes ~11 and ~5.5 hours real time, respectively.

The following is the code used for this research, where each block of code was split due to the functions performed. Comments in the code are preceded by a # or preceded and ending with """. Python 2.7, GAMESS, and PBS/Torque must be installed on the computer.

This program is designed to run in the terminal window. Prior to running this program, an input file for the solute must be created and placed in a new working folder as well as the input data for a full *ab initio* calculation, in .txt format, with all the input information preceding the first line of coordinates. The full *ab initio* file should be named “abinitioMCBH.txt”. These files must be created using Cartesian coordinates. The first input file must use a B3LYP DFT energy method.

It is best to create this folder with the naming convention “bh\_7W\_1”, where 7 is the number of waters and 1 would be the first trial. Placing this folder in a directory corresponding to the solute is advisable. The user should be in this working folder when starting the program. All of the source code below must be placed in a separate folder for operation. The program is initiated with the following command:

```
“python /[program location]/bh37.py”
```

After entering this command in the terminal window, the user should follow the prompts at the screen.

---

```

FILENAME: bh37.py
# --- This corresponds to Step 1 in the flow chart.

#!/usr/bin/env python

# BH Input Script
# 2/27/2011
# Morgyn Stryker
# Run EFP water around a central chosen molecule through GAMESS.
# Get output file, and process.

import os,sys,time,getpass, random
from mcpointsB import *
# mcpointsB.py is listed below

def pbs_job_file(input_data):
    ### Get job file to submit to PBS queue
    block1 = "#!/bin/bash\n#\n"

    jobstuff = "#PBS -N " + input_data + \
        "\n#PBS -q bigjob" + \
        "\n#PBS -M morgyn.stryker@gmail.com" + \
        "\n#PBS -m ae" + \
        "\n#PBS -l procs=4" + \
        "\n#PBS -l walltime=120:00:00" + \
        "\n#PBS -e " + workdir + "pbs.err" + \
        "\n#PBS -o " + workdir + "pbs.out\n\n" + \
        "# set job name\nJOB="+input_data+"\n"

    work_dir = "# set up working dir\ndeclare -x GAMWORKDIR="+workdir+"\n\n"

    info_bit = "cd $PBS_O_WORKDIR\n" + \
        "echo #####Begin Job Information#####\n" + \
        "echo Working directory is $PBS_O_WORKDIR\n" + \
        "echo Running on host `hostname`\n" + \
        "echo Time is `date`\n" + \
        "echo Directory is `pwd`\n" + \
        "echo This jobs runs on the following processors:\n" + \
        "echo `cat $PBS_NODEFILE`\n" + \
        "NPROCS=`wc -l < $PBS_NODEFILE`\n" + \
        "echo This job has allocated $NPROCS cpus\n" + \
        "echo #####End Job Information#####\n" + \
        "# GAMESS executable parameter for monte carlo\n" + \
        "VERNO=00 NCPUS=$NPROCS\n" + \
        "cd $GAMWORKDIR\n"

    prospath = "python /home/mstryker/programs/monte2/bh37_scatter.py\n"

    pbs_write = open(workdir+"pbs_jobfile.qjob","w")
    pbs_write.write(block1)
    pbs_write.write(jobstuff)
    pbs_write.write(work_dir)
    pbs_write.write(info_bit)
    pbs_write.write(prospath)
    pbs_write.close()
    return

```

```

def submitJob(jobFile):
    ### Submit the new job
    strtime = getTime()

    # open the job_record.dat data file with two lines(current job# and date of last)
    job_rec = open(submit_tool_dir + "gamjobno.dat", "r")
    jobno = int(job_rec.readline()) + 1
    job_rec.close()
    job_rec = open(submit_tool_dir + "gamjobno.dat", "w")
    job_rec.write(" " + str(jobno) + "\n" + strtime + "\n")
    job_rec.close()

    # Append job_record file
    job_info = open(submit_tool_dir + "job_record.txt", "a")
    job_info.write(" bh_" + str(bhStart) + "-" + str(jobno) + " standard " + strtime + " " + jobFile + "\n")
    job_info.close()

    pbs_job_file(jobFile)

    # Save submit result in the lock file
    ijob = os.system("qsub "+workdir+"pbs_jobfile.qjob")

    if ijob != 0:
        print "ERROR SUBMITTING JOB TO PBS - ijob = ", ijob
        sys.exit()
    return

#####
##### MAIN PART #####
#####
seedValue = long(time.time())*257
user = getpass.getuser()          # get username
curr_wd = os.getcwd()             # get current working directory
workdir = curr_wd+"/"

while(1):
    orig_file = raw_input("What file would you like to use? ")
    if os.path.isfile(orig_file):
        try:
            the_file = open(orig_file, 'r')
        except OSError:
            pass
        the_file.close()
        break
    else:
        print "File does not exist!"
while(1):
    n_water = raw_input("How many water molecules do you want? ")
    try:
        n_water = int(n_water)
        if isinstance(n_water, int):
            break
    except:
        print "Value is not an integer!"
while(1):
    max_d = raw_input("Maximum distance from molecule(angstrom)? ")
    try:
        max_d = float(max_d)

```

```

        if isinstance(max_d,float):
            break
    except:
        print "Value is not a float!"
while(1):
    trials = raw_input("How many iterations do you want? ")
    try:
        trials = int(trials)
        if isinstance(trials,int):
            break
    except:
        print "Value is not an integer!"
while(1):
    Teff = raw_input("What effective Temp do you want? ")
    try:
        Teff = float(Teff)
        if isinstance(Teff,float):
            break
    except:
        print "Value is not a float!"
while(1):
    WalkPt = raw_input("What average displacement do you want? ")
    try:
        WalkPt = float(WalkPt)
        if isinstance(WalkPt,float):
            break
    except:
        print "Value is not a float!"

computer = "thundra"
gam_work = "/home/" + user + "/gam_work/"
gamscr_file = "/home/" + user + "/gamscr/"
submit_tool_dir = gam_work + "submit_tools/"

# Get new basin hopping folder number/data
bh_rec = open(submit_tool_dir+"bhno.dat","r")
bhStart = int(bh_rec.readline()) + 1
bh_rec.close()
bh_rec = open(submit_tool_dir+"bhno.dat","w")
bh_rec.write(" "+str(bhStart)+"\n"+getTime()+"\n")
bh_rec.close()

# Write data to input dataMC file
dataFile = open(workdir+"dataMCinp.txt","w")
dataFile.write(str(computer)+"\n"+str(bhStart)+"\n"+str(gam_work)+"\n"+str(workdir))
FileData = "\nFile:\t"+str(orig_file)+\
"\nWalkPt:\t"+str(WalkPt)+\
"\nWaters:\t"+str(n_water)+\
"\nMax_d:\t"+str(max_d)+\
"\nTrials:\t"+str(trials)+\
"\nTeff:\t"+str(Teff)+\
"\nSeed:\t"+str(seedValue)
dataFile.write(FileData)
dataFile.close()

jobname = "MCBH_"+str(bhStart)
submitJob(jobname)

```

---

```

FILENAME: bh37_scatter.py
# --- This corresponds to Step 2-6 in the flow chart.

#!/usr/bin/env python

# BH Main Section of Code
# 9/7/2012
# Morgyn Stryker
# Run EFP water around a central chosen molecule through GAMESS.
# Get output file, and process.

##### THIS VERSION SETUP TO RUN ON DESKTOP COMPUTER

import random,time
from mcpointsB import *
from math import *
import os,sys,csv,getpass,operator

def getpoint():
    # Generate a random point using uniform distribution of sphere, returns unit vector
    # via Wolfram (http://mathworld.wolfram.com/SpherePointPicking.html)
    i = 0
    while i != 1:
        u = random.uniform(-1,1)
        v = random.uniform(-1,1)
        if u**2 + v**2 < 1:
            i += 1

    x = 2*u*((1 - u**2 - v**2)**0.5)
    y = 2*v*((1 - u**2 - v**2)**0.5)
    z = 1 - 2*(u**2 + v**2)

    return Point( x, y, z )

def metropolis(d_E):
    ### Metropolis Rule ###
    # --- This corresponds to Step 5a in the flow chart.

    boltzC = 1.380650e-23      # J/K
    conversionFactor = 4.359744e-18  # J/hartree

    metrop = exp(-d_E*conversionFactor/(boltzC*Teff))
    x = random.uniform(0,1)

    if metrop <= x:          # accepted
        N_highE = 0
    else:                   # rejected
        N_highE = 1

    return N_highE, metrop

def addRekurs(self):
    # Gets number of iterations needed to scan atom distances (lower diagonal matrix)
    value = 0
    for k in range(0,self):
        value += k

```

```

return value

def SaundersKick(atomStarting,atomTotal,atomCoordinate):
    # Arbitrary step
    # --- This corresponds to Step 6 in the flow chart.

    minDistance = 0.5
    minWaterDistance = 2.0
    m = 0
    kickCoord = atomCoordinate

    while m < addRecurs(atomTotal):
        #Use this if you want to adjust central molecule coordinates with a kick
        #for i in range(0,atomStarting-1):
        #    kick = random.uniform(0,0.5)
        #    dCoordinate = getpoint()*kick
        #    kickCoord[i] = atomCoordinate[i] + dCoordinate

        water = 0
        while water < n_water:
            k = atomStarting-1 + 3*water
            kickCoord[k],kickCoord[k+1],kickCoord[k+2] = randomWalk(atomCoordinate[k])
            test = kickCoord[k]
            waterDistance = orig.distance(test)

            if waterDistance > (dmax + max_d):
                kickCoord[k] = kickCoord[k]*(dmax/waterDistance)
                kickCoord[k+1] = kickCoord[k+1] - kickCoord[k]
                kickCoord[k+2] = kickCoord[k+2] - kickCoord[k]
            water += 1

        for i in range(0,k,1):
            samespace = atomCoordinate[i].distance(kickCoord[k])
            if (samespace < minWaterDistance):
                water -= 1
                break

        for i in range(0,atomTotal-2):
            for k in range(i+1,atomTotal-1):
                dist = kickCoord[i].distance(kickCoord[k])
                if dist < minDistance:
                    m = 0
                    break
            else:
                m += 1

        ##### This keeps track of the time of each GAMESS job submission
        #subFile.close()

    return kickCoord

def waterCoord(maxDisplacement):
    n = random.uniform(0,maxDisplacement)
    A = getpoint()*n
    B = getpoint()          # First vector
    C = getpoint()          # Second vector

    Vn = B.cross_p(C)       # cross product = normal vector

```

```

Vn_minus = C.cross_p(B)
d_norig = Vn.distance(orig) # normal vector distance from A

# convert to unit vector then adjust point distance
O1 = A + B * O_Hdistance * cos(ang_HOH/2)
H2 = A + (Vn.unit_v(d_norig)) * O_Hdistance * sin(ang_HOH/2)
H3 = A + (Vn_minus.unit_v(d_norig)) * O_Hdistance * sin(ang_HOH/2)

return O1,H2,H3

def randomWalk(walkPoint):
    O1,H2,H3 = waterCoord(random.uniform(0,2*WalkPt))
    O1 = O1 + walkPoint
    H2 = H2 + walkPoint
    H3 = H3 + walkPoint

    return O1,H2,H3

def getInput():
    # Get initial data
    i,samespace = 0,0

    inp_file = open_file(workdir+orig_file, "r") # open files
    thing1 = inp_file.readlines() # reads entire file, can't search after this
    n_lines = len(thing1)
    inp_file.close()

    for data_num,line in enumerate(thing1): # searches file for molecule origin
        if "$DATA" in line: break

    input_data = data_num + 3
    inputData = thing1[:input_data] # initial data from input files
    atomStarting = int(n_lines - input_data) # atoms in initial file/molecule
    atomTotal = int(atomStarting + n_water*3) # overall total number of atoms

    atom,atomNumber,atomCoordinate = [],[],[]
    cartCoordinates = []

    coord_start = input_data
    while coord_start < n_lines-1:
        cartCoordinates.append(thing1[coord_start])
        coord_start += 1 # reads in coordinates

    j = 0
    for line in cartCoordinates:
        atom.append(str(line.split()[0]))
        atomNumber.append(str(line.split()[1]))
        x = float(line.split()[2])
        y = float(line.split()[3])
        z = float(line.split()[4])
        atomCoordinate.append(Point(x,y,z))
        j += 1

    #orig_m = getCenter(x,y,z) # get molecule center
    orig_m = atomCoordinate[0]

    # sets center coordinate of molecule to (0,0,0)
    dmax = 0

```

```

for j in range(0,atomStarting-1,1):
    atomCoordinate[j] -= orig_m
    d = atomCoordinate[j].distance(orig)
    if d > dmax: dmax = d

# Gets the water coordinates for each new water molecule
# --- This corresponds to Step 2 in the flow chart.

water = 0
while water < n_water:
    O1,H2,H3 = waterCoord(dmax+max_d-2)
    # check that waters don't occupy same space - Naive
    for i in range(0,atomStarting-1,1):
        samespace = atomCoordinate[i].distance(O1)
        if (samespace < 1.5):
            break

    if (samespace >= 1.5):
        atom.append("O1")
        atomCoordinate.append(O1)
        atom.append("H2")
        atomCoordinate.append(H2)
        atom.append("H3")
        atomCoordinate.append(H3)
        water += 1

return atomStarting,atomTotal,inpData,atom,atomNumber,atomCoordinate,dmax

def newWaterCoord(atomCoordinate,dmax):
    ### Used for initiation only
    water = 0
    while water < 3*n_water:
        atomCoordinate.pop()          # removes water coordinates
        water += 1

    water = 0
    while water < n_water:
        O1,H2,H3 = waterCoord(dmax+max_d)
        # check that waters don't occupy same space - Naive
        for i in range(0,atomStarting-1,1):
            samespace = atomCoordinate[i].distance(O1)
            if (samespace < 2.0):
                break

        if (samespace >= 2.0):
            atom.append("O1")
            atomCoordinate.append(O1)
            atom.append("H2")
            atomCoordinate.append(H2)
            atom.append("H3")
            atomCoordinate.append(H3)
            water += 1

    return atomCoordinate

def getOutput(atomStarting,atomTotal,outFileName,calcType):
    iteration,finalEnergy,energyLineNumber,minEnergy,coord_start = 0,0,0,0,0

```



```

outputData = getOutFile(outFileName)
finalEnergy = float(-0.000666)

nserch = -1
nserchLine = "BEGINNING GEOMETRY SEARCH"
for data_num,line in enumerate(outputData):
    if nserchLine in line:
        nserch += 1

for data_num,line in enumerate(outputData): # searches file for minimum and final energy
    if "FINAL" in line:
        finalEnergy = float(line.split()[4])
        if finalEnergy < minEnergy:
            minEnergy = finalEnergy
        iteration += 1

iteration -= 1
if iteration == -1:
    iteration = 0

### Go through file to find convergence
locateTxt = "No location/failure"
wallClockTime = "Fatal Termination"
if finalEnergy != str(-0.000666):
    for data_num,line in enumerate(outputData):
        if "EQUILIBRIUM GEOMETRY LOCATED" in line:
            locateTxt = "Located"
            break
        elif "THE GEOMETRY SEARCH IS NOT CONVERGED!" in line:
            locateTxt = "Not Converged"
            break

cpuTimeStart = 0
for data_num,line in enumerate(outputData):
    if "CPU timing information for all processes" in line:
        cpuTimeStart = data_num + 2
        wallClockTime = 0
        break
    elif "ddikick.x: application process 0 quit unexpectedly" in line:
        break

terminated = testOutputFailure(outFileName)
if terminated == "Normally":
    for i in range(0,100,1):
        line = outputData[cpuTimeStart+i]
        if "-----" in line:
            break
        wallClockTime += float(line.split()[5])

### Find final coordinates, same for all
energyLineSearch = "BEGINNING GEOMETRY SEARCH POINT NSERCH=%4i ..."%(nserch)
for data,line in enumerate(outputData):
    if energyLineSearch in line:
        coord_start = data + 5
        break

xyzCoordinates(outFileName, calcType, coord_start, atomStarting, atomTotal)

```

```

return coord_start,finalEnergy,locateTxt,outputData,wallClockTime

def newInput(coord_start,outputData,calcType):
    atom,atomNumber,atomCoordinate = [],[],[]
    cartCoordinates = []

    waterStart = coord_start+atomStarting+2

    ##### TESTING CENTER
    #subFile = open_file(workdir+"progOutput.dat","a")

    if calcType == "EFP":
        for k in range(coord_start,coord_start+atomStarting-1,1):
            cartCoordinates.append(outputData[k])
        j = 0
        for i in range(waterStart,waterStart+4*n_water,1):
            if j%4 != 0:
                cartCoordinates.append(outputData[i])
                j +=1
            else:
                j += 1
        # Adds data from output file to lists
        j = 0
        for line in cartCoordinates:
            if j < atomStarting-1:
                atom.append(str(line.split()[0]))
                atomNumber.append(str(line.split()[1]))
                x = float(line.split()[2])
                y = float(line.split()[3])
                z = float(line.split()[4])
                atomCoordinate.append(Point(x,y,z))
                j += 1
            else:
                atom.append(str(line.split()[0]))
                xW = float(line.split()[1])
                yW = float(line.split()[2])
                zW = float(line.split()[3])
                atomCoordinate.append(Point(xW,yW,zW))
                j += 1
        else:
            for m in range(coord_start,coord_start+atomTotal-1,1):
                cartCoordinates.append(outputData[m])
            # Adds data from output file to lists
            j = 0
            for line in cartCoordinates:
                atom.append(str(line.split()[0]))
                atomNumber.append(str(line.split()[1]))
                x = float(line.split()[2])
                y = float(line.split()[3])
                z = float(line.split()[4])
                atomCoordinate.append(Point(x,y,z))
                j += 1

            #orig_m = getCenter(x,y,z)
            orig_m = atomCoordinate[0]

            """ Used to compare isomers """
            dmax = 0

```

```

for j in range(0,atomTotal-1,1):
    atomCoordinate[j] -= orig_m
    d = atomCoordinate[j].distance(orig)
    if d > dmax: dmax = d

return atom,atomNumber,atomCoordinate

def writeInput(atomStarting,atomTotal,atom,atomNumber,atomCoordinate,inpData):
    ### Write to New Input File ###
    updated_file = pre_file + "%d-%dWEFP-%d.inp" %(bhStart, n_water, Number)
    new_file = open_file(workdir+updated_file, "w")
    new_file.writelines([item for item in inpData])

    # Central molecule coordinates
    for j in range(0,atomStarting-1,1):
        new_file.write(atom[j]+" "+atomNumber[j]+" "+s_con(atomCoordinate[j])+"\n")

    new_file.write(" $END\n $EFRAG\nCOORD=CART\n")

    # EFP water coordinates
    j,l = 0,0
    waterLoop = atomTotal-atomStarting
    while j < waterLoop:
        k = j + atomStarting-1
        if (l%4 == 0):
            new_file.write(label)
            l += 1
        else:
            new_file.write(atom[k]+" "+s_con(atomCoordinate[k])+"\n")
            l += 1
            j += 1

    new_file.write(" $END")
    new_file.close()

    submitGAMESS(updated_file)

    return

def submitGAMESS(updated_file):
    ### Submit to GAMESS
    # --- This corresponds to Step 3 in the flow chart.

    submitTime = getTime()
    gamess = "/opt/gamess/rungms "+updated_file[:-4]+" 00 "+str(cpus)+" >& "+updated_file[:-4]+".out"

    # This keeps track of the time of each GAMESS job submission
    subFile.write(updated_file[:-4]+".out\t"+submitTime+"\n")

    os.system(gamess)
    # If output file doesn't exist, it repeats the gamess submission
    if os.path.exists(updated_file[:-4]+".out") == False:
        os.system("rm " + pre_file + "*WEFP-*.out")
        os.system(gamess)

    os.system("rm /home/mstryker/gamscr/"+updated_file[:-4]+"*")

    return

```

```

def pbs_job_file(jobName,inputFile):
    subFile.write("PBS job:\t"+str(jobName)+"\t"+str(getTime())+"\n")

    ### Run ab initio separately
    pbs_write = open(workdir+"pbs_"+jobName+".qjob","w")
    pbs_write.write("#!/bin/bash\n#\n" + \
        "\n#PBS -N " + jobName + \
        "\n#PBS -q bigjob" + \
        "\n#PBS -m ae" + \
        "\n#PBS -l procs=4" + \
        "\n#PBS -l walltime=120:00:00" + \
        "\n#PBS -e " + workdir + "pbs_"+jobName+".err" + \
        "\n#PBS -o " + workdir + "pbs_"+jobName+".out\n#\n" + \

        "\n# set job name\nJOB="+inputFile+"\n" + \
        "\n# set up working dir \ndeclare -x GAMWORKDIR="+workdir+"\n\n"
        "cd $PBS_O_WORKDIR\n" + \
        "echo #####Begin Job Information#####\n" + \
        "echo Working directory is $PBS_O_WORKDIR\n" + \
        "echo Running on host `hostname`\n" + \
        "echo Time is `date`\n" + \
        "echo Directory is `pwd`\n" + \
        "echo This jobs runs on the following processors:\n" + \
        "echo `cat $PBS_NODEFILE`\n" + \
        "NPROCS=`wc -l < $PBS_NODEFILE`\n" + \
        "echo This job has allocated $NPROCS cpus\n" + \
        "echo #####End Job Information#####\n" + \
        "\n# GAMESS executable parameter for monte carlo\n" + \
        "VERNO=00 NCPUS=$NPROCS\n" + \
        "cd $GAMWORKDIR\n" + \
        "/opt/games/rungms $JOB $VERNO $NCPUS >& $JOB.out\n")
    pbs_write.close()

    return

#####
##### --- MAIN SECTION OF PROGRAM --- #####
#####
orig = Point(0,0,0)
# constants from GAMESSrefs manual for H2ORHF EFP
O_Hdistance = 0.9438636
ang_HOH = (106.70327)*pi/180
wallClockTime = 0
kJmolEnergy = 2625.5

user = getpass.getuser()
curr_wd = os.getcwd()

### Get data for run from data file generated from input script
# --- This corresponds to Step 1 in the flow chart.

inpFile = open_file("dataMCinp.txt","r")
computer = str(inpFile.readline().split()[0])
bhStart = int(inpFile.readline().split()[0])
gam_work = str(inpFile.readline().split()[0])
workdir = str(inpFile.readline().split()[0])

```

```

orig_file = str(inpFile.readline().split())[1]
WalkPt = float(inpFile.readline().split()[1])
n_water = int(inpFile.readline().split()[1])
max_d = float(inpFile.readline().split()[1])
trials = int(inpFile.readline().split()[1])
Teff = float(inpFile.readline().split()[1])
seedValue = float(inpFile.readline().split()[1])
inpFile.close()

random.seed(seedValue)
cpus = 4

submit_tool_dir = gam_work + "submit_tools/"

subFile = open_file(workdir+"progOutput.txt","a")
pre_file = orig_file[:orig_file.find('.')] # returns filename before .
data_f = open_file(workdir+"data.csv", "w")
data_f.write("FileName,WallClockTime,Terminated,Geometry,Angle,Final
Energy(hartree),dEnergy,Energy(kJ/mol),Metropolis,N_Energy,Isomer Distance,N_Isomer\n")
data_f.close()

finalEnergy,previousEnergy,metrop,isomerEquivalence = 0,0,0,0
N_E,N_I,lastSuccessN = 0,0,0
label = "FRAGNAME=H2ODFT\n"
terminated = "Normally"
failedTrials = workdir+"/Fails/"
os.system("mkdir "+failedTrials)

# Get initial data from input file
atomStarting,atomTotal,inpData,atom,atomNumber,atomCoordinate,dmax = getInput()

inpFile = open_file("dataMCinp.txt","a")
inpFile.write("\nStartingAtoms:\t"+str(atomStarting))
inpFile.close()

previousH,previousC,previousN,previousO = getInitialDist(atom,atomCoordinate,atomStarting)
previousAtom = atom
previousAtomNumber = atomNumber
previousCoordinate = atomCoordinate

for Number in range(0,trials+1,1):
    N_highE,N_unsucc = 0,0

    if Number == 0:
        k = 0
        for i in range(0,10,1):
            writeInput(atomStarting,atomTotal,atom,atomNumber,atomCoordinate,inpData)
            terminated = testOutputFailure(pre_file + "%d-%dWEFP-%d.out" %(bhStart, n_water, Number))
            ### This will exclude the fatal error files generated by GAMESS, hopefully
            if terminated == "Normally":
                break
            else: ### Hopefully this will get rid of repeating the initial file with error endlessly
                movedFail = "mv "+pre_file+"%d*.out " %(bhStart) + failedTrials+pre_file+"fail%d.out" %(k)
                os.system(movedFail)
                k += 1
                os.system("rm " + pre_file + "%d*" %(bhStart))
                atomCoordinate = newWaterCoord(atomCoordinate,dmax)

```

```

else:
    fileName = pre_file + "%d-%dWEFP-%d.out" %(bhStart,n_water, Number-1)
    coord_start,finalEnergy,locateTxt,outputData,wallClockTime = getOutput(atomStarting,atomTotal,fileName,"EFP")
    if Number == 1:
        previousEnergy = finalEnergy
    if finalEnergy != str(-0.000666):
        prev_coord_start = coord_start
        prev_outputData = outputData

    atom,atomNumber,atomCoordinate = newInput(prev_coord_start,prev_outputData, "EFP")

    # Determine acceptance of new optimization
    # 1 = too high Boltzmann dist, 0 = accept
    metropAuto = "reject"
    d_Energy = finalEnergy - previousEnergy
    if d_Energy > 0:
        N_highE,metrop = metropolis(d_Energy)
    else:
        metropAuto = "accept" # --- This corresponds to Step 4 in the flow chart.

    N_E += N_highE

    # 1 = isomers equivalent, 0 = differ
    N_unsucc,isomerEquivalence,H,C,N,O =
atomIsomerComp(atom,atomCoordinate,atomStarting,previousH,previousC,previousN,previousO)

    angle = threeAtomIsomerComp(0, 1, 2, atomCoordinate)
    angleVariation = angleComp(0,1,7, atomCoordinate,previousCoordinate)
    dihedralVariation = torsionComp(0,1,7,7,8,9, atomCoordinate,previousCoordinate)
    dihedralAng = dihedral(1,1,7,7,8,9, atomCoordinate)

    if dihedralVariation > 7.0 and N_unsucc == 1:
        N_unsucc = 0

    N_I += N_unsucc

    # If acceptance, create the new input file
    # --- This corresponds to Step 4 and 5 in the flow chart.
    if metropAuto == "accept" or (N_highE == 0 and N_unsucc == 0) and terminated == "Normally":
        prev_coord_start = coord_start
        prev_outputData = outputData
        lastSuccessN = Number
        previousH,previousC,previousN,previousO = H,C,N,O
        previousEnergy = finalEnergy
        previousAtom = atom
        previousAtomNumber = atomNumber
        previousCoordinate = atomCoordinate
        previousEnergy = finalEnergy

        data_f = open_file(workdir+"data.csv", "a")
        data_f.write(str(fileName):-
4))+".xyz,"+str(wallClockTime)+","+terminated+","+locateTxt+","+str(angle)+","+str(finalEnergy)+","+str(d_Energy)+","+str(
finalEnergy*kJmolEnergy)+","+str(metrop)+","+str(N_highE)+","+str(isomerEquivalence)+","+str(N_unsucc)+"\n")
        data_f.close()
    else:
        trials -= 1

    if Number != (trials):

```

```

        newCoordinate = SaundersKick(atomStarting,atomTotal,previousCoordinate)
        writeInput(atomStarting,atomTotal,atom,atomNumber,newCoordinate,inpData)
        terminated = testOutputFailure(pre_file + "%d-%dWEFP-%d.out" %(bhStart, n_water, Number))

os.system("rm " + pre_file + "*WEFP-*.out")
os.system("rm /home/mstryker/gamscr/"+pre_file+"%d*" %(bhStart))

inpFile = open_file("dataMCinp.txt","a")
inpFile.write("\nCPUs: \t"+str(cpus))
inpFile.write("\nN high E: \t"+str(N_E))
inpFile.write("\nN unsucc: \t"+str(N_I))
inpFile.close()

os.system("mkdir inputFiles")
input_files = workdir + "/inputFiles/"
os.system("mv " + pre_file + "%d*.inp" %(bhStart) + " " + input_files)

#####
##### --- Start full ab-initio --- #####
#####
SortCsvFile("data.csv", 5, ",", "forward")    #Sort column 5 in forward order

csvFile = open_file("data.csv", "r")
csvlines = csvFile.readlines()
NumcsvLines = len(csvlines)    #Length of data file including header line

##### ERROR CHECKING #####
subFile.write("CSV lines:\t"+str(NumcsvLines)+"\n")

lowE_Coord = open_file("MCBH.xyz", "a")
revE_Coord = open_file("MCBHrev.xyz", "a")

##### --- SWITCH TO 50 kJ/mol -----
### 1 hartree = 627.503 kcal/mol
### 1 hartree = 2625.5 kJ/mol
if trials <= 20:
    iterations = trials + 1
else:
    iterations = 20 + 1

j, k, kEnergyDiff = 0, 0, 0
newiterations = 0

##### ERROR CHECKING #####
while j < trials:
    subFile.write("While loop,j:\t"+str(j)+"\t"+"Trials:"+str(trials)+"\n")
    subFile.write("CsvFileLineNumber:\t"+str(NumcsvLines-j)+"\n")

    filename = str(csvlines[j+1].split(",")[0])    #Returns the filename
    filenameRev = str(csvlines[NumcsvLines-j-1].split(",")[0])    #Returns the filename

    jobtermination = str(csvlines[NumcsvLines-j-1].split(",")[3])
    kEnergy = float(csvlines[NumcsvLines-j-1].split(",")[7])

    subFile.write("Energy_kJ:\t"+str(csvlines[NumcsvLines-j-1].split(",")[7]+"n"))

    if j == 0:
        initial_kJ = kEnergy

```

```

kJEnergyDiff = initial_kJ - kJEnergy

subFile.write("EnergyDiff_kJ:\t"+str(kJEnergyDiff)+"\n")

outFile = open_file(filename, "r")
outputData = outFile.readlines()      #Read xyz coordinate data

outFileRev = open_file(filenameRev, "r")
outputDataRev = outFileRev.readlines()      #Read xyz coordinate data

##### ERROR CHECKING #####
subFile.write("OutputFile:\t"+str(filename)+"\nOutputFileRev:\t"+str(filenameRev)+"\n")

if j == 0:
    lines = len(outputData)
    if jobtermination != "No location/failure":
        newiterations += 1
        for i in range(0,lines,1):
            if i == 1:
                lowE_Coord.write("Frame: "+str(newiterations)+"\t"+str(outputData[i]))
                revE_Coord.write("Frame: "+str(newiterations)+"\t"+str(outputDataRev[i]))
            else:
                lowE_Coord.write(str(outputData[i]))
                revE_Coord.write(str(outputDataRev[i]))

outFile.close()
outFileRev.close()

if kJEnergyDiff > 50 or newiterations == iterations:
    subFile.write("Break, newiterations:\t"+str(newiterations)+"\n")
    break

k += 1
j += 1

subFile.write("While loop,j:\t"+str(j)+"\n")
subFile.write("NewIterations:\t"+str(newiterations)+"\n")

csvFile.close()
lowE_Coord.close()
revE_Coord.close()

EFP_Coord = open_file("MCBHrev.xyz", "r")      #Run from lowest to highest energy

abData = open_file(workdir+"abinitioMCBH.txt", "r")
new_inputData = abData.readlines()
abFiles = open_file(workdir+"scattterFiles.txt", "a")
abFiles.write(str(newiterations)+"\n")

for i in range(0, newiterations, 1):
    newinp_file = str(EFP_Coord.readline().split()[1])+"-%dabi.inp"%(i)
    abFiles.write(newinp_file[:-4]+".out\n")

    subFile.write("NewInputFile:\t"+newinp_file+"\n")
    inpFile = open_file(newinp_file, "w")
    inpFile.writelines([item for item in new_inputData])

    for j in range(0, 1, 1):

```



```

    line = EFP_Coord.readline()
    oldwallClockTime = line.split()[2] #####
    if oldwallClockTime == "Fatal Termination":
        oldwallClockTime = 0
    for j in range(0,atomStarting-1, 1):
        line = EFP_Coord.readline()
        inpFile.write(line)
    for j in range(0,atomTotal-atomStarting, 1):
        atomEFP = str(EFP_Coord.readline())
        atom = atomEFP[:3]    #prints first character
        atomCoordinate = atomEFP[17:62]
        if atom == " H":
            atomNumber = "1.0"
        else:
            atomNumber = "8.0"
        atomData = atom+"      "+atomNumber+atomCoordinate+"\n"
        inpFile.write(atomData)
    inpFile.write(" $END")
    inpFile.close()

    jobName = "BH-"+str(bhStart)+'-'+str(i)

    pbs_job_file(jobName,newinp_file[:-4])    # Submit batch job for each ab initio run separately

subFile.write("First newiterations loop,i:\t"+str(i)+"\n")

abFiles.write("\n Finished file loop!\n")
abFiles.close()
subFile.write("BHCloseup reached!\n")

# Submit all the PBS files, except closeup
for i in range(0, newiterations, 1):
    jobName = "BH-"+str(bhStart)+'-'+str(i)
    p=os.popen("qsub pbs_" +jobName+".qjob")
    jobid=p.readline()
    p.close()

    if i == 0:
        jobString = str(jobid)
    else:
        jobString = str(jobString[:-1])+" "+str(jobid)

subFile.write("Second newiterations loop,i:\t"+str(i)+"\n")
subFile.write("JobString:\t"+jobString+"\n")
subFile.write(workdir+"pbs_BHCloseup.qjob")

# FINAL PBS BATCH DEPENDANCY JOB
pbs_f = open('pbs_BHCloseup.qjob', 'w')
pbs_f.write("#!/bin/bash\n#\n" + \
    "#PBS -N " + "BHClose" + str(bhStart)+ \
    "\n#PBS -q bigjob" + \
    "\n#PBS -W depend=afterany:" + jobString + "\n" + \
    "\n#PBS -m ae" + \
    "\n#PBS -l procs=4" + \
    "\n#PBS -l walltime=120:00:00" + \
    "\n#PBS -e " + workdir + "pbs_BHClose.err" + \
    "\n#PBS -o " + workdir + "pbs_BHClose.out\n\n" + \

```



```

for j in range(0,atomNumber-1):
    if atom[j] == "H":
        H.append(atomCoordinate[j].distance(finalOrigin))
    elif atom[j] == "C":
        C.append(atomCoordinate[j].distance(finalOrigin))
    elif atom[j] == "N":
        N.append(atomCoordinate[j].distance(finalOrigin))
    elif atom[j] == "O":
        O.append(atomCoordinate[j].distance(finalOrigin))

atomDistanceSort(H)
atomDistanceSort(C)
atomDistanceSort(N)
atomDistanceSort(O)

return H,C,N,O

def getInput():
    # Get initial data
    i,samespace = 0,0

    inp_file = open_file(workdir+orig_file, "r") # open files
    thing1 = inp_file.readlines() # reads entire file, can't search after this
    n_lines = len(thing1)
    inp_file.close()

    for data_num,line in enumerate(thing1): # searches file for molecule origin
        if "$DATA" in line: break

    input_data = data_num + 3
    inpData = thing1[:input_data] # initial data from input files
    atomStarting = int(n_lines - input_data) # atoms in initial file/molecule
    atomTotal = int(atomStarting + n_water*3) # overall total number of atoms

    return atomStarting,atomTotal

def getOutput(atomStarting,atomTotal,outFileName,calcType):
    iteration,finalEnergy,energyLineNumber,minEnergy,coord_start = 0,0,0,0,0

    imaginary = 'real'
    outputData = getOutFile(outFileName)
    finalEnergy = 666

    nserch = -1
    nserchLine = "BEGINNING GEOMETRY SEARCH"
    for data_num,line in enumerate(outputData):
        if nserchLine in line:
            nserch += 1

    for data_num,line in enumerate(outputData): # searches file for minimum and final energy
        if "FINAL" in line:
            finalEnergy = float(line.split())[4]
            if finalEnergy < minEnergy:
                minEnergy = finalEnergy
            iteration += 1

    iteration -= 1
    if iteration == -1:

```

```

iteration = 0

### Go through file to find convergence
locateTxt = "No location/failure"

if finalEnergy != 666:
    for data_num,line in enumerate(outputData):
        if "EQUILIBRIUM GEOMETRY LOCATED" in line:
            locateTxt = "Located"
            break
        elif "THE GEOMETRY SEARCH IS NOT CONVERGED!" in line:
            locateTxt = "Not Converged"
            break

### Find final coordinates, same for all
energyLineSearch = "BEGINNING GEOMETRY SEARCH POINT NSERCH=%4i ..."%(nserch)
for data,line in enumerate(outputData):
    if energyLineSearch in line:
        coord_start = data + 5
        break

xyzCoordinates(outFileName, calcType, coord_start, atomStarting, atomTotal)

for data_num,line in enumerate(outputData):
    if "IMAGINARY" in line:
        imaginary = "IMAGINARY freq"

return coord_start,finalEnergy,locateTxt,outputData,imaginary

def newInput(coord_start,outputData):
    atom,atomNumber,atomCoordinate = [],[],[]
    x,y,z,cartCoordinates = [],[],[],[]

    waterStart = coord_start+atomStarting+2

    for m in range(coord_start,coord_start+atomTotal-1,1):
        cartCoordinates.append(outputData[m])
    # Adds data from output file to lists
    j = 0
    for line in cartCoordinates:
        atom.append(str(line.split()[0]))
        atomNumber.append(str(line.split()[1]))
        x.append(float(line.split()[2]))
        y.append(float(line.split()[3]))
        z.append(float(line.split()[4]))
        atomCoordinate.append(Point(x[j],y[j],z[j]))
        j += 1

    orig_m = atomCoordinate[0]

    """ Used to compare isomers """
    dmax = 0
    for j in range(0,atomTotal-1,1):
        atomCoordinate[j] -= orig_m
        d = atomCoordinate[j].distance(orig)
        if d > dmax: dmax = d

    return atom,atomNumber,atomCoordinate

```

```

def submitGAMESS(updated_file):
    ### Submit to GAMESS
    submitTime = getTime()
    gamess = "/opt/gamess/rungms "+updated_file[:-4]+" 00 "+str(cpus)+" >& "+updated_file[:-4]+".out"

    # This keeps track of the time of each GAMESS job submission
    subFile = open_file(workdir+"progOutput.dat", "a")
    subFile.write(updated_file[:-4]+".out\t"+submitTime+"\n")
    subFile.close()

    os.system(gamess)
    # If output file doesn't exist, it repeats the gamess submission
    if os.path.exists(updated_file[:-4]+".out") == False:
        os.system("rm " + pre_file + "*WEFP-*.out")
        os.system(gamess)

    os.system("rm "+workdir+updated_file[:-4]+".dat")

    return

#####
##### --- MAIN BLOCK TO WRAP UP JOBS --- #####
#####
kJmolEnergy = 2625.5
orig = Point(0,0,0)
cpus = 4
user = getpass.getuser()
curr_wd = os.getcwd()
wallClockTime = 0
imaginary='real'

### Get data for run from data file
inpFile = open_file("dataMCinp.txt", "r")
computer = str(inpFile.readline().split())[0]
bhStart = int(inpFile.readline().split()[0])
gam_work = str(inpFile.readline().split()[0])
workdir = str(inpFile.readline().split()[0])
orig_file = str(inpFile.readline().split()[1])
WalkPt = float(inpFile.readline().split()[1])
n_water = int(inpFile.readline().split()[1])
max_d = float(inpFile.readline().split()[1])
trials = int(inpFile.readline().split()[1])
Teff = float(inpFile.readline().split()[1])
seedValue = float(inpFile.readline().split()[1])
inpFile.close()

Finaldata_f = open_file(workdir+"Finaldata.csv", "w")
Finaldata_f.write("FileName,Frequency,WallClockTime,Terminated,Geometry,Angle,Final
Energy(hartree),Energy(kJ/mol)\n")
Finaldata_f.close()

atomStarting,atomTotal=getInput()
abData=open_file(workdir+"abinitioMCBH.txt", "r")
new_inputData=abData.readlines()
abData.close()

# Get output files

```

```

abFiles=open_file(workdir+"scattterFiles.txt", "r")
newiterations = int(abFiles.readline())

subFile=open_file(workdir+'progOutput.dat', 'a')

for i in range(0, newiterations, 1):
    output_file = str(abFiles.readline()[:-1])
    terminated = testOutputFailure(output_file)

    if terminated != "Normally":
        coord_start,finalEnergy,locateTxt,outputData,imaginary = getOutput(atomStarting,atomTotal,output_file,"abinitio")
        atom,atomNumber,atomCoordinate = newInput(coord_start,outputData)
        # Testing
        subFile.write(str(coord_start)+"\n")
        for m in range(0,atomTotal-1,1):
            subFile.write(atom[m]+" "+atomNumber[m]+" "+s_con(atomCoordinate[m])+"\n")

        os.system("rm "+output_file)
        os.system("rm /home/mstryker/gamsr/"+output_file[:-4]+"*")

        inpFile = open_file(output_file[:-4]+".inp", "w")
        inpFile.writelines([item for item in new_inputData])
        ### DOESN'T WRITE THIS
        for n in range(0,atomTotal-1,1):
            inpFile.write(atom[n]+" "+atomNumber[n]+" "+s_con(atomCoordinate[n])+"\n") #Change new_file.write*
        inpFile.write("$END")
        inpFile.close()

        submitGAMESS(output_file)

    # Get output for analysis
    wallClockTime = wallTime(output_file)
    coord_start,finalEnergy,locateTxt,outputData,imaginary = getOutput(atomStarting,atomTotal,output_file,"abinitio")
    subFile.write(str(output_file)+"\t"+str(outputData[coord_start])+"\n")
    atom,atomNumber,atomCoordinate = newInput(coord_start,outputData) #-----

    ###
    angle = threeAtomIsomerComp(0, 1, 2, atomCoordinate)
    H,C,N,O = getInitialDist(atom,atomCoordinate,atomTotal)
    Finaldata_f = open_file(workdir+"Finaldata.csv", "a")
    # Add angle -----angle not changing
    Finaldata_f.write(str(output_file[:-
4]))+".xyz,"+str(imaginary)+","+str(wallClockTime)+","+str(terminated)+","+locateTxt+","+str(angle)+","+str(finalEnergy)+","+str
(finalEnergy*kJmolEnergy)+"\n")
    Finaldata_f.close()

SortCsvFile(workdir+"Finaldata.csv", 6, "", "forward")
Finaldata_f = open_file(workdir+"Finaldata.csv", "r")
header = Finaldata_f.readline()
lowFinalE_Coord = open_file("MCBH_Final.xyz", "a")

for m in range(0,newiterations,1):
    # This loop writes the xyz coordinate data to the MCBH.xyz file in order from highest to lowest energy
    filename = str(Finaldata_f.readline().split(",")[0])

    # This keeps track of the time of each GAMESS job submission
    outFile = open_file(filename, "r")
    outputData = outFile.readlines()

```

```

lines = len(outputData)
for i in range(0,lines,1):
    if i == 1:
        lowFinalE_Coord.write("Frame: "+str(m)+"\t"+str(outputData[i]))
    else:
        lowFinalE_Coord.write(str(outputData[i]))
outFile.close()

Finaldata_f.close()
lowFinalE_Coord.close()
subFile.close()
os.system("mkdir abinitioInputFiles")
input_files = workdir + "/abinitioInputFiles/"
os.system("mkdir xyzCoordinates")
coord_files = workdir + "xyzCoordinates"
os.system("mv " + orig_file[:-4] + "%d*.inp" %(bhStart) + " " + input_files)
os.system("mv " + orig_file[:-4] + "%d-%dWEFP-*.xyz" %(bhStart, n_water) + " " + coord_files)
os.system("rm " + orig_file[:-4] + "/*.out")

os.system("mkdir rstFiles")
os.system("mv *.rst rstFiles")
os.system("mkdir pbsFiles")
os.system("mv pbs_BHabopt* pbsFiles")

os.system("rm *.dat")

```

---

FILENAME: bh37\_sort.py

#!/usr/bin/env python

# Sort final MC files

# 12/3/2012

# Used to resort final MC simulation files

```

import time
from mcpointsB import *
from math import *
import os,sys, csv, getpass, operator

```

### Get data for run from data file

```

inpFile = open_file("dataMCinp.txt", "r")
computer = str(inpFile.readline().split())[0]
bhStart = int(inpFile.readline().split()[0])
gam_work = str(inpFile.readline().split()[0])
workdir = str(inpFile.readline().split()[0])
orig_file = str(inpFile.readline().split()[1])
WalkPt = float(inpFile.readline().split()[1])
n_water = int(inpFile.readline().split()[1])
max_d = float(inpFile.readline().split()[1])
trials = int(inpFile.readline().split()[1])
Teff = float(inpFile.readline().split()[1])
seedValue = float(inpFile.readline().split()[1])
atomStarting = int(inpFile.readline().split()[1])
inpFile.close()

```

```

while(1):
    n_column = raw_input("What column would you like to sort by? ")
    try:
        n_column = int(n_column)

```

```

        if isinstance(n_column,int):
            break
    except:
        print "Value is not an integer!"

n_column -= 1
atomTotal = atomStarting + 3*n_water

Final_Coord = open_file("MCBH_Final.xyz", "r")
mcbh_Coord = Final_Coord.readlines()
final_length = len(mcbh_Coord)

loop = final_length/(atomTotal+2)

errTrack = open("errTrack.txt", "a")

for i in range(0,final_length,1):
    line = mcbh_Coord[i]
    if i%(atomTotal+1) == 0:
        errTrack.write("mod:\t"+str(i%(atomTotal+1)))
        xyz_file = str(line.split()[1])+".xyz"
        fileName = open_file(xyz_file, "w")
        fileName.write(line)
        errTrack.write(str(i)+str("\t"+mcbh_Coord[i]))
    else:
        errTrack.write(str(i)+str("\t"+mcbh_Coord[i]))
        fileName.write(mcbh_Coord[i])

Final_Coord.close()

SortCsvFile("Finaldata.csv", n_column, ",", "forward")
Finaldata_f = open_file("Finaldata.csv", "r")
header = Finaldata_f.readline()
Final_Coord = open_file("MCBH_Final.xyz", "w")

for m in range(0,loop+2,1):
    # This loop writes the xyz coordinate data to the MCBH.xyz file in order from highest to lowest energy
    filename = str(Finaldata_f.readline().split(",")[0])

    # This keeps track of the time of each GAMESS job submission
    outFile = open_file(filename, "r")
    outputData = outFile.readlines()
    lines = len(outputData)
    for i in range(0,lines,1):
        if i == 1:
            Final_Coord.write(str(m)+" "+str(outputData[i]))
        else:
            Final_Coord.write(str(outputData[i]))
    outFile.close()

Final_Coord.close()

```

---

```

FILENAME: bh_popSearch.py
#!/usr/bin/env python

```

```

# Monte Carlo Conformation Search
# 10/19/2012

```



```
# Morgyn Stryker
```

```
"""
```

Scan coordinate files to determine equivalence and populations. Make a new population file everytime the geometry is unique, append if not. When all files have been read and appended create a final population file with the lowest energy structure and population count for each structure.

Test atom angle 1,2,8 for glycine!

```
threeAtomIsomerComp(1,2,8, atomCoordinate)
```

```
"""
```

```
from mcpointsB import *
```

```
from math import *
```

```
import os,sys,csv,getpass,operator
```

```
def atomIsomerComp(at,at_Coordinate,previousH,previousC,previousN,previousO):
```

```
    ### Compare Coordinates of Previous File ###
```

```
    isomerAccept = 0.0001      # distance in Angstrom
```

```
    H,C,N,O = getInitialDist(at,at_Coordinate,atomStarting)
```

```
    if len(H)!=len(previousH) or len(C)!=len(previousC) or len(N)!=len(previousN) or len(O)!=len(previousO):
```

```
        print "Lists different size!"
```

```
        sys.exit()
```

```
    else:
```

```
        isomerEquivalence = 0
```

```
        isomerEquivalence += isomer(H,previousH)
```

```
        isomerEquivalence += isomer(C,previousC)
```

```
        isomerEquivalence += isomer(N,previousN)
```

```
        isomerEquivalence += isomer(O,previousO)
```

```
    if (isomerEquivalence < isomerAccept):
```

```
        # Same isomer as a previous, relaxed
```

```
        N_unsucc = 1
```

```
    else:
```

```
        # New isomer
```

```
        N_unsucc = 0
```

```
    return N_unsucc,isomerEquivalence,H,C,N,O
```

```
def isomer(self,previous):
```

```
    # Determine if two isomers are equivalent
```

```
    isomerEquivalence = 0
```

```
    for i in range(1,len(self)-1,1):
```

```
        isomerEquivalence += (previous[i] - self[i])**2/(previous[i]**2 + self[i]**2)
```

```
    return isomerEquivalence
```

```
def atomPtArray(outputData):
```

```
    cartCoordinates = []
```

```
    atoMNumber = []
```

```
    atoM, atoM_Coordinate = [], []
```

```
    errFile = open_file(coord_files+'errFile.txt','a')
```

```

for k in range(2, atomStarting + 1,1):
    errFile.write(str(outputData[k]))
    cartCoordinates.append(outputData[k])

for k in range(waterStart-1, (waterStart + 3*n_water-1),1):
    errFile.write(str(outputData[k]))
    cartCoordinates.append(outputData[k])
errFile.write('End\n')
errFile.close()

# Adds data from output file to lists
n = 0
for line in cartCoordinates:
    if n < atomStarting-1:
        atoM.append(str(line.split()[0]))
        atoMNumber.append(str(line.split()[1]))
        x = float(line.split()[2])
        y = float(line.split()[3])
        z = float(line.split()[4])
        atoM_Coordinate.append(Point(x,y,z))
        n += 1
    else:
        atoM.append(str(line.split()[0]))
        x = float(line.split()[1])
        y = float(line.split()[2])
        z = float(line.split()[3])
        atoM_Coordinate.append(Point(x,y,z))
        n += 1

return atoM, atoM_Coordinate

#####
##### --- MAIN SECTION OF PROGRAM --- #####
#####
orig = Point(0,0,0)
atom, atomCoordinate = [], []
conformer = 1

### Get data for run from data file
inpFile = open_file("dataMCinp.txt", "r")
computer = str(inpFile.readline().split()[0])
bhStart = int(inpFile.readline().split()[0])
gam_work = str(inpFile.readline().split()[0])
workdir = str(inpFile.readline().split()[0])
orig_file = str(inpFile.readline().split()[1])
WalkPt = float(inpFile.readline().split()[1])
n_water = int(inpFile.readline().split()[1])
max_d = float(inpFile.readline().split()[1])
trials = int(inpFile.readline().split()[1])
Teff = float(inpFile.readline().split()[1])
seedValue = float(inpFile.readline().split()[1])
atomStarting = int(inpFile.readline().split()[1])
inpFile.close()

waterStart = atomStarting+2

coord_files = workdir + "xyzCoordinates/"
os.system('cp '+str(workdir)+'data.csv ' + str(coord_files) + 'data.csv')

```

```

os.system('rm '+str(coord_files)+"bh*")

csvFile = open_file(str(coord_files) + "data.csv", "r")
csvlines = csvFile.readlines()
NumbcsvLines = len(csvlines)    #Length of data file including header line
csvFile.close()

totalStructures = open_file(str(coord_files)+'bhCompleteSet.xyz', 'a')

for j in range(0, NumbcsvLines-1, 1):
    # Last filename listed should be the lowest energy
    filename = str(csvlines[j+1].split(",")[0])    #Returns the filename
    outFile = open_file(str(coord_files)+filename, "r")
    outputData = outFile.readlines()    #Read xyz coordinate data
    outFile.close()
    lines = len(outputData)

    for i in range(0,lines,1):
        if i == 1:
            totalStructures.write("Frame:\t"+str(j+1)+"\t"+str(outputData[i]))
        else:
            totalStructures.write(str(outputData[i]))

# Keep track of unique geometries
uniqueGeo = 0
angleVariation = 0.0
dihedralVariation = 0.0

finalCSVCoordFile = open_file(str(coord_files)+'bhCoordCSV.csv', 'a')
finalCSVCoordFile.write('FileName,Energy(kJ/mol),GeometryGroup,ConfEquivalence,Dihedral,DihedralVariation\n')

for j in range(0, NumbcsvLines-1, 1):
    addedToFile = 0
    isomerEq = 0
    # Last filename listed should be the lowest energy
    filenameRev = str(csvlines[NumbcsvLines-j-1].split(",")[0])    #Returns the filename

    outFileRev = open_file(str(coord_files)+filenameRev, "r")
    outputData = outFileRev.readlines()    #Read xyz coordinate data
    outFileRev.close()

    # Initiate new geometry file
    if j == 0:
        print 'FirstFile:\t'+filenameRev
        popCoordFile = open_file(str(coord_files)+"bhPop_"+str(bhStart)+"_"+str(uniqueGeo)+".xyz", 'a')
        lines = len(outputData)
        stepSize = lines
        atomOld, atomCoordinateOld = atomPtArray(outputData)

        dihedralAng = dihedral(0,1,7,7,8,9,atomCoordinateOld)
        for i in range(0,lines,1):
            if i == 1:
                popCoordFile.write("\tIsomer Equivalence:\t"+str(isomerEq)+"\tDihedral:\t"+str(dihedralAng)+'\n')
            else:
                popCoordFile.write(str(outputData[i]))
        else:
            # Get new coordinate data from file of suspect

```

```

atom, atomCoordinate = atomPtArray(outputData)

if j == 1:
    previousH,previousC,previousN,previousO = getInitialDist(atomOld,atomCoordinateOld,atomStarting)

    conformer,isomerEq,H,C,N,O = atomIsomerComp(atom,atomCoordinate,atomStarting,
previousH,previousC,previousN,previousO)
    angleVariation = angleComp(0,1,7, atomCoordinate,atomCoordinateOld)
    dihedralVariation = torsionComp(0,1,7,7,8,9, atomCoordinate,atomCoordinateOld)
    dihedralAng = dihedral(0,1,7,7,8,9, atomCoordinate)

    # If there exists a conformer match, the coordinates get added to the file and breaks from the loop
    if conformer == 1 and dihedralVariation< 2.0:
        popCoordFile = open_file(str(coord_files)+"bhPop_"+str(bhStart)+"_0.xyz", 'a')
        for i in range(0,lines,1):
            if i == 1:
                popCoordFile.write("\tIsomer Equivalence:\t"+str(isomerEq)+"\tDihedral:\t"+str(dihedralAng)+'\n')
            else:
                popCoordFile.write(str(outputData[i]))
        popCoordFile.close()
        addedToFile = 1
        break
    else:
        for k in range(0, uniqueGeo, 1):
            popCoordFile = open_file(str(coord_files)+"bhPop_"+str(bhStart)+"_"+str(k)+".xyz", 'r')
            popLines = popCoordFile.readlines()
            numbXYZLines = len(popLines)
            popCoordFile.close()

            popLinesFirstSet = []
            for coordLine in range(0, stepSize, 1):
                popLinesFirstSet.append(popLines[coordLine])

            atomOld, atomCoordinateOld = atomPtArray(popLinesFirstSet)
            previousH,previousC,previousN,previousO = getInitialDist(atomOld,atomCoordinateOld,atomStarting)

            conformer,isomerEq,H,C,N,O =
atomIsomerComp(atom,atomCoordinate,previousH,previousC,previousN,previousO)

            angleVariation = angleComp(0,1,7, atomCoordinate,atomCoordinateOld)
            dihedralVariation = torsionComp(0,1,7,7,8,9, atomCoordinate,atomCoordinateOld)
            dihedralAng = dihedral(0,1,7,7,8,9, atomCoordinate)

            # If there exists a conformer match, the coordinates get added to the file and breaks from the loop
            if conformer == 1 and dihedralVariation < 5.0:
                popCoordFile = open_file(str(coord_files)+"bhPop_"+str(bhStart)+"_"+str(k)+".xyz", 'a')
                for i in range(0,lines,1):
                    if i == 1:
                        popCoordFile.write("\tIsomer Equivalence:\t"+str(isomerEq)+"\tDihedral:\t"+str(dihedralAng)+'\n')
                    else:
                        popCoordFile.write(str(outputData[i]))
                popCoordFile.close()
                addedToFile = 1
                break

        # Checks to see if coordinates were added to file during loop. If not, a new population file is made.
        if addedToFile == 0:
            uniqueGeo += 1

```

```

popCoordFile = open_file(str(coord_files)+"bhPop_"+str(bhStart)+"_"+str(uniqueGeo)+".xyz", 'a')

for i in range(0,lines,1):
    if i == 1:
        popCoordFile.write("\tIsomer Equivalence:\t"+str(isomerEq)+"\tDihedral:\t"+str(dihedralAng)+'\n')
    else:
        popCoordFile.write(str(outputData[i]))

if addedToFile == 1:
    finalCSVCoordFile.write(str(csvlines[NumbcsvLines-j-1].split(",")[0]) +','+str(csvlines[NumbcsvLines-j-1].split(",")[7])+','+str(k)+'/'+str(isomerEq)+'/'+str(dihedralAng)+'/'+str(dihedralVariation)+'\n')
else:
    finalCSVCoordFile.write(str(csvlines[NumbcsvLines-j-1].split(",")[0]) +','+str(csvlines[NumbcsvLines-j-1].split(",")[7])+','+str(uniqueGeo)+'/'+str(isomerEq)+'/'+str(dihedralAng)+'/'+str(dihedralVariation)+'\n')

# All unique geometry files should be completed, take the population, first geometrical structure, and group in one final file.
finalFile = open_file(str(coord_files)+"bhPop_Final.xyz", 'a')

for j in range(0, uniqueGeo, 1):
    popCoordFile = open_file(str(coord_files)+"bhPop_"+str(bhStart)+"_"+str(j)+".xyz", 'r')
    #print str(coord_files)+"bhPop_"+str(bhStart)+"_"+str(j)+".xyz"

    popLines = popCoordFile.readlines()
    numbLines = len(popLines)
    popCoordFile.close()
    popNumber = numbLines/stepSize

    for i in range(0,lines,1):
        if i == 1:
            finalFile.write("Population: "+str(popNumber)+"\t"+str(popLines[i]))
        else:
            finalFile.write(str(popLines[i]))

finalFile.close()

```

---

FILENAME: mcpointsB.py

#!/usr/bin/env python

# BH Library for import

# Methods to be used with Monte Carlo simulation

import random,time

from math import \*

import os,sys,csv,getpass,operator

class Point(object):

def \_\_init\_\_(self, x=0.0, y=0.0, z=0.0):

self.x = float(x)

self.y = float(y)

self.z = float(z)

def \_\_str\_\_(self): # displays 5 points after decimal and space for positive

return '(% 0.5f, % 0.5f, % 0.5f)' % (self.x, self.y, self.z)

def \_\_add\_\_(self, other): # adding 2 vectors

return Point(self.x + other.x, self.y + other.y, self.z + other.z)

```

def __sub__(self, other):    # subtracting 2 vectors
    return Point(self.x - other.x, self.y - other.y, self.z - other.z)

def __mul__(self, f):       # multiplication by f
    return Point(self.x * f, self.y * f, self.z * f)

def dot_prod(self, other):  # dot product [self.dot_prod(other)]
    return self.x*other.x + self.y*other.y + self.z*other.z

def unit_v(self, d):        # unit vector [self.unit_v(d)]
    return Point(self.x / d, self.y / d, self.z / d)

def cross_p(self, other):   # cross product [self.cross_p(other)]
    return Point(self.y*other.z - self.z*other.y, self.z*other.x - self.x*other.z, self.x*other.y - self.y*other.x)

def distance(self, other):  # distance between two points (same as magnitude)
    return (((self.x-other.x)**2 + (self.y-other.y)**2 + (self.z-other.z)**2)**0.5)

def getXYorZ(self, value):
    if value == 0:
        return self.x
    elif value == 1:
        return self.y
    else:
        return self.z

def getCenter(x,y,z):
    # Sorts the x,y,z lists to find min and max, first and last, respectively
    x.sort()
    y.sort()
    z.sort()
    # Must do len(x)-1 since list starts at 0
    xcenter = x[len(x)-1] + (x[len(x)-1]-x[0])/2
    ycenter = y[len(y)-1] + (y[len(y)-1]-y[0])/2
    zcenter = z[len(z)-1] + (z[len(z)-1]-z[0])/2

    return Point(xcenter, ycenter, zcenter)

def threeAtomIsomerComp(i, j, k, atomCoordinate):
    dist_a = atomCoordinate[i].distance(atomCoordinate[j])
    dist_b = atomCoordinate[i].distance(atomCoordinate[k])
    dist_c = atomCoordinate[j].distance(atomCoordinate[k])
    ques_c = (dist_a**2+dist_b**2)**0.5

    cosAng_C = (dist_a**2+dist_b**2-dist_c**2)/(2*dist_a*dist_b)
    angle_C = (acos(cosAng_C))*180/pi

    return angle_C

def angleComp(i, j, k, newCoord, oldCoord):
    angleOld = threeAtomIsomerComp(i, j, k, oldCoord)
    angleNew = threeAtomIsomerComp(i, j, k, newCoord)

    angleDiff = fabs(fabs(angleOld) - fabs(angleNew))

    return angleDiff

```

```

def dihedral(a, b, c, d, e, f, atomCoordinate):
    dist_ab = atomCoordinate[a].distance(atomCoordinate[b])
    dist_bc = atomCoordinate[b].distance(atomCoordinate[c])

    dist_de = atomCoordinate[d].distance(atomCoordinate[e])
    dist_ef = atomCoordinate[e].distance(atomCoordinate[f])

    vect_ab = (atomCoordinate[a] - atomCoordinate[b]).unit_v(dist_ab)
    vect_bc = (atomCoordinate[b] - atomCoordinate[c]).unit_v(dist_bc)

    vect_de = (atomCoordinate[d] - atomCoordinate[e]).unit_v(dist_de)
    vect_ef = (atomCoordinate[e] - atomCoordinate[f]).unit_v(dist_ef)

    plane_abc = vect_ab.cross_p(vect_bc)
    plane_def = vect_de.cross_p(vect_ef)

    cos_theta = plane_abc.dot_prod(plane_def)/((plane_abc.distance(Point(0, 0, 0)))*(plane_def.distance(Point(0, 0, 0))))

    dihedral_angle = (acos(cos_theta))*180/pi

    if dihedral_angle > 90.0:
        dihedral_angle = 180 - dihedral_angle

    return dihedral_angle

def torsionComp(i,j,k,m,n,p, newCoord, oldCoord):
    angleOld = dihedral(i,j,k,m,n,p, oldCoord)
    angleNew = dihedral(i,j,k,m,n,p, newCoord)

    torsional_diff = fabs(angleOld - angleNew)

    return torsional_diff

def s_con(self):
    """ May get rid of, if values are processed without commas.
    new_string = str(self)
    new_string = new_string.replace("(", "")
    new_string = new_string.replace(")", " ")
    new_string = new_string.replace(", ", " ")

    return new_string

def atomDistanceSort(self):
    # Sorted lists, lowest to highest
    k = 0
    self.sort()
    self.reverse()
    for line in self:
        k += 1

    return

def getInitialDist(atom,atomCoordinate,atomNumber):
    # Atoms in initial file
    H,C,N,O = ["H"],["C"],["N"],["O"]

    x,y,z = [],[],[]
    """ get x,y,z

```

```

for j in range(0,atomNumber-1):
    atomCoordinate[j]
    x.append(atomCoordinate[j].getXorZ(0))
    y.append(atomCoordinate[j].getXorZ(1))
    z.append(atomCoordinate[j].getXorZ(2))
#finalOrigin = getCenter(x,y,z)
finalOrigin = atomCoordinate[0]

for j in range(0,atomNumber-1):
    if atom[j] == "H":
        H.append(atomCoordinate[j].distance(finalOrigin))
    elif atom[j] == "C":
        C.append(atomCoordinate[j].distance(finalOrigin))
    elif atom[j] == "N":
        N.append(atomCoordinate[j].distance(finalOrigin))
    elif atom[j] == "O":
        O.append(atomCoordinate[j].distance(finalOrigin))

atomDistanceSort(H)
atomDistanceSort(C)
atomDistanceSort(N)
atomDistanceSort(O)

return H,C,N,O

def atomIsomerComp(atom,atomCoordinate,atomStarting,previousH,previousC,previousN,previousO):
    ### Compare Coordinates of Previous File ###
    isomerAccept = 0.0001    # distance in Angstrom

    H,C,N,O = getInitialDist(atom,atomCoordinate,atomStarting)

    if len(H)!=len(previousH) or len(C)!=len(previousC) or len(N)!=len(previousN) or len(O)!=len(previousO):
        print "Lists different size!"
        sys.exit()
    else:
        isomerEquivalence = 0
        isomerEquivalence += isomer(H,previousH)
        isomerEquivalence += isomer(C,previousC)
        isomerEquivalence += isomer(N,previousN)
        isomerEquivalence += isomer(O,previousO)

    if (isomerEquivalence < isomerAccept):
        # Same isomer as a previous, relaxed
        N_unsucc = 1
    else:
        # New isomer
        N_unsucc = 0

    return N_unsucc,isomerEquivalence,H,C,N,O

def isomer(self,previous):
    # Determine if two isomers are equivalent
    isomerEquivalence = 0

    for i in range(1,len(self)-1,1):
        isomerEquivalence += (previous[i] - self[i])**2/(previous[i]**2 + self[i]**2)

    return isomerEquivalence

```



```

def open_file(filename, mode):
    # exits program if file doesn't exist
    try:
        the_file = open(filename, mode)
    except(IOError), e:
        print "Unable to open the file", filename, "Ending program.\n", e
        sys.exit()
    else:
        return the_file

def getOutFile(outFileName):
    # get previous run output file
    out_file = open_file(outFileName, "r")
    outputData = out_file.readlines()      # reads entire file, can't search after this
    out_file.close()

    return outputData

def getTime():
    # localtime gives time: hour:minute:second month/day/year
    localtime = time.localtime(time.time())
    startTime = str(localtime[3])+':' +str(localtime[4])+':' +str(localtime[5])+ ' ' + str(localtime[1])+ '/' +str(localtime[2])+ '/' +str(localtime[0])

    return startTime

def SortCsvFile(filename, fieldNo, separator, order):
    delimiter=separator
    ### New version of sorting due to errors with previous 2/21/2012
    if order == "reverse":
        direction = True      #Sort from lowest to highest
    else:
        direction = False     #Sort from highest to lowest

    fileN = open(filename, "r")
    lineData = fileN.readlines()
    header = lineData[0]      #Gets header
    lines = len(lineData)     #Gets number of lines total
    fileN.close()

    #Rewrite file without header
    fileN = open(filename, "w")
    for i in range(1,lines,1):
        fileN.write(lineData[i])
    fileN.close()

    #Sort file
    fileN = open(filename,"r")
    data = csv.reader(fileN,delimiter=separator)
    sortedList = sorted(data, key=operator.itemgetter(fieldNo), reverse=direction)
    fileN.close()

    #Write sorted file again with header
    fileN = open(filename,"w")
    fileN.write(str(header))
    for row in sortedList:
        fileN.write(delimiter.join(row)+"\n")
    fileN.close()

```

```

return

def testOutputFailure(outFileName):
    outputData = getOutFile(outFileName)
    terminated = "Normally"

    for line in outputData:
        if "EXECUTION OF GAMESS TERMINATED -ABNORMALLY-" in line or "ddikick.x: application process 0 quit
unexpectedly" in line:
            terminated = "Abnormally"

    return terminated

def wallTime(outFileName):
    wallClockTime = "Fatal Termination"
    outputData = getOutFile(outFileName)

    cpuTimeStart = 0
    for data_num,line in enumerate(outputData):
        if "CPU timing information for all processes" in line:
            cpuTimeStart = data_num + 2
            wallClockTime = 0
            break
        elif "ddikick.x: application process 0 quit unexpectedly" in line:
            break

    terminated = testOutputFailure(outFileName)
    if terminated == "Normally":
        for i in range(0,100,1):
            line = outputData[cpuTimeStart+i]
            if "-----" in line:
                break
            wallClockTime += float(line.split()[5])

    return wallClockTime

def xyzCoordinates(f, calcType, coord_start, atomStarting, atomTotal):
    minEnergy = 0
    outFile = open_file(f, "r")
    outputData = outFile.readlines()
    outFile.close()

    n_water = (atomTotal-atomStarting)/3
    xyzFile = open_file(f[:-4]+".xyz", "w")

    iteration = 0
    for data_num,line in enumerate(outputData):
        if "FINAL" in line:
            finalEnergy = float(line.split()[4])
            if finalEnergy < minEnergy:
                minEnergy = finalEnergy
            iteration +=1

    wallClockTime = wallTime(f)

    xyzFile.write(str(atomTotal-1)+"\t"+f[:-4)+"\t\t"+str(minEnergy)+"\n")
    xyzFile.write("Wallclock time:\t"+str(wallClockTime)+"\n")

```

```

waterStart = coord_start+atomStarting+2
if calcType == "EFP":
    for k in range(coord_start,coord_start+atomStarting-1,1):
        xyzFile.write(" "+str(outputData[k]))
    j = 0
    for k in range(waterStart,waterStart+4*n_water,1):
        if j%4 != 0:
            xyzFile.write(" "+str(outputData[k]))
            j += 1
        else:
            j += 1
    else:
        for k in range(coord_start,coord_start+atomTotal-1,1):
            xyzFile.write(" "+str(outputData[k]))

xyzFile.close()
return

def clearall():
    """clear all globals"""
    for uniquevar in [var for var in globals().copy() if var[0] != "_" and var != 'clearall']:
        del globals()[uniquevar]

```

## B. ENERGY DATA FOR OPTIMIZED STRUCTURES

Table 25 lists the energies obtained after ZPE correction with the corresponding basis set for methylammonium water system, where the Frame number corresponds to the coordinates in Appendix B.

Table 25. Energy values obtained after ZPE correction for CH <sub>3</sub> NH <sub>3</sub> <sup>+</sup> .		
Frame	ID	B3LYP/6-311++G** (kJ/mol)
1	H <sub>2</sub> O	-200591
2	CH <sub>3</sub> NH <sub>3</sub> <sup>+</sup>	-252324
3	1A	-452988
4	2A	-653640
5	3A	-854283
6	3B	-854278
7	3C	-854265
8	4A	-1054921
9	4B	-1054912
10	4C	-1054910
11	4D	-1054909
12	5A	-1255555
13	5B	-1255555
14	5C	-1255549
15	5D	-1255548
16	6A	-1456184
17	6B	-1456179
18	6C	-1456177
19	6D	-1456175

Table 26 lists the energies obtained after ZPE correction with the corresponding basis set for methylcarbamic acid water system, where the Frame number corresponds to the coordinates in Appendix B.

**Table 26.** Energy values obtained after ZPE correction for methylcarbamic acid.

<b>Frame</b>	<b>ID</b>	<b>B3LYP/6-31+G* (kJ/mol)</b>	<b>B3LYP/6-311++G** (kJ/mol)</b>	<b>MP2/6-311++G** (kJ/mol)</b>
20	zM-1A	-946758.414	-947090.2058	-942266.6019
21	zM-2A	-1147279.513	-1147703.056	-1141893.684
22	zM-2B	-1147271.21	-1147684.575	-1141837.46
23	zM-2C	-1147269.075	-1147682.287	-1141836.324
24	zM-2D	-1147262.577	-1147674.457	-1141834.832
25	zM-3A	-1347802.37	-1348306.428	-1341477.765
26	zM-3B	-1347802.226	-1348305.767	-1341477.455
27	zM-3C	-1347800.225	-1348303.199	-1341477.891
28	zM-3D	-1347798.82	-1348302.447	-1341480.024
29	zM-4A	-1548334.68	-1548929.271	-1541116.601
30	zM-4B	-1548334.327	-1548929.065	-1541124.175
31	zM-4C	-1548333.331	-1548926.623	-1541115.634
32	zM-4D	-1548333.105	-1548927.836	-1541115.365
33	zM-5A	-1748872.983	-1749555.48	-1740759.603
34	zM-5B	-1748871.573	-1749557.334	-1740765.983
35	zM-5C	-1748870.296	-1749555.248	-1740762.286
36	zM-5D	-1748868.883	-1749552.587	-1740756.622
37	zM-6A	-1949404.615	-1950179.713	-1940401.842
38	zM-6B	-1949404.121	-1950181.821	-1940406.632
39	zM-6C	-1949401.03	-1950176.822	-1940400.112
40	zM-6D	-1949400.938	-1950176.36	-1940400.965
41	zM-7A	-2150815.992	-2150815.992	-2140046.738
42	zM-7B	-2150812.732	-2150812.732	-2140042.188
43	zM-7C	-2150810.818	-2150810.806	-2140043.714
44	zM-7D	-2150810.745	-2150810.745	-2140042.615
45	zM-8A	-2351436.642	-2351436.642	-2339678.355
46	zM-8B	-2351428.477	-2351428.477	-2339681.302
47	zM-8C	-2351426.378	-2351426.378	-2339679.291
48	zM-8D	-2351423.43	-2351423.43	-2339675.91

Table 27 lists the energies obtained after ZPE correction with the corresponding basis set for glycine water system, where the Frame number corresponds to the coordinates in Appendix B.

<b>Table 27.</b> Energy values obtained after ZPE correction for glycine.			
<b>Frame</b>	<b>ID</b>	<b>B3LYP/6-311++G** (kJ/mol)</b>	<b>Comments</b>
49	zG-7A	-2150805.938	
50	zG-7B	-2150803.638	
51	zG-7C	-2150799.134	
52	zG-7D	-2150798.882	
53	nG-7E	-2150798.108	Intramolecular H transfer
54	nG-7G	-2150779.103	Intermolecular H transfer
55	zG-8A	-2351444.188	
56	zG-8B	-2351442.722	
57	zG-8C	-2351438.15	
58	zG-8D	-2351434.794	

## C. COORDINATE DATA FOR OPTIMIZED STRUCTURES

Coordinates corresponding to Table 25-27.

3 ATOMS H2O				H 1.0 0.41861 -3.89528 0.88797	H 1.0 0.78531 -3.20832 -0.38570	
Frame: 1				O 8.0 -3.09431 0.00357 -1.01820	H 1.0 -0.75586 -3.24248 -0.14420	
O 8.0 -2.70434 1.44997 -0.23814				H 1.0 -3.51006 -0.28986 -1.83731	O 8.0 0.26740 1.91669 1.36243	
H 1.0 -2.81456 2.40527 -0.27701				H 1.0 -3.73677 0.58008 -0.58794	H 1.0 1.17858 1.98896 1.66899	
H 1.0 -3.59568 1.09954 -0.14431				17 ATOMS 3B (CH3NH3_3W)		
8 ATOMS CH3NH3				Frame: 6		
Frame: 2				N 7.0 -0.51204 -0.07336 -0.08837	20 ATOMS 4B (CH3NH3_4W)	
N 7.0 -0.34183 0.04223 -0.27385				H 1.0 -0.71709 0.87821 -0.49096	Frame: 9	
H 1.0 0.37500 0.62146 0.17520				H 1.0 0.24340 -1.96147 -0.70545	N 7.0 -0.55276 -0.35121 -0.07198	
H 1.0 -0.75278 -0.92400 -2.12085				H 1.0 -1.38242 -0.45383 0.33353	H 1.0 -0.21726 -1.08961 0.54744	
H 1.0 -1.22278 0.56566 -0.24156				H 1.0 0.16808 0.05206 0.66251	H 1.0 -0.45871 0.25330 -2.09021	
H 1.0 -0.46325 -0.79625 0.30341				C 6.0 0.01907 -0.99196 -1.14614	H 1.0 -0.22045 0.57046 0.31510	
C 6.0 0.04174 -0.31385 -1.69664				H 1.0 -0.74058 -1.10293 -1.91713	H 1.0 -1.59458 -0.33983 -0.02473	
H 1.0 0.15860 0.60989 -2.25936				H 1.0 0.91992 -0.55699 -1.57474	C 6.0 -0.07569 -0.55804 -1.47437	
H 1.0 0.97789 -0.86729 -1.66876				O 8.0 -2.87246 -1.16507 1.02324	H 1.0 1.01279 -0.54982 -1.48993	
11 ATOMS 1A (CH3NH3_1W)				H 1.0 -3.77681 -0.94535 0.76986	H 1.0 -0.44492 -1.51167 -1.84742	
Frame: 3				H 1.0 -2.94624 -1.80764 1.73880	O 8.0 0.06818 2.05967 0.98746	
N 7.0 -0.34016 0.04105 -0.27500				O 8.0 -0.96719 2.36172 -1.22168	H 1.0 -0.61315 2.41640 1.59930	
H 1.0 0.37981 0.61342 0.17288				H 1.0 -1.43681 3.11629 -0.85136	H 1.0 0.76628 2.71569 0.90067	
H 1.0 -0.75016 -0.92160 -2.11703				H 1.0 -0.58810 2.65264 -2.08015	O 8.0 -3.33814 -0.15853 0.05598	
H 1.0 -1.24340 0.57843 -0.24518				O 8.0 0.14747 3.06144 -3.62266	H 1.0 -3.80852 0.55441 0.52484	
H 1.0 -0.45494 -0.79723 0.30014				H 1.0 0.92795 3.61783 -3.72571	H 1.0 -4.01024 -0.73354 -0.32490	
C 6.0 0.04410 -0.31445 -1.68801				H 1.0 -0.33295 3.11563 -4.45675	O 8.0 -2.00977 2.75343 2.59113	
H 1.0 0.15701 0.60697 -2.25532				17 ATOMS 3C (CH3NH3_3W)		
H 1.0 0.98033 -0.86874 -1.67489				Frame: 7		
O 8.0 -2.70757 1.45197 -0.23797				N 7.0 0.10955 -0.72600 -0.27765	H 1.0 -2.93070 2.61334 2.31834	
H 1.0 -2.80883 2.41112 -0.27780				H 1.0 -0.35656 0.00316 0.28758	H 1.0 -2.02315 3.19969 3.44375	
H 1.0 -3.59819 1.09169 -0.14370				H 1.0 1.43958 -0.91955 -1.91555	O 8.0 -4.60986 2.01417 1.48657	
14 ATOMS 2A (CH3NH3_2W)				H 1.0 -0.66402 -1.29287 -0.71492	H 1.0 -5.28574 1.75837 2.12814	
Frame: 4				H 1.0 0.64157 -1.32799 0.35218	H 1.0 -5.02050 2.69468 0.93676	
N 7.0 0.27037 0.52919 -0.19844				C 6.0 0.98525 -0.11991 -1.33371	20 ATOMS 4C (CH3NH3_4W)	
H 1.0 1.06186 0.95402 -0.72686				H 1.0 0.35840 0.49662 -1.97481	Frame: 10	
H 1.0 -1.46502 -0.64718 -0.53568				H 1.0 1.75851 0.48362 -0.86160	N 7.0 -0.22880 0.05319 -0.56186	
H 1.0 -0.22662 1.28494 0.32011				O 8.0 -1.56134 1.32659 1.05482	H 1.0 0.57476 0.32101 0.03417	
H 1.0 0.67111 -0.10754 0.49252				H 1.0 -1.23619 2.19217 1.33584	H 1.0 -1.02878 0.40980 -2.48684	
C 6.0 -0.66371 -0.19880 -1.11982				H 1.0 -2.16037 1.04118 1.75782	H 1.0 -1.11148 0.25238 -0.05925	
H 1.0 -1.07902 0.51727 -1.82604				O 8.0 -2.10593 -1.76300 -1.38962	H 1.0 -0.24942 -0.97760 -0.66317	
H 1.0 -0.11086 -0.97075 -1.65179				H 1.0 -2.53135 -2.53428 -1.77718	C 6.0 -0.17985 0.73632 -1.88895	
O 8.0 2.41063 1.61639 -1.66302				H 1.0 -2.55361 -0.95371 -1.70202	H 1.0 -0.22412 1.81392 -1.74042	
H 1.0 3.31982 1.71586 -1.35562				O 8.0 -2.52831 0.91556 -1.71104	H 1.0 0.74862 0.47254 -2.39217	
H 1.0 2.40398 1.92887 -2.57564				H 1.0 -3.04079 1.48038 -2.30022	O 8.0 -5.06280 0.51710 -0.70695	
O 8.0 -1.11425 2.54516 1.17544				H 1.0 -2.53359 1.33758 -0.83953	H 1.0 -5.46934 1.39041 -0.67309	
H 1.0 -1.78012 2.43775 1.86528				20 ATOMS 4A (CH3NH3_4W)		
H 1.0 -0.99029 3.49612 1.07143				Frame: 8		
17 ATOMS 3A (CH3NH3_3W)				N 7.0 -0.77572 0.02913 -0.52493	H 1.0 -1.27579 -3.44417 -0.52058	
Frame: 5				H 1.0 -0.29040 -0.88506 -0.44683	H 1.0 -2.04932 -2.19490 0.02060	
N 7.0 -0.48095 -0.61464 -0.05286				H 1.0 -1.16919 1.55220 -1.94231	O 8.0 2.03730 0.77492 1.01401	
H 1.0 -0.28439 -0.01396 0.76863				H 1.0 -0.41329 0.66931 0.20230	H 1.0 2.42420 1.64645 1.15720	
H 1.0 0.30497 -1.03838 -1.97398				H 1.0 -1.76995 -0.18865 -0.32182	H 1.0 2.59765 0.15455 1.49474	
H 1.0 -1.44136 -0.40665 -0.38278				C 6.0 -0.61579 0.61646 -1.88857	20 ATOMS 4D (CH3NH3_4W)	
H 1.0 -0.45312 -1.60350 0.25662				H 1.0 0.43968 0.80419 -2.07716	Frame: 11	
C 6.0 0.51750 -0.37215 -1.13991				H 1.0 -1.00365 -0.08509 -2.62470	N 7.0 0.27452 0.13847 0.09649	
H 1.0 0.44464 0.66334 -1.46743				O 8.0 -2.56316 -3.89935 0.21898	H 1.0 1.00438 -0.53130 -0.22116	
H 1.0 1.51710 -0.56842 -0.75643				H 1.0 -2.91760 -4.52197 -0.42940	H 1.0 -1.04844 1.62281 -0.63544	
O 8.0 0.13679 1.04199 2.18753				H 1.0 -2.68396 -4.32493 1.07795	H 1.0 0.73323 0.72574 0.80330	
H 1.0 0.15805 0.76089 3.10988				O 8.0 -3.29333 -1.10608 -0.04864	H 1.0 -0.43615 -0.38971 0.63761	
H 1.0 0.39045 1.97239 2.18905				H 1.0 -3.24106 -2.06899 0.07642	C 6.0 -0.31207 0.92416 -1.02860	
O 8.0 -0.36529 -3.35928 0.72005				H 1.0 -4.21436 -0.84984 0.06402	H 1.0 0.48098 1.46807 -1.53879	
H 1.0 -1.11886 -3.94871 0.84141				O 8.0 -0.00278 -2.65773 -0.33500	H 1.0 -0.79288 0.23794 -1.72356	

O 8.0 2.48107 -1.49959 -0.36119  
H 1.0 2.84184 -2.05235 -1.06215  
H 1.0 2.99532 -1.67343 0.44524  
O 8.0 3.61199 -1.58442 2.30462  
H 1.0 4.52740 -1.29311 2.41023  
H 1.0 3.55070 -2.41770 2.79040  
O 8.0 -1.04503 -1.01348 2.20988  
H 1.0 -0.40891 -0.70798 2.87399  
H 1.0 -1.79845 -1.40221 2.66548  
O 8.0 1.44725 0.30765 2.78062  
H 1.0 1.63296 0.97965 3.44676  
H 1.0 2.22841 -0.27438 2.75855

#### 23 ATOMS 5A (CH3NH3\_5W)

Frame: 12

N 7.0 0.16723 0.47766 0.75473  
H 1.0 0.53785 -0.48680 0.76672  
H 1.0 0.32357 2.29399 1.82384  
H 1.0 -0.87061 0.44847 0.80203  
H 1.0 0.37019 0.91246 -0.16741  
C 6.0 0.73781 1.28874 1.87011  
H 1.0 0.47923 0.82594 2.82090  
H 1.0 1.82068 1.33054 1.76649  
O 8.0 -3.91354 1.24059 -4.20134  
H 1.0 -4.75968 0.78000 -4.21917  
H 1.0 -3.63411 1.33190 -5.11885  
O 8.0 -2.61407 2.23054 -1.99597  
H 1.0 -3.07338 1.85624 -2.78092  
H 1.0 -2.91367 3.14531 -1.93048  
O 8.0 1.30731 -2.15565 0.93703  
H 1.0 2.04078 -2.50746 0.41969  
H 1.0 0.99606 -2.88667 1.48299  
O 8.0 0.16106 1.81782 -1.68385  
H 1.0 0.74965 2.15021 -2.36826  
H 1.0 -0.75542 2.02527 -1.95187  
O 8.0 -2.57899 0.79854 0.43901  
H 1.0 -3.42138 0.59513 0.85662  
H 1.0 -2.76653 1.27164 -0.39514

#### 23 ATOMS 5B (CH3NH3\_5W)

Frame: 13

N 7.0 0.16734 0.25205 0.35707  
H 1.0 0.59715 -0.66451 0.14968  
H 1.0 0.31409 1.84823 1.73199  
H 1.0 -0.85372 0.12592 0.50477  
H 1.0 0.23241 0.86250 -0.48174  
C 6.0 0.80479 0.89537 1.54332  
H 1.0 0.69425 0.24419 2.40853  
H 1.0 1.86166 1.05977 1.34036  
O 8.0 -2.61129 0.39389 0.41555  
H 1.0 -3.38148 0.02340 0.85725  
H 1.0 -2.92986 1.01175 -0.27122  
O 8.0 -0.21908 2.03603 -1.74177  
H 1.0 -1.17326 2.21199 -1.85737  
H 1.0 0.24794 2.46573 -2.46485  
O 8.0 1.41746 -2.30071 -0.07237  
H 1.0 2.30385 -2.47630 -0.40819  
H 1.0 0.99924 -3.16334 0.03019  
O 8.0 -4.14854 4.66670 -1.09234  
H 1.0 -3.87627 5.52190 -1.44303  
H 1.0 -4.92755 4.83075 -0.54944  
O 8.0 -3.03647 2.22867 -1.68659  
H 1.0 -3.43089 3.10226 -1.46614  
H 1.0 -3.51731 1.91159 -2.46050

#### 23 ATOMS 5C (CH3NH3\_5W)

Frame: 14

N 7.0 0.04879 0.14502 -0.11356  
H 1.0 0.75360 -0.59210 -0.30374

H 1.0 -0.25901 1.89173 1.04103  
H 1.0 -0.81650 -0.34214 0.18482  
H 1.0 -0.15502 0.57093 -1.03781  
C 6.0 0.51824 1.14508 0.88716  
H 1.0 0.73359 0.64000 1.82740  
H 1.0 1.42099 1.62671 0.51526  
O 8.0 -2.69597 -3.08818 -2.08643  
H 1.0 -3.51903 -2.89126 -2.55364  
H 1.0 -2.73084 -4.03456 -1.89279  
O 8.0 -0.28381 -2.13258 -3.36937  
H 1.0 -0.11158 -2.53741 -4.22787  
H 1.0 -1.10799 -2.53337 -3.03343  
O 8.0 1.52810 -1.98293 -1.15525  
H 1.0 2.36491 -2.45583 -1.10816  
H 1.0 1.11146 -2.19496 -2.00844  
O 8.0 -2.28840 -1.38832 0.29558  
H 1.0 -2.52264 -1.99347 -0.42695  
H 1.0 -2.94355 -1.50557 0.99111  
O 8.0 -0.50415 0.65969 -2.80316  
H 1.0 -0.52869 1.34671 -3.47640  
H 1.0 -0.41141 -0.19809 -3.25338

#### 23 ATOMS 5D (CH3NH3\_5W)

Frame: 15

N 7.0 -0.36058 0.01029 -0.34050  
H 1.0 -0.18030 0.91626 0.14971  
H 1.0 0.53742 -1.28650 -1.75852  
H 1.0 -1.24327 0.09733 -0.84455  
H 1.0 -0.47928 -0.72422 0.40635  
C 6.0 0.76152 -0.34352 -1.26304  
H 1.0 0.89032 0.44564 -2.00187  
H 1.0 1.66916 -0.44818 -0.67220  
O 8.0 -1.17075 3.22319 5.56018  
H 1.0 -1.62723 4.07111 5.53110  
H 1.0 -1.49456 2.76531 6.34381  
O 8.0 0.69636 2.22211 3.82767  
H 1.0 0.01237 2.59054 4.43239  
H 1.0 1.52759 2.63495 4.09179  
O 8.0 0.67043 -0.62708 3.86700  
H 1.0 0.72724 0.34371 3.94199  
H 1.0 1.32521 -1.00382 4.46285  
O 8.0 0.16465 2.33378 1.07455  
H 1.0 0.33867 2.34523 2.03914  
H 1.0 0.16319 3.24937 0.77733  
O 8.0 -0.46116 -1.76754 1.69610  
H 1.0 -1.00232 -2.53451 1.90726  
H 1.0 -0.06231 -1.43895 2.53502

#### 26 ATOMS 6A (CH3NH3\_6W)

Frame: 16

N 7.0 -0.74520 0.51493 0.04196  
H 1.0 0.23726 0.83590 0.14958  
H 1.0 -2.69444 1.25128 -0.33829  
H 1.0 -0.98873 0.01664 0.92096  
H 1.0 -0.74057 -0.20112 -0.71022  
C 6.0 -1.68282 1.64200 -0.24234  
H 1.0 -1.64024 2.35503 0.57916  
H 1.0 -1.38832 2.13135 -1.16931  
O 8.0 2.07598 0.83337 0.40450  
H 1.0 2.35828 0.17754 -0.28196  
H 1.0 2.63012 1.61461 0.29188  
O 8.0 -0.94306 -1.11598 2.38681  
H 1.0 -0.03987 -0.91941 2.74205  
H 1.0 -1.55693 -1.05260 3.12792  
O 8.0 -0.35561 -1.78020 -1.62664  
H 1.0 -0.59820 -2.45690 -0.94563  
H 1.0 -0.78971 -2.03886 -2.44803  
O 8.0 1.55190 -0.32725 2.92214

H 1.0 1.96723 0.08076 2.13985  
H 1.0 2.24945 -0.57620 3.53641  
O 8.0 2.35997 -1.02393 -1.48936  
H 1.0 1.51171 -1.46454 -1.68234  
H 1.0 3.07043 -1.58238 -1.81952  
O 8.0 -1.02873 -3.22608 0.51621  
H 1.0 -0.98559 -4.15251 0.77253  
H 1.0 -1.01671 -2.68561 1.32765

#### 26 ATOMS 6B (CH3NH3\_6W)

Frame: 17

N 7.0 0.24705 0.52208 -0.24238  
H 1.0 1.25572 0.32262 -0.36283  
H 1.0 -1.20282 1.94916 -0.82033  
H 1.0 0.08441 0.63211 0.78147  
H 1.0 -0.29887 -0.29334 -0.58702  
C 6.0 -0.14100 1.76284 -0.97276  
H 1.0 0.43547 2.60135 -0.58614  
H 1.0 0.06059 1.63449 -2.03506  
O 8.0 3.06818 0.11352 -0.40621  
H 1.0 3.68114 0.26839 0.33297  
H 1.0 3.59113 -0.25221 -1.12696  
O 8.0 4.79753 0.63359 1.88865  
H 1.0 5.57490 1.17886 1.71088  
H 1.0 5.12967 -0.14718 2.35075  
O 8.0 -1.28273 -1.58510 -1.27645  
H 1.0 -2.06842 -1.48552 -1.85120  
H 1.0 -1.20601 -2.52360 -1.07801  
O 8.0 -3.49699 -1.22756 -2.90667  
H 1.0 -4.41387 -1.20692 -2.61117  
H 1.0 -3.52338 -1.39754 -3.85470  
O 8.0 2.55524 1.86225 3.30562  
H 1.0 3.37695 1.52511 2.91476  
H 1.0 2.79051 2.58298 3.89813  
O 8.0 0.08680 0.99713 2.48095  
H 1.0 0.92402 1.32052 2.87502  
H 1.0 -0.48111 0.71191 3.20287

#### 26 ATOMS 6C (CH3NH3\_6W)

Frame: 18

N 7.0 -0.45373 0.24281 -0.47435  
H 1.0 0.07540 0.18921 -1.36120  
H 1.0 -1.78426 1.53642 0.54327  
H 1.0 0.21081 0.14396 0.32616  
H 1.0 -1.09536 -0.56751 -0.48455  
C 6.0 -1.21566 1.52520 -0.38499  
H 1.0 -0.51133 2.35538 -0.39110  
H 1.0 -1.88796 1.59644 -1.23806  
O 8.0 1.27441 0.10269 1.67895  
H 1.0 1.28714 0.70745 2.44987  
H 1.0 1.97074 -0.54590 1.82334  
O 8.0 -2.15250 -1.97331 -1.26415  
H 1.0 -2.45199 -1.48679 -2.06566  
H 1.0 -2.91715 -2.42992 -0.89602  
O 8.0 0.38246 -2.85458 -2.11934  
H 1.0 0.65427 -3.74749 -2.35490  
H 1.0 -0.53947 -2.89729 -1.80786  
O 8.0 0.51248 -0.37327 -3.13865  
H 1.0 0.65067 -1.33290 -2.95900  
H 1.0 1.24078 -0.07212 -3.69321  
O 8.0 1.24791 1.84703 3.81788  
H 1.0 0.86991 1.65656 4.68373  
H 1.0 1.85913 2.58087 3.94654  
O 8.0 -2.27662 -0.24467 -3.33876  
H 1.0 -1.34039 -0.21117 -3.60994  
H 1.0 -2.81238 -0.12602 -4.13004

#### 26 ATOMS 6D (CH3NH3\_6W)

Frame: 19



N 7.0 -0.24149 0.02834 0.02967  
H 1.0 0.67494 0.05568 -0.47086  
H 1.0 -1.96298 1.22211 0.34028  
H 1.0 0.01614 -0.06469 1.03759  
H 1.0 -0.76630 -0.80757 -0.26794  
C 6.0 -1.02660 1.27101 -0.21340  
H 1.0 -0.44586 2.12727 0.12471  
H 1.0 -1.23623 1.36561 -1.27763  
O 8.0 2.35657 0.16023 -0.86070  
H 1.0 2.97807 0.11492 -0.11828  
H 1.0 2.87575 0.22993 -1.68326  
O 8.0 3.71518 -0.00690 1.74750  
H 1.0 4.25298 -0.78776 1.92958  
H 1.0 4.24948 0.74538 2.03149  
O 8.0 3.85922 0.37571 -3.21038  
H 1.0 4.15258 -0.33915 -3.78549  
H 1.0 4.09740 1.19575 -3.65611  
O 8.0 -1.87380 -2.19169 -0.87284  
H 1.0 -1.74916 -2.71992 -1.66927  
H 1.0 -2.60040 -2.60592 -0.39373  
O 8.0 0.87905 -0.06995 2.53644  
H 1.0 1.84676 -0.06056 2.48153  
H 1.0 0.63184 -0.11821 3.47860  
O 8.0 0.17568 -0.18655 5.24257  
H 1.0 -0.00011 -0.97478 5.76736  
H 1.0 -0.02254 0.56569 5.81063

#### 13 ATOMS zM-1A

Frame: 20

N 7.0 0.04515 -0.55268 0.08539  
C 6.0 -0.64505 -0.68016 1.37644  
C 6.0 1.66717 1.68919 -0.01442  
O 8.0 2.21093 1.41673 0.98257  
O 8.0 1.20017 2.07316 -1.02153  
H 1.0 0.62352 -1.37413 -0.08556  
H 1.0 -0.62351 -0.49770 -0.68347  
H 1.0 -1.26456 -1.58620 1.47725  
H 1.0 0.09626 -0.67935 2.18352  
H 1.0 -1.29426 0.19027 1.52139  
O 8.0 -1.17099 0.74768 -2.35668  
H 1.0 -1.47301 1.05661 -3.22332  
H 1.0 -0.42452 1.31844 -2.10529

#### 16 ATOMS zM-2A

Frame: 21

N 7.0 -0.15970 -0.44099 0.35360  
C 6.0 -0.81494 -0.36457 1.66757  
C 6.0 1.69920 1.49713 0.16371  
O 8.0 1.97229 1.41495 1.29643  
O 8.0 1.55205 1.70191 -0.98474  
H 1.0 0.35355 -1.31889 0.27782  
H 1.0 -0.86287 -0.45882 -0.39101  
H 1.0 -1.50103 -1.20049 1.88125  
H 1.0 -0.05391 -0.33302 2.45503  
H 1.0 -1.38944 0.56667 1.72669  
O 8.0 -0.44285 1.61418 -3.17365  
H 1.0 0.22859 1.69245 -2.47143  
H 1.0 0.04389 1.38491 -3.97977  
O 8.0 -2.07652 -0.35598 -2.04718  
H 1.0 -3.01533 -0.11981 -2.06606  
H 1.0 -1.60580 0.35133 -2.54280

#### 16 ATOMS zM-2B

Frame: 22

N 7.0 0.36646 -0.08870 0.36224  
C 6.0 -0.97428 -0.36436 0.91435  
C 6.0 1.13279 1.42572 0.93924  
O 8.0 2.22802 1.48355 0.40295  
O 8.0 0.39253 1.98617 1.70367

H 1.0 1.03389 -0.82374 0.60073  
H 1.0 0.34729 -0.02575 -0.67235  
H 1.0 -1.36502 -1.31373 0.53181  
H 1.0 -0.91735 -0.39364 2.00413  
H 1.0 -1.64028 0.45018 0.62499  
O 8.0 0.37732 0.29274 -2.49458  
H 1.0 1.18909 0.84765 -2.61768  
H 1.0 0.38753 -0.36958 -3.20116  
O 8.0 2.67622 1.75606 -2.33232  
H 1.0 2.68195 1.73360 -1.34850  
H 1.0 2.73031 2.69134 -2.57953

#### 16 ATOMS zM-2C

Frame: 23

N 7.0 -0.57337 0.21437 -0.17147  
C 6.0 -1.50584 0.56588 0.92093  
C 6.0 1.08856 0.75354 0.12649  
O 8.0 1.44473 0.30830 1.20644  
O 8.0 1.44505 1.42435 -0.80998  
H 1.0 -0.53807 -0.81503 -0.28898  
H 1.0 -0.84135 0.64152 -1.05870  
H 1.0 -2.50936 0.17439 0.72280  
H 1.0 -1.11921 0.13519 1.84622  
H 1.0 -1.55065 1.65332 1.02358  
O 8.0 -0.24258 -2.64036 -0.16919  
H 1.0 0.51569 -2.73382 0.46213  
H 1.0 -0.86162 -3.35655 0.03397  
O 8.0 1.81073 -2.41732 1.62049  
H 1.0 2.71870 -2.68270 1.41069  
H 1.0 1.80431 -1.43305 1.60751

#### 16 ATOMS zM-2D

Frame: 24

N 7.0 0.18787 0.19509 0.71975  
C 6.0 -0.50488 -0.36402 1.91140  
C 6.0 1.76850 0.62758 1.07018  
O 8.0 2.52970 -0.30966 0.86581  
O 8.0 1.82138 1.78356 1.47097  
H 1.0 0.22308 -0.51352 -0.02935  
H 1.0 -0.29759 1.04730 0.39969  
H 1.0 -1.52727 -0.64835 1.64795  
H 1.0 0.04439 -1.24296 2.25457  
H 1.0 -0.52465 0.39657 2.69438  
O 8.0 -0.55189 2.93651 0.69730  
H 1.0 0.36220 2.88640 1.05886  
H 1.0 -0.60122 3.73807 0.15649  
O 8.0 1.11576 -2.08498 -0.69046  
H 1.0 1.43094 -2.41522 -1.54441  
H 1.0 1.89253 -1.70272 -0.22231

#### 19 ATOMS zM-3A

Frame: 25

N 7.0 -0.30344 -0.17206 1.24172  
C 6.0 -1.71550 -0.12563 1.67938  
C 6.0 0.33637 1.29011 0.69271  
O 8.0 1.55909 1.23382 0.82437  
O 8.0 -0.52976 2.03169 0.28441  
H 1.0 0.32391 -0.49003 1.98244  
H 1.0 -0.18061 -0.82943 0.43968  
H 1.0 -2.08432 -1.13597 1.88122  
H 1.0 -1.79653 0.48910 2.57859  
H 1.0 -2.29788 0.34059 0.88355  
O 8.0 3.28849 1.14850 -1.31101  
H 1.0 3.18427 1.93685 -1.86426  
H 1.0 2.73183 1.30320 -0.51053  
O 8.0 2.41878 -1.25788 -2.24661  
H 1.0 2.76274 -0.36869 -1.96466  
H 1.0 3.16601 -1.86966 -2.16949  
O 8.0 0.12280 -1.90952 -0.94514

H 1.0 -0.53728 -2.00710 -1.64768  
H 1.0 0.97731 -1.69934 -1.41152

#### 19 ATOMS zM-3B

Frame: 26

N 7.0 -0.12699 -0.13171 -0.03946  
C 6.0 -1.35629 0.27265 -0.76015  
C 6.0 1.28416 0.63342 -0.53297  
O 8.0 1.04919 1.81296 -0.78852  
O 8.0 2.21265 -0.14854 -0.49506  
H 1.0 -0.21216 0.09957 0.97563  
H 1.0 0.05955 -1.13246 -0.12062  
H 1.0 -2.23488 -0.18511 -0.29602  
H 1.0 -1.42848 1.35954 -0.71084  
H 1.0 -1.28080 -0.03465 -1.80605  
O 8.0 2.25245 3.74172 0.76218  
H 1.0 1.90032 3.11666 0.08389  
H 1.0 3.21318 3.61513 0.76486  
O 8.0 -0.38134 0.69743 2.64493  
H 1.0 0.04139 1.58878 2.78274  
H 1.0 -0.07363 0.14918 3.38213  
O 8.0 0.85307 3.09129 3.01020  
H 1.0 1.40589 3.35642 2.22746  
H 1.0 0.29955 3.85884 3.21724

#### 19 ATOMS zM-3C

Frame: 27

N 7.0 -0.06440 0.21481 0.06633  
C 6.0 -1.10034 0.09935 1.11828  
C 6.0 0.69521 1.67162 -0.01886  
O 8.0 0.31373 2.45283 0.82720  
O 8.0 1.51610 1.66172 -0.94971  
H 1.0 0.67421 -0.50305 0.18859  
H 1.0 -0.46264 0.02719 -0.87302  
H 1.0 -1.51749 -0.91176 1.10467  
H 1.0 -0.64905 0.31247 2.08789  
H 1.0 -1.88365 0.83485 0.93064  
O 8.0 2.06899 -0.37435 -2.56238  
H 1.0 1.95241 0.45983 -2.01738  
H 1.0 2.75634 -0.19658 -3.22186  
O 8.0 1.98731 -1.80715 -0.16513  
H 1.0 2.23057 -1.50870 -1.07014  
H 1.0 2.81533 -1.87189 0.33344  
O 8.0 -0.71983 -0.53126 -2.65022  
H 1.0 0.22845 -0.57285 -2.90768  
H 1.0 -1.15130 0.04671 -3.29725

#### 19 ATOMS zM-3D

Frame: 28

N 7.0 -0.12956 0.09192 -0.06589  
C 6.0 -0.47304 0.10506 1.37983  
C 6.0 0.68189 1.43273 -0.53053  
O 8.0 1.73778 1.54758 0.08983  
O 8.0 0.07917 2.05812 -1.39755  
H 1.0 0.48721 -0.72186 -0.26663  
H 1.0 -0.98138 0.02331 -0.64660  
H 1.0 -0.95388 -0.83845 1.65272  
H 1.0 0.44558 0.23924 1.95273  
H 1.0 -1.15598 0.93494 1.57538  
O 8.0 -2.32681 0.78914 -1.75080  
H 1.0 -2.66352 0.54347 -2.62490  
H 1.0 -1.62250 1.46307 -1.89152  
O 8.0 3.80981 -0.27425 0.04056  
H 1.0 4.48141 -0.01213 -0.60677  
H 1.0 3.19388 0.49177 0.12084  
O 8.0 1.77505 -1.97207 -0.56404  
H 1.0 2.62804 -1.49710 -0.38528  
H 1.0 1.85689 -2.85094 -0.16607

#### 22 ATOMS zM-4A

```

Frame: 29
N 7.0 -0.03916 -0.65806 0.04772
C 6.0 -1.16411 -0.65067 1.01377
C 6.0 1.23789 0.27610 0.42984
O 8.0 0.90110 1.26921 1.07031
O 8.0 2.26325 -0.19935 -0.04803
H 1.0 0.33423 -1.59700 -0.10682
H 1.0 -0.35828 -0.29844 -0.88814
H 1.0 -2.01118 -1.20337 0.59757
H 1.0 -0.84268 -1.10755 1.95230
H 1.0 -1.43870 0.38818 1.19844
O 8.0 -0.79569 0.32386 -2.44087
H 1.0 -0.01880 0.76045 -2.89623
H 1.0 -1.52237 0.96092 -2.51233
O 8.0 1.27678 1.60942 -3.56507
H 1.0 1.67839 1.29469 -4.38817
H 1.0 2.03226 1.76615 -2.92623
O 8.0 2.47492 3.50785 0.37834
H 1.0 3.14465 3.72537 1.04363
H 1.0 1.91650 2.80138 0.77481
O 8.0 3.26523 1.88914 -1.77270
H 1.0 3.08583 2.61376 -1.12927
H 1.0 3.19538 1.08034 -1.22650
22 ATOMS zM-4B
Frame: 30
N 7.0 0.27341 1.06681 0.69661
C 6.0 -0.71870 1.33894 1.77242
C 6.0 1.13112 2.36802 0.31814
O 8.0 0.41868 3.27781 -0.11153
O 8.0 2.33376 2.23037 0.53547
H 1.0 0.90288 0.29206 0.98712
H 1.0 -0.24106 0.76847 -0.15807
H 1.0 -1.29039 0.42955 1.97450
H 1.0 -0.18800 1.64497 2.67672
H 1.0 -1.38084 2.13934 1.44065
O 8.0 -1.45970 3.16042 -2.11693
H 1.0 -0.81035 3.36027 -1.40034
H 1.0 -1.11463 3.58490 -2.91659
O 8.0 -1.23606 0.49448 -1.65914
H 1.0 -2.06547 0.00080 -1.73510
H 1.0 -1.41886 1.41313 -1.98796
O 8.0 4.12755 0.34819 1.45199
H 1.0 3.60313 1.10264 1.09365
H 1.0 4.66267 0.70465 2.17625
O 8.0 1.85667 -1.10297 1.71221
H 1.0 1.92963 -1.94994 1.24772
H 1.0 2.77173 -0.71943 1.72634
22 ATOMS zM-4C
Frame: 31
N 7.0 0.12864 -0.50023 0.54042
C 6.0 0.85566 -1.09165 1.69032
C 6.0 0.78044 0.84849 -0.08618
O 8.0 1.15044 1.62042 0.80085
O 8.0 0.70480 0.84337 -1.30910
H 1.0 0.02013 -1.15962 -0.23308
H 1.0 -0.83510 -0.18272 0.83988
H 1.0 0.27688 -1.92115 2.10574
H 1.0 1.83545 -1.44484 1.36013
H 1.0 0.98516 -0.31072 2.43986
O 8.0 -3.40891 0.73960 -1.20817
H 1.0 -2.80801 1.45140 -1.55669
H 1.0 -4.30479 0.95404 -1.50616
O 8.0 -2.28023 0.63103 1.28613
H 1.0 -2.85629 0.61813 0.48149
H 1.0 -2.05246 1.57455 1.41240
O 8.0 -1.54804 2.64669 -1.79646
H 1.0 -1.42178 3.07291 -0.92407
H 1.0 -0.71990 2.14700 -1.94209
O 8.0 -1.04986 3.26712 1.01166
H 1.0 -0.13451 2.90127 1.05565
H 1.0 -1.03687 4.12408 1.46410
22 ATOMS zM-4D
Frame: 32
N 7.0 -0.97663 -0.18334 0.12983
C 6.0 -1.83374 -0.76705 1.19107
C 6.0 0.10317 0.91938 0.65072
O 8.0 0.60060 0.59793 1.72178
O 8.0 0.20741 1.82591 -0.17664
H 1.0 -0.40020 -0.93883 -0.31910
H 1.0 -1.51982 0.27125 -0.60657
H 1.0 -2.43723 -1.58011 0.77750
H 1.0 -1.17717 -1.14064 1.97743
H 1.0 -2.48395 0.00843 1.60250
O 8.0 3.45305 0.60276 1.10776
H 1.0 3.40193 1.46461 0.63195
H 1.0 2.61261 0.56640 1.60572
O 8.0 3.22544 -1.35474 -0.74925
H 1.0 3.36257 -0.64910 -0.05254
H 1.0 3.69481 -1.04553 -1.53831
O 8.0 0.66281 -2.11190 -1.02290
H 1.0 1.61985 -1.82487 -0.95930
H 1.0 0.65533 -3.04684 -0.76931
O 8.0 2.78891 2.89991 -0.34058
H 1.0 2.86757 3.78376 0.04753
H 1.0 1.82937 2.67754 -0.32750
25 ATOMS zM-5A
Frame: 33
N 7.0 -0.47647 -0.17540 0.25669
C 6.0 -0.42804 -0.93158 1.53315
C 6.0 0.76235 0.78079 0.00511
O 8.0 0.42674 1.93845 -0.26640
O 8.0 1.83719 0.17344 0.06318
H 1.0 -0.45718 -0.82603 -0.56573
H 1.0 -1.34167 0.39239 0.14708
H 1.0 -1.27130 -1.62586 1.57258
H 1.0 0.51456 -1.47855 1.57574
H 1.0 -0.48742 -0.23066 2.36929
O 8.0 0.84286 1.99258 -3.12148
H 1.0 0.84292 2.29492 -2.18956
H 1.0 1.48960 1.25578 -3.13193
O 8.0 -0.33308 -1.48956 -2.22258
H 1.0 0.59271 -1.35640 -2.51157
H 1.0 -0.86420 -0.85936 -2.75653
O 8.0 -2.45115 1.64088 -0.66837
H 1.0 -2.36394 1.40484 -1.61766
H 1.0 -1.73908 2.29191 -0.52627
O 8.0 2.33914 -0.41016 -2.57398
H 1.0 3.20152 -0.71924 -2.89045
H 1.0 2.42654 -0.24371 -1.60354
O 8.0 -1.58247 0.82931 -3.29270
H 1.0 -0.71407 1.32646 -3.33881
H 1.0 -2.04939 0.99631 -4.12561
25 ATOMS zM-5B
Frame: 34
N 7.0 -0.25474 -0.35104 0.33784
C 6.0 -0.38192 -1.15380 1.58145
C 6.0 0.54911 1.00571 0.48372
O 8.0 1.40886 0.95535 1.36678
O 8.0 0.20762 1.85121 -0.34829
H 1.0 0.29093 -0.89524 -0.37063
H 1.0 -1.19335 -0.15956 -0.06553
H 1.0 -0.84741 -2.11193 1.33646
H 1.0 0.61134 -1.30707 2.00198
H 1.0 -1.00287 -0.61035 2.29636
O 8.0 1.43296 -1.77451 -1.40642
H 1.0 2.36597 -1.44045 -1.30746
H 1.0 1.26308 -1.78842 -2.36026
O 8.0 3.95098 -0.77562 -1.15676
H 1.0 3.98078 0.06599 -0.62722
H 1.0 4.63275 -1.35155 -0.78002
O 8.0 3.91704 1.48253 0.37478
H 1.0 3.89653 2.32200 -0.10904
H 1.0 3.04856 1.42534 0.84301
O 8.0 -2.05777 2.45416 -1.78656
H 1.0 -2.37328 3.33469 -1.53410
H 1.0 -1.18339 2.33742 -1.34326
O 8.0 -2.91176 0.16582 -0.61759
H 1.0 -2.80153 1.02835 -1.09583
H 1.0 -3.42835 -0.40868 -1.20178
25 ATOMS zM-5C
Frame: 35
N 7.0 0.15616 -0.53946 0.13562
C 6.0 0.80034 -1.38333 1.17135
C 6.0 1.10970 0.52137 -0.56637
O 8.0 2.12643 0.77214 0.05489
O 8.0 0.59550 0.93945 -1.62778
H 1.0 -0.31357 -1.12927 -0.58223
H 1.0 -0.60110 0.03493 0.56430
H 1.0 0.04604 -2.04478 1.60564
H 1.0 1.59612 -1.97183 0.71152
H 1.0 1.23412 -0.73805 1.93484
O 8.0 -2.42736 2.79433 -0.97343
H 1.0 -3.13062 3.45100 -1.08832
H 1.0 -1.57805 3.21848 -1.27916
O 8.0 -1.81413 1.21601 1.22622
H 1.0 -1.59806 1.77725 1.98556
H 1.0 -2.11512 1.82592 0.51021
O 8.0 -1.97002 0.42941 -2.67284
H 1.0 -1.02632 0.62760 -2.48310
H 1.0 -2.44707 1.14188 -2.20545
O 8.0 0.00703 3.58631 -1.85334
H 1.0 0.62720 4.17903 -1.40366
H 1.0 0.43138 2.69138 -1.85415
O 8.0 -1.59220 -2.00232 -1.55623
H 1.0 -1.45847 -2.68153 -2.23372
H 1.0 -1.93573 -1.19844 -2.03007
25 ATOMS zM-5D
Frame: 36
N 7.0 -0.22663 -0.26237 0.34252
C 6.0 0.49805 -1.22117 -0.52691
C 6.0 0.58513 1.05404 0.73929
O 8.0 0.10522 1.57220 1.76966
O 8.0 1.47193 1.33877 -0.04382
H 1.0 -1.10239 0.07906 -0.13472
H 1.0 -0.58084 -0.71259 1.19368
H 1.0 -0.17399 -2.04148 -0.79260
H 1.0 0.82624 -0.69177 -1.42092
H 1.0 1.37259 -1.60427 0.00292
O 8.0 -2.71437 0.52195 -0.61577
H 1.0 -2.90651 1.43701 -0.31332
H 1.0 -3.13046 -0.06571 0.04840
O 8.0 -3.03627 3.04676 0.67607
H 1.0 -3.73996 3.70217 0.55447
H 1.0 -2.18015 3.55436 0.73686
O 8.0 -2.47784 1.32677 2.92588
H 1.0 -2.87896 1.98330 2.31904
H 1.0 -1.51687 1.41097 2.74388
O 8.0 -0.59465 4.15491 1.09953

```

H 1.0 -0.01794 4.48946 0.39579  
H 1.0 -0.14738 3.34421 1.44424  
O 8.0 -2.80828 -1.05261 1.73990  
H 1.0 -3.34747 -1.71284 2.20054  
H 1.0 -2.79840 -0.23181 2.31109

28 ATOMS zM-6A

Frame: 37

N 7.0 0.39360 -0.33275 0.67431  
C 6.0 0.86330 -0.72026 2.02811  
C 6.0 0.60673 1.17923 0.26296  
O 8.0 1.52898 1.72311 0.87368  
O 8.0 -0.15668 1.52726 -0.64609  
H 1.0 0.90281 -0.86458 -0.06400  
H 1.0 -0.59866 -0.61926 0.51986  
H 1.0 0.74345 -1.80066 2.14251  
H 1.0 1.90851 -0.43188 2.13513  
H 1.0 0.26636 -0.19500 2.77600  
O 8.0 3.53865 2.25471 -0.96595  
H 1.0 2.86306 2.20791 -0.24795  
H 1.0 3.31006 3.03882 -1.48744  
O 8.0 -1.04379 -0.23415 -2.50503  
H 1.0 -0.81634 0.52454 -1.90500  
H 1.0 -1.37236 0.14670 -3.33334  
O 8.0 -1.94472 -1.67240 -0.17202  
H 1.0 -1.46509 -2.52179 -0.28634  
H 1.0 -1.99902 -1.29434 -1.07182  
O 8.0 0.03908 -3.66216 -0.51491  
H 1.0 0.64886 -3.12776 -1.07316  
H 1.0 0.03334 -4.56000 -0.87743  
O 8.0 3.56469 -0.11749 -2.34703  
H 1.0 4.46702 -0.46565 -2.29839  
H 1.0 3.57720 0.75402 -1.87043  
O 8.0 1.32501 -1.55857 -1.73505  
H 1.0 0.60917 -1.15017 -2.27172  
H 1.0 2.17526 -1.13044 -2.01552

28 ATOMS zM-6B

Frame: 38

N 7.0 0.03384 -0.07850 1.20632  
C 6.0 0.33551 -1.22578 2.10003  
C 6.0 1.12792 0.25433 0.14328  
O 8.0 1.96702 -0.63164 -0.01293  
O 8.0 0.92897 1.35677 -0.40170  
H 1.0 -0.83439 -0.28474 0.64981  
H 1.0 -0.16722 0.77822 1.76430  
H 1.0 -0.51302 -1.37496 2.77278  
H 1.0 0.49984 -2.11483 1.49245  
H 1.0 1.23888 -1.00877 2.67221  
O 8.0 -2.13045 -0.83416 -0.37183  
H 1.0 -1.89121 -0.79699 -1.34135  
H 1.0 -2.99687 -0.40674 -0.29805  
O 8.0 0.13567 3.80492 0.62434  
H 1.0 0.46017 3.00996 0.14002  
H 1.0 -0.45388 4.27560 0.01655  
O 8.0 2.98502 -0.91553 -2.61243  
H 1.0 2.77227 -0.86028 -1.65221  
H 1.0 3.89308 -0.59150 -2.70541  
O 8.0 -1.44842 -0.64699 -2.97510  
H 1.0 -1.34837 -1.46593 -3.48259  
H 1.0 -0.60940 -0.12317 -3.12540  
O 8.0 0.84165 0.78027 -3.21590  
H 1.0 0.90603 1.20951 -2.33838  
H 1.0 1.62383 0.18113 -3.22656  
O 8.0 -0.68486 2.25049 2.67968  
H 1.0 -0.47456 2.97412 2.03356  
H 1.0 -0.35558 2.54602 3.54132

28 ATOMS zM-6C

Frame: 39

N 7.0 -0.30070 -1.20632 0.54728  
C 6.0 -0.25849 -1.39669 2.02005  
C 6.0 0.44935 0.05522 0.01107  
O 8.0 1.46834 0.32964 0.64784  
O 8.0 -0.06213 0.52666 -1.01994  
H 1.0 0.21555 -1.99798 0.06339  
H 1.0 -1.27999 -1.22877 0.20259  
H 1.0 -0.71423 -2.36030 2.26084  
H 1.0 0.78051 -1.37438 2.34649  
H 1.0 -0.81251 -0.59034 2.50536  
O 8.0 2.07393 0.02653 -3.04989  
H 1.0 1.45343 0.52861 -2.48724  
H 1.0 2.75179 -0.29466 -2.42152  
O 8.0 -1.85535 -0.54151 -2.77217  
H 1.0 -1.23043 -0.04841 -2.17964  
H 1.0 -2.20254 0.10012 -3.41049  
O 8.0 0.46451 -2.25261 -3.39705  
H 1.0 -0.42818 -1.85769 -3.37208  
H 1.0 1.07000 -1.47531 -3.46412  
O 8.0 1.24112 -2.99087 -0.86334  
H 1.0 2.14075 -2.60919 -0.82699  
H 1.0 0.94969 -2.87076 -1.80746  
O 8.0 3.45239 -1.04597 -0.72797  
H 1.0 2.97191 -0.45484 -0.10350  
H 1.0 4.39203 -1.00604 -0.49362  
O 8.0 -2.98900 -1.48129 -0.46236  
H 1.0 -2.84126 -1.20781 -1.39826  
H 1.0 -3.43502 -2.34083 -0.49225

28 ATOMS zM-6D

Frame: 40

N 7.0 -0.31565 -0.00231 -0.16712  
C 6.0 -1.77854 0.12309 0.04957  
C 6.0 0.55197 1.20750 0.30972  
O 8.0 1.70792 1.14692 -0.17830  
O 8.0 0.00591 1.96583 1.09580  
H 1.0 0.07109 -0.81118 0.38976  
H 1.0 -0.10821 -0.19673 -1.17146  
H 1.0 -2.25945 -0.79534 -0.29580  
H 1.0 -1.97248 0.28023 1.11034  
H 1.0 -2.15522 0.97840 -0.51392  
O 8.0 0.99524 -1.82899 1.46778  
H 1.0 1.83644 -1.92913 0.97916  
H 1.0 1.21685 -1.22753 2.22271  
O 8.0 1.62887 0.21271 3.26823  
H 1.0 2.43881 0.62366 3.88728  
H 1.0 0.94123 0.88998 3.15152  
O 8.0 0.08257 -0.73830 -2.87563  
H 1.0 0.16464 -0.05660 -3.55956  
H 1.0 0.90173 -1.29285 -2.95007  
O 8.0 2.41553 -2.17411 -2.84663  
H 1.0 2.82862 -1.90788 -1.99023  
H 1.0 2.36611 -3.14171 -2.84062  
O 8.0 3.65659 1.41440 1.71473  
H 1.0 4.04378 2.26586 1.96690  
H 1.0 3.00783 1.59819 0.99293  
O 8.0 3.14843 -1.18365 -0.37684  
H 1.0 4.02811 -0.97959 -0.02039  
H 1.0 2.66349 -0.31778 -0.38476

31 ATOMS zM-7A

Frame: 41

N 7.0 -0.01512 -0.74239 0.68758  
C 6.0 -0.13333 -1.76261 1.76304  
C 6.0 0.53251 0.63865 1.17870  
O 8.0 1.61708 0.53531 1.76411  
O 8.0 -0.18493 1.59949 0.87560

H 1.0 0.66396 -1.07383 -0.04487  
H 1.0 -0.91864 -0.61024 0.18154  
H 1.0 -0.44197 -2.71192 1.31835  
H 1.0 0.83734 -1.86678 2.24889  
H 1.0 -0.88117 -1.43354 2.48840  
O 8.0 1.63544 -1.44418 -1.45875  
H 1.0 0.99420 -1.46591 -2.20240  
H 1.0 2.20381 -0.66392 -1.63802  
O 8.0 -2.21188 -0.51597 -1.10887  
H 1.0 -1.77218 -0.80198 -1.93868  
H 1.0 -2.35412 0.44510 -1.22858  
O 8.0 -0.44658 -0.94554 -3.31000  
H 1.0 -0.18024 0.01981 -3.38588  
H 1.0 -0.62677 -1.26734 -4.20622  
O 8.0 2.85086 1.10530 -1.93047  
H 1.0 3.57698 1.26547 -2.55263  
H 1.0 3.13553 1.50605 -1.06081  
O 8.0 0.31372 1.61035 -3.16801  
H 1.0 -0.34243 2.01242 -2.55615  
H 1.0 1.17384 1.60298 -2.69431  
O 8.0 -1.66953 2.28165 -1.26634  
H 1.0 -2.12033 3.13441 -1.17318  
H 1.0 -1.16677 2.12605 -0.42502  
O 8.0 3.49538 2.09405 0.51823  
H 1.0 4.35935 1.87813 0.89991  
H 1.0 2.82824 1.66055 1.10882

31 ATOMS zM-7B

Frame: 42

N 7.0 0.51004 -0.19290 1.02492  
C 6.0 -0.41493 -0.80781 2.00966  
C 6.0 1.21973 1.13019 1.47622  
O 8.0 0.60512 1.74600 2.35380  
O 8.0 2.26367 1.34244 0.84640  
H 1.0 1.24277 -0.86643 0.71770  
H 1.0 -0.01088 0.04623 0.14226  
H 1.0 -0.86183 -1.69723 1.55857  
H 1.0 0.14331 -1.08197 2.90672  
H 1.0 -1.18457 -0.08295 2.27328  
O 8.0 2.27372 -2.05300 -0.22524  
H 1.0 3.21588 -1.97316 -0.01129  
H 1.0 2.17356 -1.69408 -1.13948  
O 8.0 1.24432 4.40727 2.04561  
H 1.0 2.20090 4.43776 1.88932  
H 1.0 1.05373 3.46274 2.27665  
O 8.0 1.71982 -0.70868 -2.59806  
H 1.0 1.98889 -0.99119 -3.48514  
H 1.0 2.11448 0.20661 -2.43902  
O 8.0 -0.21946 4.90488 -0.17829  
H 1.0 -1.09949 5.13682 0.15377  
H 1.0 0.34452 4.75545 0.63194  
O 8.0 2.53897 1.71710 -1.91922  
H 1.0 1.76062 2.30134 -2.05237  
H 1.0 2.60314 1.62522 -0.94185  
O 8.0 0.00929 3.02537 -2.08752  
H 1.0 -0.24163 3.46552 -2.91389  
H 1.0 -0.09857 3.71358 -1.36832  
O 8.0 -0.69243 0.34107 -1.44981  
H 1.0 -0.05787 -0.12602 -2.03285  
H 1.0 -0.60638 1.29010 -1.69096

31 ATOMS zM-7C

Frame: 43

N 7.0 -0.16838 0.33438 0.59367  
C 6.0 -0.96958 0.23750 1.84034  
C 6.0 1.14818 1.16973 0.68093  
O 8.0 1.51830 1.40934 1.83125  
O 8.0 1.62271 1.40707 -0.44460

H	1.0	0.14639	-0.62354	0.29600
H	1.0	-0.75492	0.68906	-0.19390
H	1.0	-1.82530	-0.41415	1.64706
H	1.0	-0.34647	-0.17286	2.63403
H	1.0	-1.31104	1.23293	2.12897
O	8.0	4.33107	0.58175	-0.45460
H	1.0	3.49126	1.06581	-0.61001
H	1.0	4.52910	0.78431	0.49100
O	8.0	-1.74585	0.84662	-1.68729
H	1.0	-2.23024	1.63592	-1.97128
H	1.0	-0.98977	0.73341	-2.32904
O	8.0	3.54892	-1.90680	-0.90362
H	1.0	3.91129	-0.98131	-0.71023
H	1.0	4.26380	-2.53073	-0.70568
O	8.0	1.71446	-2.03515	-2.96428
H	1.0	2.54187	-2.01356	-2.42501
H	1.0	1.93279	-2.46341	-3.80573
O	8.0	0.60108	0.60382	-2.96538
H	1.0	0.95781	-0.30137	-3.08981
H	1.0	1.07821	0.95915	-2.18453
O	8.0	4.28395	1.29792	2.23185
H	1.0	4.67725	2.13797	2.51141
H	1.0	3.30905	1.44406	2.21215
O	8.0	0.82421	-2.17454	-0.17828
H	1.0	0.62147	-2.36110	-1.11386
H	1.0	1.80701	-2.14083	-0.17418

31 ATOMS zW-7D

Frame: 44

N	7.0	0.45623	-0.53654	0.36297
C	6.0	0.54600	-1.07927	1.74405
C	6.0	0.96969	0.91470	0.19417
O	8.0	0.47449	1.71118	1.01369
O	8.0	1.75603	1.05642	-0.74937
H	1.0	0.94629	-1.13415	-0.33470
H	1.0	-0.55742	-0.52836	0.04262
H	1.0	0.10419	-2.07828	1.75784
H	1.0	1.59484	-1.13173	2.04598
H	1.0	0.00169	-0.41489	2.41574
O	8.0	0.42941	1.76865	-3.16377
H	1.0	1.05106	1.61171	-2.42095
H	1.0	-0.10478	2.54164	-2.87669
O	8.0	-0.77975	-0.61748	-3.22503
H	1.0	-0.39352	0.30615	-3.30021
H	1.0	-1.12730	-0.85257	-4.09885
O	8.0	-2.07541	-0.42911	-0.68373
H	1.0	-1.87950	-0.51847	-1.64005
H	1.0	-2.34286	0.51267	-0.53509
O	8.0	0.99781	4.24995	0.01762
H	1.0	0.89434	3.38457	0.48835
H	1.0	1.90399	4.23780	-0.32755
O	8.0	-2.30428	2.22882	0.01669
H	1.0	-2.00165	2.84966	-0.67992
H	1.0	-1.55342	2.18838	0.63873
O	8.0	-0.91582	3.90513	-1.85298
H	1.0	-1.28964	4.72437	-2.21196
H	1.0	-0.19860	4.16946	-1.20936
O	8.0	1.29646	-2.01798	-1.91829
H	1.0	0.59005	-1.64257	-2.49444
H	1.0	2.12532	-1.64289	-2.25467

34 ATOMS zM-8A

Frame: 45

N	7.0	-0.33060	-0.28665	0.72527
C	6.0	-1.71967	-0.81348	0.82267
C	6.0	-0.17608	1.25341	0.75547
O	8.0	-1.04754	1.86913	0.12657
O	8.0	0.83100	1.59655	1.39193

H	1.0	0.26081	-0.64512	1.47870
H	1.0	0.11729	-0.62762	-0.21486
H	1.0	-1.67929	-1.90520	0.80958
H	1.0	-2.18519	-0.45965	1.74520
H	1.0	-2.28710	-0.46482	-0.03881
O	8.0	-1.76296	-1.23763	-2.81201
H	1.0	-1.94604	-0.25929	-2.84113
H	1.0	-2.03060	-1.60297	-3.66774
O	8.0	2.07470	0.86797	-2.75058
H	1.0	1.48015	1.59300	-3.03587
H	1.0	2.78400	1.29592	-2.22004
O	8.0	2.34553	3.69472	0.59919
H	1.0	2.69560	4.21254	1.33997
H	1.0	1.81211	2.95530	0.99815
O	8.0	0.19693	2.96526	-3.33334
H	1.0	0.19449	3.55188	-2.52150
H	1.0	0.20484	3.55130	-4.10519
O	8.0	-2.07059	1.44415	-2.56594
H	1.0	-1.91377	1.59178	-1.61194
H	1.0	-1.34734	1.94878	-3.00024
O	8.0	0.04671	4.27677	-0.99434
H	1.0	0.89824	4.31220	-0.51061
H	1.0	-0.47852	3.61196	-0.49836
O	8.0	3.97256	2.24322	-1.18177
H	1.0	3.49649	2.81746	-0.53999
H	1.0	4.58896	2.81631	-1.66095
O	8.0	0.68396	-1.14220	-1.55791
H	1.0	-0.08982	-1.30220	-2.15026
H	1.0	1.24042	-0.44634	-2.00690

34 ATOMS zM-8B

Frame: 46

N	7.0	-0.92717	0.71004	-0.24952
C	6.0	-2.23425	1.34796	-0.54958
C	6.0	0.22863	1.68279	0.06323
O	8.0	1.34576	1.07760	0.07447
O	8.0	-0.07231	2.84375	0.25978
H	1.0	-1.03055	0.05491	0.58045
H	1.0	-0.62293	0.07611	-1.02692
H	1.0	-2.95577	0.55882	-0.77368
H	1.0	-2.55820	1.92243	0.31787
H	1.0	-2.12490	2.01710	-1.40424
O	8.0	2.62391	-0.19911	-2.00915
H	1.0	3.20326	0.22711	-2.65870
H	1.0	2.28255	0.50825	-1.41490
O	8.0	2.39825	-1.44642	0.62222
H	1.0	2.96752	-1.41896	-0.16922
H	1.0	1.89936	-0.59646	0.55456
O	8.0	3.18125	2.08890	1.93611
H	1.0	2.91067	2.93034	2.33355
H	1.0	2.52880	1.91188	1.22038
O	8.0	2.90945	-0.35777	3.28083
H	1.0	3.00958	-0.92357	2.49141
H	1.0	3.08411	0.55488	2.94917
O	8.0	-1.22682	-1.15679	1.72917
H	1.0	-0.72919	-0.98939	2.56991
H	1.0	-0.84253	-1.96488	1.33194
O	8.0	0.24057	-3.02487	0.11303
H	1.0	0.32097	-3.98629	0.20638
H	1.0	1.11688	-2.62618	0.35775
O	8.0	-0.05177	-1.22255	-2.11732
H	1.0	-0.02211	-2.01494	-1.54107
H	1.0	0.88339	-1.00960	-2.31862
O	8.0	0.28253	-0.55669	3.93153
H	1.0	0.18446	-0.92541	4.82118
H	1.0	1.26127	-0.47441	3.75861

34 ATOMS zM-8C

Frame: 47

N	7.0	-0.53999	0.79156	-0.48743
C	6.0	-1.88438	1.36311	-0.20716
C	6.0	0.65097	1.78574	-0.43535
O	8.0	0.56601	2.64187	0.45212
O	8.0	1.53802	1.52386	-1.26506
H	1.0	-0.30142	0.05929	0.22173
H	1.0	-0.55627	0.25382	-1.39423
H	1.0	-2.61368	0.55008	-0.23456
H	1.0	-1.88290	1.84049	0.77239
H	1.0	-2.12300	2.09941	-0.97727
O	8.0	2.05104	-1.03995	-2.51919
H	1.0	2.41102	-1.39932	-1.68475
H	1.0	1.99110	-0.08062	-2.34024
O	8.0	2.96895	-1.41816	0.26589
H	1.0	3.65079	-2.05867	0.51941
H	1.0	3.39769	-0.51860	0.29458
O	8.0	0.35375	-1.28406	1.24236
H	1.0	1.30651	-1.39672	1.01125
H	1.0	0.30940	-0.99279	2.18951
O	8.0	-0.33825	2.42572	3.03493
H	1.0	-0.01341	2.57192	2.11213
H	1.0	-0.00055	3.16575	3.56101
O	8.0	-0.72610	-1.07965	-2.51552
H	1.0	0.22908	-1.26655	-2.67989
H	1.0	-1.04150	-1.80424	-1.93406
O	8.0	0.16158	-0.18289	3.73580
H	1.0	-0.01445	0.77845	3.57106
H	1.0	-0.52827	-0.48710	4.34414
O	8.0	-1.45755	-2.70555	-0.30524
H	1.0	-1.53619	-3.66473	-0.20039
H	1.0	-0.80501	-2.39333	0.36314
O	8.0	3.90403	1.12401	0.08282
H	1.0	3.16414	1.49343	-0.45742
H	1.0	3.99606	1.71523	0.84505

34 ATOMS zM-8D

Frame: 48

N	7.0	-0.60607	-0.08654	0.35621
C	6.0	-1.90828	0.03618	1.06912
C	6.0	0.39067	1.10460	0.51500
O	8.0	1.37957	0.98385	-0.23064
O	8.0	0.06891	1.93009	1.37063
H	1.0	-0.07990	-0.91803	0.71000
H	1.0	-0.75747	-0.27745	-0.65485
H	1.0	-2.54382	-0.79450	0.75181
H	1.0	-1.72080	-0.01767	2.14129
H	1.0	-2.37051	0.98821	0.80581
O	8.0	3.67081	-1.56222	1.05236
H	1.0	3.71601	-0.60361	1.35575
H	1.0	4.43127	-2.02002	1.44142
O	8.0	1.56786	1.41092	3.61124
H	1.0	0.94947	1.79595	2.94458
H	1.0	1.57962	2.01673	4.36781
O	8.0	2.71418	-1.13242	-1.57708
H	1.0	2.21051	-0.34452	-1.27244
H	1.0	3.28924	-1.34211	-0.81004
O	8.0	0.85699	-3.02464	-1.64434
H	1.0	1.08688	-3.80661	-2.16895
H	1.0	1.62463	-2.37562	-1.71443
O	8.0	3.57583	1.03185	1.63694
H	1.0	3.05093	1.21165	2.44741
H	1.0	2.98842	1.31186	0.90321
O	8.0	0.97134	-2.36212	1.15916
H	1.0	1.93193	-2.14209	1.19128
H	1.0	0.86448	-2.90431	0.35037
O	8.0	-1.14720	-1.19525	-2.19715

H	1.0	-1.03652	-0.78253	-3.06686
H	1.0	-0.50836	-1.94725	-2.16004
O	8.0	-0.08797	-1.03601	3.60498
H	1.0	0.33188	-1.71512	3.04845
H	1.0	0.58544	-0.34546	3.75514

31 ATOMS zG-7A

Frame: 49

C	6.0	-0.68019	0.63399	-0.17264
N	7.0	0.19871	1.84453	-0.11670
H	1.0	0.15644	2.25176	0.82606
H	1.0	-0.05610	2.51801	-0.85563
H	1.0	1.21006	1.58883	-0.26685
H	1.0	-1.72167	0.95898	-0.20015
H	1.0	-0.52635	0.06914	0.74553
C	6.0	-0.38919	-0.27570	-1.38791
O	8.0	0.37462	0.19412	-2.28478
O	8.0	-0.91903	-1.39577	-1.35533
O	8.0	2.90358	1.25674	-0.23575
H	1.0	3.09508	0.55029	-0.87907
H	1.0	3.11835	0.88411	0.63743
O	8.0	-0.17721	2.82338	-2.70244
H	1.0	-0.74756	3.08970	-3.42834
H	1.0	0.05283	1.87657	-2.83833
O	8.0	0.69908	1.73223	2.66107
H	1.0	1.42552	1.07421	2.66968
H	1.0	0.31927	1.73510	3.54375
O	8.0	2.26179	-2.17499	0.70961
H	1.0	2.44029	-1.95295	-0.22161
H	1.0	1.37497	-2.59716	0.72960
O	8.0	2.77093	-0.10337	2.26539
H	1.0	2.55364	-0.93764	1.75546
H	1.0	3.46242	-0.33593	2.89291
O	8.0	2.73854	-1.02176	-1.92573
H	1.0	3.20589	-1.40989	-2.67161
H	1.0	1.87239	-0.68594	-2.26802
O	8.0	-0.34846	-3.08186	0.67620
H	1.0	-0.61824	-3.98736	0.49745
H	1.0	-0.70415	-2.53694	-0.06427

31 ATOMS zG-7B

Frame: 50

C	6.0	0.28688	-0.00871	0.54257
N	7.0	0.74445	1.41744	0.54897
H	1.0	0.75226	1.80179	1.49965
H	1.0	0.06640	1.98078	0.02434
H	1.0	1.68917	1.47214	0.05143
H	1.0	-0.50097	-0.11848	1.28920
H	1.0	1.13137	-0.64259	0.80458
C	6.0	-0.27515	-0.35707	-0.85507
O	8.0	-1.07772	0.48486	-1.34065
O	8.0	0.11921	-1.42144	-1.37824
O	8.0	-0.05487	-1.74886	-4.12302
H	1.0	-0.78136	-2.30381	-4.42144
H	1.0	-0.10193	-1.74417	-3.14207
O	8.0	2.97854	1.25340	-0.90727
H	1.0	2.77380	1.64845	-1.78464
H	1.0	3.12900	0.29893	-1.06249
O	8.0	-2.15280	2.45953	0.16807
H	1.0	-2.66160	3.10411	-0.33211
H	1.0	-1.95459	1.70963	-0.44353
O	8.0	-0.80907	2.44557	2.60367
H	1.0	-1.19971	2.68451	3.44814
H	1.0	-1.49300	2.56611	1.91697
O	8.0	-0.23105	1.06623	-4.00205
H	1.0	-0.70717	0.94675	-3.15944
H	1.0	-0.12078	0.14741	-4.31362
O	8.0	2.88235	-1.54353	-1.12666

H	1.0	1.92301	-1.68854	-1.27061
H	1.0	3.33862	-2.11752	-1.74912
O	8.0	2.01334	2.35013	-3.21420
H	1.0	2.42318	2.82830	-3.93984
H	1.0	1.22008	1.88521	-3.58556

31 ATOMS zG-7C

Frame: 51

C	6.0	0.18151	0.77518	0.51714
N	7.0	0.77024	2.06860	0.04485
H	1.0	0.96880	2.69610	0.82295
H	1.0	0.15609	2.56718	-0.64020
H	1.0	1.68529	1.91309	-0.49814
H	1.0	-0.64863	0.99033	1.19360
H	1.0	0.94668	0.22127	1.05951
C	6.0	-0.35138	-0.06230	-0.66976
O	8.0	-0.76263	0.57761	-1.66703
O	8.0	-0.32353	-1.30598	-0.52738
O	8.0	0.78782	-0.29328	-3.74493
H	1.0	0.46433	-1.17115	-4.00838
H	1.0	0.12169	-0.01777	-3.07567
O	8.0	-0.52005	3.29710	-2.15819
H	1.0	0.15683	3.23852	-2.85403
H	1.0	-0.97626	2.44039	-2.22509
O	8.0	-0.50069	-2.68851	-3.00467
H	1.0	-1.35299	-3.10043	-3.17260
H	1.0	-0.57462	-2.27194	-2.12027
O	8.0	3.14212	-0.81818	-2.04819
H	1.0	2.39605	-0.83486	-2.67360
H	1.0	2.90315	-1.43142	-1.32563
O	8.0	2.14432	-2.28570	0.15011
H	1.0	2.21593	-3.23789	0.26546
H	1.0	1.19141	-2.09301	0.01665
O	8.0	1.63084	2.31877	-3.97075
H	1.0	1.96607	2.66738	-4.80299
H	1.0	1.37753	1.38524	-4.14359
O	8.0	2.96870	1.79353	-1.44818
H	1.0	2.71909	2.12525	-2.32921
H	1.0	3.18284	0.83639	-1.60388

31 ATOMS zG-7D

Frame: 52

C	6.0	-0.29340	-0.89347	-0.17103
N	7.0	0.50761	0.33189	0.16494
H	1.0	0.68920	0.42662	1.18448
H	1.0	0.00511	1.19396	-0.11994
H	1.0	1.39072	0.38445	-0.36792
H	1.0	-1.11498	-0.94839	0.54240
H	1.0	0.35449	-1.76333	-0.08254
C	6.0	-0.80978	-0.71741	-1.62028
O	8.0	-1.78362	0.09830	-1.73276
O	8.0	-0.19019	-1.28830	-2.52432
O	8.0	-0.00470	2.95473	-0.96163
H	1.0	0.92500	2.87981	-1.23112
H	1.0	-0.51688	2.89540	-1.79255
O	8.0	1.14925	0.71066	-3.94857
H	1.0	0.73342	-0.14026	-3.70450
H	1.0	0.39049	1.31253	-4.03951
O	8.0	-1.28937	2.14301	-3.34816
H	1.0	-2.00336	2.48283	-3.89621
H	1.0	-1.62924	1.32879	-2.88892
O	8.0	-1.32022	2.83587	1.64483
H	1.0	-0.91614	3.23053	0.85564
H	1.0	-2.05504	2.28904	1.29790
O	8.0	0.53261	1.22872	2.75863
H	1.0	-0.17430	1.88465	2.53368
H	1.0	0.47450	1.03898	3.69863
O	8.0	-3.07283	1.01805	0.46268

H	1.0	-2.69742	0.66977	-0.38440
H	1.0	-4.02927	1.01637	0.36841
O	8.0	2.38336	1.44392	-1.70055
H	1.0	3.33209	1.56111	-1.80999
H	1.0	2.02439	1.17097	-2.59227

31 ATOMS nG-7E

Frame: 53

C	6.0	0.63559	-0.20004	-1.17205
N	7.0	2.03019	0.15534	-1.44402
C	6.0	0.19354	0.44094	0.14638
O	8.0	1.10483	1.28905	0.64015
O	8.0	-0.86015	0.21092	0.69291
H	1.0	1.84819	1.24002	-0.03083
H	1.0	2.19296	0.38595	-2.41570
H	1.0	2.65538	-0.60236	-1.15601
H	1.0	0.46539	-1.27691	-1.08629
H	1.0	-0.03154	0.17707	-1.95186
O	8.0	1.66558	0.61611	3.42900
H	1.0	1.61351	1.16509	2.63250
H	1.0	1.65017	1.21502	4.20886
O	8.0	-0.87733	-0.46986	3.47198
H	1.0	-1.18458	-0.24160	2.57652
H	1.0	0.04546	-0.12296	3.47915
O	8.0	1.11138	2.11591	5.65121
H	1.0	0.17306	1.85219	5.77399
H	1.0	1.53274	2.03243	6.51156
O	8.0	0.11422	-3.09237	3.37522
H	1.0	-0.06109	-3.62212	4.15846
H	1.0	-0.40645	-2.26975	3.47553
O	8.0	-1.46473	1.12001	5.67489
H	1.0	-2.26621	1.64677	5.60405
H	1.0	-1.45278	0.52618	4.89332
O	8.0	2.61006	-2.00224	2.59032
H	1.0	2.48844	-1.11724	2.97239
H	1.0	1.81467	-2.50153	2.85917
O	8.0	3.26415	-2.07814	-0.05092
H	1.0	3.04454	-2.07392	0.91214
H	1.0	3.97873	-2.71190	-0.15540

31 ATOMS nG-7G

Frame: 54

C	6.0	0.37375	-0.34567	0.03324
N	7.0	-0.16576	1.02078	0.11953
C	6.0	-0.04272	-1.20811	1.21203
O	8.0	0.08163	2.08707	2.68569
O	8.0	0.70698	-1.92536	1.82832
H	1.0	0.04746	3.04622	2.62707
H	1.0	-1.16989	1.01186	-0.04530
H	1.0	0.23614	1.58506	-0.63424
H	1.0	1.46242	-0.30433	0.04372
H	1.0	0.07294	-0.87353	-0.88502
O	8.0	-0.99507	0.68222	4.77360
H	1.0	-0.62078	1.20527	4.02580
H	1.0	-0.44201	0.87830	5.53538
O	8.0	-3.54222	0.50188	-2.66350
H	1.0	-3.86327	-0.31194	-3.06230
H	1.0	-3.52233	0.34814	-1.69732
O	8.0	0.57590	2.56791	-2.33818
H	1.0	1.33705	2.60131	-2.92430
H	1.0	-0.20777	2.46673	-2.91937
O	8.0	-1.35784	-1.10479	1.49996
H	1.0	-1.53395	-1.51130	2.41044
H	1.0	0.02300	1.74637	1.74824
O	8.0	-3.41472	0.15094	0.07903
H	1.0	-2.77962	-0.36511	0.61216
H	1.0	-4.01825	0.57094	0.69920
O	8.0	-1.71602	2.19620	-3.84987

H 1.0 -2.23723 2.95665 -4.12386  
H 1.0 -2.35477 1.56912 -3.44795  
O 8.0 -1.69101 -1.83119 3.97740  
H 1.0 -1.45507 -0.96243 4.38326  
H 1.0 -0.96467 -2.42734 4.19264

#### 34 ATOMS zG-8A

Frame: 55

C 6.0 1.10211 -0.06114 -0.97134  
N 7.0 0.91364 1.38337 -0.63364  
C 6.0 2.09429 -0.68655 0.03255  
O 8.0 2.88592 0.11985 0.57523  
O 8.0 1.99238 -1.92183 0.22877  
H 1.0 0.52999 1.45294 0.34461  
H 1.0 0.25768 1.84551 -1.29975  
H 1.0 1.83006 1.83373 -0.61341  
H 1.0 1.48479 -0.14108 -1.99059  
H 1.0 0.12898 -0.54570 -0.92211  
O 8.0 2.49175 0.37501 3.19245  
H 1.0 2.86591 0.26604 2.28140  
H 1.0 3.19555 0.71681 3.75210  
O 8.0 -2.84584 0.22522 -1.99849  
H 1.0 -3.33842 -0.23347 -2.68485  
H 1.0 -2.58376 -0.46517 -1.33832  
O 8.0 1.12675 -2.14025 2.91674  
H 1.0 1.65344 -1.41023 3.28625  
H 1.0 1.52776 -2.28536 2.03670  
O 8.0 -1.29736 -1.08922 2.47863  
H 1.0 -1.91131 -1.30738 3.18735  
H 1.0 -0.44477 -1.56514 2.68396  
O 8.0 0.05055 1.36995 2.00360  
H 1.0 0.83269 1.20188 2.56101  
H 1.0 -0.56781 0.64484 2.21822  
O 8.0 -1.07414 2.21364 -2.38404  
H 1.0 -1.54726 3.05013 -2.42079  
H 1.0 -1.76339 1.50652 -2.31086  
O 8.0 -2.03670 -1.64263 -0.23814  
H 1.0 -1.88469 -1.48259 0.71094  
H 1.0 -1.41440 -2.35320 -0.49928  
O 8.0 0.02039 -3.34229 -0.97537  
H 1.0 0.81037 -2.91497 -0.56299  
H 1.0 0.13114 -4.29129 -0.86928

#### 34 ATOMS zG-8B

Frame: 56

C 6.0 0.10405 0.54128 -0.30698  
N 7.0 0.40712 1.86387 0.35456  
C 6.0 0.79217 -0.60265 0.46794  
O 8.0 0.26240 -0.91802 1.57597  
O 8.0 1.83340 -1.07970 -0.02450  
H 1.0 1.17473 1.78250 1.06867  
H 1.0 -0.43818 2.23131 0.83375  
H 1.0 0.74203 2.52594 -0.36762  
H 1.0 0.47296 0.58758 -1.32833  
H 1.0 -0.97746 0.40899 -0.31348  
O 8.0 -2.04102 2.42108 1.61287  
H 1.0 -2.23446 1.54168 2.01466  
H 1.0 -2.25017 3.08303 2.2783  
O 8.0 4.87953 0.80210 1.08018  
H 1.0 4.69615 -0.14817 1.32221  
H 1.0 5.80451 0.97440 1.28118  
O 8.0 2.43332 1.68745 2.26439  
H 1.0 2.32243 0.94965 2.89176  
H 1.0 3.33336 1.57438 1.90435  
O 8.0 4.00443 -1.64855 1.66661  
H 1.0 3.54219 -1.57243 2.51837  
H 1.0 3.26885 -1.70021 1.02220  
O 8.0 2.03590 -0.84789 3.58605

H 1.0 1.31971 -1.04174 2.93030  
H 1.0 1.70184 -1.10012 4.45223  
O 8.0 3.47897 0.81349 -1.40390  
H 1.0 2.93111 0.05366 -1.12546  
H 1.0 4.12053 0.91111 -0.67656  
O 8.0 1.88489 2.95737 -1.70884  
H 1.0 2.55777 2.22987 -1.71115  
H 1.0 1.95415 3.42785 -2.54319  
O 8.0 -2.12856 -0.11954 2.51679  
H 1.0 -1.27324 -0.48271 2.17328  
H 1.0 -2.79037 -0.80853 2.40977

#### 34 ATOMS zG-8C

Frame: 57

C 6.0 -0.23430 0.41418 -0.95654  
N 7.0 0.28574 1.79797 -0.72092  
C 6.0 -1.58016 0.2298 -0.22747  
O 8.0 -2.20943 1.28382 0.01798  
O 8.0 -1.89264 -0.94797 0.07805  
H 1.0 0.31236 1.99449 0.31711  
H 1.0 -0.34856 2.48358 -1.12911  
H 1.0 1.25658 1.88186 -1.08298  
H 1.0 0.51341 -0.28293 -0.58434  
H 1.0 -0.35056 0.25801 -2.03071  
O 8.0 -0.63608 -3.19902 -0.72972  
H 1.0 -1.20106 -3.66382 -1.35249  
H 1.0 -1.12774 -2.38918 -0.44493  
O 8.0 0.33658 2.24077 2.00220  
H 1.0 0.77166 1.49544 2.45879  
H 1.0 -0.57371 2.25683 2.35197  
O 8.0 2.93910 1.28739 -1.20351  
H 1.0 2.99107 0.53681 -0.55929  
H 1.0 3.81713 1.67174 -1.26738  
O 8.0 -2.42747 1.74649 2.62351  
H 1.0 -2.59356 1.60510 1.65650  
H 1.0 -3.17691 2.23603 2.97553  
O 8.0 2.07083 -3.24099 -0.37270  
H 1.0 2.30103 -4.04610 0.09958  
H 1.0 1.10299 -3.29115 -0.53981  
O 8.0 -1.40231 -0.92751 2.89854  
H 1.0 -1.67732 -1.17147 1.99433  
H 1.0 -1.87795 -0.09345 3.05961  
O 8.0 2.73812 -0.72841 0.56086  
H 1.0 2.27805 -0.63208 1.41368  
H 1.0 2.57614 -1.64668 0.25427  
O 8.0 1.20092 -0.27679 2.96135  
H 1.0 0.26463 -0.61871 2.97065  
H 1.0 1.58728 -0.51521 3.81027

#### 34 ATOMS zG-8D

Frame: 58

C 6.0 0.03803 -0.10748 0.52590  
N 7.0 0.68044 0.91551 -0.36531  
C 6.0 -1.28936 0.48437 1.05731  
O 8.0 -1.17444 1.38635 1.95127  
O 8.0 -2.33092 0.09634 0.51411  
H 1.0 1.16127 1.67846 0.17644  
H 1.0 -0.03558 1.40315 -0.94192  
H 1.0 1.34611 0.49648 -1.03992  
H 1.0 0.73435 -0.34294 1.33136  
H 1.0 -0.16632 -0.99572 -0.06837  
O 8.0 0.16478 1.55929 -4.08254  
H 1.0 -0.45130 1.27206 -4.76298  
H 1.0 -0.36906 2.02694 -3.40414  
O 8.0 1.79691 3.10540 0.91411  
H 1.0 1.16128 3.80585 0.68664  
H 1.0 1.69407 2.96369 1.87642  
O 8.0 1.96250 0.01625 -2.69881

H 1.0 2.85634 0.07022 -3.04902  
H 1.0 1.38332 0.50557 -3.32537  
O 8.0 0.89819 2.33933 3.39500  
H 1.0 1.10414 1.88753 4.21724  
H 1.0 0.12504 1.88461 2.98457  
O 8.0 -3.53948 2.21745 -0.80050  
H 1.0 -3.54274 2.85218 -0.06660  
H 1.0 -3.37113 1.36325 -0.35192  
O 8.0 -2.51017 3.63514 1.55469  
H 1.0 -2.10478 2.75676 1.81084  
H 1.0 -2.91664 3.99155 2.35091  
O 8.0 -0.48724 4.69495 0.02776  
H 1.0 -0.51697 5.65160 -0.07554  
H 1.0 -1.22163 4.45477 0.64203  
O 8.0 -1.08295 2.62331 -1.86685  
H 1.0 -2.02983 2.47508 -1.59999  
H 1.0 -0.83620 3.46807 -1.44692

## REFERENCES

- [1] A. Leach, *Molecular Modelling: Principles and Applications*, United Kingdom: Pearson Prentice Hall, 2001.
- [2] M. Garey and D. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*, San Francisco, CA: Freeman, 1979.
- [3] N. Metropolis, A. Rosenbluth, M. Rosenbluth and A. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087-1092, 1953.
- [4] J. Head, "Predicting Nanocluster Structures," *Handbook of Nanophysics: Principles and Methods*, vol. 1, pp. 5/1-5/12, 2011.
- [5] E. Izgorodina, C. Lin and M. Coote, "Energy-directed tree search: an efficient systematic algorithm for finding the lowest energy conformation of molecules," *Phys. Chem. Chem. Phys.*, vol. 9, pp. 2507-2516, 2007.
- [6] S. Kirkpatrick, C. Gellatt Jr. and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [7] F. Calvo, "Non-genetic global optimization methods in molecular science: An overview," *Comput. Mater. Sci.*, vol. 45, pp. 8-15, 2009.
- [8] L. Cheng, Y. Feng, J. Yang and J. Yang, "Funnel hopping: Searching the cluster potential energy surface over the funnels," *J. Chem. Phys.*, vol. 130, p. 214112, 2009.
- [9] D. Wales and J. Doye, "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms," *J. Phys. Chem. A*, vol. 101, no. 28, pp. 5111-5116, 1997.
- [10] Z. Li and H. Scheraga, "Monte Carlo-minimization approach to the multiple-minima problem in protein folding," *Proc. Natl. Acad. Sci.*, vol. 84, pp. 6611-6615, 1987.
- [11] R. Gehrke and K. Reuter, "Assessing the efficiency of first-principles basin-hopping sampling," *Phys. Rev. B*, vol. 79, p. 085412, 2009.
- [12] D. Deaven and K. Ho, "Molecular Geometry Optimization with a Genetic Algorithm," *Phys. Rev. Lett.*, vol. 75, pp. 288-291, 1995.
- [13] P. Day, J. Jensen, M. Gordon, S. Webb, W. Stevens, M. Krauss, D. Garmer, H. Basch and D. Cohen, "An effective fragment method for modeling solvent effects in quantum mechanical calculations," *J. Chem. Phys.*, vol. 105, no. 5, pp. 1968-1986, 1996.

- [14] M. Schmidt, K. Baldridge, J. Boatz, S. Elbert, M. Gordon, J. Jensen, S. Koseki, N. Matsunaga, K. Nguyen, S. Su, T. Windus, M. Dupuis and J. Montgomery Jr., "General Atomic and Molecular Electronic Structure System," *J. Comput. Chem.*, vol. 14, pp. 1347-1363, 1993.
- [15] C. Roothaan, "A Study of Two-Center Integrals Useful in Calculations on Molecular Structure. I," *J.Chem.Phys.*, vol. 19, pp. 1445-1458, 1951.
- [16] G. Hall, "The molecular orbital theory of chemical valency VIII. A method of calculating ionization potentials," *Proc.Roy.Soc.London Ser.A*, vol. 205, pp. 541-552, 1951.
- [17] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, no. 3, pp. B864-B871, 1964.
- [18] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects\*," *Phys. Rev.*, vol. 140, pp. A1133-A1138, 1965.
- [19] A. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior," *Phys. Rev.*, vol. A38, no. 6, pp. 3098-3100, 1988.
- [20] A. Becke, "Density-functional thermochemistry. I. The effect of the exchange-only gradient correction," *J. Chem. Phys.*, vol. 96, no. 3, p. 2155, 1992.
- [21] A. Becke, "Density-functional thermochemistry. II. The effect of the Perdew-Wang generalized-gradient correlation correction," *J. Chem. Phys.*, vol. 97, no. 12, p. 9173, 1992.
- [22] A. Becke, "Density-functional thermochemistry. III. The role of exact exchange," *J. Chem. Phys.*, vol. 98, no. 7, p. 5648, 1993.
- [23] A. Becke, "Density-functional thermochemistry. V. Systematic optimization of exchange-correlation functionals," *J. Chem. Phys.*, vol. 107, no. 20, p. 8554, 1997.
- [24] C. Lee, W. Yang and R. Parr, "Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density," *Phys. Rev. B*, vol. 37, no. 2, pp. 785-789, 1988.
- [25] S. Vosko, L. Wilk and M. Nusair, " Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis," *Can. J. Phys.*, vol. 58, no. 8, p. 1200, 1980.
- [26] C. Cramer and D. Truhlar, "AM1-SM2 and PM3-SM3 parameterized SCF solvation models for free energies in aqueous solution," *J. Comput. Aided Mol. Des.*, vol. 6, pp. 629-666, 1992.
- [27] P. Smith and B. Pettitt, "Modeling Solvent in Biomolecular Systems," *J.Phys.Chem.*, vol. 98, no. 39, pp. 9700-9711, 1994.



- [28] M. Gordon, L. Slipchenko, H. Li and J. Jensen, "The effective fragment potential: A general method for predicting intermolecular interactions," *Ann.Rep.Comp.Chem.*, vol. 3, pp. 177-193, 2007.
- [29] M. Gordon, M. Freitag, P. Bandyopadhyay, J. Jensen, V. Kairys and W. Stevens, "The effective fragment potential method: A QM-based approach to modeling environmental effects in chemistry," *J.Phys.Chem.A*, vol. 105, pp. 293-307, 2001.
- [30] D. Ghosh, D. Kosenkov, V. Vanovschi, C. Williams, J. Herbert, M. Gordon, M. Schmidt, L. Slipchenko and A. Krylov, "Non-covalent interactions in extended systems described by the effective fragment potential method: Theory and applications to nucleobase oligomers," *J.Phys.Chem.A.*, vol. 114, pp. 12739-12754, 2010.
- [31] D. Ghosh, O. Isayev, L. Slipchenko and A. Krylov, "The effect of solvation on vertical ionization energy of thymine: From microhydration to bulk," *J.Phys.Chem.A.*, vol. 115, pp. 6028-6038, 2011.
- [32] L. Slipchenko, "Solvation of the Excited States of Chromophores in Polarizable Environment: Orbital Relaxation versus Polarization," *J.Phys.Chem.A*, vol. 114, pp. 8824-8830, 2010.
- [33] G. Merrill and M. Gordon, "Study of Small Water Clusters Using the Effective Fragment Potential Model," *J. Phys. Chem. A*, vol. 102, pp. 2650-2657, 1998.
- [34] A. Stone and M. Alderton, "Distributed multipole analysis methods and applications," *Molec.Phys.*, vol. 56, pp. 1047-1064, 1985.
- [35] P. Day, R. Pachter, M. Gordon and G. Merrill, "A study of water clusters using the effective fragment potential and Monte Carlo simulated annealing," *J. Chem. Phys.*, vol. 112, no. 5, pp. 2063-2073, 2000.
- [36] C. Aikens and M. Gordon, "Incremental Solvation of Nonionized and Zwitterionic Glycine," *J. Am Chem. Soc.*, vol. 128, pp. 12835-12850, 2006.
- [37] I. Adamovic, M. Freitag and M. Gordon, "Density functional theory based effective fragment potential method," *J. Chem. Phys.*, vol. 118, no. 15, pp. 6725-6732, 2003.
- [38] P. Bandyopadhyay and M. Gordon, "A combined discrete/continuum solvation model: Application to glycine," *J. Chem. Phys.*, vol. 113, no. 3, pp. 1104-1109, 2000.
- [39] D. Yevick, *A First Course in Computational Physics and Object-Oriented Programming with C++*, New York: Cambridge University Press, 2005.
- [40] H. P. Langtangen, *A Primer on Scientific Programming with Python*, New York: Springer,

2011.

- [41] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans.Model.Comput.Simul.*, vol. 8, no. 1, pp. 3-30, 1998.
- [42] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes: The Art of Scientific Computing 3ed*, New York: Cambridge University Press, 2007.
- [43] G. Marsaglia, "Choosing a Point from the Surface of a Sphere," *Ann.Math.Stat.*, vol. 43, no. 2, pp. 645-646, 1972.
- [44] M. Saunders, "Stochastic Search for Isomers on a Quantum Mechanical Surface.," *J. Comput. Chem.*, vol. 25, pp. 621-626, 2004.
- [45] S. Maheshwary, N. Patel, N. Sathyamurthy, A. Kulkarni and S. Gadre, "Structure and Stability of Water Clusters (H<sub>2</sub>O)<sub>n</sub>, n=8-20: An Ab Initio Investigation," *J.Phys.Chem.A.*, vol. 105, pp. 10525-10537, 2001.
- [46] J. Jiang, H.-C. Chang, Y. Lee and S. Lin, "Ab initio Studies of NH<sub>4</sub>+(H<sub>2</sub>O)<sub>1-5</sub> and the influence of Hydrogen-Bonding Nonadditivity on Geometries and Vibrations," *J.Phys.Chem.A*, vol. 103, pp. 3123-3133, 1999.
- [47] J. Douady, F. Calvo and F. Spiegelman, "Structure, stability, and infrared spectroscopy of (H<sub>2</sub>O)<sub>n</sub>NH<sub>4</sub><sup>+</sup> clusters: A theoretical study at zero and finite temperature," *J.Chem.Phys.*, vol. 129, pp. 154305(1-14), 2008.
- [48] E. Adams, "Relations between the constants of dibasic acids and of amphoteric electrolytes.," *J.Am.Chem.Soc.*, vol. 38, pp. 1503-1510, 1916.
- [49] H. Borsook and D. MacFayden, "The Effect of Isoelectric Amino Acids on the pH of a Phosphate Buffer Solution.," *J.Gen.Physiol.*, vol. 13, pp. 509-527, 1930.
- [50] L. Harris and T. Birch, "Zwitterions," *Biochem.J.*, vol. 24, pp. 1080-1097, 1930.
- [51] P. Day, J. Jensen, M. Gordon, S. Webb, W. Stevens, M. Krauss, D. Garmer, H. Basch and D. Cohen, "An effective fragment method for modeling the solvent effects in quantum mechanical calculations," *J.Chem.Phys.*, vol. 105, pp. 1968-1986, 1996.
- [52] J. Jensen and M. Gordon, "On the Number of Water Molecules Necessary to Stabilize the Glycine Zwitterion," *J.Am.Chem.Soc.*, vol. 117, pp. 8159-8170, 1995.
- [53] F. Tortonda, J. Pascual-Ahuir, E. Silla and I. Tunon, "Why is glycine a zwitterion in aqueous

solution? A theoretical study of solvent stabilising factors," *Chem. Phys. Lett.*, vol. 260, no. 1, pp. 21-26, 1996.

- [54] B. Balta and V. Aviyente, "Solvent Effects of Glycine. I. A Supermolecule Modeling of Tautomerization via Intramolecular Proton Transfer.," *J. Comput. Chem.*, vol. 24, pp. 1789-1802, 2003.
- [55] W. Wang, X. Pu, W. Zheng, N. Wong and A. Tian, "Some theoretical observations on the 1:1 glycine zwitterion–water complex," *J. Mol. Struct.*, vol. 626, pp. 127-132, 2003.
- [56] S. Bachrach, "Microsolvation of Glycine: A DFT Study.," *J. Phys. Chem. A*, vol. 112, pp. 3722-3730, 2008.
- [57] H. Basch and W. Stevens, "The structure of glycine-water hydrogen-bonded complexes," *Chem.Phys.Lett.*, vol. 169, pp. 275-280, 1990.
- [58] E. Kassab, J. Langlet, E. Evleth and Y. Akacem, "Theoretical study of solvent effect on intramolecular proton transfer of glycine," *J.Mol.Struct.*, vol. 531, pp. 267-282, 2000.
- [59] A. Chaudhari and S.-L. Lee, "Computational study of glycine–(water)<sub>3</sub> complex by density functional method," *Chem. Phys.*, vol. 310, pp. 281-285, 2005.
- [60] A. Chaudhari, P. Sahu and S.-L. Lee, "Many-body interaction in glycine-(water)<sub>3</sub> complex using density functional theory method," *J.Chem.Phys.*, vol. 120, pp. 170-174, 2004.
- [61] S.-W. Park, S. Im, S. Lee and C. Desfrancois, "Structure and stability of glycine-(H<sub>2</sub>O)<sub>3</sub> cluster and anion: zwitterion vs. canonical glycine," *Int.J.Quantum Chem.*, vol. 107, pp. 1316-1327, 2007.
- [62] R. Ramaekers, J. Pajak, B. Lambie and G. Maes, "Neutral and zwitterionic glycine. H<sub>2</sub>O complexes: A theoretical and matrix-isolation Fourier transform infrared study," *J.Chem.Phys.*, vol. 120, pp. 4182-4193, 2004.
- [63] J. Sun, D. Bousquet, H. Forbert and D. Marx, "Glycine in aqueous solution: solvation shells, interfacial water, and vibrational spectroscopy from ab initio molecular dynamics," *J. Chem. Phys.*, vol. 133, pp. 114508-114517, 2010.
- [64] P. Holtom, C. Bennett, Y. Osamura, N. Mason and R. Kaiser, "A combined experimental and theoretical study on the formation of the amino acid glycine (NH<sub>2</sub>CH<sub>2</sub>COOH) and its isomer (CH<sub>3</sub>NHCOOH) in extraterrestrial ices," *Astrophys.J.*, vol. 626, pp. 940-952, 2005.
- [65] J. Bossa, F. Duvernay, P. Theule, F. Borget, L. d'Hendecourt and T. Chiavassa, "Methylammonium methylcarbamate thermal formation in interstellar ice analogs: a

glycine salt precursor in protostellar environments," *Astron. Astrophys.*, vol. 24, pp. 601-608, 2009.

- [66] J. Bossa, F. Borget, F. Duvernay, P. Theule and T. Chiavassa, "How a usual carbamate can become an unusual intermediate: a new chemical pathway to form glycinate in the interstellar medium," *J.Phys.Org.Chem.*, vol. 23, pp. 333-339, 2010.
- [67] H. Kayi, R. Kaiser and J. Head, "A computational study on the structures of methylamine-carbon dioxide-water clusters: evidence for the barrier free formation of the methylcarbamic acid zwitterion ( $\text{CH}_3\text{NH}_2+\text{COO}^-$ ) in interstellar water ices," *Phys.Chem.Chem.Phys.*, vol. 13, pp. 11083-11098, 2011.
- [68] H. Kayi, R. Kaiser and J. Head, "A theoretical investigation of the low energy conformers of the isomers glycine and methylcarbamic acid and their role in the interstellar medium," *Phys.Chem.Chem.Phys.*, vol. 13, pp. 15774-15784, 2011.
- [69] H. Kayi, R. Kaiser and J. Head, "A theoretical investigation of the relative stability of hydrated glycine and methylcarbamic acid-from water clusters to interstellar ices," *Phys.Chem.Chem.Phys.*, vol. 14, pp. 4942-4958, 2012.
- [70] R. Khanna and M. Moore, "Carbamic acid: molecular structure and IR spectra," *Spectrochim. Acta A*, vol. 55, pp. 961-967, 1999.
- [71] C. Moller and M. Plesset, "Note on the approximation treatment for many-electron systems," *Phys.Rev.*, vol. 46, pp. 618-622, 1934.
- [72] A. McLean and G. Chandler, "Contracted Gaussian basis sets for molecular calculations. I. Second row atoms,  $Z = 11-18$ ," *J.Chem.Phys.*, vol. 72, pp. 5639-5648, 1980.
- [73] R. Krishnan, J. Binkley, R. Seeger and J. Pople, "Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions," *J.Chem.Phys.*, vol. 72, pp. 650-654, 1980.
- [74] T. Clark, J. Chandrasekhar, G. Spitznagel and P. Schleyer, "Efficient diffuse function-augmented basis sets for anion calculations. III. The 3-21 + G basis set for first-row elements, lithium to fluorine," *J.Comput.Chem.*, vol. 4, pp. 294-301, 1983.
- [75] M. Frisch, J. Pople and J. Binkley, "Self-consistent molecular orbital methods. 25. Supplementary functions for Gaussian basis sets," *J.Chem.Phys.*, vol. 80, pp. 3265-3269, 1984.
- [76] G. Alagona, C. Ghio and P. Kollman, "Monte Carlo Simulation Studies of the Solvation of Ions. 1. Acetate Anion and Methylammonium Cation," *J.Am.Chem.Soc.*, vol. 108, pp. 185-

191, 1986.

- [77] M. Masamura, "Ab initio study of the structures of  $\text{CH}_3\text{NH}_3^+$  and  $\text{CH}_3\text{PH}_3^+$  in aqueous solution," *J. Phys.Org.Chem.*, vol. 14, pp. 257-264, 2001.
- [78] M. Meot-Ner (Mautner), "The Ionic Hydrogen Bond and Ion Solvation. 2. Solvation of Onium Ions by One to Seven  $\text{H}_2\text{O}$  Molecules. Relations between Monomolecular, Specific, and Bulk Hydration," *J.Am.Chem.Soc.*, vol. 106, pp. 1265-1272, 1984.
- [79] D. McQuarrie, Statistical Thermodynamics, Mill Valley, CA: University Science Books, 1973.
- [80] K. Riley, M. Pitonak, P. Jurecka and P. Hobza, "Stabilization and Structure Calculations for Noncovalent Interactions in Extended Molecular Systems Based on Wave Function and Density Functional Theories," *Chem.Rev.*, vol. 110, pp. 5023-5063, 2010.
- [81] D. DellAmico, F. Calderazzo, L. Labella, F. Marchetti and G. Pampaloni, "Converting Carbon Dioxide into Carbamate Derivatives," *Chem.Rev.*, vol. 103, pp. 3857-3897, 2003.
- [82] M. Caplow, "Kinetics of Carbamate Formation and Breakdown," *J.Am.Chem.Soc.*, vol. 90, pp. 6795-6802, 1968.
- [83] L. Blanchard, Z. Gu and J. Brennecke, "High-Pressure Phase Behavior of Ionic Liquid/ $\text{CO}_2$  Systems," *J.Phys.Chem.B*, vol. 105, pp. 2437-2444, 2001.
- [84] E. Bates, R. Mayton, I. Ntai and J. Davis Jr., " $\text{CO}_2$  Capture by a Task-Specific Ionic Liquid," *J.Am.Chem.Soc.*, vol. 124, pp. 926-927, 2002.
- [85] K. Jorgensen, T. Cundari and A. Wilson, "Interaction Energies of  $\text{CO}_2$ -Amine Complexes: Effects of Amine Substituents," *J.Phys.Chem.*, vol. 116, pp. 10403-10411, 2012.
- [86] P. Hariharan and J. Pople, "Influence of polarization functions on MO hydrogenation energies," *Theor.Chim.Acta*, vol. 28, pp. 213-222, 1973.
- [88] A. Szabo and N. Ostlund, Modern Quantum Chemistry, Mineola: Dover Publications, 1996.
- [89] G. Munoz Caro, U. Meierhenrich, W. Schutte, B. Barbier, A. Arcones Segovia, H. Rosenbauer, W.-P. Thiemann, A. Brack and J. Greenberg, "Amino acids from ultraviolet irradiation of interstellar ice analogues," *Nature*, vol. 416, pp. 403-406, 2002.
- [90] M. Bernstein, J. Dworkin, S. Sandford, G. Cooper and L. Allamandola, "Racemic amino acids from the ultraviolet photolysis of interstellar ice analogues," *Nature*, vol. 416, pp. 401-403, 2002.

