Reactive molecular dynamics simulations of the surface interactions between atomic oxygen and poly(2,5)-benzimidazole carbon nanotube composite

EH Ellis 27643158

orcid.org 0000-0001-9194-5963

Dissertation accepted in partial fulfilment of the requirements for the degree *Master of Science in Astrophysical Sciences* at the North-West University

Supervisor: Dr LC Square

Graduation July 2023

27643158

Acknowledgements

I would firstly like to acknowledge the National Research Foundation (NRF) under the grant UID 134251, and the National Astrophysics and Space Science Programme (NASSP) for funding this work. I would also like to include the Centre for Space Research (CSR) for always going above and beyond to help students whether it is financially or otherwise.

Secondly, I would like to thank my supervisor, Dr. Lynndle Square, for all the guidance and support that she provided for the last two years. Thanks for all the encouragement and patience that you showed, even when I would derail my whole project pursuing some new detour. It is these detours that led to this project as it stands.

Lastly, I would like to thank my family and friends for their absolute love and support throughout the last two years, including their infinite patience for my ramblings regarding materials, simulations and the effects of atomic oxygen. And a final special thanks to my girlfriend, Henriëtte Vorster, for being a pillar of support in my life and always believing in me. This work is proof of all the support of the people closest to me. Thank you.

Abstract

The low Earth orbit (LEO) environment is home to many species, of which atomic oxygen (AO) is the most abundant. AO is formed via the photo-disassociation of molecular oxygen by solar radiation. In the LEO environment, AO collides with spacecraft material and causes the degradation of the material. Organic materials such as polymers used in space applications are most susceptible to the degradation caused by AO. In this investigation, computational modelling in the form of reactive molecular dynamics (MD) is employed to simulate the interaction between AO and polymer materials and their composite variations. The reactive MD simulations were performed using the reactive force field ReaxFF, which can perform MD simulations of sufficiently large systems within the realm of MD while including the chemistry of the reactions at a somewhat low computational runtime. Using these simulations, the polymer poly(2,5)-benzimidazole (ABPBI) and carbon nanotube (CNT) loaded variations were exposed to AO bombardment to test both the efficacy of pristine ABPBI and the effects of adding CNT against degradation caused by AO bombardment. The simulations in this study indicate that ABPBI and CNT composite variations show high stability against AO bombardment. A second study considered the effects of low temperature in combination with AO bombardment. The outcome of this study shows a decrease in degradation caused by AO bombardment when combined with a decreased simulation temperature for the pristine ABPBI system. The conclusions of this work have given insight into the extent of degradation to ABPBI because of AO bombardment. The favourable results of adding CNTs to the polymer matrix indicate that its addition can further enhance the materials' properties and shows promise for space applications.

Keywords: Poly(2,5)-benzimidazole; Atomic oxygen; Molecular dynamics simulations; Reactive molecular dynamics; ReaxFF; Carbon nanotube composites; LEO space environment.

Contents

List of Figures

List of Tables

Table of abbreviations

A table containing a list of abbreviations that will be used throughout this paper.

ABPBI	Poly(2,5)-benzimidazole
AO	Atomic oxygen
CNT	Carbon nanotube
COMPASS	Condensed-phase optimised molecular potentials for atomistic simulation studies
EOM	Equations of motion
ISS	International Space Station
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
LEO	Low Earth orbit
LJ potential	Lennard-Jones potential
MD	Molecular dynamics
MISSE	Materials International Space Station Experiment
MS	Materials Studio
MWCNT	Multi-walled carbon nanotube
NASA	National Aeronautics and Space Administration
PBI	Polybenzimidazole
PEMFC	Polymer electrolyte membrane fuel cells
QM	Quantum mechanics
SRIM	The Stopping and Range of Ions in Matter
SWCNT	Single-walled carbon nanotube
TRIM	Transport of Ions in Matter

Chapter 1

Polymer Nanocomposite for Space Application

The idea of prolonged human space travel seems like a distant future, but multiple advances have transpired with this singular goal in mind. The scientific community often debates the low Earth orbit (LEO) region to be between 100 km (the lowest altitude for an orbiting spacecraft) and the generally accepted upper limit of LEO to be 2000 km altitude [2]. The risks associated with space travel for extended periods are plenty and multifaceted, from muscle and bone atrophy due to prolonged exposure to microgravity [3] to degradation of the spacecraft structure due to the harsh environment of space. The National Aeronautics and Space Administration (NASA) assessed the risks of human space exploration [4]. It concluded that one of the highest priority problems is finding radiation shielding to protect astronauts and spacecraft [5]. The LEO environment is home to various species and types of radiation. To date, the most commonly used radiation shielding material is aluminium, one of the lightest metals known to man, with a density of only $\rho = 2,7\,\mathrm{g/cm^3}$. Adding aluminium of proper thickness to insulate the spacecraft from radiation adds significant weight to the spacecraft. In 1997, the estimated required shielding to protect against the LEO environment exceeded $50\,\mathrm{g/cm^2}$ [6], adding $500\,\mathrm{kg/m^2}$ of aluminium to the total weight of the spacecraft. Therefore the search for a replacement material that could provide roughly the same properties as, for example, aluminium, with a lower density commenced.

Since the development of synthetic polymers, replacing traditional materials used daily with a polymer that provides the same mechanical properties while reducing the overall weight and cost has become standard practice. In 2001 NASA in cooperation with other entities and agencies, by means of a project called Materials International Space Station Experiment (MISSE), implemented the search for replacements and component development [7]. The MISSE programme has flown over nine missions with over 630 samples tested with variable duration and orientations [8].

1.1 Poly(2,5)-benzimidazole (ABPBI)

The material chosen for this project is poly(2,5)-benzimidazole (ABPBI) (see Figure 1.1). The reason for this choice is twofold: ABPBI is inexpensive to synthesise, compared to other more complex polymers, requiring only a single commercially available monomer (3,4-diaminobenzoic acid (DABA)) and polyphosphoric acid (PPA) as described by Asensio et al. [9]. In addition ABPBI is a thermosetting polymer demonstrating high thermal stability due to a high glass transition temperature (T_g). Further favourable characteristics are that ABPBI has no freezing or melting point, is insoluble in water or organic solvents, and has high chemical resistance [10]. ABPBI and other polymers of the benzimidazole group have been studied for use in Polymer Electrolyte Membrane Fuel Cells (PEMFC) because they are good proton conductors when doped with acceptable acids [11, 12, 13, 14]. Bhowmik and Benedictus [15] state that the polybenzimidazole (PBI) systems have a high resistance to radiation environments due to the abundance of hydrogen groups and their high heat resistance. For further information regarding the favourability of a profusion of hydrogen groups, see Wilson et al. [6] and Adams Jr et al. [16]. The properties mentioned above make ABPBI a good candidate for radiation shielding utilised in space applications and warrant further investigation.

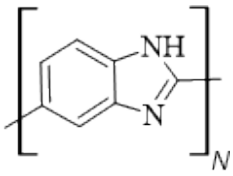
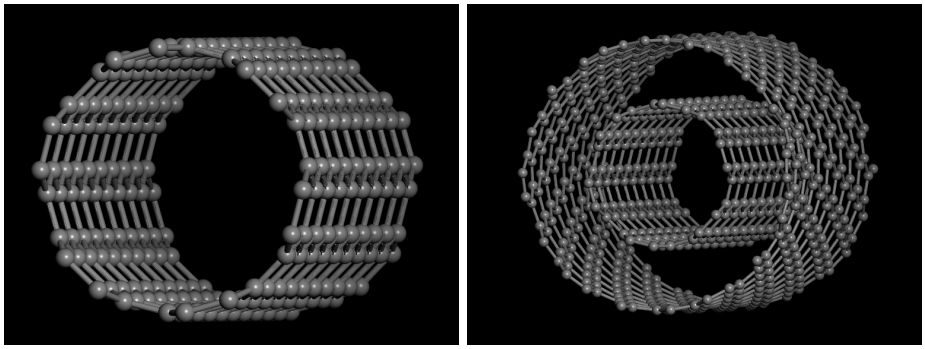


Figure 1.1: The structure of ABPBI.

1.1.1 Carbon Nanotube Composites

Carbon nanotubes (CNTs) were discovered in 1991 by Iijima [17]. Carbon nanotubes can be visualised as a sheet of carbon neatly rolled into cylinders. These cylinders can come in various configurations that favour different properties, for example, metallic or semiconducting electronic properties. Carbon nanotubes also come in varying arrangements, and the most common are single-walled carbon nanotubes (SWCNT) and multi-walled carbon nanotubes (MWCNT), see Figure 1.2. The example in Figure 1.2 of the MWCNT is that of a double-walled CNT and is one of the configurations under the umbrella term MWCNT. When computationally testing the effects of MWCNTs, the double-walled CNT configuration is employed to save on computational resources. The discovery of CNTs sparked research in various fields; physics, chemistry, electrical engineering, and material science [18].

Figure 1.3 shows that since its discovery, there has been a steady increase in research publications about CNT applications over the last three decades due to the extraordinary properties of CNTs: high tensile strength while having low mass density, versatile



(a) Single-walled CNT (b) Double-walled CNT

Figure 1.2: Examples of (a) Single-walled CNT and (b) Double-walled CNT. The double-walled CNT is an example of multi-walled CNT.

electronic behaviour (metallic or semiconducting properties), high heat conductivity and other similarly valuable properties [19, 20, 21]. The adaptable nature of CNTs has made them the perfect candidate for various studies where CNT form part of a constituent in composite materials. There has been a steady increase in using CNT-based composite for electrical, mechanical, space, and other applications [21].

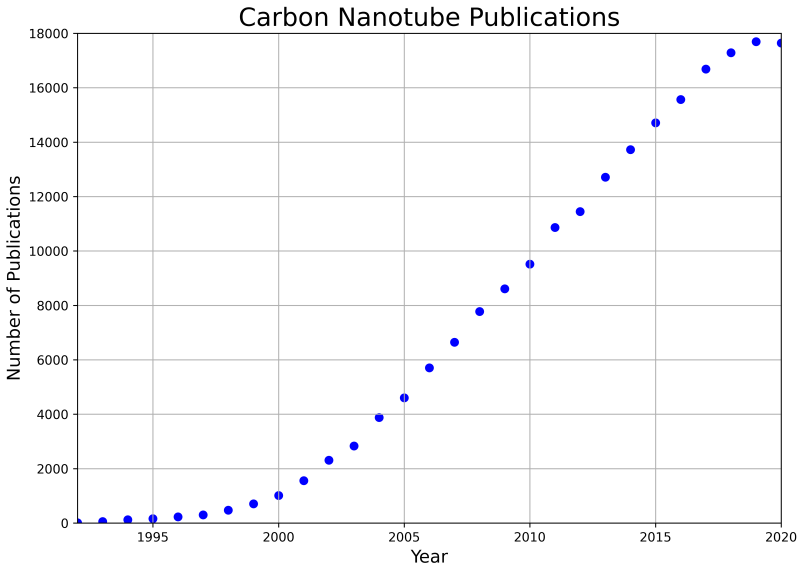


Figure 1.3: A plot of the number of publications regarding CNT from 1991, the year of discovery, to 2020. Data collected from the Science Citation Index (ISI)

1.2 Atomic Oxygen

The low Earth orbit (LEO) environment is home to various threats against external materials of a spacecraft like charge particles, ultraviolet radiation, thermal cycling, debris impacts, micrometeoroid, and atomic oxygen (AO) [1]. Table 1.1 is a list of the LEO environment's species, with the most abundant species being AO [5]; as explained by De-

ver et al. [5], AO is formed via photo disassociation of diatomic oxygen (O_2) from solar radiation with sufficient energy to break the diatomic bond of O_2:

$$O_2 + h\nu \rightarrow 2O^{\cdot}. \tag{1.1}$$

The radical oxygen atom $O^{\cdot}$ is also known as atomic oxygen. The mean free path ($10^4 m$ [22]) at the LEO range is sufficiently long that the re-association and or formation of O_2 or ozone (O_3) from AO is improbable [5, 22]. The impact energy of AO relies on three factors: The orbital velocity of the spacecraft, the thermal velocity of the AO, and the co-rotation velocity of the Earth's atmosphere [5, 23]. In this area, one can consider a circular orbit with an orbital velocity of $V_{sc} \approx 7.7\, km/s$ [5] and an inclination angle of 28.5 degrees [24]. The AO thermal velocity is calculated at $T \approx 1000\, K$ as the average temperature of the thermosphere. The average impact energy of AO is $4.5\, eV \pm 1\, eV$ for an orbit at an altitude of 400 km, and the impact energy decreases with the increase in altitude [5]. The flux of AO also relies on external factors, for example, the solar cycle. The flux of AO recorded by the International Space Station (ISS) at an altitude of 400 km (averaged over a complete solar cycle) is approximately $5.23 \times 10^{13}\, atoms/cm^2 s$ [25]. Atomic oxygen (AO) is reactive and has sufficiently high energy to cause damage to most organic materials, including polymers, via bond breaking [5].

Species	Density (cm^3)	Flux ($atoms/cm^2 s$)	Energy (eV)
O	5×10^8	4×10^{14}	5.3
O_2	1.2×10^7	10^{13}	10.6
O^+	10^5	8×10^{10}	5.3
O_2^+	10^4	8×10^9	10.6
UV (175 nm-190 nm)	$3 \times 10^{-3}\, mW/m^2$		

Table 1.1: Species found at 300 km altitude [1]

1.3 Thermal Cycling

A spacecraft in the LEO environment experiences thermal cycling where the spacecraft surface temperatures can range from approximately $-170°C$ to $123°C$ ($103\,K$ to $396\,K$) [1]. Thermal cycling depends on the material thermo-optical properties and the periods of solar eclipse and solar illumination experienced by the spacecraft [1, 26, 27]. Three sources contribute to the radiant thermal energy a spacecraft would experience: solar radiation, reflected solar radiation from the Earth, and longwave radiation emitted by the Earth's atmosphere heating [1]. Any craft that orbits in the LEO range ($100 - 2000\,km$) has an orbital period of between 90 and 120 minutes, which amounts to 12 to 16 thermal cycles a day, and this can cause thermal stresses on the material, degrading the mechanical properties of the material [27]. A study done by George and Dursch [28] observed thermal cycling-induced micro-cracking in some of the organic composites. The combination of thermal cycling and AO bombardment can cause extended degradation on a polymer

material, such as ABPBI.

1.4 Ion Bombardment and Molecular Dynamics

In this investigation, the initial consideration should be to test ABPBI and composite variations against AO. In the MISSE (mission), test materials were attached to passive experiment containers that were, in turn, connected to the outside of the International Space Station (ISS) in various orientations (ram, zenith and wake) for different periods [29]. Each of these orientations exposes the materials to diverse effects: Ram was in the direction of travel and encounters the highest level of atomic oxygen (AO), wake faces away from the direction of travel and has little to no AO exposure, and Zenith is the direction away from the Earth and thus, had the highest solar exposure [30]. Testing the materials in this manner would require many resources. At the very least, testing the materials would require simulating the LEO environment via experimental apparatus, such as particle accelerators, and therefore, would be costly and time-consuming. As an alternative and allowing for quick and repetitive testing, exploring, and changing various parameters, computational modelling was, therefore, employed. Various types of modelling allow for different time and length scales and material property considerations. For this specific investigation, molecular dynamics (MD) is utilised. MD simulate the position, trajectories, and potential forces of atoms or molecules in a system by numerically solving Newton's equations of motion. Because of the reactive nature of AO concerning polymers, the classical form of MD would not be able to model the interaction between AO and ABPBI adequately, and therefore, a reactive MD model was employed instead. Ashraf et al. [31] have successfully implemented MD simulations for AO bombardment using a reactive empirical-based force field ReaxFF [32, 33], that considers the chemical interactions and bond order (see 2.4.1 for more information). The ReaxFF reactive force field allows for large systems (10^6 particles), including chemical interactions making it the perfect force field for investigating AO bombardment.

The investigation aims to test the ABPBI polymer as a possible radiation shield against atomic oxygen. The testing will be performed using reactive MD simulations to simulate the AO bombardment that the ABPBI system would experience when used as radiation shielding in LEO. Factors considered in this work include small-molecule production, the erosion yield of ABPBI, the effects of a thermostat on the simulation, and the overall mass loss of the ABPBI system due to AO bombardment. The investigation considered the effects of different weight percentages (%wt) of MWCNTs added to the ABPBI systems concerning degradation caused by AO. Lastly, the effects of thermal cycling in combination with AO bombardment are also considered by conducting an AO bombardment simulation at the temperature minimum to compare to the simulation at room temperature. There have been very few computational and experimental reports regarding the ABPBI and its composite for LEO applications; consequently, the outcome of this study will assist in understanding the ability of the polymer composites to withstand the LEO environment

and how it compares with existing shielding materials. Implementing reactive MD using ReaxFF to simulate atomic oxygen bombardment of ABPBI, has to the best of the authors knowledge, not been done before.

Chapter 2

Molecular Dynamics

2.1 Classical Molecular Dynamics

The laws of classical mechanics were initially presumed to describe the motion of planetary objects, but surprisingly they approximated the molecular scale adequately. This approximation was the start of a whole computational methodology, known as *molecular dynamics* (MD). In its simplest form, classical molecular dynamics (MD) refers to the numerical solution of classical equations of motion (EOM) for a group of atoms/particles [34]. This kind of study of a many-bodied system is employed to extract the calculated dynamics from some specific experimental observables [35]. The most accurate representation of a microscopic system, is depicted by quantum mechanics, but quantum mechanical methods are computationally expensive, thus, creating the need for more simplistic approaches.

Consider a system of N particles existing in three dimensions, where the particles are influenced by forces between them, for the classical formulation. The spatial positions and velocities of the particles, as a function of time, are denoted by $\{\mathbf{r}_1,...,\mathbf{r}_N\}$ and $\{\mathbf{v}_1,...,\mathbf{v}_N\}$, respectively. If the forces, $\mathbf{F}_1,...,\mathbf{F}_N$, experienced by the N particles are specified, the classical motion of the system is measurable by Newton's second law

$$m_i\ddot{\mathbf{r}}_i = \mathbf{F}_i, \tag{2.1}$$

where m_i and $\mathbf{v}_i = \dot{\mathbf{r}}_i$ are the mass and velocity of the ith particle (where $i = 1,...,N$); the *dot* notation is used throughout to denote the time derivative of the variable. In principle, the force on the ith particle is a function of all N particle positions, $\mathbf{F}_i = \mathbf{F}_i(\mathbf{r}_1,...,\mathbf{r}_N)$; thus, equation 2.1 comprises a set of dN, where d describes the number of spatial dimensions, coupled second-order differential equations. A set of initial conditions, $\{\mathbf{r}_1(0),...,\mathbf{r}_N(0), \mathbf{v}_1(0),...,\mathbf{v}_N(0)\}$, is required to find a unique solution for equation 2.1, which determines the complete set of positions and velocities of the system at a time t. The complete set of positions and velocities, as a function of time, is determined by

Newton's equation, and therefore the *classical state* of the system is defined at time t. The above EOM, equation 2.1, can only be solved analytically for special cases. Thus, an approximate solution is obtained using iterative numerical procedures [35, 36, 37]. The numerical procedure is known as a *numerical integrator* or a *map*. The accuracy of the numerical solution, is dictated by the time discretisation, t_{step}, more commonly known as the time step. If a real solution could be given for a specific set of initial conditions and be compared to the numerical solution, the two would not agree after enough iterations of the map. The difference is due to the forces, $\mathbf{F}_i(\mathbf{r}_1,...,\mathbf{r}_N)$, being sufficiently nonlinear functions of position [35, 36]. One such numerical solution is, however, still statistically equivalent to the real solution within a bounded error and can sufficiently describe the physical observables as an average [36].

To show the conditions with this statistical equivalence between the numerical and real solution can be achieved, it is required to rewrite equations 2.1 in Hamiltonian form. The Hamiltonian for an N-particle system that only experiences interparticle interactions is

$$\mathcal{H}(\mathbf{p},\mathbf{r}) \equiv \mathcal{H}(\mathbf{p}_1,...,\mathbf{p}_N,\mathbf{r}_1,...,\mathbf{r}_N) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}_1,...,\mathbf{r}_N) \tag{2.2}$$

where $\mathbf{p}_1,...,\mathbf{p}_N$ are the momenta of the particles defined by $\mathbf{p}_i = m_i\mathbf{v}_i$, and $U(\mathbf{r}_1,...,\mathbf{r}_N)$ is the interparticle potential. It is important to note that equation 2.2 is not a fundamental formulation but is derived from the Legendre transformation of the Lagrangian [35]. The Hamiltonian described by equation 2.2 is not explicitly dependent on time and thus, describes the total energy of the system:

$$\mathcal{H}(\mathbf{p},\mathbf{r}) = E. \tag{2.3}$$

The interparticle potential is related to the forces $\mathbf{F}_i$:

$$\mathbf{F}_i = -\frac{\partial U}{\partial \mathbf{r}_i}, \tag{2.4}$$

where $\mathbf{F}_i$ is considered a conservative force [38]. By using Hamilton's equations,

$$\dot{\mathbf{r}}_i = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}_i}{m_i} \tag{2.5}$$

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} = -\frac{\partial U}{\partial \mathbf{r}_i} = \mathbf{F}_i(\mathbf{r}_1,...,\mathbf{r}_N), \tag{2.6}$$

where both equations 2.5 and 2.6 are equivalent to equation 2.1. The classical system can, thus, be described in its entirety by the full set of particle positions and momenta. The full set can be described as a single vector, $\mathbf{x} = (\mathbf{r}_1,...,\mathbf{r}_N,\mathbf{p}_1,...,\mathbf{p}_N)$, also known as the *phase space vector*. The vector, is characterised in a $2d\,N$-dimensional $(6N)$ phase space, where

d is the number of spatial dimensions. A single point in phase space describes a classical state of a system; the phase space is the combination of all possible classical states of a system. There are two principle properties of the EOM equation 2.1 to consider: The EOM is time reversible, meaning they take the same form when, for example, a $t \to -t$ transformation is performed. The importance of the time reversibility is that microscopic physics is independent of the direction of time [36]. The second property is that Hamilton's equations conserve the total Hamiltonian (equation 2.2). This can be demonstrated by taking the total time derivative of $\mathcal{H}$ and using equations 2.5 and 2.6:

$$\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}t} = \sum_{i=1}^{N} \left[\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} \dot{\mathbf{r}}_i + \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} \dot{\mathbf{p}}_i \right] + \frac{\partial \mathcal{H}}{\partial t} = \sum_{i=1}^{N} \left[\frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} - \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} \frac{\partial \mathcal{H}}{\partial \mathbf{r}_i} \right] = 0. \tag{2.7}$$

Because $\mathcal{H}$ is the total energy of a system, equation 2.7 is equivalent to the energy conservation of the system [35, 36]; the energy conservation also provides a critical link between molecular dynamics (MD) and statistical mechanics.

2.2 Ensembles

Statistical mechanics originated from the realisation that the observable macroscopic properties of a system are not heavily reliant on the exact and fully detailed motion of each particle within the system. A method whereby many individual microscopic configurations of a macroscopic system led to the same macroscopic observable properties, also known as *Gibbs' ensemble*, was developed [35]. Gibbs' ensemble is the most general ensemble where both energy and other extensive variables fluctuate. An *ensemble* is an imaginary collection of systems all described by the same Hamiltonian while each system is in a unique microscopic state at any given time. Thus, averaging over a large number of identical systems, each in a different microscopic configuration, is sufficient in describing the observable properties of a system (or the macroscopic observables formulated in terms of the ensemble averages) [36].

A unique state function is required to describe the thermodynamics of an ensemble of systems with specified values of the number of particles N, volume V, and energy E. A state function relies only on the initial and final states of thermodynamic transformation, meaning its change does not rely on the specific thermodynamic path taken between the two states [35] and is obtained by using the First Law of Thermodynamics:

$$E = Q + W, \tag{2.8}$$

where Q is the heat absorbed by the system and W is work done on the system. To examine the change in energy, when a small amount of heat ($\mathrm{d}Q$) is added to the system, combined with a small amount of work ($\mathrm{d}W$) done on the system and because E is considered a

state function, any path may be chosen for the thermodynamic transformation. In this instance, the reversible path is considered

$$\mathrm{d}E = \mathrm{d}Q_{rev} + \mathrm{d}W_{rev}. \tag{2.9}$$

Both Q and W are not state functions and are denoted with the "rev" subscript. The change in entropy of the system is related to the heat absorbed by the system,

$$\Delta S = \int \frac{\mathrm{d}Q_{rev}}{T}, \qquad \mathrm{d}S = \frac{\mathrm{d}Q_{rev}}{T} \tag{2.10}$$

where T is the system temperature, and therefore $\mathrm{d}Q_{rev} = T\mathrm{d}S$. Any work done on the system is measured with respect to two control variables V and N. $P(V)$ is the pressure of the system at a volume V. Mechanical work can be performed on the system by compressing it from a volume V_1 to a volume V_2, where $V_2 < V_1$,

$$W_{12}^{(mech)} = -\int_{V_1}^{V_2} P(V)\mathrm{d}V \tag{2.11}$$

The negative sign in equation 2.11 shows that compression on a system is positive work. The small volume change $\mathrm{d}V$ is equal to an amount of work $\mathrm{d}W_{rev}^{(mech)} = -P\mathrm{d}V$. The explicit dependence of P on V is dropped for simplicity. Be mindful that P is dependent on not only V but also N and E throughout the text. Chemical work can be performed on the system by increasing the number of particles. The chemical potential, μ, of the system describes the energy that can be absorbed or released due to a change in number of particles N of the system, and is also dependent on V and E. By increasing the number of particles from N_1 to N_2 (where $N_1 < N_2$), chemical work

$$W_{12}^{(chem)} = \sum_{N_i=N_1}^{N_2} \mu N_i \tag{2.12}$$

is done on the system. The change in the number of particles must be integer values, ΔN, but when comparing the small changes considered with the total number of particles ($N \sim 10^{23}$), the changes can approximately be regarded as changes $\mathrm{d}N$ on a continuous variable. Thus the small change $\mathrm{d}N$ is described by $\mathrm{d}W_{rev}^{(chem)} = \mu\mathrm{d}N$. From equations 2.11 and 2.12, the total work done on the system is given as

$$\mathrm{d}W_{rev} = \mathrm{d}W_{rev}^{(mech)} + \mathrm{d}W_{rev}^{(chem)} = -P\mathrm{d}V + \mu\mathrm{d}N. \tag{2.13}$$

The total change in energy of the system is:

$$\mathrm{d}E = T\mathrm{d}S - P\mathrm{d}V + \mu\mathrm{d}N. \tag{2.14}$$

Equation 2.14 can be rewritten as:

$$\mathrm{d}S = \frac{1}{T}\mathrm{d}E + \frac{P}{T}\mathrm{d}V - \frac{\mu}{T}\mathrm{d}N. \tag{2.15}$$

Equation 2.15 shows the infinitesimal change in entropy S of the system is directly related to the three control variables of the ensemble, consequently, the state function is just the change in entropy of the system, $S = S(N, V, E)$. The entropy S is a function of N, V, and E, where small changes in N, V, and E causes changes in S, therefore

$$\mathrm{d}S = \left(\frac{\partial S}{\partial E}\right)_{N,V} \mathrm{d}E + \left(\frac{\partial S}{\partial V}\right)_{N,E} \mathrm{d}V + \left(\frac{\partial S}{\partial N}\right)_{V,E} \mathrm{d}N. \tag{2.16}$$

A comparison between equations 2.15 and 2.16 shows that the thermodynamic quantities T, P and μ, obtained by taking the partial derivatives of S concerning each of the control variables:

$$\frac{1}{T} = \left(\frac{\partial S}{\partial E}\right)_{N,V}, \quad \frac{P}{T} = \left(\frac{\partial S}{\partial V}\right)_{N,E}, \quad \frac{\mu}{T} = \left(\frac{\partial S}{\partial N}\right)_{V,E} \tag{2.17}$$

From statistical arguments, the statistical entropy S is related to the number of microscopic states of the system Ω [35],

$$S(N, V, E) = k_B \ln \Omega(N, V, E). \tag{2.18}$$

Because the entropy S is a function of N, V, and E, the number of microscopic states Ω must be the same. The constant k_B, in equation 2.18 is known as the Boltzmann constant. If $\Omega(N, V, E)$ can be determined using the microscopic description of the system, equation 2.18 provides a connection between the microscopic description of a system and the thermodynamic observables [35].

$$\left(\frac{\partial \ln \Omega}{\partial E}\right)_{N,V} = \frac{1}{k_B T} \tag{2.19}$$

The microcanonical ensemble describes a collection of microscopic states with fixed energy and size isolated and following Hamilton's equations of motion [35]. Bearing in mind equation 2.7, a system that obeys Hamilton's equations conserves the total Hamiltonian,

$$\mathcal{H}(\mathbf{x}) = E, \tag{2.20}$$

where E is the total energy of the system. With regard to the microcanonical ensemble, each state is represented as a unique phase space vector $\mathbf{x} = (\mathbf{r}_1,...,\mathbf{r}_N, \mathbf{p}_1,...,\mathbf{p}_N)$, that satisfies 2.20. A function of the Hamiltonian of the system $\mathbf{F}(\mathcal{H}(\mathbf{x}))$, must be chosen that

obeys equation 2.20, and can be achieved by the use of the Dirac δ-function

$$\mathbf{F}(\mathcal{H}(\mathbf{x})) = \mathcal{N}\delta(\mathcal{H}(\mathbf{x}) - E) \tag{2.21}$$

to express the conservation of energy condition, where $\mathcal{N}$ is a normalisation constant. Equation 2.20 defines the hypersurface known as the *constant-energy surface* occurring in phase space. Thus, equation 2.21 expresses that all phase space points of a microcanonical ensemble must be found on the hypersurface and are equally probable [35]. The *ergodic hypothesis* assumes that if a system has infinite time, the entire constant energy hypersurface will be covered [36]. The number of microscopic states $\Omega(N, V, E)$, quantifies the phase space available to a given system and must therefore be proportional to the fraction of phase space defined by the constant-energy hypersurface. The fraction of phase space is defined by equation 2.20, the $(6N-1)$-dimensional area. The partition function of the microcanonical ensemble is obtained by integration of equation 2.21 over the entire phase space. This integration will be over the momentum $\mathbf{p}_i$ and position $\mathbf{r}_i$ of each particle in the system, consequently a $6N$-dimensional integration. The integration over the momentum $\mathbf{p}_i$ is infinite, but the integration of the position $\mathbf{r}_i$ is limited by the constant containing volume defined in a region of space, denoted by $L(V)$. Therefore,

$$\Omega(N, V, E) = M \int \mathrm{d}\mathbf{p}_1 ... \int \mathrm{d}\mathbf{p}_N \int_{L(V)} \mathrm{d}\mathbf{r}_1 ... \int_{L(V)} \mathrm{d}\mathbf{r}_N \delta(\mathcal{H}(\mathbf{p}, \mathbf{r}) - E), \tag{2.22}$$

where M is the overall constant. Equation 2.22 defines the partition function of the microcanonical ensemble. By using the phase space vector $\mathbf{x}$, a simplified version of equation 2.22 is:

$$\Omega(N, V, E) = M \int \mathrm{d}\mathbf{x}\delta(\mathcal{H}(\mathbf{x}) - E). \tag{2.23}$$

The normalisation constant M derivation can be found in Tuckerman [35] as:

$$M = \frac{E_0}{N! h^{3N}}, \tag{2.24}$$

where E_0 defines a thin energy shell above the constant-energy hypersurface and h is Planck's constant. It is worth noting that the simplified version still refers to equation 2.22. Ensembles are described by the variables that are kept constant and the variables that are allowed to fluctuate, also known as the independent and dependent variables of the system, respectively. Table 2.1 defines the three major thermodynamical ensembles pertaining to this investigation.

The partition functions for all other ensembles are derived in much the same way as done for the microcanonical ensemble above. The variables of interest in the ensembles are the number of particles N, volume V, temperature T, energy E, pressure P chemical potential

Ensemble	Independent Variables	Dependent Variables
Microcanonical	NVE	μPT
Canonical	NVT	μPE
Isothermal-isobaric	NPT	μVH

Table 2.1: Different thermodynamic ensembles.

μ, and enthalpy H. The other ensembles besides the microcanonical ensemble, used in this investigation are the canonical ensemble (or NVT ensemble) and the isothermal-isobaric ensemble (or NPT ensemble); consequently N, V, and T are held constant for the canonical ensemble and N, P, and T for the isothermal-isobaric ensemble. For the interested reader, the derivations and explanations of the other ensembles can be found in *Statistical Mechanics: Theory and Molecular Simulations* by Tuckerman [35].

2.2.1 Thermostats

As previously stated, an MD simulation works by integrating Newton's equations of motion of a molecular system. A thermostat is an algorithm that varies the fundamental MD model to keep the temperature constant (on average) [39]. The reason for such an application is numerous, for example, to better the comparison between an MD simulation and experimental simulations or to study a process that is temperature dependent [39]. The temperature T is related to the average kinetic energy of the system. The instantaneous internal kinetic energy $\mathcal{K}$ is

$$\mathcal{K} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i}. \tag{2.25}$$

Thus, by the equipartition theorem,

$$\langle \mathcal{K} \rangle = \left\langle \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} \right\rangle = \frac{1}{2} k_B N_{df} T, \tag{2.26}$$

where k_B is the Boltzmann constant and N_{df} is the number of internal degrees of freedom calculated by,

$$N_{df} = 3N - N_c - N_r. \tag{2.27}$$

N_c is the number of external geometric constraints placed on the system, and N_r is the external degrees of freedom. For equation 2.27, $N_r = 0$ under the influence of frictional and stochastic forces, $N_r = 3$ if periodic boundary conditions are applied and $N_r = 6$ under vacuum boundary conditions. There are several methods with which temperature control can be achieved, such as the constraint method, the extended system method, and the stochastic method. The extended system method was initially formulated by Nosé [40].

Nosé-Hoover Thermostat

The original formulation of a thermostat introduced an artificial dynamical variable $\tilde{s}$, which describes an additional degree of freedom, acting as an external system coupled to the real system. The artificial variable $\tilde{s}$ has an associated "mass" $Q > 0$ and velocity $\dot{\tilde{s}}$ [39]. The added variable $\tilde{s}$ extends the real system by stretching the timescale, creating an extended system[1]. The timescale stretching at time $\tilde{t}$ in the extended system is described by,

$$\mathrm{d}t = \tilde{s}^{-1}(\tilde{t})\mathrm{d}\tilde{t}, \tag{2.28}$$

where $\mathrm{d}\tilde{t}$ is the infinitesimal time interval of the extended system and $\mathrm{d}t$ is the time interval of the real system. For notation simplicity, the time dependence of different functions is omitted. Based on equation 2.28, we can see that all the extended system velocities are amplified by a factor of $\tilde{s}^{-1}$, while the atomic coordinates remain the same, therefore the extended system variables are related to the real system variables as follows,

$$\tilde{\mathbf{r}} = \mathbf{r}, \quad \dot{\tilde{\mathbf{r}}} = \tilde{s}^{-1}\dot{\mathbf{r}}, \quad \tilde{s} = s, \quad \text{and} \quad \dot{\tilde{s}} = \tilde{s}^{-1}\dot{s}. \tag{2.29}$$

The Lagrangian $\mathcal{L}$ of a system is defined as the difference between the total kinetic K_{total} and potential energy U_{total},

$$\mathcal{L} = K_{total} - U_{total} \tag{2.30}$$

The Lagrangian is a function of generalised coordinates and velocities, and considering the extended system, the Lagrangian is selected to be [39, 40]

$$\mathcal{L}(\tilde{\mathbf{r}}, \dot{\tilde{\mathbf{r}}}, \tilde{s}, \dot{\tilde{s}}) = \frac{1}{2}\sum_{i=1}^{N} m_i \tilde{s}^2 \dot{\tilde{\mathbf{r}}}_i^2 - U(\tilde{\mathbf{r}}) + \frac{1}{2}Q\dot{\tilde{s}}^2 - gk_BT_0 \ln \tilde{s}. \tag{2.31}$$

The first two terms of the Lagrangian define the kinetic energy minus the potential energy of the real system, and the last two terms of the Lagrangian describes the kinetic energy minus the potential energy of the extended system (relating to $\tilde{s}$), where T_0 is the specified temperature of the external system and $g = N_{df}$ (see equation 2.26). The conjugate momenta, $\tilde{\mathbf{p}}_i$ and $\tilde{p}_s$, of the extended system related to the coordinates $\tilde{\mathbf{r}}_i$ and $\tilde{s}$, are given by the partial derivatives of the Lagrangian [39, 41]:

$$\tilde{\mathbf{p}}_i = \frac{\partial \mathcal{L}(\tilde{\mathbf{r}}, \dot{\tilde{\mathbf{r}}}, \tilde{s}, \dot{\tilde{s}})}{\partial \dot{\tilde{\mathbf{r}}}} = m_i \tilde{s}^2 \dot{\tilde{\mathbf{r}}}_i \quad \text{and} \quad \tilde{p}_s = \frac{\partial \mathcal{L}(\tilde{\mathbf{r}}, \dot{\tilde{\mathbf{r}}}, \tilde{s}, \dot{\tilde{s}})}{\partial \dot{\tilde{s}}} = Q\dot{\tilde{s}}. \tag{2.32}$$

We can compare these to the real system momenta defined as,

$$\mathbf{p}_i = m_i \dot{\mathbf{r}}_i \quad \text{and} \quad p_s = Qs^{-2}\dot{s}. \tag{2.33}$$

[1] All variables related to the extended system are denoted by the tilde, for example, $\tilde{s}$.

The Hamiltonian of the extended system is obtained from the Legendre transform of the Lagrangian function in equation 2.31 is,

$$\mathcal{H}(\tilde{\mathbf{r}}, \tilde{\mathbf{p}}, \tilde{s}, \tilde{p}_s) = \sum_{i=1}^{N} \dot{\tilde{\mathbf{r}}}_i \tilde{\mathbf{p}}_i - \mathcal{L}(\tilde{\mathbf{r}}, \dot{\tilde{\mathbf{r}}}, \tilde{s}, \dot{\tilde{s}}). \tag{2.34}$$

Considering the extended system, the Hamiltonian is as follows [41]:

$$\mathcal{H}(\tilde{\mathbf{r}}, \tilde{\mathbf{p}}, \tilde{s}, \tilde{p}_s) = \frac{1}{2} \sum_{i=1}^{N} \frac{\tilde{\mathbf{p}}_i^2}{m_i \tilde{s}^2} + U(\tilde{\mathbf{r}}) + \frac{\tilde{p}_s^2}{2Q} + g k_B T_0 \ln \tilde{s}. \tag{2.35}$$

This Hamiltonian evaluates the total energy E of the extended system. From this, the equations of motion are [39]

$$\dot{\tilde{\mathbf{p}}}_i = \tilde{\mathbf{F}}_i \quad \text{and} \quad \dot{\tilde{\mathbf{r}}}_i = \frac{\tilde{\mathbf{p}}_i}{m_i \tilde{s}^2}, \tag{2.36}$$

$$\dot{\tilde{p}}_s = \tilde{s}^{-1} \left(\sum_{i=1}^{N} \frac{\tilde{\mathbf{p}}_i^2}{m_i \tilde{s}^2} - g k_B T_0 \right) \quad \text{and} \quad \dot{\tilde{s}} = \frac{\tilde{p}_s}{Q}. \tag{2.37}$$

Equation 2.36 relates to the physical variables, and equation 2.37 to the $\tilde{s}$-variable. Converting both equations 2.36 and 2.37 to the real-system variables is achieved by [40, 42]

$$\begin{aligned} s &= \tilde{s}, \quad \dot{s} = \tilde{s}\dot{\tilde{s}}, \\ \mathbf{r} &= \tilde{\mathbf{r}}, \quad \dot{\mathbf{r}} = \tilde{s}\dot{\tilde{\mathbf{r}}}, \\ p_s &= \tilde{s}^{-1}\tilde{p}_s, \quad \dot{p}_s = \dot{\tilde{p}}_s - Q^{-1}\tilde{s}^{-1}\tilde{p}_s^2, \\ \mathbf{p} &= \tilde{s}^{-1}\tilde{\mathbf{p}}, \quad \dot{\mathbf{p}} = \dot{\tilde{\mathbf{p}}} - Q^{-1}\tilde{s}^{-1}\tilde{p}_s\tilde{\mathbf{p}}, \\ \text{and} \quad \mathbf{F} &= \tilde{\mathbf{F}}. \end{aligned} \tag{2.38}$$

This yields the real-system variables for equations 2.36 and 2.37, i.e. the Nosé equations of motion:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i}, \tag{2.39}$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \mathbf{p}_i \frac{\tilde{p}_s}{Q}, \tag{2.40}$$

$$\dot{p}_s = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} - g k_B T_0 - p_s \dot{s}, \tag{2.41}$$

$$\dot{s} = \frac{p_s}{Q} s^2. \tag{2.42}$$

The Nosé-Hoover thermostat shows that canonical distributions can generate smooth, deterministic, and time-reversible trajectories and appear under the assumption that the

system dynamics are ergodic. For small and stiff MD systems, which are not dynamically ergodic, the Nosé-Hoover thermostat does not generate the correct distributions, and various solutions were presented to this shortcoming. An interpretation by Martyna et al. [43] states that applying a thermostat to the thermostat momenta p_s, thus regulating the first thermostat, would allow its application to non-ergodic systems. A third thermostat added to govern the momenta p_s of the second thermostat creates a chained system of thermostats. The benefit of such an addition is that even with large systems, the cost of the added chain is lower due to it being one-dimensional. It is only the first thermostat that interacts with an N amount of particles [43]. For the complete derivation of the Nosé-Hoover extended chain thermostat see the *Nosé-Hoover chains: The canonical ensemble via continuous dynamics* by Martyna et al. [43].

Langevin Thermostat

A stochastic model describes a system for which the evolution in time is (partially) random. The stochastic differential equation was the original concept on which the description of a system in contact with an external environment was based [44]. This concept is now more commonly referred to as a thermostat. The initial formulation of Langevin dynamics was a system with a "central" oscillator and a bath of many "small" oscillators linearly coupled to the central one. The central oscillator can be described as a particle that follows Brownian motion, namely a Brownian particle. Ullersma [45] derives the Langevin thermostat by choosing the number of small oscillators to be infinitely large. The one-dimensional motion of a particle in a potential field $V(r)$, with added friction and stochastic terms produces the Langevin equation

$$m\dot{r} = p, \qquad \dot{p} = -\nabla V(r) - \gamma p + \sqrt{2D} f(t), \tag{2.43}$$

where γ is the friction coefficient and D is the intensity of the external random force $f(t)$. γ and D are related by

$$D = m\gamma k_B T_0. \tag{2.44}$$

Here the external force $f(t)$ is a generalised Gaussian stochastic process, "white noise" [46]. Hünenberger [39] states that integrating the Langevin equation of motion produces a canonical distribution of microstates at a temperature T_0. The Langevin equation of motion creates smooth, non-deterministic, time-irreversible trajectories [39].

Samoletov et al. [46] proposed a set of dynamical equations rather than the Nosé-Hoover chain thermostatting dynamics,

$$\dot{\mathbf{r}} = \frac{\mathbf{p}}{m}, \quad \dot{\mathbf{p}} = -\nabla V(\mathbf{r}) - \zeta \mathbf{p}, \quad \dot{\zeta} = g(\mathbf{p}) - \gamma\zeta + f(t). \tag{2.45}$$

As described above $f(t)$ is the generalised Gaussian random process. The above equa-

tion presents a combination between the Nosé-Hoover extended chains thermostat and the Langevin thermostat and is achieved due to the governing thermostatting equation (ζ) being the stochastic differential equation. The difference between this variation on the Nosé-Hoover thermostat and a Langevin thermostat is that although the thermostat variable ζ is subject to stochastic perturbations, the dynamic variables $(\mathbf{r}, \mathbf{p})$ are not directly stochastically perturbed. The new thermostat is referred to as the Nosé-Hoover-Langevin thermostat [46]. The complete derivation and explanation of the Nosé-Hoover-Langevin thermostat are found in, *Thermostats for "Slow" Configurational Modes* by Samoletov et al. [46].

2.2.2 Barostats

Many condensed-phase experiments require a constant pressure, thus reproducing the isothermal-isobaric ensemble (N, P, T) is important in the field of MD [35]. The thermostat maintains control over the temperature of a system, and a barostat provides control over the pressure of a system.

Berendsen Barostat

The Berendsen barostat, derived by Berendsen et al. [47], uses the same principles as the Langevin thermostat by coupling a system to an external pressure bath. The system pressure is regulated by changing the volume V of the system, such that the average internal pressure $\langle P_{int} \rangle$ equals the applied external pressure P_{ext} [36]. The instantaneous internal pressure is

$$P_{int} = \frac{1}{\mathrm{d}V} \left[\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} + \sum_{i=1}^{N} \mathbf{r}_i \mathbf{F}_i - (\mathrm{d}V) \frac{\partial U}{\partial V} \right]. \tag{2.46}$$

The isothermal-isobaric ensemble is created by applying a thermostat and a barostat to an MD simulation. The NPT ensemble has a specific set of equations of motion that includes both a thermostat variable and a barostat variable. There are different ways in which the thermostat and barostat connect to a system, including methods where a thermostat chain is added to the particles and the barostat separately [36]. A more in-depth description of the Berendsen barostat and applications thereof can be found in *Molecular dynamics with coupling to an external bath.* by Berendsen et al. [47].

2.2.3 Monte Carlo Method

Some early formulations of the Monte Carlo method were published by Metropolis and Ulam [48] and Metropolis et al. [49]. An exact definition for the Monte Carlo method has always proven difficult to pin down. Ripley [50] defined methods that rely on probabilistic

solutions as stochastic simulations. The Monte Carlo method relies on randomness to solve problems that might be deterministic in nature. This is achieved by repeated random sampling to find a approximated numerical result. The method in MD applies a random walk through configuration space such that:

$$\mathbf{r}^N \propto \exp\left[-\beta U(\mathbf{r}_1, ..., \mathbf{r}_N)\right], \tag{2.47}$$

where $\exp\left[-\beta U(\mathbf{r}^N)\right]$ is the Boltzmann Factor [35], and $U(\mathbf{r}_1,...,\mathbf{r}_N)$ is the potential energy. β is the reciprocal temperature as described by

$$\beta = \frac{1}{k_B T}. \tag{2.48}$$

The acceptance probability of a trial move from $\mathbf{r} \equiv \mathbf{r}_1, ..., \mathbf{r}_N$ to $\mathbf{r}'$ is given by [49]:

$$A(\mathbf{r}'|\mathbf{r}) = \min\left[1, \mathrm{e}^{-\beta[U(\mathbf{r}')-U(\mathbf{r})]}\right]. \tag{2.49}$$

Thus, the acceptance probability is determined exclusively by the change in potential energy ΔU that results from the move $\mathbf{r}$ to $\mathbf{r}'$. If a trial move decreases the potential energy, the movement will be accepted (with a probability of 1), and if the energy increases the acceptance probability decreases exponentially with the change in energy. This allows a particle within a system to move under the rule that the move lowers the potential energy of the system, thereby producing a dynamic system. Many more examples of the applications of the Monte Carlo method, including variations thereof, appear in *Statistical Mechanics: Theory and Molecular Simulation* by Tuckerman [35].

2.2.4 Verlet Method

The Verlet algorithm is one of the simplest finite difference methods of numerically solving equations of motion [35]. The derivation for the Verlet algorithm starts with a Taylor series expansion to second order of particle i at time $t + \Delta t$ in terms of its position $\mathbf{r}_i$, velocity $\dot{\mathbf{r}}_i$, and acceleration $\ddot{\mathbf{r}}_i$ at time t:

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &\approx \mathbf{r}_i(t) + \Delta t \dot{\mathbf{r}}_i(t) + \frac{1}{2}\Delta t^2 \ddot{\mathbf{r}}_i(t) \\ &\approx \mathbf{r}_i(t) + \Delta t \dot{\mathbf{r}}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t). \end{aligned} \tag{2.50}$$

A similar expansion, written for a velocity-independent variation, starts with $\mathbf{r}_i(t - \Delta t)$:

$$\mathbf{r}_i(t - \Delta t) = \mathbf{r}_i(t) - \Delta t \dot{\mathbf{r}}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t). \tag{2.51}$$

The combination of equations 2.50 and 2.51 produce

$$\mathbf{r}_i(t+\Delta t) + \mathbf{r}_i(t-\Delta t) = 2\mathbf{r}_i(t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t). \tag{2.52}$$

Rearranged, it becomes:

$$\mathbf{r}_i(t+\Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \frac{\Delta t^2}{2m_i}\mathbf{F}_i(t). \tag{2.53}$$

Equation 2.53 is known as the *Verlet algorithm*. Another variation of the Verlet algorithm is derived in the same way that explicitly evolves the positions and velocities. This variation is known as the *velocity Verlet* algorithm:

$$\dot{\mathbf{r}}_i(t+\Delta t) = \mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{2m_i}\left[\mathbf{F}_i(t) + \mathbf{F}_i(t+\Delta t)\right]. \tag{2.54}$$

Both the Verlet and velocity Verlet algorithms are used in MD to calculate the trajectory and velocity of particles within a system. The Verlet and velocity Verlet algorithms are time-reversible, a fundamental symmetry of Hamilton's equations.

2.3 Molecular Dynamics Programme: Materials Studio

Materials Studio (MS) is a software package for simulating and modelling materials [51], and the package has various tools and software components used for molecular modelling and dynamics. Varying force fields can be employed within the packages depending on the task. One such force field is COMPASS [52].

2.3.1 COMPASS Force Field

A force field describes the interatomic interactions in a system. Various force fields were used in this investigation, depending on the required outcome of the simulations. The force field used for all the optimisation steps (see 3.1.1) is the Condensed-phase Optimised Molecular Potential for Atomistic Simulation Studies (particularly COMPASS III). The COMPASS III force field was developed by analysing large molecule databases and systematically fixing deficiencies in parametrisation [53]. The energy expression for COMPASS III is of the consistent force field (CFF)-type,

$$E_{total} = E_{non-bond} + E_{valence}. \tag{2.55}$$

The non-bond energy term ($E_{non-bond}$) is a varied "soft" Lennard-Jones (LJ) potential with an added Coulomb term given by the following equation,

$$E_{non-bond} = \sum_{i>j} \left[D_{0,ij} \left(2 \left(\frac{R_{0,ij}}{R_{ij}} \right)^9 - 3 \left(\frac{R_{0,ij}}{R_{ij}} \right)^6 \right) + \frac{q_i q_j}{4\pi\epsilon_0 R_{ij}} \right], \tag{2.56}$$

where R_{ij} describes the distance between atoms i and j, ϵ_0 is the vacuum permittivity, and q_i is the partial charge on atom i. The LJ parameters ($R_{0,ij}$ and $D_{0,ij}$) are derived from a sixth power combination rule using atom-based parameters [53]. The valence energy term ($E_{valence}$) given below comprises both diagonal and cross terms:

$$\begin{aligned} E_{diagonal} &= \sum_R \frac{1}{2} K_s (R - R_0^2)(1 + C_s(R - R_0) + D_s(R - R_0)^2) \\ &+ \sum_\theta \frac{1}{2} K_b (\theta - \theta_0)^2 (1 + C_b(\theta - \theta_0) + D_b(\theta + \theta_0)^2) \\ &+ \sum_\phi \frac{1}{2} \sum_j B_j (1 - d_j \cos j\phi) + \sum_\omega \frac{1}{2} K_i \omega_{av}^2, \end{aligned} \tag{2.57}$$

$$\begin{aligned} E_{cross} &= \sum_{(R,R')} K_{ss}(R - R_0)(R' - R'_0) + \sum_{(R,R'')} K'_{ss}(R - R_0)(R'' - R''_0) \\ &+ \sum_{(R,\phi)} (R - R_0) \sum_j U_j \cos j\phi + \sum_{(\theta,\theta')} K_{bb}(\theta - \theta_0)(\theta' - \theta'_0) \\ &+ \sum_{(R,\theta,R')} (\theta - \theta_0)(K_{sbs}(R - R_0) + K'_{sbs}(R' - R'_0)) \\ &+ \sum_{(\theta,\theta',\phi)} K_{tbb}(\theta - \theta_0)(\theta' - \theta'_0) \cos\phi \\ &+ \sum_{(\theta,\theta',\phi)} \sum_j \left(V_j(\theta - \theta_0) + V'_j(\theta' - \theta'_0) \right) \cos j\phi \\ &+ \sum_{(R,R',\phi)} \sum_j \left(W_j(R - R_0) + W'_j(R' - R'_0) \right) \cos j\phi. \end{aligned} \tag{2.58}$$

In equations 2.57 and 2.58, R defines the bond length, θ the valence angle, ϕ the torsion angle and ω_{av} the inversion angle. The remaining symbols, defined as the force field parameters, are trained and corrected by comparing them to three massive databases, Maybridge, PoLyInfo and ILThermo. The training process includes COMPASS III simulating all molecules found in the three databases and systematically fixing any deficiencies found in the force field parameters. For more information on the training method and optimisation of the parametrisation for the COMPASS III force field, see Akkermans et al. [53]. COMPASS III is a well-trained force field that works well for the optimisation simulations used in this investigation.

2.3.2 Forcite

Forcite is a module in MS consisting of a collection of molecular mechanics (MM) tools used to investigate a molecular or periodic system [51]. Forcite relies on a force field to approximate the potential energy surface on which atomic nuclei move. The COMPASS III force field (see section 2.3.1) was used in this project. Forcite allows for various molecular mechanics calculations, such as annealing, confined shear, geometric optimisation, quenching, dynamics and even mechanical properties calculations.

Geometry Optimisation

The geometric optimisation task found in Forcite uses the Monte Carlo method to refine the geometry of the structure, until it satisfies a specified criterion, for example, energy convergence and, therefore, adjusts the atomic coordinates until a minimum potential energy surface is achieved. The geometry optimisation task allows the choice between a few algorithms: adjusted basis set Newton-Raphson (ABNR), conjugate gradient, steepest descent, quasi-Newton and Smart algorithm [51]. Some simulations might require a different algorithm to be employed; for example, a structure far from equilibrium would best suit the steepest descent algorithm. A better approach would be a combination, or cascade of different algorithms, where each algorithm gets applied to the most appropriate section in the calculations. The Smart algorithm is a cascade of three methods; steepest descent, ABNR and quasi-Newton [51, 54].

Dynamics

The dynamics task produces the time evolution of a system under thermodynamic constraints. Whichever ensemble is applied, a thermostat (see 2.2.1) and barostat (see 2.2.2) will be applied to the system. The dynamics task is often used to equilibrate a system. Equilibrium is considered to be achieved when either the energy or temperature fluctuates about a constant average. An integration fail-safe is maintained to ensure that if the system energy/temperature should diverge and the simulation becomes unphysical, it is terminated [51].

Anneal

The annealing task uses a dynamics calculation combined with a gradual increase in temperature to a user-specified temperature and then back to the initial temperature. The multiple iterations temperature cycling and the lowest energy structure in each cycle are (optionally geometrically optimised) saved with all relevant data. The annealing process is applied to overcome energy barriers and avoid a structural configuration stuck in a local energy minimum [51]. In Figure 2.1, the temperature cycling of an example annealing

process is visible where the example completes ten temperature cycles between $\sim 300\,\mathrm{K}$ and $\sim 500\,\mathrm{K}$.

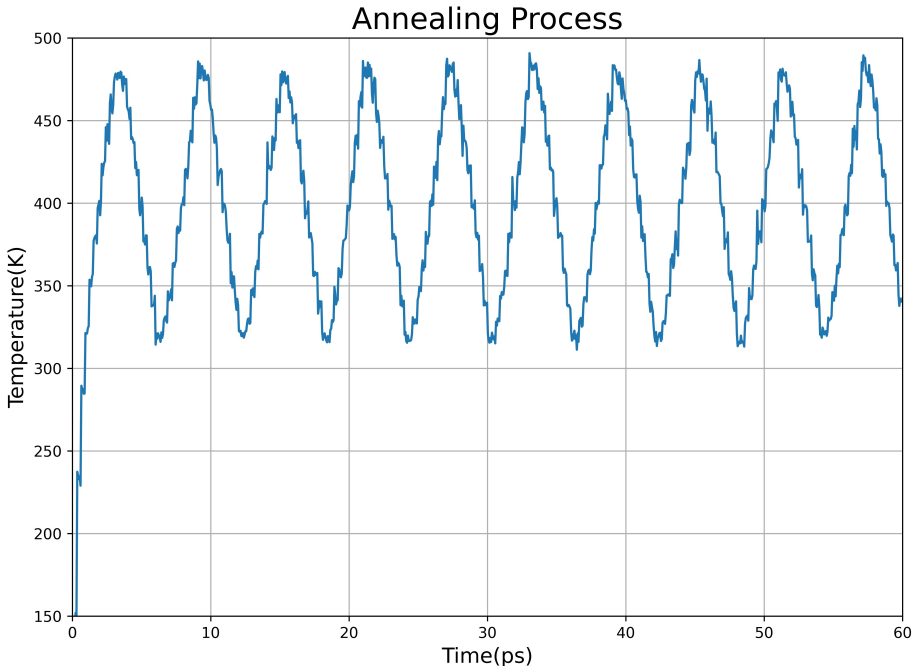


Figure 2.1: A representative graph of the temperature fluctuations and cycling of the system throughout an annealing process.

2.4 Molecular Dynamics Programme: MedeA

The *MedeA* software package [55] provides an environment for atomistic simulations of materials. Numerous simulation engines including Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [56], Vienna Ab Initio Simulation Package (VASP) and others, are integrated into the MedeA environment. The MedeA package introduces a graphical user interface to better visualise complex systems that rely on engines like LAMMPS. The MedeA environment also allows access, for the implementation of different force fields, to the system; one of these force field sets is ReaxFF.

2.4.1 Reactive Force Field (ReaxFF)

The force field used in the bombardment calculations is an empirical interatomic potential that considers bond order to describe the system's chemical kinetics [57]. The ReaxFF reactive force field was created to bridge the gap between system size and chemical reactions. The simulation methods used for large systems are non-reactive empirical methods that do not consider the system's chemical kinetics, and therefore, more chemically accurate systems, and quantum mechanics (QM) methods such as *ab initio* or density functional theory (DFT) are used. The QM methods are computationally expensive and are thus not implemented in large systems. The reactive force field ReaxFF is an empirical poten-

tial that uses structure and energy data from QM methods to train ReaxFF to include chemical reactions [57, 58]. ReaxFF thus allows for large systems without being computationally expensive and includes the necessary parameters to simulate the system's chemical kinetics. ReaxFF is still considered a empirical force field, and thus, the system energy is described by partial energy contributions,

$$E_{\text{system}} = E_{\text{bond}} + E_{\text{over}} + E_{\text{angle}} + E_{\text{tors}} + E_{\text{vdWaals}} + E_{\text{Coulomb}} + E_{\text{specific}}. \tag{2.59}$$

The energy associated with bonds forming between atoms is described by E_{bond}, as a continuous function of the interatomic distance. E_{angle} describes the energies of three-body valence angle strain, while E_{tors} describes four-body torsional angle strain. E_{over} is an energy penalty to prevent over-coordination (the imposition of penalties if carbon, for example, tries to form more that four bonds). E_{vdWaals} and E_{Coulomb} are the non-bond contributions appearing as dispersive and electrostatic forces that do not rely on connectivity but on the distance between atoms. E_{specific} is a system-specific variable that can describe any extra energy consideration required for a particular task, for example, C_2 corrections or lone-pair calculations [58]. For this investigation, the E_{specific} includes a short-range repulsion energy term due to the high-velocity impact. This modification was completed by Ashraf et al. [31] on a previously modified version of ReaxFF used by Vashisth et al. [59] and Vashisth et al. [60]. The modification is required to account for the high-velocity impact that occurs due to AO bombardment. The result should be a strong, short-range repulsive force, which the previous variations of ReaxFF do not include. The short-range repulsive energy, based on the Ziegler-Biersack-Littmark (ZBL) potential and the training and preparation of this specific force field, can be found in Ashraf et al. [31].

2.4.2 Large-scale Atomic/Molecular Massively Parallel Simulator

LAMMPS is an open-source classical molecular dynamics programme [61] from Sandia National Laboratories [1]. The LAMMPS engine is a powerful parallelised programme that allows the calculation of large systems with multiple processing cores [61]. The LAMMPS code permits flexibility regarding the system's treatment, such as the thermostat used for the NVT ensemble.

An explanation concerning some of the commands implemented in this investigation are:

NVE Command

The fix *NVE* command uses the velocity-Verlet time integration algorithm (see 2.2.4). This command updates the position and velocity of the particles to create a system tra-

[1] https://www.lammps.org

jectory that is in accordance with the *NVE* ensemble. A system trajectory describes the time evolution translation of a system given a specified set of initial conditions. The *NVE* command is usually used in combination with a thermostat such as the Langevin thermostat.

NVT Command

The *NVT* ensemble command (denoted as *fix NVT*) has a few arguments to customise the ensemble applied to a specific system. *Temp* and *drag* are the two main arguments of the *NVT* command for this investigation. The temp argument allows the user to set an initial and final external temperature with a temperature-damping parameter (in units of time). The second argument, drag, adds a drag factor to the applied thermostat that prevents or lessens the undesirable effect of oscillating system temperature caused by the thermostat. The larger the drag value, the more damping occurs, but this also alters the Nosé-Hoover equations (see 2.2.1) and must be used sparingly. The fix *NVT* command uses the non-Hamiltonian Nosé-Hoover thermostat for time integration, and gives rise to calculated positions and velocities at every time step.

Langevin Command

The *fix Langevin* command applies the Langevin thermostat; this adds the Langevin thermostat as described by Schneider and Stoll [62] and is usually used in combination with, for example, the fix *NVE* command. The application models an interaction with a background implicit solvent. The calculation for the total force on each atom is as follows:

$$\begin{aligned} F &= F_c + F_f + F_r, \\ F_f &= -\frac{m}{\text{damp}} v, \\ F_r &\propto \sqrt{\frac{k_B T m}{\mathrm{d}t \,\text{damp}}}. \end{aligned} \tag{2.60}$$

Here F_c is the conservative force computed by the inter-particle interactions, F_f is the frictional drag proportional to the particle velocity v, and F_r is the force due to the solvent atoms (at temperature T) bumping the particle randomly. F_r is also reliant on the time step $\mathrm{d}t$ of the simulation. The initial and final temperatures and the damping parameter are the arguments for the command. The first two arguments define the desired temperature for the system at the start and at the end of the simulation. The damp parameter (specified in units of time) determines how fast the system temperature is relaxed. If the damp parameter is set to be 100 time units, it signifies the specified time in which the temperature should be relaxed by the thermostat.

Complete explanations of all commands used within the LAMMPS programme can be found in the LAMMPS manual[1].

[1] https://docs.lammps.org/Manual.html

Chapter 3

Poly(2,5)-benzimidazole and Composite Systems

3.1 Computational Details

The following tasks are required to model the material for AO bombardment: firstly, creating and optimising the materials and variations thereof; secondly, creating a simulation programme that can accurately describe the AO bombardment with the proper energy taken into consideration; the final task is to properly describe the interactions between the materials and AO, that includes the chemical reactions that are caused by AO. These tasks and the solutions to them, are described below. The computational resources and national license for a portion of the above-mentioned tasks were provided by the Centre for High-Performance Computing (CHPC). Other tasks, unable to use the CHPC computational nodes were completed on an 8-core desktop processor. Due to the difference in performance, the latter has a longer overall runtime for the more significant simulations.

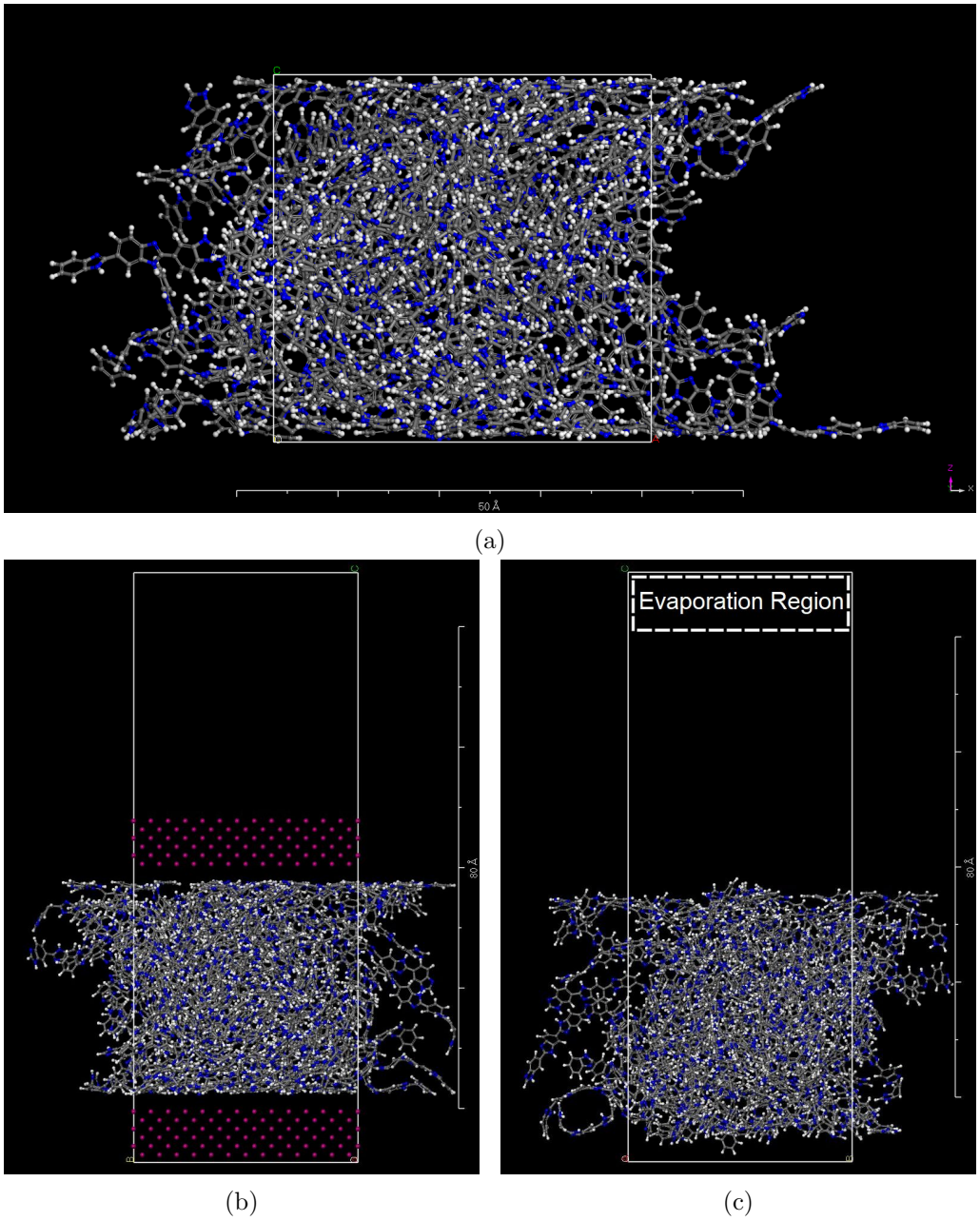
3.1.1 Simulation Cells

To replicate the material, three-dimensional (3D) simulation cells are created using a Dassault Systèmes BIOVIA software programme, Materials Studio (MS) [51]. Each of the 3D cells contained ABPBI with a chain length of 10 repeating units with random torsion angles. Two neatly packed iron layers were created to stack on the bottom and top of the ABPBI systems; a layer with $13 \times 13 \times 6$ iron atoms (cell size $37\,\text{Å} \times 37\,\text{Å} \times 9\,\text{Å}$), and a layer with $14 \times 14 \times 6$ iron atoms (cell size $40\,\text{Å} \times 40\,\text{Å} \times 9\,\text{Å}$). The cells' horizontal (x and y) dimensions are set to approximately match the x and y dimensions of either one of the two iron layers. Starting with a concentration of $0.6\,\text{g/cm}^3$ the cells were congested until a specified density ($2.0\,\text{g/cm}^3$) was obtained, and four cells were created; pristine ABPBI, a 10%wt multi-walled carbon nanotube (MWCNT) loaded ABPBI, a 15%wt MWCNT

loaded ABPBI and a 20%wt MWCNT loaded ABPBI. The specific loading percentages of MWCNT added to the systems are selected to establish the mitigation effects of MWCNT for discreet loading percentages. A precise size of CNT is added with varying amounts of ABPBI chains (see Table 3.1) to achieve the correct weight percentage for each combination. To save on computational resources a double-walled CNT is used which is referred to as an MWCNT. The MWCNTs, are added to the amorphous cell with random orientations. Rahmani et al. [63] showed that randomly oriented nanoparticles, for example, MWCNTs, generally perform better. Some areas in which improved performance have been reported are erosion yields, surface damage, and a reduction in the AO penetration depth [63]. The cells have fixed boundary conditions along the z-axis and periodic boundary conditions along the x and y-axis. This configuration of boundary conditions forces the amorphous ABPBI composites to form a flat surface in preparation for the ion bombardment (see Figure 3.1a). To perform a bombardment simulation, sufficient vacuum space is required above the ABPBI composites to accommodate the added particles and leave some space for reactions. New cells created with the Build Layer function provided in MS, the above-specified iron layer added to both sides of the ABPBI composites in the z-direction, and 40 Å vacuum space is added to the top layer of iron (see Figure 3.1b). The newly created cells have periodic boundary conditions along all three axes. However, by locking the iron atoms to their Cartesian coordinates, the iron layers act as fixed boundaries for the amorphous ABPBI, retaining the flat surface of the ABPBI composites.

For this investigation, the cells were subjected to the following order of optimisation calculations; geometric optimisation, annealing, and two separate dynamics calculations. The Forcite package, included in BIOVIA Materials Studio, provides practical tasks, for example, geometric optimisation, annealing, dynamics and confined shear. The first dynamics calculation is a 100ps canonical (NVT) analysis at room temperature $T = 298\,\mathrm{K}$. Both iron layers were removed from the cells, leaving only the ABPBI composites and vacuum space in the z-direction before the final dynamics calculation. The last dynamics task performed was a 1 ns NPT calculation at room temperature $T = 298\,\mathrm{K}$ and atmospheric pressure $P = 1\,\mathrm{atm}$, to allow the ABPBI composites to relax and fill some of the vacuum space. This, in turn, lowers the density of the composites close to the determined density of ABPBI, $\rho \approx 1.5\,\mathrm{g/cm^3}$. Both the NVT and NPT calculations use the Nosé-Hoover-Langevin thermostat, with the NPT calculation employing the Berendsen barostat for pressure control.

All optimisation calculations used the COMPASS III force field. The COMPASS III force field is a general all-atom *ab initio* force field that is suitable for the systems under investigation (see 2.3.1 for more about COMPASS III). The result is a well-optimised simulation cell with a somewhat flat surface and adequate space for a bombardment simulation, an illustration hereof can, be seen in Figure 3.1c. Table 3.1 shows all the final dimensions, densities and composite ratios of the optimised structures. The final densities, as seen in Table 3.1, are approximated by measuring the "cubic" space that the ABPBI composites occupy within the cell and calculating the density by utilising the empirical mass of each



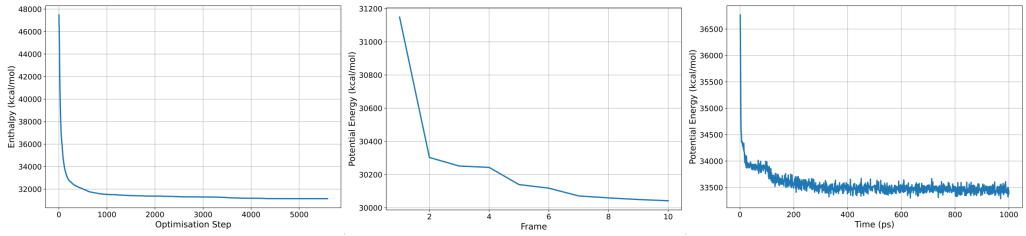
(a)

(b) (c)

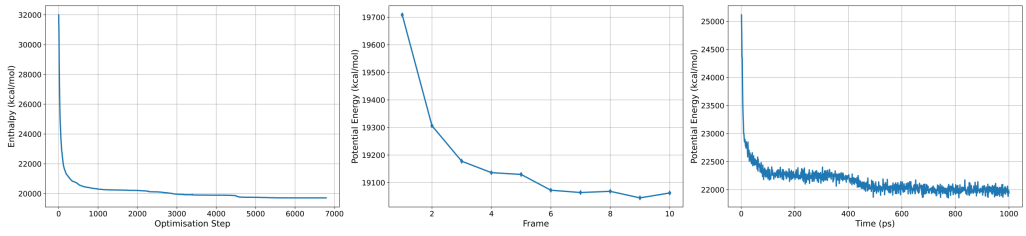
Figure 3.1: The pristine ABPBI simulation cell at different stages of creation: (a) The two-directional periodic simulation cell filled with ABPBI to a density of $\rho = 2.0\text{g/cm}^3$ (the z-direction has fixed boundary conditions), (b) the simulation cell after the build layer step where iron layers are added, above and below the ABPBI polymer acting as *fixed boundaries*, with added vacuum space needed for the bombardment simulation and (c) the simulation cell after all optimisations are completed, with only the ABPBI remaining. The blue, grey, white and magenta spheres represent nitrogen, carbon, hydrogen and iron, respectively.

system.

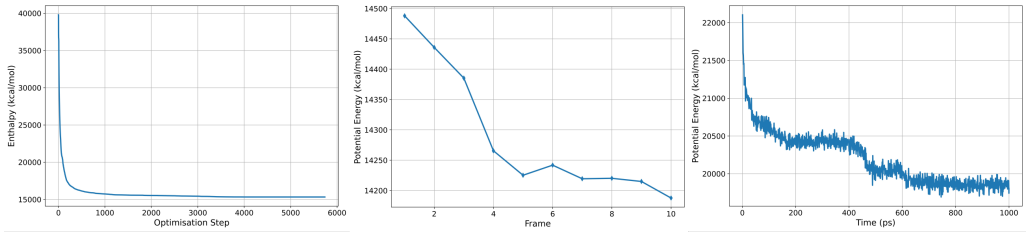
The results of the optimisation steps can be found in Figure 3.2. The figure shows the energy during three different steps of the cell optimisation process. The first graph on the left for all four systems is the geometric optimisation energy (enthalpy), which shows a simple energy minimisation leading to a better-optimised structure. The middle graph is



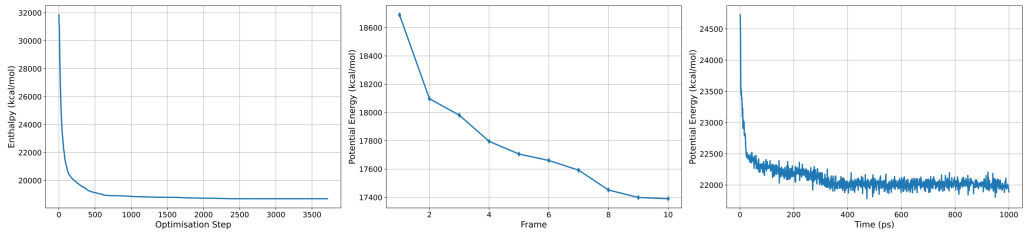
(a) ABPBI



(b) ABPBI/MWCNT10



(c) ABPBI/MWCNT15



(d) ABPBI/MWCNT20

Figure 3.2: Optimisation energies for the four systems via geometric optimisation followed by annealing process and dynamics calculation (*NPT*).

the annealing energy as a function of annealing cycles; up to this point, and for all four systems the energy is the lowest at the end of the annealing process. Keep in mind that in this step of the optimisation process, the iron layers have not yet been removed. After the removal of the iron layers, the final two steps are completed. The final graph of Figure 3.2 is the energy as a function of calculation time and shows the energy minimisation of the structure after the iron layers have been removed. At the end of the final optimisation step, the lowest energy structure is the ABPBI/MWCNT15, while the highest energy system is the pristine ABPBI. The final minimised energy at the end of the *NPT* calculation for all systems is much higher than the final minimised energy at the end of the annealing process. This difference in energy is because after the iron layers were removed a whole new volume could be occupied by the composite system leading to different configurations that consist of higher potential energies. To fully optimise all four structures to the lowest possible energy achievable, the composites will move apart until it fills the entire vacuum space and has no coherent structure. The compromise for total energy minimisation is a

System	Simulation Cell Size ($Å^3$)	Number of Particles (ABPBI/MWCNT)	Approx. Simulated Density (g/cm^3)
pristine ABPBI	$39 \times 39 \times 102$	6864/0	1.393 ± 0.073
ABPBI/MWCNT10	$37 \times 37 \times 100$	5148/416	1.468 ± 0.086
ABPBI/MWCNT15	$38 \times 38 \times 99$	5280/694	1.449 ± 0.085
ABPBI/MWCNT20	$39 \times 39 \times 103$	6072/1110	1.435 ± 0.072

Table 3.1: Simulation cell details of all four ABPBI composite systems.

structure with a somewhat flat surface and vacuum space to allow for atoms to be added to the system, as well as room for the composite systems to move as the force fields dictate.

3.1.2 Bombardment Simulation

After creating and optimising the simulation cells, the cells are ready for molecular dynamics simulations (MD). Although BIOVIA Materials Studio provides many helpful tools in molecular modelling, computational chemistry and tools for MD, a different programme, namely MedeA [55], is used. MedeA employs the LAMMPS engine [56], which is more suited for large-scale MD simulations. LAMMPS also includes code that is required to run MD simulations using a custom ReaxFF force field, which is crucial for this investigation. The computational method used for the bombardment simulation (see Appendix A.2) was implemented to produce similar results as previous works by Ashraf et al. [31], Rahmani et al. [63], and Rahnamoun and Van Duin [64]. The simulation cell was imported into the MedeA Environment [55], which includes tools and engines needed to run large-scale MD simulations. The composites are anchored to the simulation cell by "freezing" the bottom $\sim 20\%$ of the composites' atoms (frozen atoms cannot move during a simulation). The reason for anchoring the composite systems is to prevent the material from freely moving around in the cell. A final NVT dynamics simulation is performed for a total of $t = 10\,\text{ps}$ to ensure the system is equilibrated before the bombardment simulation (see Appendix A.1). The NVT uses a Nosé-Hoover thermostat with $T_{\text{initial}} = T_{\text{final}} = 298\,\text{K}$, $t_{\text{damp}} = 100\,\text{fs}$, and $f_{\text{drag}} = 1.0$ as thermostat variables (see 2.4.2).

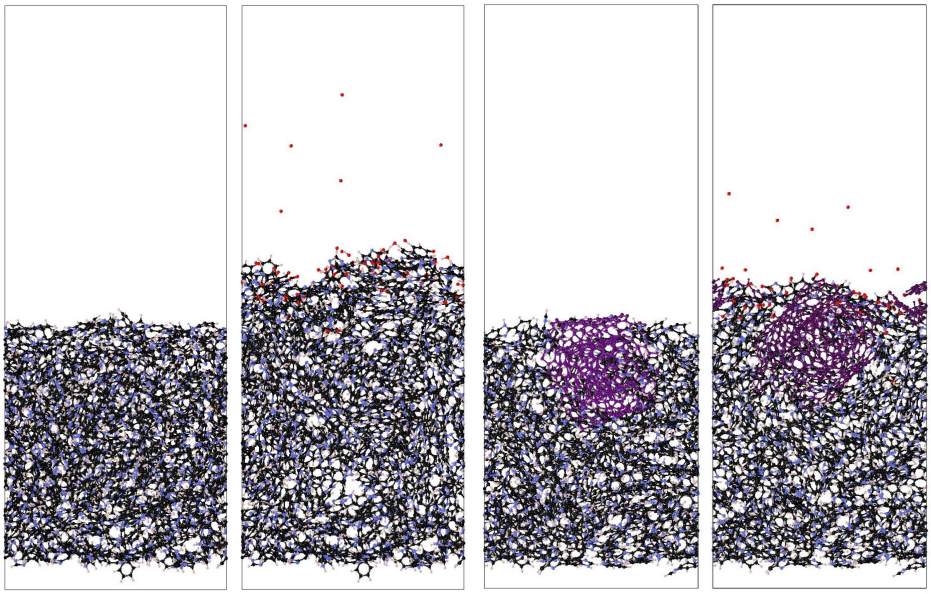
To regulate the temperature during the bombardment simulation a suitable thermostat is required. Thus, for the bombardment simulation, an NVE ensemble is used with a Langevin thermostat included. The ReaxFF forcefield used in this simulation was created and trained for use by Ashraf et al. [31]. The total simulation time t_s for the bombardment simulation is $t_s = 40\,\text{ps}$ with a time step of $t_{\text{step}} = 0.1\,\text{fs}$ (this equals a total of 400000 time steps). The time step must be significantly small when using the ReaxFF force field to ensure accurate simulations and convergences of the system energy [65]. An AO was added to the system every 200 fs with an energy of $E_{AO} = 4.5\,\text{eV}$. A total of 200 AOs were bombarded onto each of the four systems, giving a fluence of $f_{AO} \approx 10^{15}\,\text{AO/cm}^2$ (f_{AO} varies between systems, see Table 3.2). The AO fluence is calculated as the amount of AO per bombarded area. The bombardment simulation uses an NVE/Langevin thermostat

ensemble with thermostat variables as $T_{\text{initial}} = T_{\text{final}} = 298\,\text{K}$ and $t_{\text{damp}} = 10\,\text{fs}$. An extra simulation is performed on the pristine ABPBI system, without a thermostat, to test the effect of a thermostat on the bombardment simulation. An evaporation region is added to the top of the simulation cell (approximately 10 Å, see Figure 3.1c) that removes any atoms/molecules found in the region every 1000 time steps. The data collected from the evaporation region is used to calculate the mass loss of the system, and the small molecule production of the systems via AO. An analysis is also done on the final structures of all four systems regarding the AO that remains at the end of the bombardment simulation. This analysis shows the penetration depth of AO in all four systems.

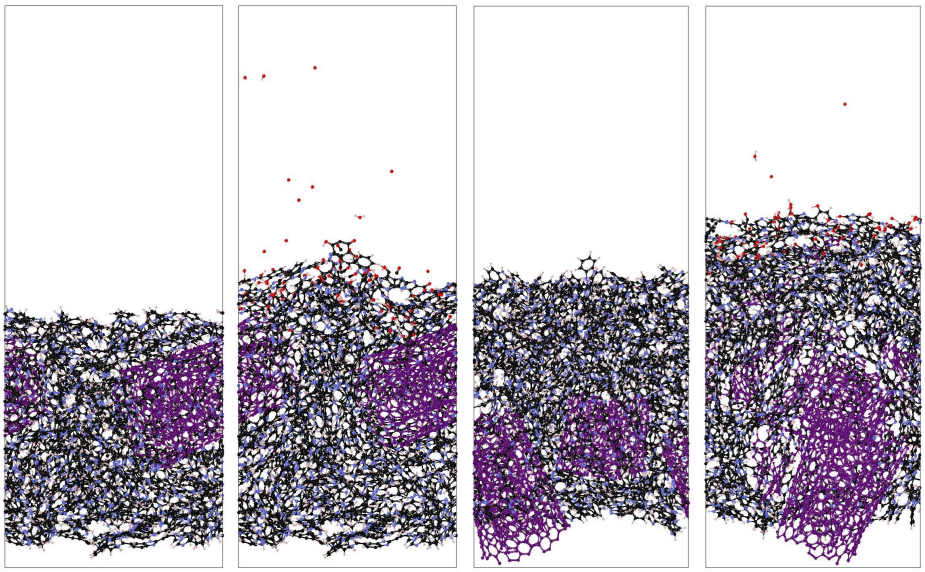
The third part of the study, is testing the effects of thermal cycling by performing an AO bombardment simulation at the extreme minimum temperature. Ashraf et al. [31] showed that performing an AO bombardment simulation at the thermal cycle maximum temperature yields a similar result as the 298 K simulation. The effects of low system temperature, in comparison, are of more interest concerning AO bombardment, and therefore the minimum temperature of 103 K is considered as part of this study.

System	Area(Å^2)	Fluence (AO/cm^2)
Pristine ABPBI	1517.95	1.318×10^{15}
ABPBI/MWCNT10	1340.00	1.493×10^{15}
ABPBI/MWCNT15	1479.79	1.352×10^{15}
ABPBI/MWCNT20	1547.09	1.293×10^{15}

Table 3.2: Fluence for all four systems.



(a) ABPBI (b) ABPBI/MWCNT10



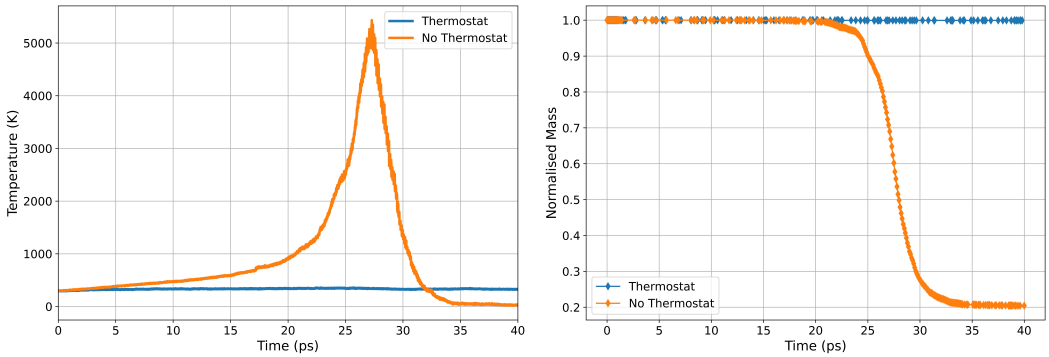
(c) ABPBI/MWCNT15 (d) ABPBI/MWCNT20

Figure 3.3: The four systems' initial structures and final structures at the end of the bombardment simulation. The black, blue, white, red, and magenta spheres represent carbon, nitrogen, hydrogen, oxygen, and MWCNTs.

3.2 The Effects of a Thermostat

A thermostat controls the temperature of a system by varying the kinetic energy of the particles that the thermostat is applied to. Rahnamoun and Van Duin [64] conducted a test to quantify the effects of the Berendsen thermostat with different damping coefficients and concluded that the addition of a thermostat with any damping coefficient keeps the system at a constant temperature which caused less degradation of their simulated materials, Kapton, Teflon, Silica, Kapton-POSS. The effect of a thermostat is thus essential in mimicking a more realistic system. Figure 3.4a displays the difference in system tempera-

ture for two simulations of the optimised pristine ABPBI, one with and the other without a Langevin thermostat (as described in 3.1.2). The system without a thermostat shows an exponential increase in temperature to a maximum of $\sim$ 5433K and then a sharp decrease to 0 K. The increased temperature is due to the AO impact on the system, where severe degradation causes mass loss via the evaporation region. The sharp drop-off is due to most of the material leaving the system until nothing of the system remains except the "frozen" atoms at the bottom of the simulation cell. In Figure 3.4b, the material loss is observed starting from $t_s \approx$ 20ps and continuing until no mass remains at $t_s \approx 33$ ps. As a result of the frozen atoms immobility, the system temperature drops to 0 K in Figure 3.4a, and the final mass of the system does not decrease to zero in Figure 3.4b. The simulation with a Langevin thermostat (with a damping coefficient of $t_{\text{damp}} = 10$ fs) shows a much more consistent temperature with an average of $[331.38 \pm 11.18]$ K over the total simulation time.



(a) (b)

Figure 3.4: A representative graph of the (a) temperature difference between a system with a thermostat added and one without. The peak for the system without a thermostat is at $t_s = 27.24$ ps, where the total loss of particles is high enough that the system temperature decreases to 0 K due to total material loss; (b) A graph of the difference in normalised mass between the two systems.

System	Average Temperature (K)	Final Temperature (K)
pristine ABPBI	331.38 ± 11.18	328.12
ABPBI/MWCNT10	350.40 ± 15.24	338.31
ABPBI/MWCNT15	347.28 ± 16.22	342.20
ABPBI/MWCNT20	345.82 ± 18.51	347.05

Table 3.3: System temperature for all four systems

Figure 3.5 shows the temperature for all four systems under consideration as a function of simulation time. The average temperature of the four systems over the bombardment simulation is in Table 3.3. The oscillating effect of the Langevin thermostat can be seen in Figure 3.5, where the four systems have a temperature increase from the initially set temperature $T_{\text{initial}} = 298$ K. The temperatures slightly decrease over the rest of the bombardment simulation until the four systems reach a final temperature of approximately

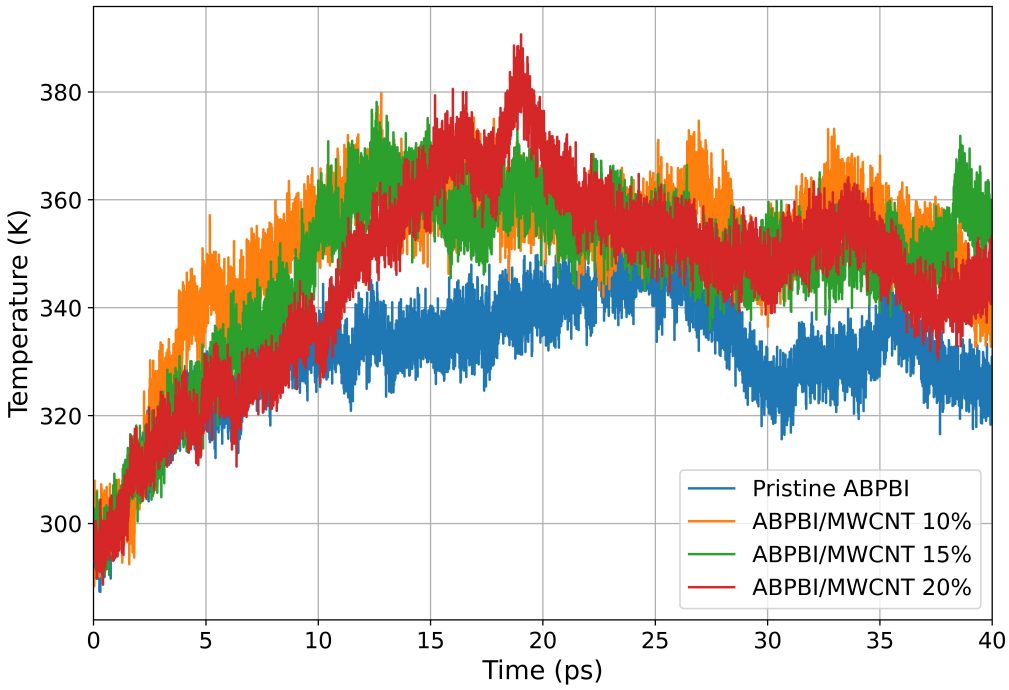


Figure 3.5: System temperature of the four simulations as a function of simulation time.

337 K. The temperature increase is due to the incident AO on the systems, followed by a slight decrease in temperature over time. The lowest system temperature is shown by the pristine ABPBI system, with the overall lowest average temperature and the lowest final temperature (see Table 3.3). This could be due to the absence of an MWCNT or because the ABPBI system is the smallest.

3.3 Atomic Oxygen Penetration Depth

The atomic oxygen (AO) distribution, remaining within the ABPBI systems, was analysed after the bombardment simulation. In Figure 3.6 the distribution across the z-axis of the simulation cells for all four systems is observed; the figure indicates that most AO remaining in the ABPBI systems appear within the first ~ 15 Å from the surface of the ABPBI systems (indicated by the black line found at $z = 0$ Å). The trend shows approximately the same narrow Gaussian graph for all four systems. The ABPBI/MWCNT15 system gives a slightly lower peak maximum and a broader distribution overall. This system experienced a deeper penetration depth of AO compared to the other three. The ABPBI/MWCNT10 graph shows the narrowest Gaussian, indicating that most of the AO bombarded onto the system was stopped within the first 10 Å of the material. The ABPBI/MWCNT20 system also shows that most of the AO were stopped well within the first 11 Å of the material. The system also has a peak at a shallow depth of $z = 2.6972$ Å. Although the differences between the four systems are notable, they are still of the same order of magnitude.

An external programme, was utilised for a proportionate comparison of the penetration

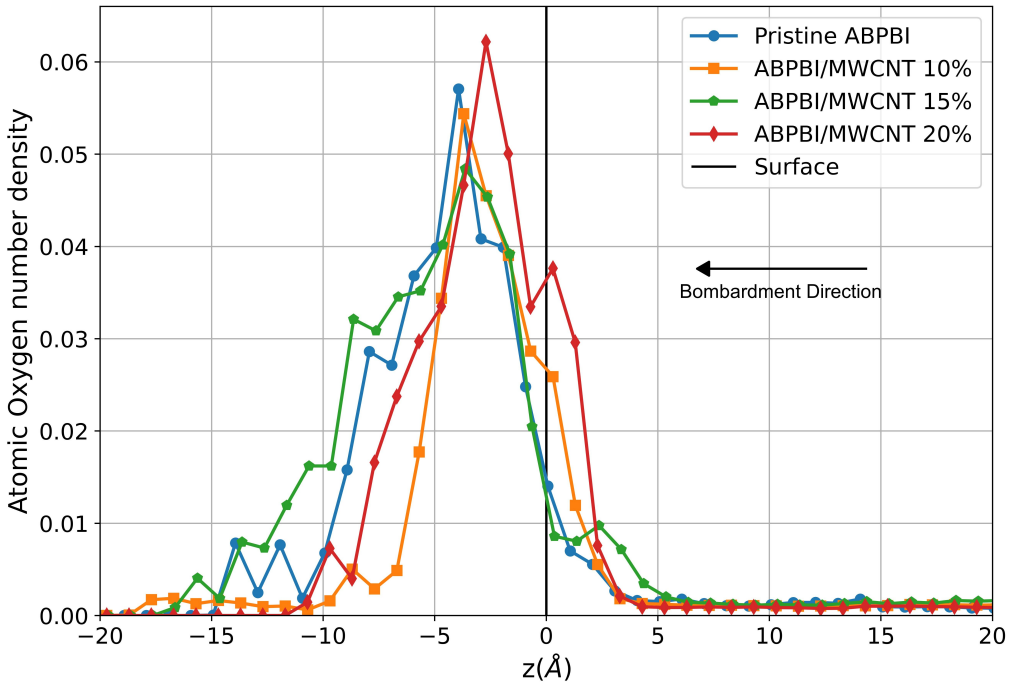
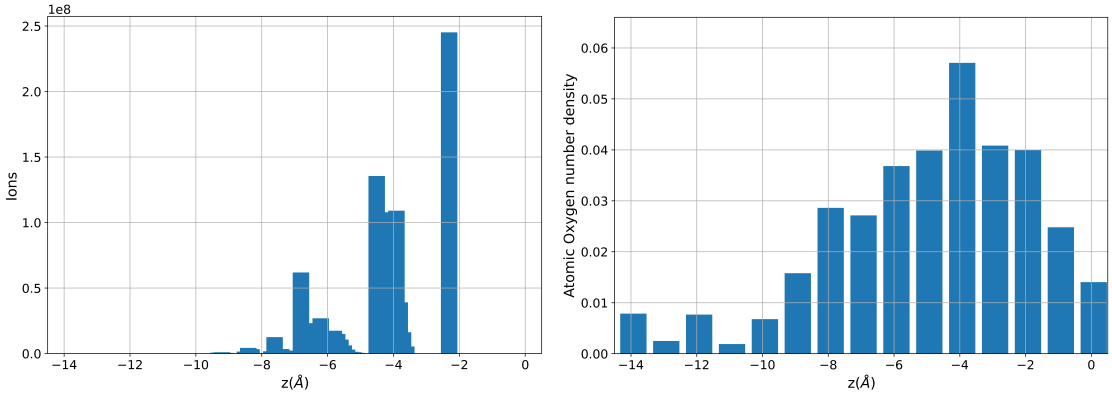


Figure 3.6: The number density of AO remaining in the four systems under consideration at the end of the bombardment simulation (t=40 ps). The surface of all four systems at $z = 0$ Å.

depth of AO in the systems. The Stopping and Range of Ions in Matter (SRIM) programme and a sub-programme, Transport of Ions in Matter (TRIM), use Monte Carlo simulations to calculate the interactions of energetic ions with a specified amorphous material [66, 67]. Figure 3.7a shows the penetration depth of AO in amorphous ABPBI, as simulated using the SRIM/TRIM programme. The surface of the simulated ABPBI is found at $z = 0$ Å and towards the negative integers is deeper into the material. TRIM simulates bombarding the amorphous ABPBI with 99999 AO at an energy of 4.5 eV, and the result shows that the AO does not reach further than 8Å into the material. The input value of 99999 AO was used to get a well-defined shape to compare with the ReaxFF MD simulated penetration depth profile. A deeper penetration depth was discovered, from the bombardment simulations of the ABPBI system, as shown in Figure 3.7b. This difference can be due to various reasons, for example, the build methods, the bombardment calculations and the reaction at impact site calculation. A general conclusion from these two figures is that most of the AO, after bombardment simulation, will be found within the first ~ 9 Å of the material. The maximum peak of AO for the pristine ABPBI simulation appears at $z = 4$ Å and $z = 2.3$ Å for the TRIM calculation. It is important to note that the TRIM programme does not consider the reactions occurring due to the AO impact but predicts the stopping power of the specified material, in this case, ABPBI. Regardless, a moderately similar distribution occurs in both simulations, with most of the AO only reaching a depth of $z \approx 10$ Å; in both simulations.



(a) TRIM simulation of ABPBI (b) ReaxFF MD simulation of ABPBI

Figure 3.7: (a) The penetration depth of AO into an ABPBI system simulated using the SRIM/TRIM programme and (b) the penetration depth of AO of our ABPBI system. The surfaces of the ABPBI systems were found at $z = 0\,\text{Å}$.

3.4 Mass Loss and Small Molecule Production

The small molecules produced by the AO bombardment simulation appear in Table 3.4. Only molecules or atoms with sufficient energy to move into the evaporation region are counted towards the produced molecules and, by extension, the mass loss of the systems. Some of the molecules produced have little to no energy left to move away from the main body of atoms towards the evaporation region. The molecules produced by the AO bombardment are H_2O, OH and single H atoms. In both the ABPBI/MWCNT10 and ABPBI/MWCNT20 systems, a single O_2 is produced. The production of the single O_2 for both systems is due to the scale length of the simulated systems being smaller than would be found in the real world; thus, AO is more likely to recombine into O_2 when two AOs are close. The reason for the production of H_2O, OH and single H atoms relies on the binding energy of the different atoms found within the ABPBI systems. The lowest binding energies are that of hydrogen bonds. The three main mechanisms whereby AO bombardment degrades a polymer are hydrogen abstraction, oxygen replacement and oxygen addition [5].

It is revealed in Table 3.4 that more than half of the AO bombarded remains in the system at the end of the bombardment simulation; therefore, polymer composite system surfaces have an increased oxygen content after being exposed to AO.

The data collected from the evaporation region provides the molecules that are removed from the systems every 1000th time step. A mass loss calculation, is performed from this data as a function of simulation time. Figure 3.8 reflects the comparison for the normalised mass loss of all four systems during the AO bombardment simulation. The mass loss for each system, is normalised at each time step concerning their respective total mass of the initial structures. At the end of the bombardment simulation, the ABPBI/MWCNT15 system shows the most significant mass deficit due to the AO impact with a total mass loss of 0.0973% at $t_s = 40\,\text{ps}$. All four systems show the same linear downward trend, resulting

System	AO added	AO reacted	H_2O	OH	H
Pristine ABPBI	66	42		7	8
ABPBI/MWCNT10	66	47	3	8	2
ABPBI/MWCNT15	66	56	1	13	2
ABPBI/MWCNT20	66	49	1	11	2
Pristine ABPBI	132	78	1	12	6
ABPBI/MWCNT10	132	77		14	7
ABPBI/MWCNT15	132	95	2	8	3
ABPBI/MWCNT20	132	84*	1	9	3
Pristine ABPBI	200	107	1	5	3
ABPBI/MWCNT10	200	102*		7	4
ABPBI/MWCNT15	200	132	1	10	9
ABPBI/MWCNT20	200	117	2	6	2

*: a single O_2 was produced by the system.

Table 3.4: The removed fragments of all four systems at three different time steps in the bombardment simulation. H_2O is the largest molecule produced in all four systems.

in a final mass loss below 0.1%. Figure 3.8 demonstrates that the ABPBI/MWCNT20 system has the best resistance against the AO bombardment of the four systems tested.

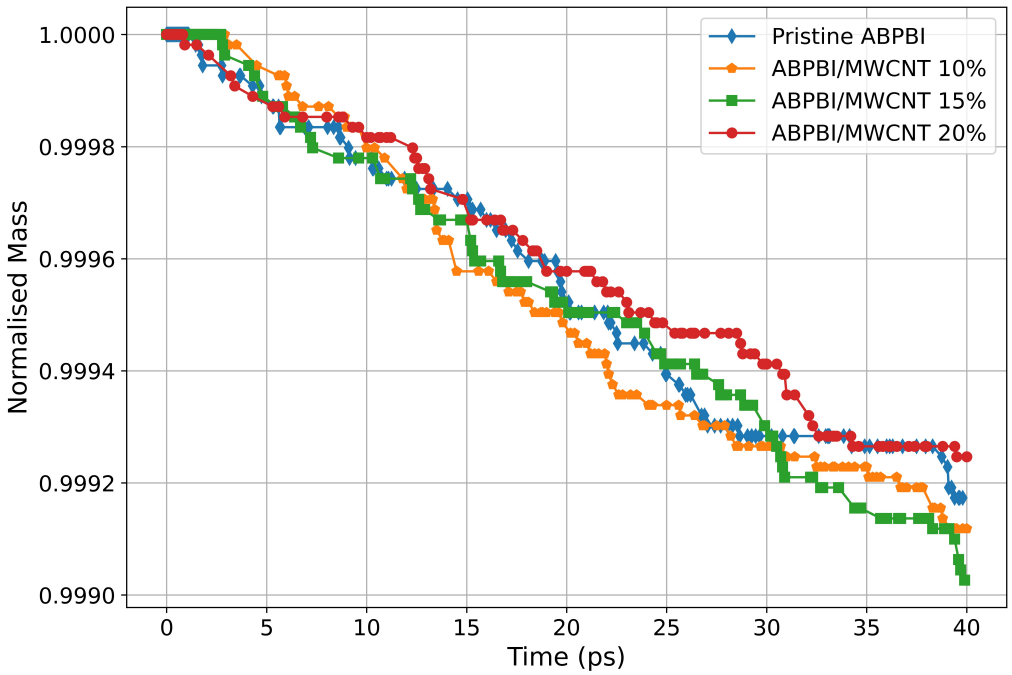


Figure 3.8: Normalised mass of the four systems as a function of simulation time.

From the mass loss results, the erosion yield can be calculated for the systems at different times through the bombardment simulation. Table 3.5 depicts the erosion yields after 80, 120, 160, and 200 AOs are added to the simulation. Again, the ABPBI/MWCNT20 system shows the lowest erosion yield at the end of the simulation, $E_y = 3.43 \times 10^{-25}$ g/AO.

	Erosion Yield ($\times 10^{-25}$ g/AO)				
n_{AO}[a]	40	80	120	160	200
System					
pristine ABPBI	3.77	3.77	4.18	4.08	3.78
ABPBI/MWCNT10	2.93	4.81	5.02	4.24	4.02
ABPBI/MWCNT15	4.67	4.66	4.04	4.47	4.44
ABPBI/MWCNT20	3.35	3.77	3.77	3.87	3.43

[a] The number of AO added to the bombardment simulation.

Table 3.5: The erosion yield for all four systems after a specified number of atomic oxygen (AO).

3.5 Thermal Cycling

An additional simulation was conducted to test the effects of thermal cycling on the AO bombardment. The setup for the simulation is identical to the arrangement explained above (see 3.1.2) except for the temperature of the simulation. The thermal cycle extreme minimum (103 K) was the initial temperature for the simulation, and similar to the above, a 10 ps NVT calculation was performed before the bombardment simulation. Figure 3.9 indicates the system temperature for pristine ABPBI at two initial temperatures: 103 K and 298 K. The two simulations show a similar trend, where a minor increase in temperature occurs due to the energy added to the system by AO, followed by a decrease in temperature, and both systems reflect a final temperature higher than the initial temperature. Table 3.6 indicates the average and final temperatures of the two simulations.

The results of an analysis conducted on the AO distribution within the system after a bombardment simulation are shown in Figure 3.10 below.

Simulation	Average Temperature (K)	Final Temperature (K)
$T = 103$K	138.96 ± 15.71	125.28
$T = 298$K	331.38 ± 11.18	328.12

Table 3.6: System temperature averages and final for two simulations at different initial temperatures.

In Figure 3.10, the 103 K simulation shows a narrow AO distribution with a sharp maximum peak found at $z = 3$ Å, and a lower overall AO penetration depth, with a maximum of $z = 11$ Å. At low temperatures, the kinetic energy of the particles is low, causing the structure to be more rigid. Here a combination of effects occurs where some of the AO is scattered elastically from the surface, and some of the AO transfers energy to the material, thus getting trapped at the surface with no energy left. The 298 K simulation shows a broader distribution of AO combined with a higher AO penetration depth. The reasoning is the same as for the 103 K simulation, where higher temperatures equate to higher particle kinetic energy leading to a more relaxed and spread-out structure that allows for higher AO penetration depths. The already available kinetic energy combined with the added energy from impacting AO equates to easier bond-breaking and, therefore, higher

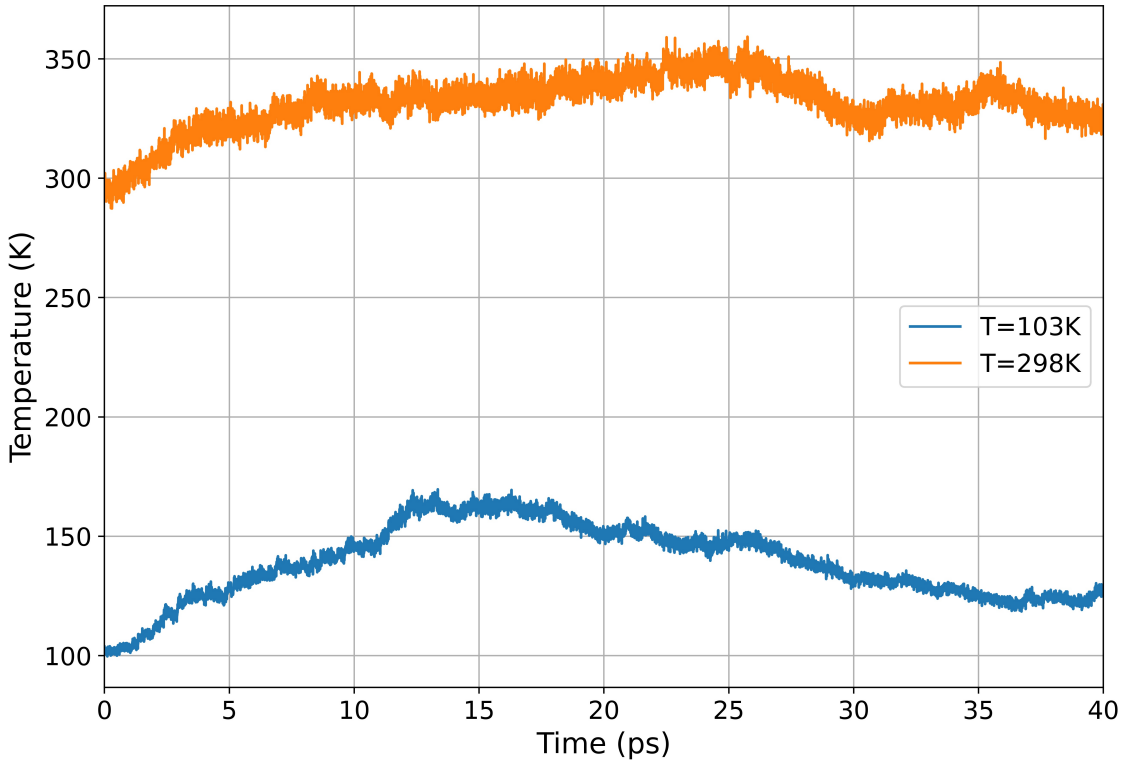


Figure 3.9: The system temperature for two simulations as a function of simulation time.

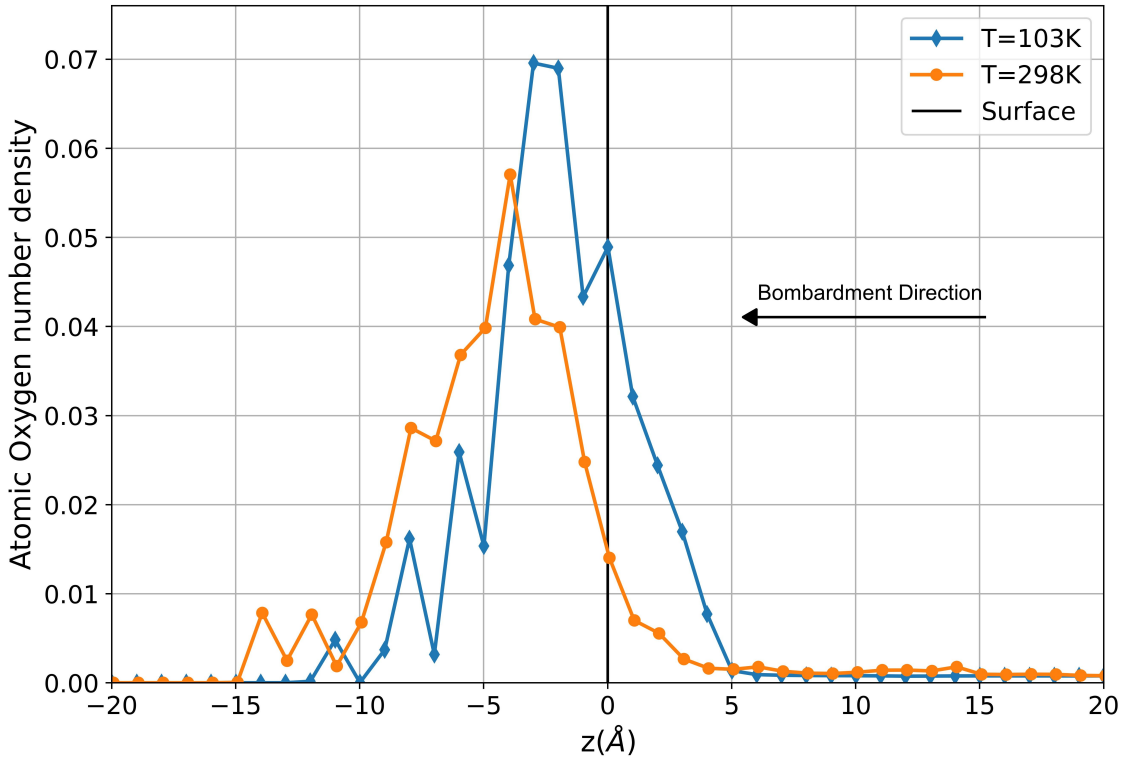


Figure 3.10: The number density of AO remaining within the two simulations at the end of the AO bombardment simulation (t=40ps). The surfaces of two ABPBI systems are at $z = 0$Å.

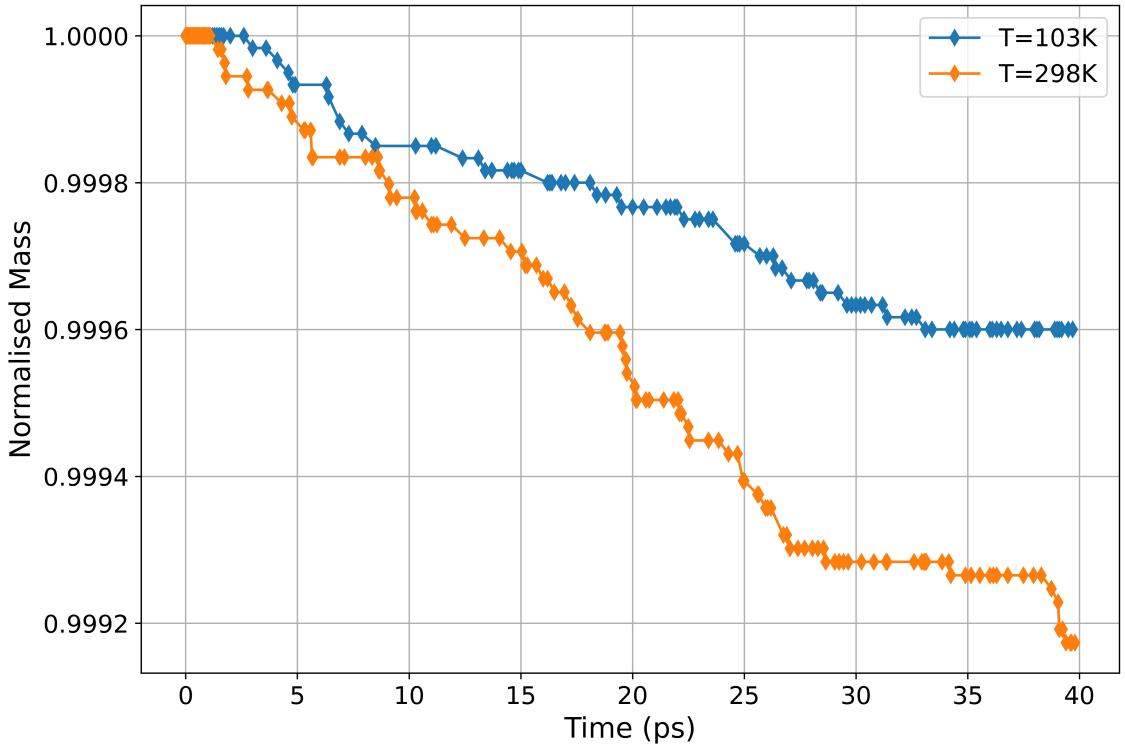


Figure 3.11: Normalised mass of the two systems as a function of simulation time.

degradation rates.

As a final analysis of the effects of temperature on the ABPBI system, the results of the normalised mass as a function of simulation time appear in Figure 3.11. The mass, therefore is normalised at each time step by dividing with the mass of the initial ABPBI structure. Figure 3.11 shows that the normalised mass loss for the 103K simulation is the lowest of the two simulations, with a total mass loss at the end of the bombardment simulation of 0.04%. The 298 K simulation shows a higher mass loss rate with a final mass loss of 0.08%. Thus, the 298 K simulation shows double the total mass loss of the 103 K simulation. The erosion yield for the 103 K simulation is $E_y = 2.01 \times 10^{-25}$ g/AO.

Figure 3.11 shows the 103 K and the 298 K simulations have a linear downward trend. The small molecule production of the 103 K simulation consists of 17 OH and 2 H_2O molecules with a couple of loose H atoms removed from the system; therefore, the ratio of volatile species produced is similar to those of the 298K simulation.

Chapter 4

Discussion and Conclusion

In this study poly(2,5)-benzimidazole (ABPBI) was tested as radiation-shielding against atomic oxygen (AO) by use of molecular dynamics (MD) simulations. The creation of the MD simulations was to accurately simulate the impact conditions of atomic oxygen (AO) that a material would experience when orbiting in low Earth orbit (LEO). Reactive MD simulation was employed to simulate the chemical kinetics that a material would experience when interacting with AO and achieved by implementing a reactive empirical-based force field, namely ReaxFF. The force field was adapted to investigate the impact of AO on materials [31]. Further investigations of the AO bombardment on ABPBI included testing the effects of a thermostat on a bombardment simulation, the penetration depth of AO in ABPBI, the small molecule production due to AO bombardment, and the overall mass loss of a system due to AO bombardment. All these investigations were applied to four variations of the ABPBI system, three of which were loaded with specific weight percentages (% wt) of multi-walled carbon nanotubes (MWCNT). The reason for the creation of the four systems was to test the efficacy of % wt of MWCNT against AO bombardment. The bombardment simulation was conducted on all four optimised and energy-minimised systems while retaining a semi-flat surface.

In testing the effects of a thermostat on the temperature of the pristine ABPBI system, it is clear that the thermostat is crucial in maintaining the temperature at some specified constant value. In Figure 3.4, the ABPBI system without the thermostat succumbs to non-physical disintegration, showing the importance of the material temperature when considering the stability of the material. Rahnamoun and Van Duin [64] conducted a similar study in which four simulations of each material, Kapton, Kapton-POSS, Silica, and Teflon, were completed, where one of the simulations had no thermostat and three had thermostats with varying damping coefficients. The results revealed that the effect of a thermostat that keeps a constant temperature, regardless of the damping coefficient, is crucial in AO bombardment simulations.

AO has a low-impact energy dependent on a spacecraft ramming into the AO while orbiting the Earth. AO is also a large ion in comparison to some of the other single-atom species

found in the LEO environment, including but not limited to protons, electrons, and alpha particles. The penetration depth relies on these factors as well as the density of the material under investigation. Vacuum spaces exist within the polymer structure because of the amorphous nature of the ABPBI polymer. The penetration depths of AO, concerning the ABPBI systems, are essential to ensure that most of the erosion caused by AO is limited to the surface of the material. The maximum penetration depth for the ABPBI systems is $z = 18\,\text{Å}$ for the ABPBI/MWCNT10 system. Similar studies by Ashraf et al. [31] and Rahmani et al. [63] discuss the AO penetration depth of various other polymers bombarded with AO to a fluence of $\sim 10^{15}\,\text{AO/cm}^2$. The study by Ashraf et al. [31] reflects a maximum AO penetration depth of $z \approx 35\,\text{Å}$ for the four systems, while Rahmani et al. [63] has a maximum at an approximate depth of $z \approx 40\,\text{Å}$. The contributing factors for this difference can be varied, such as system size, simulated densities, and added nanoparticles, for example, MWCNT. After similar AO bombardment simulations, all four ABPBI systems showed a lower AO penetration depth.

The small molecule production due to AO bombardment is a decisive study point of the simulation. The production of volatile species such as H_2O, CO, and OH can react with the polymer surface and cause further degradation [27]. Another reason for the importance of small molecule production is to give an insight that is hard to obtain by experimental procedures. The production rate of these small molecules is directly related to the rate at which the polymer composites would deteriorate. Some CO molecules were still created during the bombardment simulation for all four systems even though there was no CO extrated during the bombardment simulation. Whether due to lack of energy after bonding or bond breaking, all the CO created for the four systems remained within the system. As per Table 3.4, OH is the most produced molecule for all four systems, with a few loose H atoms were produced by the bombardment simulation. The four systems of Ashraf et al. [31] show that the most produced molecule is H_2O, while the simulations in this study show that some of the molecules produced are H_2O but much fewer than that of Ashraf et al. [31]. As previously noted, H_2O is considered a volatile species and, thus, producing less of it bodes well. Molecule production relies heavily on the composition of the polymers and composites under investigation. The abundance of hydrogen groups lessens the degradation due to AO bonding with hydrogen rather than degrading the main structure of the ABPBI polymer.

The erosion yield associated with AO bombardment at a fluence of $\sim 10^{15}\,\text{AO/cm}^2$ is of the order $\sim 10^{-25}\,\text{g/AO}$. Rahmani et al. [63] produced erosion yields for various systems after AO bombardment with a fluence of $\sim 10^{14}\,\text{AO/cm}^2$. The highest erosion yield recorded by Rahmani et al. [63] is that of the pristine polyimide (PI) at $E_y = 8.99 \times 10^{-23}\,\text{g/AO}$, while the lowest is $E_y = 5.6 \times 10^{-25}\,\text{g/AO}$ for a PI system enhanced with 15 wt % randomly aligned CNTs. The simulated erosion yields for Kapton and Teflon after AO bombardment, as studied by Rahnamoun and Van Duin [64], are $E_y = 2.00 \times 10^{-23}\,\text{g/AO}$ and $E_y = 5.86 \times 10^{-25}\,\text{g/AO}$, respectively. The four ABPBI systems show a slightly lower erosion yield than that of Rahmani et al. [63] and Rahnamoun and Van Duin [64].

Comparatively, all four indicate a low erosion yield. The ABPBI systems show promise with regards to degradation caused by AO. The effects of CNTs on a system such as ABPBI/MWCNT20 do seem to show some improvement against degradation and warrants further investigation into the impact of CNT on degradation caused by AO.

The effects of thermal cycling on a material, such as ABPBI, combined with the influence of AO, were tested with an additional AO bombardment simulation conducted at the thermal cycling minimum temperature. In no way does this simulate the actual effects of thermal cycling, but it does give insight on the effects of temperature concerning AO bombardment. Higher system temperatures cause deeper AO penetration into the material, leading to higher levels of degradation, as can be concluded from Figure 3.10 and Figure 3.11. A similar simulation was conducted by Ashraf et al. [31] regarding the minimum and maximum temperature of thermal cycling in LEO. Their bisphenol A epoxide cured with diethyltoluenediamine (bisA/DETDA) system revealed that at low temperatures, less degradation occurs. They also stated that a considerable amount of energy is required from the impacting source, AO, to start the material degradation. Ashraf et al. [31] showed that two simulations at high temperatures (298 K and 396 K) yield similar results due to both high temperatures being well below the glass transition temperature of the polymer. Similar results would thus most likely occur for the ABPBI polymer with a glass transition temperature of $T_g = 758\,\mathrm{K}$.

Given the results discussed above; low AO penetration depth, minimal small molecule production, and low erosion yield, the ABPBI systems show promise when considering radiation shielding against AO.

Chapter 5

Future Works

The construction of the AO bombardment simulation was the first step in a series to properly simulate the LEO environment and test materials for applications therein. For this investigation a total of ~ 26 seconds of real world fluence was simulated. A lengthier simulation with a higher fluence is required to get more detailed information about the effects of AO on materials. Currently, the computational resources at our disposal are limited, leaving room for improvement; this includes optimising the programme for multi-threaded processing. During the investigation, multiple errors regarding parallel processing occurred due to the specific simulation cell configuration. Improving the code to account for this configuration would allow parallel processing, reducing the total runtime of a bombardment simulation. Another area for future work would be including another species found in LEO. The combination of species would have a different effect compared to isolating each species. Although, considering the other species found in LEO in isolation would be a step in the right direction as well. The physical interactions caused by AO are well-known, but a detailed investigation is needed to research the exact chemical mechanisms that occur. These mechanisms can give insight into the weak points of the material. Lastly, a comparison between the simulation and experimental data to validate and improve the simulation configuration is essential.

Bibliography

[1] Paulo Gordo, Tiago Frederico, Rui Melício, Sophie Duzellier, and Antonio Amorim. System for space materials evaluation in leo environment. *Advances in Space Research*, 66(2):307–320, 2020.

[2] Nicholas L Johnson. Medium earth orbits: is there a need for a third protected region? In *61st International Astronautical Congress*, number JSC-CN-21489, 2010.

[3] Krishna Kandarpa, Victor Schneider, and Krishnan Ganapathy. Human health during space travel: An overview. *Neurology India*, 67(8):176, 2019.

[4] Guy Fogleman, Lauren Leveton, and John Charles. The bioastronautics roadmap: A risk reduction strategy for human exploration. In *1st Space Exploration Conference: Continuing the Voyage of Discovery*, page 2526, 2005.

[5] Joyce Dever, Bruce Banks, Kim de Groh, and Sharon Miller. Degradation of spacecraft materials. In *Handbook of environmental degradation of materials*, pages 465–501. Elsevier, 2005.

[6] JW Wilson, J Miller, A Konradi, and FA Cucinotta. Shielding strategies for human space exploration. Technical report, 1997.

[7] Kim K de Groh, Bruce A Banks, Anne M Hammerstrom, Erica E Youngstrom, Carolyn Kaminski, Laura M Marx, Elizabeth S Fine, Jonathan D Gummow, and Douglas Wright. Misse peace polymers: an international space station environmental exposure experiment. In *Proceedings of the AIAA Conference on International Space Station Utilization*, 2001.

[8] Kim K deGroh, Joyce A Dever, Donald A Jaworske, Sharon K Miller, Edward A Sechkar, Scott R Panko, et al. Nasa glenn research center's materials international space station experiments (misse 1-7). In *International Symposium on SM/MPAC and SEED Experiments*, number E-16690, 2008.

[9] Juan Antonio Asensio, Salvador Borrós, and Pedro Gómez-Romero. Proton-conducting membranes based on poly (2, 5-benzimidazole)(abpbi) and phosphoric acid prepared by direct acid casting. *Journal of Membrane Science*, 241(1):89–93, 2004.

[10] Keki Hormusji Gharda, Prakash D Trivedi, Tushar R Parida, and Amitkumar Biradar. Method for processing a high temperature resistant thermosetting material, March 8 2018. US Patent App. 15/676,187.

[11] Liliana A Diaz, Graciela C Abuin, and Horacio R Corti. Methanol sorption and permeability in nafion and acid-doped pbi and abpbi membranes. *Journal of membrane science*, 411:35–44, 2012.

[12] Liliana A Diaz, Graciela C Abuin, and Horacio R Corti. Water and phosphoric acid uptake of poly [2, 5-benzimidazole](abpbi) membranes prepared by low and high temperature casting. *Journal of Power Sources*, 188(1):45–50, 2009.

[13] Juan Antonio Asensio, Salvador Borrós, and Pedro Gómez-Romero. Proton-conducting membranes based on poly(2,5-benzimidazole) (abpbi) and phosphoric acid prepared by direct acid casting. *Journal of Membrane Science*, 241(1):89–93, 2004. ISSN 0376-7388. doi: https://doi.org/10.1016/j.memsci.2004.03.044. URL https://www.sciencedirect.com/science/article/pii/S0376738804003084.

[14] Juan Antonio Asensio, Salvador Borrós, and Pedro Gómez-Romero. Sulfonated poly(2,5-benzimidazole) (sabpbi) impregnated with phosphoric acid as proton conducting membranes for polymer electrolyte fuel cells. *Electrochimica Acta*, 49(25):4461–4466, 2004. ISSN 0013-4686. doi: https://doi.org/10.1016/j.electacta.2004.05.002. URL https://www.sciencedirect.com/science/article/pii/S0013468604004748.

[15] Shantanu Bhowmik and R Benedictus. Performance of space durable polymeric nano composite under electromagnetic radiation at low earth orbit. In *2007 IEEE Applied Electromagnetics Conference (AEMC)*, pages 1–4. IEEE, 2007.

[16] JH Adams Jr, DH Hathaway, RN Grugel, JW Watts, TA Parnell, JC Gregory, and RM Winglee. Revolutionary concepts of radiation shielding for human exploration of space. Technical report, 2005.

[17] Sumio Iijima. Helical microtubules of graphitic carbon. *Nature*, 354(6348):56–58, 1991.

[18] Siyuan Chen, Shruti Nambiar, Zhenhao Li, Yonghai Sun, Shen Gong, George ZH Zhu, and John TW Yeow. Polymer nanocomposite for space applications. In *14th IEEE International Conference on Nanotechnology*, pages 685–688. IEEE, 2014.

[19] Enkeleda Dervishi, Zhongrui Li, Yang Xu, Viney Saini, Alexandru R Biris, Dan Lupu, and Alexandru S Biris. Carbon nanotubes: synthesis, properties, and applications. *Particulate Science and Technology*, 27(2):107–125, 2009.

[20] Mauricio Terrones. Science and technology of the twenty-first century: synthesis, properties, and applications of carbon nanotubes. *Annual review of materials research*, 33(1):419–501, 2003.

[21] S Bellucci, C Balasubramanian, F Micciulla, and G Rinaldi. Cnt composites for aerospace applications. *Journal of Experimental Nanoscience*, 2(3):193–206, 2007.

[22] Kim K De Groh, Bruce A Banks, Catherine E Mccarthy, Rochelle N Rucker, Lily M Roberts, and Lauren A Berger. Misse 2 peace polymers atomic oxygen erosion experiment on the international space station. *High Performance Polymers*, 20(4-5): 388–409, 2008.

[23] Bruce Banks, Sharon Miller, and Kim de Groh. Low earth orbital atomic oxygen interactions with materials. In *2nd International Energy Conversion Engineering Conference*, page 5638, 2004.

[24] Kirn K de Groh and Bruce A Banks. Atomic-oxygen undercutting of long duration exposure facility atomized-kapton multilayer insulation. *Journal of Spacecraft and Rockets*, 31(4):656–664, 1994.

[25] Bruce Banks, Sharon Miller, B.M. Auer, and F. Difilippo. Materials degradation in low earth orbit (leo). *TMS Society*, pages 15–33, 01 1990.

[26] Vincent L Pisacane. *Fundamentals of space systems*. Johns Hopkins University Appli, 2005.

[27] Kwang-Bok Shin, Chun-Gon Kim, Chang-Sun Hong, and Ho-Hyung Lee. Prediction of failure thermal cycles in graphite/epoxy composite materials under simulated low earth orbit environments. *Composites Part B: Engineering*, 31(3):223–235, 2000.

[28] PE George and HW Dursch. Low earth orbit effects on organic composites flown on the long duration exposure facility. *Journal of advanced materials*, 25(3):10–19, 1994.

[29] Kim K deGroh, Donald A Jaworske, Gary Pippin, Philip P Jenkins, Robert J Walters, Sheila A Thibeault, Iwona Palusinski, and Justin R Lorentzen. Materials international space station experiment (misse): overview, accomplishments and future needs. In *Annual International Space Station Research and Development Conference*, number GRC-E-DAA-TN15048, 2014.

[30] Kim K de Groh, Bruce A Banks, and Loredana Santo. Materials international space station experiment-9 (misse-9) polymers and composites experiment. In *COSPAR Scientific Assembly*, number GRC-E-DAA-TN57206, 2018.

[31] Chowdhury Ashraf, Aniruddh Vashisth, Charles E Bakis, and Adri CT Van Duin. Reactive molecular dynamics simulations of the atomic oxygen impact on epoxies with different chemistries. *The Journal of Physical Chemistry C*, 123(24):15145–15156, 2019.

[32] Adri CT Van Duin, Siddharth Dasgupta, Francois Lorant, and William A Goddard. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A*, 105(41):9396–9409, 2001.

[33] Kimberly Chenoweth, Adri CT Van Duin, and William A Goddard. Reaxff reactive force field for molecular dynamics simulations of hydrocarbon oxidation. *The Journal of Physical Chemistry A*, 112(5):1040–1053, 2008.

[34] Dimitrios Vlachakis, Elena Bencurova, Nikitas Papangelopoulos, and Sophia Kossida. Current state-of-the-art molecular dynamics methods and applications. *Advances in protein chemistry and structural biology*, 94:269–313, 2014.

[35] Mark Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2010.

[36] Mark E Tuckerman and Glenn J Martyna. Understanding modern molecular dynamics: Techniques and applications. *The Journal of Physical Chemistry B*, 104(2): 159–178, 2000.

[37] Harvey Gould and Jan Tobochnik. *Statistical and thermal physics: with computer applications*. Princeton University Press, 2021.

[38] Robert Floyd Sekerka. *Thermal physics: thermodynamics and statistical mechanics for scientists and engineers*. Elsevier, 2015.

[39] Philippe H Hünenberger. Thermostat algorithms for molecular dynamics simulations. *Advanced computer simulation: Approaches for soft matter sciences I*, pages 105–149, 2005.

[40] Shūichi Nosé. A molecular dynamics method for simulations in the canonical ensemble. *Molecular physics*, 52(2):255–268, 1984.

[41] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

[42] William G Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Physical review A*, 31(3):1695, 1985.

[43] Glenn J Martyna, Michael L Klein, and Mark Tuckerman. Nosé–hoover chains: The canonical ensemble via continuous dynamics. *The Journal of chemical physics*, 97(4): 2635–2643, 1992.

[44] Crispin Gardiner and Peter Zoller. *Quantum noise: a handbook of Markovian and non-Markovian quantum stochastic methods with applications to quantum optics*. Springer Science & Business Media, 2004.

[45] P Ullersma. An exactly solvable model for brownian motion: I. derivation of the langevin equation. *Physica*, 32(1):27–55, 1966.

[46] Alex A Samoletov, Carl P Dettmann, and Mark AJ Chaplain. Thermostats for "slow" configurational modes. *Journal of Statistical Physics*, 128:1321–1336, 2007.

[47] Herman JC Berendsen, JPM van Postma, Wilfred F Van Gunsteren, ARHJ DiNola, and Jan R Haak. Molecular dynamics with coupling to an external bath. *The Journal of chemical physics*, 81(8):3684–3690, 1984.

[48] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[49] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[50] Brian D Ripley. *Stochastic simulation*. John Wiley & Sons, 2009.

[51] Biovia, Dassault Systèmes, Materials Studio, Release 2019. San Diego: Dassault Systèmes. https://www.3ds.com/products-services/biovia/products/molecular-modeling-simulation/biovia-materials-studio/. Accessed: 2022-10-23.

[52] Huai Sun. Compass: an ab initio force-field optimized for condensed-phase applications overview with details on alkane and benzene compounds. *The Journal of Physical Chemistry B*, 102(38):7338–7364, 1998.

[53] Reinier LC Akkermans, Neil A Spenley, and Struan H Robertson. Compass iii: Automated fitting workflows and extension to ionic liquids. *Molecular Simulation*, 47 (7):540–551, 2021.

[54] Nancy Iwamoto, Matthew MF Yuen, Haibo Fan, and Artur Wymysłowski. *Molecular modeling and multiscaling issues for electronic material applications*, volume 57. Springer, 2012.

[55] MedeA. Medea version 3.4. medea is a registered trademark of materials design, inc., san diego, usa. https://www.3ds.com/products-services/biovia/products/molecular-modeling-simulation/biovia-materials-studio/. Accessed: 2022-11-01.

[56] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.

[57] Huai Sun. Force field for computation of conformational energies, structures, and vibrational frequencies of aromatic polyesters. *Journal of Computational Chemistry*, 15(7):752–768, 1994.

[58] Thomas P Senftle, Sungwook Hong, Md Mahbubul Islam, Sudhir B Kylasa, Yuanxia Zheng, Yun Kyung Shin, Chad Junkermeier, Roman Engel-Herbert, Michael J Janik, Hasan Metin Aktulga, et al. The reaxff reactive force-field: development, applications and future directions. *npj Computational Materials*, 2(1):1–14, 2016.

[59] Aniruddh Vashisth, Chowdhury Ashraf, Weiwei Zhang, Charles E Bakis, and Adri CT Van Duin. Accelerated reaxff simulations for describing the reactive cross-linking of polymers. *The Journal of Physical Chemistry A*, 122(32):6633–6642, 2018.

[60] Aniruddh Vashisth, Chowdhury Ashraf, Charles E Bakis, and Adri CT van Duin. Effect of chemical structure on thermo-mechanical properties of epoxy polymers: Comparison of accelerated reaxff simulations and experiments. *Polymer*, 158:354–363, 2018.

[61] Aidan P Thompson, H Metin Aktulga, Richard Berger, Dan S Bolintineanu, W Michael Brown, Paul S Crozier, Pieter J in't Veld, Axel Kohlmeyer, Stan G Moore, Trung Dac Nguyen, et al. Lammps-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171, 2022.

[62] T Schneider and E Stoll. Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions. *Physical Review B*, 17(3):1302, 1978.

[63] Farzin Rahmani, Sasan Nouranian, Xiaobing Li, and Ahmed Al-Ostaz. Reactive molecular simulation of the damage mitigation efficacy of poss-, graphene-, and carbon nanotube-loaded polyimide coatings exposed to atomic oxygen bombardment. *ACS Applied Materials & Interfaces*, 9(14):12802–12811, 2017.

[64] A Rahnamoun and ACT Van Duin. Reactive molecular dynamics simulation on the disintegration of kapton, poss polyimide, amorphous silica, and teflon during atomic oxygen impact using the reaxff reactive force-field method. *The Journal of Physical Chemistry A*, 118(15):2780–2787, 2014.

[65] Benjamin D Jensen, Ananyo Bandyopadhyay, Kristopher E Wise, and Gregory M Odegard. Parametric study of reaxff simulation parameters for molecular dynamics modeling of reactive carbon gases. *Journal of chemical theory and computation*, 8(9): 3003–3008, 2012.

[66] James F Ziegler and Jochen P Biersack. The stopping and range of ions in matter. In *Treatise on heavy-ion science*, pages 93–129. Springer, 1985.

[67] James F Ziegler, Matthias D Ziegler, and Jochen P Biersack. Srim–the stopping and range of ions in matter (2010). *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 268(11-12):1818–1823, 2010.

Appendix A

Simulation Code

A.1 *NVT* Code

The code was created using the MedeA graphical workflow editor. The code performs the final $t = 10$ps NVT calculation.

```
#  This is the control script for LAMMPS.

echo                    both
log                     2.1_Initialize.out

#-----------------------------------------------------------
# Stage 2.1: Initialize LAMMPS run for 3-d periodic
#-----------------------------------------------------------

units                   real
boundary                p p p
atom_style              full


box                     tilt large
read_data               structure.dat

pair_style              reax/c NULL checkqeq no safezone 10.0
   mincap 1000
pair_coeff              * * parameters.dat C H N O

variable                R               equal 0.00198722
variable                sysvol          equal vol
```

```
variable        sysmass         equal mass(all)
  /6.0221367e+023
variable        sysdensity      equal v_sysmass/v_sysvol
  /1.0e-24
variable        coulomb         equal ecoul+elong
variable        etotal          equal etotal
variable        pe              equal pe
variable        ke              equal ke
variable        evdwl           equal evdwl
variable        epair           equal epair
variable        ebond           equal ebond
variable        eangle          equal eangle
variable        edihed          equal edihed
variable        eimp            equal eimp
variable        lx              equal lx
variable        ly              equal ly
variable        lz              equal lz
variable        Nthermo         equal 0
variable        cella           equal lx
variable        cellb           equal sqrt(ly*ly+xy*xy)
variable        cellc           equal sqrt(lz*lz+xz*xz+
  yz*yz)
variable        cellalpha       equal acos((xy*xz+ly*yz)
  /(v_cellb*v_cellc))
variable        cellbeta        equal acos(xz/v_cellc)
variable        cellgamma       equal acos(xy/v_cellb)
variable        p               equal press
variable        pxx             equal pxx
variable        pyy             equal pyy
variable        pzz             equal pzz
variable        pyz             equal pyz
variable        pxz             equal pxz
variable        pxy             equal pxy
variable        sxx             equal -pxx
variable        syy             equal -pyy
variable        szz             equal -pzz
variable        syz             equal -pyz
variable        sxz             equal -pxz
variable        sxy             equal -pxy
variable        fmax            equal fmax
variable        fnorm           equal fnorm
variable        time equal step*dt+0.000001
```

```
variable                surfacetension equal 0.5*v_lz*(0.5*(
   v_sxx+v_syy)-v_szz)

thermo_style            custom step v_time press vol
   v_sysdensity temp ebond eangle edihed eimp evdwl ecoul etail
   elong pe ke
thermo_modify           flush yes


#
# Set up the fixed and movable groups
#

group           fixed id 133 134 135 136 137 138 139 140 141 142
group           fixed id 143 144 145 146 147 148 149 150 151 152
group           fixed id 153 154 155 156 157 158 159 160 161 162
                                 ⋮
group           fixed id 6828 6829 6830 6831 6832 6833 6834 6835
    6836 6837
group           fixed id 6838 6839 6840 6841 6842 6843 6844 6845
    6846 6847
group           fixed id 6848 6854

group                   movable subtract all fixed

compute                 sum_f1 movable reduce sum fx fy fz
variable                sum_fx equal c_sum_f1[1]
variable                sum_fy equal c_sum_f1[2]
variable                sum_fz equal c_sum_f1[3]


uncompute           thermo_temp
compute             thermo_temp movable temp

#
# Subsets
#

group                   subset_Thermostat id 1 2 3 4 5 6 7 8 9
   10
group                   subset_Thermostat id 11 12 13 14 15 16
   17 18 19 20
```

```
group                   subset_Thermostat id 21 22 23 24 25 26
   27 28 29 30
                                          ⋮
group                   subset_Thermostat id 6775 6776 6777 6778
    6779 6780 6781 6849 6850 6851
group                   subset_Thermostat id 6852 6853 6855 6856
    6857 6858 6859 6860 6861 6862
group                   subset_Thermostat id 6863 6864

group                   subset_AO id 6865

log             2.2_Custom.out
#----------------------------------------------------------
# Stage 2.2: ReaxFF Custom Parameters
#----------------------------------------------------------

pair_style              reax/c NULL
pair_coeff              * * C:/MD/data/Forcefields/custom/ffield
   .reax.AO C H N O
fix             qeq all qeq/reax 1 0.0 10.0 1e-6 reax/c


log                     2.3_Velocities.out
#----------------------------------------------------------
# Stage 2.3: Set the initial velocities for $T
#----------------------------------------------------------

velocity                all create 298 1 dist gaussian mom yes
   rot no

velocity            fixed set 0.0 0.0 0.0

log                     2.4_NVT.out
#-----------------------------------------------------------------
# Stage 2.4: NVT integration for 10 ps with a timestep of 0.1 fs
#             Temperature 298 K
#-----------------------------------------------------------------

reset_timestep          0
thermo_style            custom step v_time press vol
   v_sysdensity temp ebond eangle edihed eimp evdwl ecoul etail
   elong pe ke
```

```
thermo                  ${Nthermo}
fix                     1 movable nvt temp 298 298 100 drag 1.0
fix                     2 movable ave/time 1 10000 10000 v_time
   c_thermo_temp c_thermo_press v_sysvol v_sysdensity v_etotal
   v_pe v_ke v_evdwl v_coulomb v_sxx v_syy v_szz v_syz v_sxz
   v_sxy file 2.4_averages.txt
fix                     3 movable ave/time 10  1     10 v_time
   c_thermo_temp c_thermo_press v_sysvol v_sysdensity v_etotal
   v_pe v_ke v_evdwl v_coulomb v_sxx v_syy v_szz v_syz v_sxz
   v_sxy file 2.4_instantaneous.txt
fix                     4 fixed setforce 0.0 0.0 0.0
fix                     5 all qeq/reax 1 0.0 10.0 1.0e-6 reax/c
restart                 100000 2.4.restart
dump                    sci all custom 10000 2.4.xyz id mol type
    q xs ys zs

timestep                0.1
run                     100000
undump                  sci

restart                 0
dump                    sci all custom 100000 2.4.xyz id mol
   type q xs ys zs
run                     0
undump                  sci

unfix                   1
unfix                   2
unfix                   3
unfix                   4
unfix                   5
```

A.2 Bombardment Simulation Code

The code was created using the MedeA graphical workflow editor. The code includes four parts: Initialisation, implementation of custom ReaxFF force field and parameters, initialisation of velocities, and AO bombardment.

```
#  This is the control script for LAMMPS.

echo                    both
log                     3.1_Initialize.out
```

```
#----------------------------------------------------------
# Stage 3.1: Initialize LAMMPS run for 3-d periodic
#----------------------------------------------------------

units                   real
boundary                p p p
atom_style              full


box                     tilt large
read_data               structure.dat

pair_style              reax/c NULL checkqeq no safezone 10.0
   mincap 1000
pair_coeff              * * parameters.dat C H N O

variable                R                       equal 0.00198722
variable                sysvol          equal vol
variable                sysmass         equal mass(all)
   /6.0221367e+023
variable                sysdensity      equal v_sysmass/v_sysvol
   /1.0e-24
variable                coulomb         equal ecoul+elong
variable                etotal          equal etotal
variable                pe                      equal pe
variable                ke                      equal ke
variable                evdwl           equal evdwl
variable                epair           equal epair
variable                ebond           equal ebond
variable                eangle          equal eangle
variable                edihed          equal edihed
variable                eimp            equal eimp
variable                lx                      equal lx
variable                ly                      equal ly
variable                lz                      equal lz
variable                Nthermo         equal 0
variable                cella           equal lx
variable                cellb           equal sqrt(ly*ly+xy*xy)
variable                cellc           equal sqrt(lz*lz+xz*xz+
   yz*yz)
variable                cellalpha       equal acos((xy*xz+ly*yz)
   /(v_cellb*v_cellc))
```

```
variable                cellbeta        equal acos(xz/v_cellc)
variable                cellgamma       equal acos(xy/v_cellb)
variable                p                       equal press
variable                pxx                     equal pxx
variable                pyy                     equal pyy
variable                pzz                     equal pzz
variable                pyz                     equal pyz
variable                pxz                     equal pxz
variable                pxy                     equal pxy
variable                sxx                     equal -pxx
variable                syy                     equal -pyy
variable                szz                     equal -pzz
variable                syz                     equal -pyz
variable                sxz                     equal -pxz
variable                sxy                     equal -pxy
variable                fmax            equal fmax
variable            fnorm               equal fnorm
variable                time equal step*dt+0.000001
variable                surfacetension equal 0.5*v_lz*(0.5*(
   v_sxx+v_syy)-v_szz)

thermo_style            custom step v_time press vol
   v_sysdensity temp ebond eangle edihed eimp evdwl ecoul etail
   elong pe ke
thermo_modify           flush yes


#
# Set up the fixed and movable groups
#

group           fixed id 133 134 135 136 137 138 139 140 141 142
group           fixed id 143 144 145 146 147 148 149 150 151 152
group           fixed id 153 154 155 156 157 158 159 160 161 162
                                ⋮
group           fixed id 6828 6829 6830 6831 6832 6833 6834 6835
    6836 6837
group           fixed id 6838 6839 6840 6841 6842 6843 6844 6845
    6846 6847
group           fixed id 6848 6854

group                   movable subtract all fixed
```

```
compute                 sum_f1 movable reduce sum fx fy fz
variable                sum_fx equal c_sum_f1[1]
variable                sum_fy equal c_sum_f1[2]
variable                sum_fz equal c_sum_f1[3]


uncompute           thermo_temp
compute             thermo_temp movable temp

#
# Subsets
#

group                   subset_Thermostat id 1 2 3 4 5 6 7 8 9
   10
group                   subset_Thermostat id 11 12 13 14 15 16
   17 18 19 20
group                   subset_Thermostat id 21 22 23 24 25 26
   27 28 29 30
                                ⋮
group                   subset_Thermostat id 6775 6776 6777 6778
    6779 6780 6781 6849 6850 6851
group                   subset_Thermostat id 6852 6853 6855 6856
    6857 6858 6859 6860 6861 6862
group                   subset_Thermostat id 6863 6864

group                   subset_AO id 6865

log             3.2_Custom.out
#-------------------------------------------------------
# Stage 3.2: ReaxFF Custom Parameters
#-------------------------------------------------------

pair_style              reax/c NULL
pair_coeff              * * C:/MD/data/Forcefields/custom/ffield
   .reax.AO C H N O
fix             qeq all qeq/reax 1 0.0 10.0 1e-6 reax/c


log                     3.3_Velocities.out
#-------------------------------------------------------
```

```
# Stage 3.3: Set the initial velocities for $T
#-----------------------------------------------------------

velocity                all create 298 3051 dist gaussian mom
   yes rot no

velocity            fixed set 0.0 0.0 0.0

log       3.4_Deposition.out
#-----------------------------------------------------------
# Stage 3.4: NVE integration for Deposition for 40000 with a
   timestep of 0.1
#-----------------------------------------------------------

group           deposition_atoms_AO id  6865
delete_atoms    group deposition_atoms_AO

molecule        depositing_species_AO deposition_AO.dat
region          deposition_region_AO block EDGE EDGE EDGE EDGE
   80 89 units box

region          evaporation_region block EDGE EDGE EDGE EDGE 90
   EDGE units box
group           evaporation_group dynamic all region
   evaporation_region every 1000

uncompute       thermo_temp
compute         thermo_temp movable temp

thermo_style    custom step v_time press vol v_sysdensity temp
   ebond eangle edihed eimp evdwl ecoul etail elong pe ke
thermo          ${Nthermo}

fix             1 movable nve
fix             2 evaporation_group reax/c/species 1 1 1000
   removed_fragments.out element C H N O terse yes
fix             3 all evaporate 1000 1000 evaporation_region 1
   molecule yes
fix             dlan subset_Thermostat langevin 298 298 10.0 35
   tally yes
fix             4 movable ave/time 1 40000 40000 v_time
   c_thermo_temp c_thermo_press v_sysvol v_sysdensity v_etotal
   v_pe v_ke v_evdwl v_coulomb v_sxx v_syy v_szz v_syz v_sxz
```

```
    v_sxy f_dlan file 3.4_averages.txt
fix             5 movable ave/time 40  1     40 v_time
    c_thermo_temp c_thermo_press v_sysvol v_sysdensity v_etotal
    v_pe v_ke v_evdwl v_coulomb v_sxx v_syy v_szz v_syz v_sxz
    v_sxy f_dlan file 3.4_instantaneous.txt
fix                     6 all qeq/reax 1 0.0 10.0 1.0e-6 reax/c
fix             deposition_AO_1 deposition_atoms_AO deposit 200
    0 2000 1 region deposition_region_AO mol
    depositing_species_AO vz -0.066308929594968077
    -0.081044247282738763 vx 0.0 0.0 vy 0.0 0.0 near 1.50 id next
     units box
fix             deposition_AO_2 deposition_atoms_AO nve

dump            trj all custom 4000 3.4.Trajectory.xyz id mol
    type q xs ys zs
fix             trjE all ave/time 4000 1 4000 v_etotal v_pe v_ke
     file 3.4.energies.txt
dump            trj_AO deposition_atoms_AO custom 4000 3.4.
    trajectory_1.xyz id mol type q xs ys zs
compute         keatom all ke/atom
compute         peatom all pe/atom
dump            cfg all cfg 4000 *.cfg mass type xs ys zs id
    type q c_keatom c_peatom vx vy vz fx fy fz
dump_modify     cfg element C H N O
compute                 chunk_1 deposition_atoms_AO chunk/atom
    bin/1d z lower 1.0 units box
fix                     chunk_1 deposition_atoms_AO ave/chunk 1
    40000 40000 chunk_1 density/number norm sample file 3.4
    _distribution_AO_1.out
compute                 chunk_2 deposition_atoms_AO chunk/atom
    bin/1d z lower 1.0 units box
fix                     chunk_2 deposition_atoms_AO ave/chunk 1
    40000 40000 chunk_2 density/mass norm sample file 3.4
    _distribution_AO_2.out
timestep        0.1
run             400000

write_dump      all xyz final_3.4.xyz modify element C H N O
undump          trj
unfix   trjE
undump          trj_AO
undump          cfg
```

```
write_dump      deposition_atoms_AO custom 3.4.deposited_1.xyz
    id mol type q xs ys zs

restart         0
dump            sci all custom 400000 3.4.xyz id mol type q xs
    ys zs
run             0
undump          sci
uncompute               chunk_1
unfix                   chunk_1
uncompute               chunk_2
unfix                   chunk_2
unfix           1
unfix           2
unfix           3
unfix           4
unfix           5
unfix           6
unfix           deposition_AO_1
unfix           deposition_AO_2
unfix           dlan
```