

Molecular dynamics simulation of
dipole interactions

David.R.Gilson

Department of Physics, Hull University.

December 2000

*With thanks to my mother, Carol and my supervisor Dr. Thomas Stirner for their
patience, encouragement and support*

Table of Contents

Abstract	5
Chapter 1. Introduction	6
1.1. Early history of computer simulations	6
1.2. Background of this project	8
1.3. Description of the dipole model and simplifications	9
Chapter 2. Theory	11
2.1. Mathematical Model	11
2.1.1. Rules for Notation.....	11
2.1.2. Coordinate Systems	12
2.1.2.1. Simplification by special case of local coordinate system.....	13
2.1.2.2. Local and Global Coordinate systems	14
2.1.3. Cartesian components of systems	15
2.1.4. System of equations used for mathematical model.....	16
2.1.4.1. Total Potential Energy	17
2.1.4.2. Forces	17
2.1.4.3. Rotational Dynamics.....	18
2.1.4.3a. Definition of torque.....	18
2.1.4.3b. Definition of angular acceleration	19
2.1.4.3c. Rotational Displacement and Kinetic Energy	20
2.1.4.4. Spatial Displacement	20
2.2 Computer Model	21
2.2.1 Computer description of dipole	21
2.2.2. Program Structure	23

2.2.3 Output Files.....27

Chapter 3. Results and Discussion..... 28

3.1. Results29

3.2. Discussion.....34

3.2.1. Energetics.....34

3.2.2. Dynamics36

3.3. Review of discussions.....37

Chapter 4. Summary..... 38

4.1. Results38

4.2. Limitations and solutions38

4.3. Future Work.....40

Chapter 5. References..... 41

Abstract

This model investigates the rotation of a rigid body with the molecular dynamics (MD) method. Two ideal dipoles are modelled, both having their centre of mass (COM) fixed in space. One dipole is permitted to rotate about its COM the other is non-moving. Such a configuration leads to the moving dipole rotating under the influence of a dipolar electric field. The relative positions of the two dipoles are chosen such that the moving dipole may only rotate in one plane. This allows the use of a local Spherical Polar (SP) coordinate system (for measuring orientation in space). The mass and length of the dipoles were chosen to reflect atomic magnitudes. The dipole length was set to **5 Angstroms** and monopole mass was **$1.6 \cdot 10^{-26}$ Kg**. All monopoles are given the charge of **± 1 electron charge**.

Using a time step of **$7 \cdot 10^{-18}$** seconds and starting with **zero kinetic energy**, the simulation showed that the dipole was confined in a potential well of the dipolar field. This confinement caused an oscillatory rotation. The amplitude and period of the oscillation were due to the initial position of the two dipoles, which set the initial potential energy. Clearly, with zero kinetic energy, the initial potential energy is equivalent to the total system energy. The period of oscillation was found to be approximately **2 Pico-seconds** for a system-energy of **0.77 eV** (2 d.p.). The value of this period sets a limit on the time-step size for such models in the future.

Chapter 1. Introduction.

1.1. Early history of computer simulations.

The properties of liquids have long been of interest to experimentalists and theoreticians alike. It was not until the latter half of the last century that the computing capability to carry out realistically the large amount of required calculations existed. The first method of simulation was the *Monte Carlo* (MC) technique, which is still widely used to the present day in various applications. The first time a large computer was used for a MC simulation was by Metropolis *et al* in 1953 [1] at the Los Alamos National Laboratory, using the large MANIAC computer. The technique used in their work has since become known as the Metropolis MC method. During this initial period of computer simulations, all simulations were of highly idealised objects such as hard spheres and discs [2]. Many quantities used in computer simulations are based on empirical values gained from experiment. One such example is the Lennard-Jones (LJ) potential which approximates the Van der Waals intermolecular force, which is attractive at long range and repulsive at short range, to give an equilibrium separation between molecules. This force arises from induced dipoles in the charge density of molecules. Wood and Parker in 1959 performed the first MC simulation employing the LJ potential. This allowed experimental data to be compared to computer generated data for the first time [3].

The MC method made use of random numbers to make predictions about a physical system. The Molecular Dynamics (MD) technique simulated the behaviour of molecules by integrating classical/Newtonian physics over time. The first MD simulation modelled hard spheres that had constant velocity between perfectly elastic collisions [4,5]. It was several years before a successful MD simulation of the LJ potential was made. In 1964, Aneesu H Rahman made the first MD simulation of the LJ potential. This simulation

modelled liquid Argon and generated thermodynamic data, which could be readily compared with experimental data of liquid Argon [6].

Following this break through, the area of computer simulations was more intensively researched, and more efficient and widely used algorithms were developed. The Verlet algorithm [7] for instance was perhaps the most widely used method of integrating the equations of motion. The Verlet method has good energy conserving properties, even with long time steps. Time steps are the period of time that the algorithm integrates the equations of motion over. Long time steps are desirable because they allow for larger amounts of time to be simulated over. However, long time steps can lead to larger discontinuities in the motion of particles, which then leads to less accuracy and energy is not conserved well. As an example of the Verlet method, simulations of liquid Argon at the triple point give RMS energy fluctuations in the order of 0.01% with a time step of 10^{-14} seconds. This increases to 0.2% for a time step of approximately $4 \cdot 10^{-14}$ seconds [7,19,20]. The Verlet algorithm has been modified many times to overcome the inherent computational disadvantages. In the Verlet algorithm, velocities are difficult to compute and may needlessly introduce numerical imprecision [16]. An improved Verlet algorithm has become known as the *leapfrog* method. This is so called because it used mid-time-step points to calculate velocities [12,15]. This method is not completely satisfactory either. A further modification is known as the *Velocity Verlet*, which works by storing position, acceleration and velocity at a time, t .

Soon, simulations of rigid non-spherical bodies were of interest, in particular the water (H_2O) molecule. For such simulations intra-molecular forces are taken to be at least an order of magnitude greater than the inter-molecular forces. Bond vibrations are not appropriate for rigid body simulations. Given the size of time steps used in MD simulations the period of bond vibrations are so small that the positions of atoms would

average out to a constant value anyway. Also, the amplitude of vibrations may be so small that they are insignificant to the bond lengths and can hence be neglected. Following the work of Rahman [6] and Verlet [7], two simulations of diatomic molecular liquids were made [8,11]. During this time, there were also two interesting simulations of liquid water made; the first in 1968 was an MC simulation by Harp and Berne [9], the second in 1971 was a MD simulation by Rahman and Stillinger [14]. The Water molecule continues to be the most interesting and difficult molecule to model because of its anomalous hydrogen bonding properties [25].

A more recent alternative simulation method is Continuum Methods. Continuum methods replace individual particles by a bulk force field that approximates the presence of many molecules. This is more computationally efficient but lacks the accuracy of molecular dynamics. Work has been done to mix MD and Continuum methods for investigating acids in a water solvent. This method of mixing techniques works by using MD for the N nearest neighbouring layer of water molecules and beyond this range using Continuum Methods to approximate the rest of the bulk water. [27].

1.2. Background of this project.

This undergraduate project has been done in the Chemical physics theory group of Hull University. MD research in the group is fairly new and work has only been done for hard spherical particles. The project aims to explore and learn how to create a model of a non-spherical molecule that can freely rotate in space. Such a model, even of a simple diatomic molecule would have many uses. Eventually, a simulation of water molecules would be of interest in several areas. For example; investigating the interaction between water and certain polymer surfaces, use as the suspension for colloid simulations, use as a solvent for other molecules such as proteins. It was chosen for this project to take the

simplest possible structure to model, a dipole. Once a dipole can be successfully modelled, the charges and masses of the monopoles and the bond length between them can be set to describe a particular diatomic molecule. Further more, many dipoles could be modelled for lattice structures and liquids.

1.3. Description of the dipole model and simplifications.

In this project the dipole consists of two point charges of equal mass separated by a fixed length rigid bond.

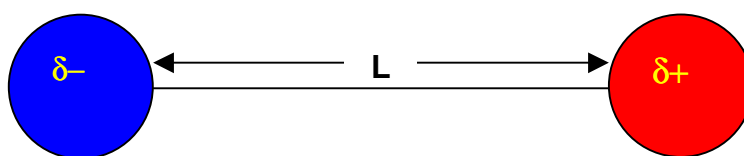


Figure 1.1. Schematic of dipole model.

Since this project is focused on the rotation of a rigid dipole, a further simplification was made. The dipoles would clearly rotate about their centre of mass (COM) under the influence of a torque. Therefore the position of the COM's are kept constant, such that the dipoles only rotate, translation through space of their COM is not permitted. The simulation simply has two dipoles, so it can be seen how one dipole affects another. Furthermore, only one dipole is allowed to rotate. By the process of Occam's razor it can be seen how the dipole's kinetic energy and potential energy is affected by the presence of a neighbouring dipolar field. See fig. 1.2 for a visualisation of how the dipoles would be arranged in space.

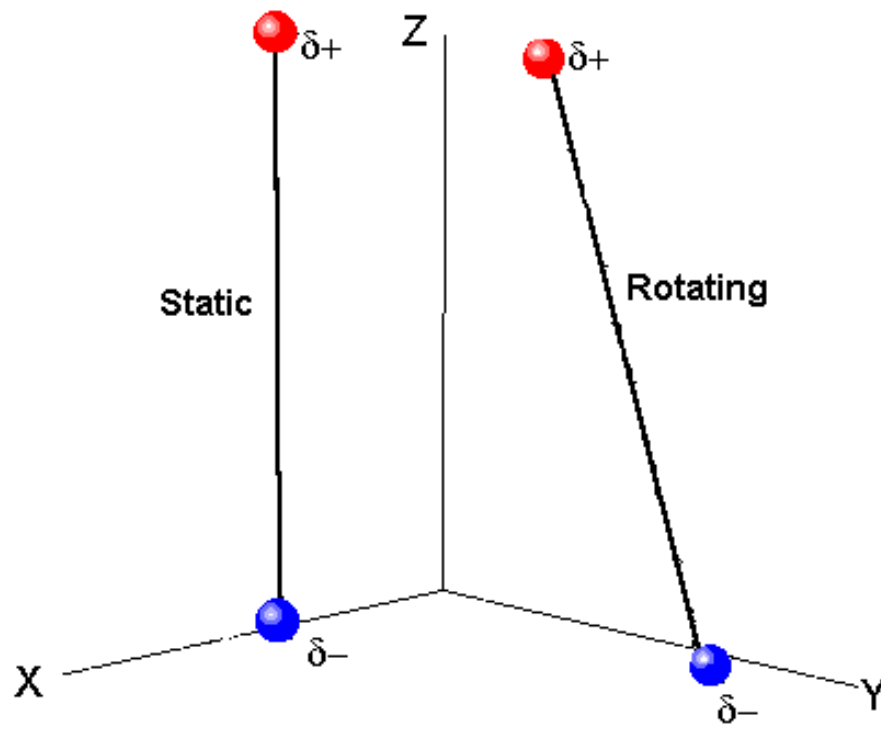


Figure 1.2. Spatial configuration of dipoles.

Chapter 2. Theory

2.1. Mathematical Model

This model has had many simplifications so that the behaviour of a single dipole may be clearly observed. The conditions of this model state that there are just two dipoles. Where one dipole is static in space and is not permitted to move in any way, therefore it has zero kinetic energy. The other dipole may only rotate, its centre of mass is fixed in space, and therefore it has zero translational kinetic energy but non-zero rotational kinetic energy. Therefore, it can be seen that all the energy in the system comes from the rotational kinetic energy of one dipole plus the total potential energy between the two dipoles. (See *fig. 1.2*).

2.1.1. Rules for Notation

This simulation is essentially a four-body problem, when considering each monopole individually. In the cases of spatial coordinates, forces and potentials, where specific monopoles are being referred to, it is important to have a clear system with which to address each monopole.

Both dipoles and monopoles are given numerical *names*. It is advantageous to name them in this way because it is completely compatible with the way in which they are addressed in the computer program. When addressing dipoles, they are simply designated **0** and **1**, where **0** is the first dipole and **1** is the second dipole. (This is clearly arbitrary, but does provide for consistent handling of quantities). For monopoles, the positive monopole is **always** designated **0** and the negative monopole is designated **1**. Therefore the name of the positive monopole on the first dipole is **00**, and the positive monopole on the second dipole is **10**, and the negative monopole on the first dipole would be called **01**. These so-

called names are often used as subscripts on quantities to denote for which dipole/monopole they correspond to. Often for a quantity such as those mentioned at the beginning of this section, the subscript has four characters. The first two always denote the monopole on the first dipole and the second pair always denotes the monopole on the second dipole. For instance, the resultant force between the negative monopole on the first dipole and the positive monopole on the second dipole would be denoted as $\mathbf{F}_{0,1,1,0}$ (shown in fig 2.1).

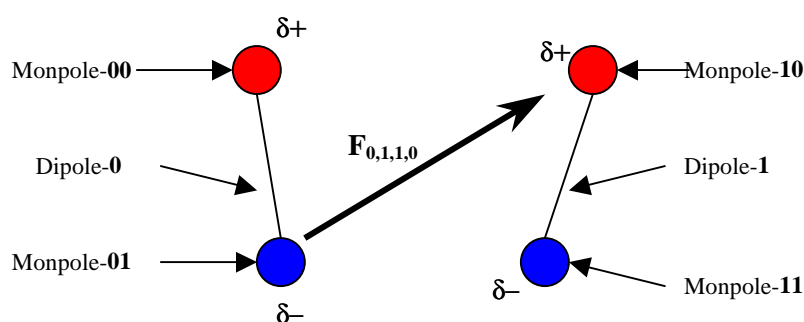


Figure 2.1. Display of the naming convention for dipoles and monopoles.

2.1.2. Coordinate Systems

The choice of coordinate system plays a very important role in the mathematical model. It is therefore important to choose the best coordinate system to fit the application. The most commonly used coordinates are Cartesian coordinates. These are simply three mutually orthogonal axes, X, Y, and Z. Spherical Polar (SP) coordinates allow any point on the surface of a sphere to be defined with two angles ϕ and θ (with optionally r , a radial distance from the origin).

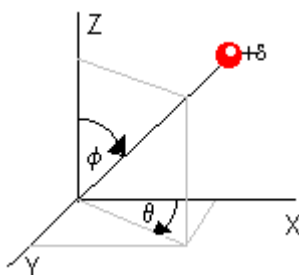


Figure 2.2. Spherical Polar coordinates.

However this system does not work well for describing rotations in multiple planes.

Euler Angles however are composed of three angles (ϕ , θ and φ). This system works very well for describing general rotations in multiple planes [21, p.86].

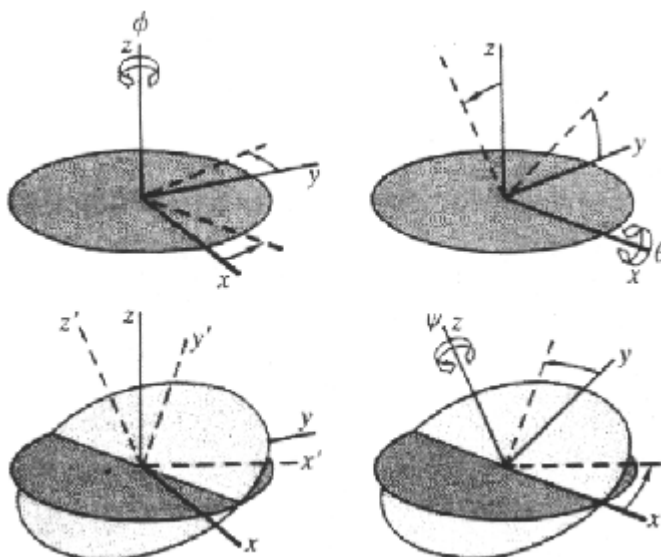


Figure 2.3. Euler Angles.

2.1.2.1. Simplification by special case of local coordinate system

Despite Euler angles being far more versatile for performing rotations of an object, it is hard to compute. If an initial configuration in space can be chosen such that it is clear the dipole can only rotate in one plane, then SP coordinates can be chosen. This simplifies the mathematical treatment as well as the required computer programming, yet still yields the same results as Euler angles in the same configuration.

Such a configuration can be easily seen to exist. If dipole-0 is held static (as discussed) and parallel to the Z-axis, then dipole-1 can be at any angle to the Z-axis as long as it completely lies in the YZ plane. It should be noted that the overall dipole torque has two contributions. Each monopole contributes to the overall rotation similar to a seesaw or a balancing scale. From such a configuration, all X-components of torques will cancel out because of the symmetry of the configuration, i.e. it is balanced. Furthermore, there are no X-components in the dipole's length; hence, all distances between the COM and the monopoles are zero in the X-direction. See section **2.1.4.3a** definitions of torques.

2.1.2.2. Local and Global Coordinate systems

With the previous section in mind, the choice of coordinate systems for the simulation is clear. A Global coordinate system will track the COM positions of the dipoles, as well as being the coordinate system in which all dipole-dipole interactions are calculated. Since the global system will only be used for point values as well as line of sight values, which have Cartesian components (e.g., force), it is logical to choose a Global Cartesian coordinate system. Having a local coordinate system simplifies the calculation of the dipole's rotation. It has already been stated that only one dipole will rotate and only then in a single plane, so to choose a SP local coordinate system is adequate (*See fig. 2.2*). In this case if θ is measured clockwise from the x-axis, then this can be held at a value constant at $\pi/2$ radians, and the angle ϕ (clockwise from the z-axis) can be used to measure the dipole's rotation in the $\theta=\pi/2$ (YZ) plane. To convert SP coordinates back into Cartesian coordinates compatible with the global system there are a standard set of transforms.

$$dx = l \cdot \sin(\phi) \cdot \cos(\theta) \quad (2.1)$$

$$dy = l \cdot \sin(\phi) \cdot \sin(\theta) \quad (2.2)$$

$$dz = l \cdot \cos(\phi) \quad (2.3)$$

Where l is the full length of the dipole.

These transforms give us local Cartesian coordinates relative to the COM. It should also be noted here, that these coordinates are the coordinates of the positive monopole since the SP angles are measured from axes to the positive monopole. Given the symmetry of the dipole, the transforms for the negative monopole are simply the negative of the positive monopole transforms.

To completely transform the monopole coordinates to the global coordinate system, simply add together the local monopole Cartesian coordinates to the global Cartesian COM coordinates. This is shown below, the additions refer to the positive monopole and the subtractions refer to the negative monopole.

$$X = X_{COM} \pm dx \quad (2.4)$$

$$Y = Y_{COM} \pm dy \quad (2.5)$$

$$Z = Z_{COM} \pm dz \quad (2.6)$$

2.1.3. Cartesian components of systems

Since it is a standard technique to divide almost all dynamic quantities into Cartesian components, there are certain geometrical issues to be considered. Of prime interest in this work is the effect on dipole torques. The overall torque is divided into component torques, each parallel to the Cartesian axes. Calculating a torque component entails looking at the dipole along the axis about which the rotation is being calculated. The dipole has a fixed length in space. This length can be said to have X, Y and Z components

in space (depending on orientation). Consider the torque parallel to the x-axis, when looking along the x-axis only the Y and Z components of its length can be seen, since the X components of its length are parallel to the line of sight and hence have zero length.

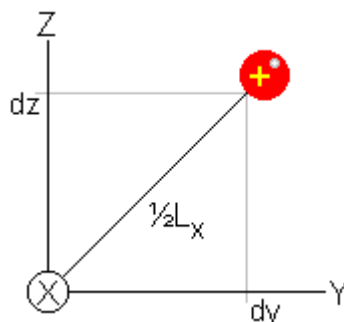


Figure 2.4. Illustration of Virtual Dipole length viewed parallel to the X-axis.

Unless the dipole is axially aligned, considering the dipole from the perspective of each axis, the whole length would never be observed. Looking along the X-axis, only the length in Y and Z are seen. Looking along the Y-axis only the X and Z lengths are seen, etc. The axial torques also see these lengths, therefore the orientation of the dipole in a given coordinate system affects the value of the torque (and all quantities derived from the torque). This leads to the concept of *Virtual dipole lengths*, which are simply **projections** of the dipole length parallel to each axis. Consider Fig. 2.4, which shows the virtual dipole length in X-direction, the length seen from the COM to monopole is clearly half the total length of the dipole, this leads us to the general form for calculating a virtual dipole length as.

$$l_i = 2 \cdot \sqrt{d_j^2 + d_k^2} \quad (2.7)$$

2.1.4. System of equations used for mathematical model.

This simulation, like any other MD simulation uses a set of equations that are circularly interdependent. Initially the potential energy of the system is evaluated by summing all the inter-dipole potentials between monopoles based on initial monopole coordinates.

Next, the forces on all monopoles are calculated and resolved into Cartesian components. The forces are then used to calculate the torques on the dipoles; in turn the angular acceleration is derived from the torques. Subsequently, the rotational kinetic energy and angular displacements are calculated from the angular acceleration. Finally, the angular displacements enable the calculation of the new monopole coordinates. The new coordinates allow calculation of the new total potential energy value, and so on.

2.1.4.1. Total Potential Energy

The total potential energy is summed over all the potentials between all inter-dipole monopole pairs. The overall expression (and algorithm is);

$$V = \sum_{i=0}^1 \sum_{j=0}^1 V_{0,i,1,j} \quad (2.8)$$

Where the monopole pair potential $V_{0,i,1,j}$ is expressed as;

$$V_{0,i,1,j} = \frac{q_1 \cdot q_2}{4\pi\epsilon_0 \cdot r_{0,i,1,j}} \quad (2.9)$$

The variable $r_{0,i,1,j}$ is the radial distance between monopole $0\mathbf{i}$ and $1\mathbf{j}$.

2.1.4.2. Forces

The expression for force obeys the general form of,

$$F = -\nabla \cdot V \quad (2.10)$$

Which when evaluated gives,

$$F_{0,i,1,j} = \frac{q_1 \cdot q_2}{4\pi\epsilon_0 \cdot (r_{0,i,1,j})^2} \quad (2.11)$$

This is a resultant force between monopoles, however Cartesian components are required.

Fortunately, they are easily evaluated, algebraically and computationally.

$$(F_{0,i,1,j})_X = \left(\frac{dx}{r} \right) \cdot F_{0,i,1,j} \quad (2.12)$$

$$(F_{0,i,1,j})_Y = \left(\frac{dy}{r} \right) \cdot F_{0,i,1,j} \quad (2.13)$$

$$(F_{0,i,1,j})_Z = \left(\frac{dz}{r} \right) \cdot F_{0,i,1,j} \quad (2.14)$$

Where dx, dy and dz are the Cartesian components of \mathbf{r} (the subscript is neglected on the spatial terms, it is implicit that they refer to the same monopoles referred to by the force subscript). Equations 2.12, 2.13 and 2.14 can be easily proved by use of Pythagoras theorem to recombine the components.

2.1.4.3. Rotational Dynamics

2.1.4.3a. Definition of torque

The general expression for torque is used here,

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F} \quad (2.15)$$

In general this would expand to,

$$\tau_X = dy \cdot F_Z - dz \cdot F_Y \quad (2.16)$$

$$\tau_Y = dz \cdot F_X - dx \cdot F_Z \quad (2.17)$$

$$\tau_Z = dx \cdot F_Y - dy \cdot F_X \quad (2.18)$$

The dx, dy and dz terms are the Cartesian transforms of the SP local coordinates. In other words, they are the distances between the monopole and the COM.

For each torque component, the same argument for virtual dipole lengths (section 2.1.3), applies to force components. Given that in this case it is known that only the x-torque will be non-zero, hence, only the X component (eqn. 2.16) shall be used. It can be seen from

section 2.1.2.2 that equations 2.17 and 2.18 will become zero since monopole contributions cancel each other out, as seen in the orientation discussion earlier.

2.1.4.3b. Definition of angular acceleration

The following relation is well known (L is angular momentum and I is the moment of inertia, ω is the angular velocity),

$$L = I \cdot \omega \quad (2.19)$$

Where for a dipole with monopole mass M and length R (projected along axis), the moment of inertia is,

$$I = 2MR^2 \quad (2.20)$$

(Obviously dipole weighs twice as much as a monopole).

It follows from eqn. 2.19 then,

$$\dot{L} = I \cdot \dot{\omega} \quad (2.21)$$

Where \dot{L} and $\dot{\omega}$ are time derivatives.

It is also well known that,

$$\dot{L} = \tau \quad (2.22)$$

Therefore,

$$\tau = I \cdot \dot{\omega} \quad (2.23)$$

Which finally leads to an expression for angular acceleration,

$$\dot{\omega} = \frac{\tau}{I} \quad (2.24)$$

2.1.4.3c. Rotational Displacement and Kinetic Energy

The instantaneous rotational velocity of the dipole about any axis for the i^{th} iteration of the simulation is calculated by,

$$\omega_i = \omega_{i-1} + \dot{\omega}_i \cdot dt \quad (2.25)$$

Where dt is the time step for the simulation. Equation 2.25 shows that angular acceleration and instantaneous rotational velocity are calculated in the same iteration. Clearly, angular acceleration must be calculated first in each iteration.

Hence the angular displacement can be expressed as,

$$d\phi = \omega \cdot dt \quad (2.26)$$

Similarly the (rotational) kinetic energy in the i^{th} iteration is calculated as,

$$KE_i = \frac{1}{2} \cdot I \cdot \omega_i^2 \quad (2.27)$$

2.1.4.4. Spatial Displacement

The spatial displacement of the monopoles is governed by the change the SP ϕ coordinate. In the i^{th} iteration of the simulation, the new value for ϕ is calculated as,

$$\phi_i = \phi_{i-1} + d\phi \quad (2.28)$$

The θ angle does not need to be re-evaluated in this case, since it is explicitly assumed that the dipole will not move beyond the YZ plane. Once ϕ has been re-evaluated, the Cartesian coordinates can be updated, both locally and globally for all monopoles. See equations 2.1 to 2.6 in section 2.1.2.1.

2.2 Computer Model

The program is written in the C programming language [22]. This language is chosen because of its power and flexibility. Other languages are often constrained in such a way that relatively simple algorithms can be complicated to implement. C however, allows you to do almost anything you want, in a simple and efficient way. The work in this project has been carried out in the Unix environment.

The program itself consists of three files; **dipole.c**: This is the main *source code* which contains the main function and all sub-routines., **dipole.h**: This is the *header file*, this contains all the numerical constants and definitions of the dipole structures (see next section), **prototypes.h**: This is the *prototypes file* which contains a list of all sub-routines. The **compiler program** (“cc”) requires information about the sub-routines prior to compilation of source code.

2.2.1 Computer description of dipole

One of the biggest challenges when writing a program with many related values, is a systematic and easy to understand naming system. The C programming language has a facility known as *structures*. A structure is similar to an array variable. However, all the elements of the structure may have individual names and do not have to be of the same variable type. Furthermore, one can nest structures within structures and even declare arrays of structures. This flexibility was capitalised on when forming the model of the dipole. Structure and element names are simply separated by a period, when being called in a program, for example;

structure.element1.element2.

Array elements are called by an index number, for example, if the above structure was declared as an array with two elements, the following two lines of C code could access both array elements.

```
structure[0].element1.element2
```

```
structure[1].element1.element2.
```

This provides the basis for a logical and systematic naming system suited to the dipole-monopole naming system.

Firstly, many quantities in the simulation have Cartesian components, so a structure named “threeD” was defined which had three elements called “X”, “Y” and “Z”, all of type *double* (a floating point variable, highest available precision). Next, a structure was defined to describe a monopole, by virtue of all its properties. Then a structure to describe a dipole was defined, this included properties that were exclusive to a dipole, as well as having an array of two monopole structures, since a dipole indeed has two monopoles. In doing this, the dipole *inherits* all the properties of the monopoles and monopole properties would then be called in the program explicitly as part of the dipole.

The monopole structure included the following quantities; Mass (constant), force (external force felt, of type ‘threeD’), distance from COM (local Cartesian coordinates, of type ‘threeD’), torque (each monopole must contribute to the overall torque of the dipole, of type ‘threeD’).

The dipole structure included the following quantities; COM coordinates (of type ‘threeD’), virtual length (projections of length discussed in section 2.1.3, of type ‘threeD’), theta and phi (SP local coordinates, of type double), rotational velocity (of type ‘threeD’), rotational kinetic energy (of type ‘threeD’), and two monopoles (see above).

The program declares an array of two dipoles, thus an array of two dipoles, which have two monopoles each, perfectly suits the naming system described in section 2.1.1.

Below is an example of this naming system showing the full name of the resultant force on monopole **10** in the y direction:

```
dipole[1].mpole[0].force.y
```

2.2.2. Program Structure

The program is based on its main function. The main function begins with a series of initialization commands. These are followed by a *loop*, which contains commands to call sub-routines, energy evaluations and output functions. The Main function is completed with commands to close all of its output files.

The initialization section of the program declares all the variables that will be used, including the dipole array. The dipole array is declared as a *global* variable, which simply means all of its elements can be accessed from any subroutine. Next, all the required output files are created, ready for data to be saved in them. A special function is then called to initialize all the values in the dipole structures; this takes values that are *hard-coded* into the dipole (such as dipole length and monopole mass and COM coordinates which are constants, and initial SP coordinates which are altered between simulation runs). All other values such as forces, torques and kinetic energies, are all set to zero, to make the system start from rest. This function also calls two sub-routines that are utilized later on by the main loop. The first calculates the local Cartesian coordinates of the monopoles (see equations **2.1** to **2.6**), the second calculates all of the virtual dipole lengths. Upon completion, control is returned to the main function. Having now set all of the coordinates, the main function then calls the potential energy function and saves the value as the total energy of the system. See section **2.1.4.1** for details on the potential energy expression. This is done because the system is explicitly started from rest, so there is zero kinetic energy. This value is then stored as the initial total energy of the system. At the end of the simulation, the initial total energy can be compared with the final total

energy, to see how well the energy is conserved through the simulation. This completes the initialization phase, the program then continues to the main simulation loop.

The main simulation loop begins by calling the force calculation sub-routine. This performs a set of calculations, which will give all the Cartesian components of the resultant force on each monopole. The force routine operates by calculating all the forces on the monopoles of the first dipole (dipole-0) from the second dipole (dipole-1). It then sums all the forces on each monopole and similarly infers (by Newton's third law) and sums all the forces on the monopoles of the second dipole. See section **2.1.4.2** for the force equations. Having calculated forces the main function loop then calls the sub-routine which contains all of the rotational dynamics calculations; these are based on the equations in section **2.1.4.3**. This sub-routine first calculates moments of inertia and torques, then angular accelerations, angular displacements and rotational kinetic energy. The main function loop then calls the monopole coordinate update and virtual dipole length subroutines to update respective variables. The main function loop then calls the potential energy sub-routine to recalculate the potential energy based on the new coordinates.

Having gone through all the main equations, the kinetic energy components are summed together to obtain the total kinetic energy of the system. The total kinetic energy and total potential energy are summed together and saved as the total energy of the system. This sum should always stay constant if the simulation is to obey the law of conservation of energy. It does not actually stay constant because of the precision of the calculations. The deviation of this value is a good gauge of the simulation's accuracy. As well as total energy, total kinetic energy and total potential energy, the change of these energies is also saved to output files so the energy derivative can be seen. If the program is operating

correctly, the change in kinetic energy should reflect the change in potential energy through time.

The main loop continues until it has performed the number of iterations specified in the header file. Once the loop exits, the main function prints to the computer's screen the initial and final total system energy and the duration of the simulation (which is the number of iterations multiplied by the time step). Finally the main function closes all the output files and terminates.

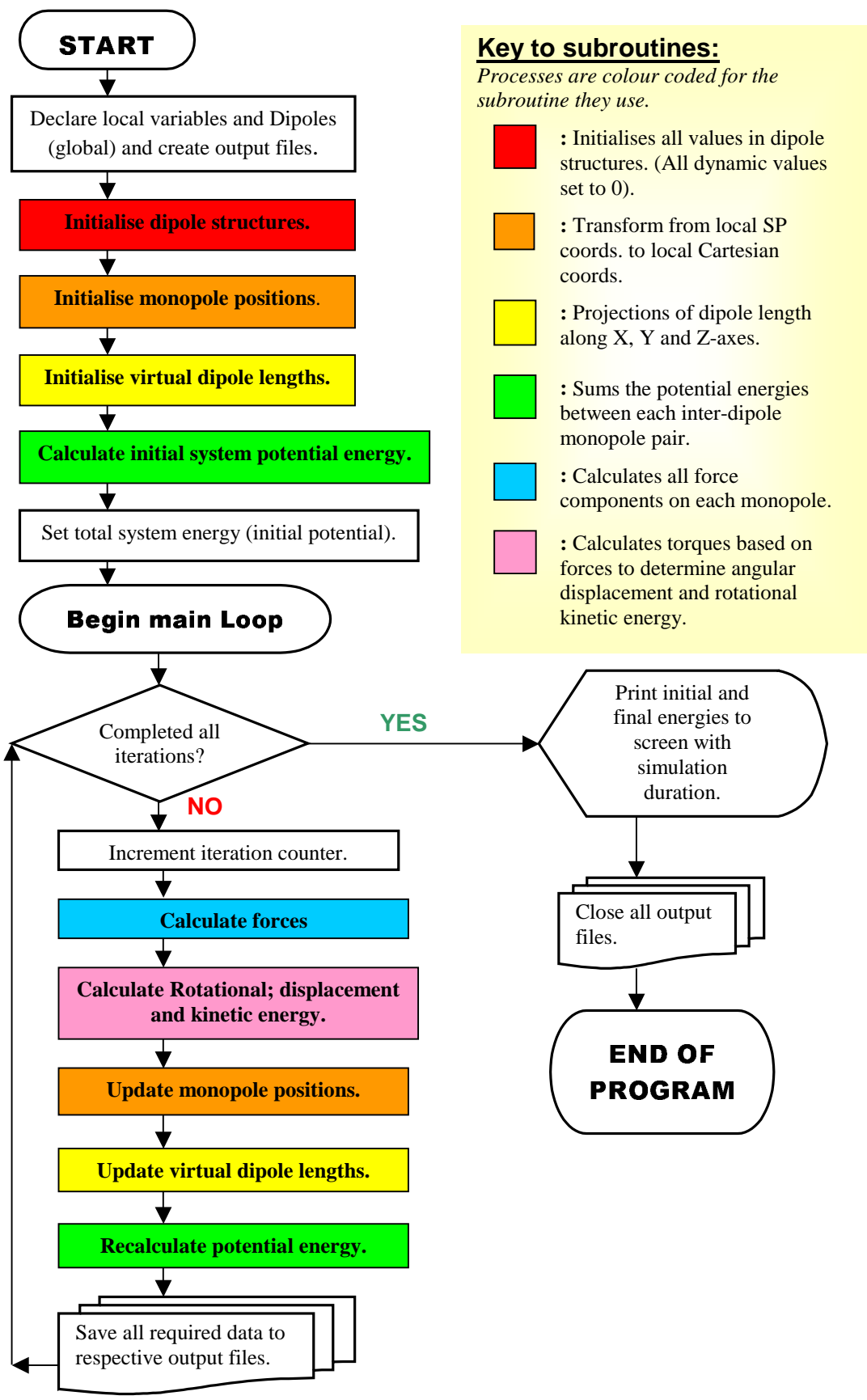


Figure 2.5. Schematic of the program structure.

2.2.3 Output Files

The output of the program is in the form of ASCII text files. These files are then taken and are loaded into commercial graph plotting software (example being XMGR for Unix work stations and Origin 4.0 for Microsoft Windows work stations).

There are two types of output file from the simulation, energy profiles and spatial trajectories. The energy profiles are two-dimensional plots, energy versus time. Total energy, change in total energy, total kinetic energy, change in total kinetic energy, total potential energy and change in total potential energy are all saved in separate files. Plots of these can be seen in Chapter 3. The spatial plots include trajectories of both monopoles on dipole-1 (**10** and **11**) through three-dimensional space. Also, since the monopoles do not move in the X direction, there are also 3D plots of the paths of monopoles **10** and **11** in the Y and Z directions through time. These are also shown in Chapter 3.

Chapter 3. Results and Discussion

Before any results are shown or discussed, the parameters of the simulation, which all results clearly depend on, should be reviewed. Despite the simulation not actually simulating any specific physical system, parameters were chosen to reflect the orders of magnitude used in MD simulations. The monopole masses were set to approximately the mass of an Oxygen atom. The monopole charges were simply set to **± 1 electron charge** ($\pm 1.6 \cdot 10^{-19}$ C) respectively and the dipole length (c.f. a bond length) was set to **5 Angstroms**. As discussed in chapters 1 and 2, the SP coordinates were chosen so that dipole-1 was perpendicular to dipole-0 with respect to the X-axis ($\theta_0=0$ and $\theta_1=\pi/2$, θ measured clockwise from X-axis). The dipole COM coordinates were also chosen so that the dipoles were close but not so close that monopoles could collide. In practice this is the effect of the LJ potential, to give an equilibrium separation of molecules. All dynamic quantities of the dipoles were initially set to zero to infer the system was starting from rest.

Other quantities in the simulation were the number of iterations and the time step. The order of magnitude of the time step was estimated by reviewing the values used in equations 2.24 to 2.26. The specific value was altered by a process of trial and error. The number of iterations was again set by trial and error. It is worth noting that whereas a large number of iterations are desirable to have a good picture of the simulation through time; a constraint arises from the considerable amount of data produced, which could be impractical to handle. Below is a table summarising all parameters for this simulation.

Constants	
Time Step	$7 \cdot 10^{-18}$ Seconds
Iterations	144458 (=2.1ps)
Monopole mass	$1.6 \cdot 10^{-26}$ Kg
Monopole charges	$\pm 1.6022 \cdot 10^{-19}$ C
Dipole length	5 Å (both dipoles)
COM coordinates	Dipole-0 (x,y,z)=3Å, 3Å, 3Å Dipole-1 (x,y,z)=9Å, 3Å, 3Å
Initial Values to variables	
Initial SP coordinates	Dipole-0 (θ, ϕ) = 0, 0 Dipole-1 (θ, ϕ) = 0, $\pi/4$
Dynamic quantities	<i>All quantities on both dipoles set to zero</i>

Table 3.1. List of constants and initial values.

3.1. Results

Initially, energy profiles for the simulation are plotted. By plotting the total system energy, it can be gauged how well energy was conserved in the simulation. Total energy should clearly remain constant over time if energy is conserved well. Also, a plot of kinetic and potential energy should reflect each other over time if total energy is to remain constant. See Figure 3.1.

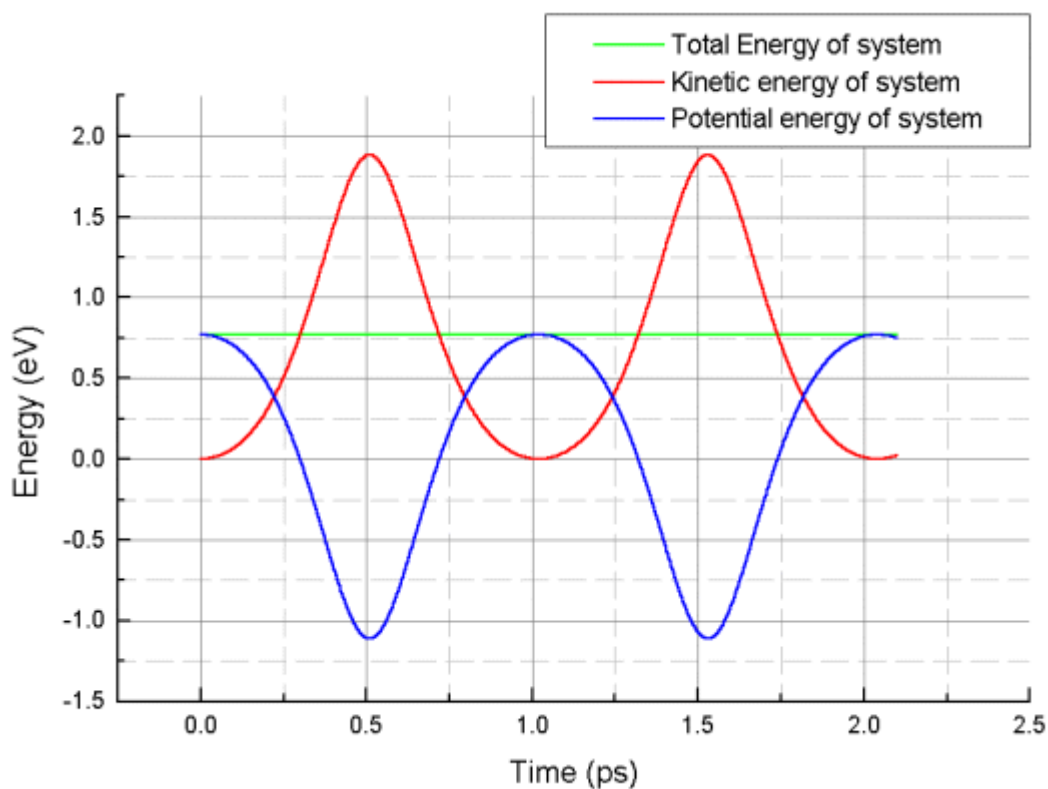


Figure 3.1. A composite plot of total system energy, kinetic and potential energy.

Summing the kinetic and potential energies in each loop of the simulation calculated the total system energy line in fig. 3.1 (shown in green).

It can also be seen how kinetic and potential energy vary not just with time but also with angular separation between the two dipoles (See Figures 3.2. and 3.3.).

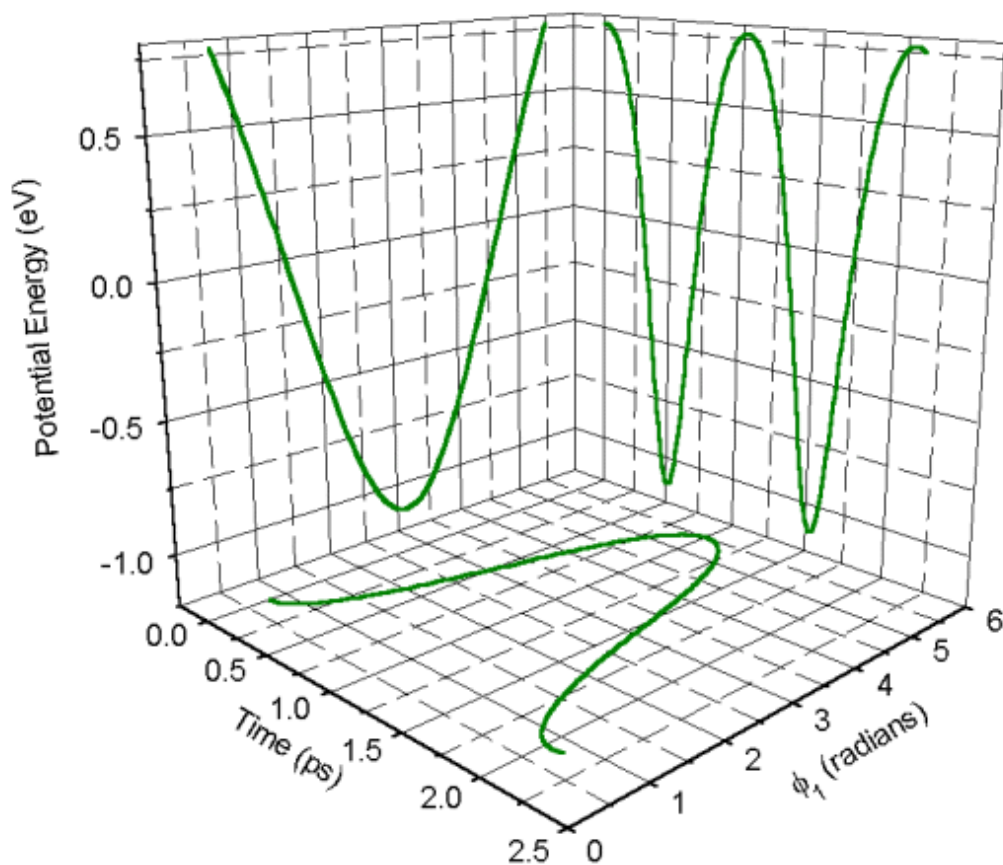


Figure 3.2. Variation of potential energy with angular separation and time.

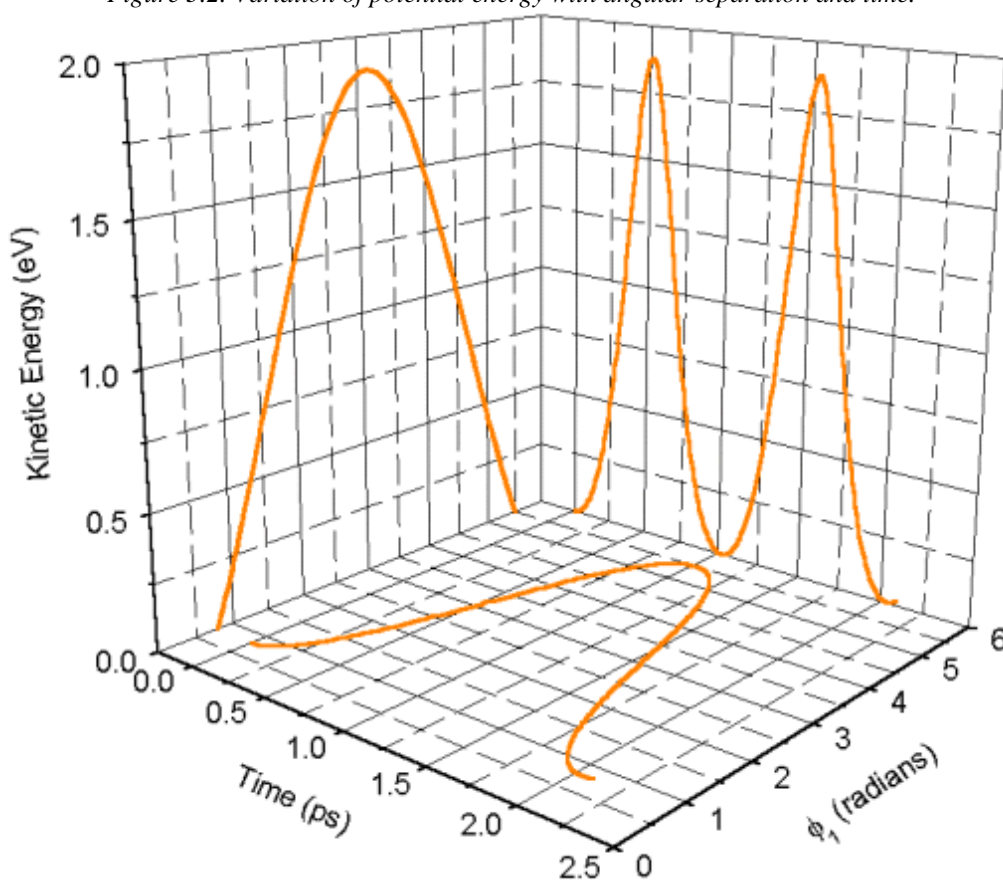


Figure 3.3. Variation of kinetic energy with angular separation and time.

Figures 3.2 and 3.3 were obtained by plotting a 3 dimensional data set of time, SP ϕ angle on dipole-1 and the potential/kinetic energy (respectively). Please note the subscript on the ϕ_1 axis label, it means this is the ϕ angle on dipole-1. Then, rather than plotting a 3 dimensional curve, projections of the data on each plane are shown to make the data more readable.

Having now considered the energetic variation of the simulation, Figs. 3.2 and 3.3 bring us on to considering the actual motion of the dipole through space. The variation of ϕ_1 with time, clearly demonstrates rotational displacement. The next figure shows the positions of both dipoles in space and how dipole-1 rotates.

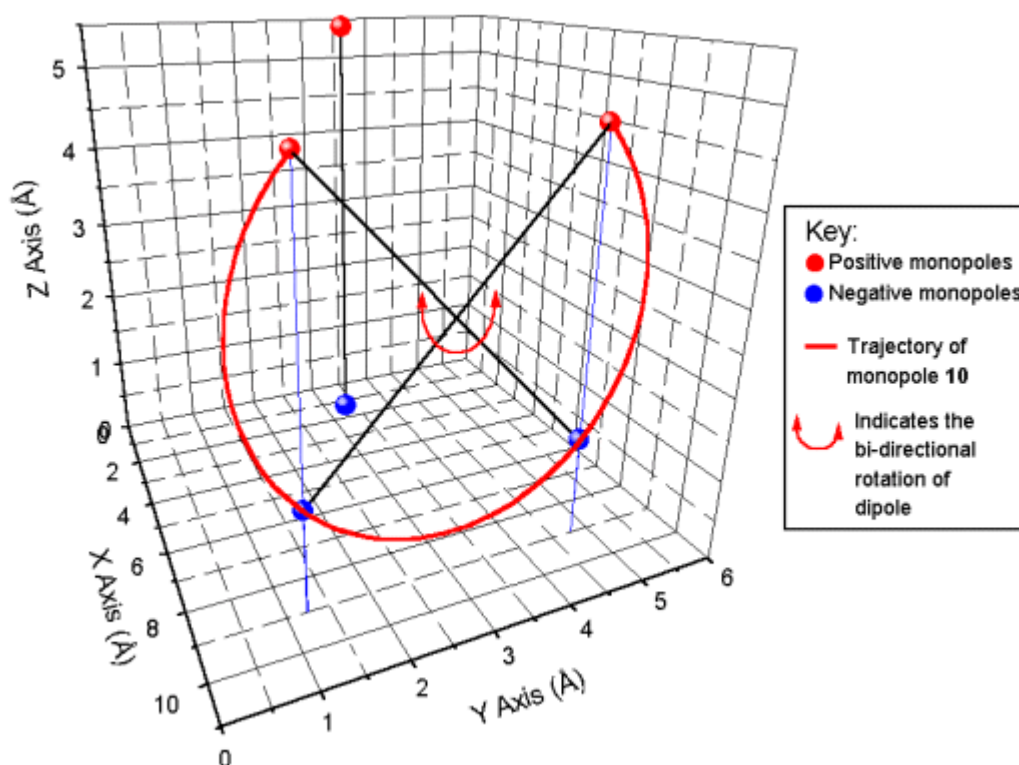


Figure 3.4. Schematic of Dipole positions and movement.

Figure 3.4 is actually a plot of data. The red curve is the trajectory of monopole **10**. The spheres represent the initial, intermediate and final coordinates of the monopoles, with the

connecting black lines representing the bond between monopoles. The blue anchor lines from the monopoles of dipole-1 (which is rotating) are there to demonstrate that the dipole has no X-component in its path.

Figures 3.2 and 3.3 show a periodic variation of ϕ_1 with time. This implies an oscillatory motion. This is represented in fig. 3.4 by the bi-directional arrow. However, this periodic motion can be seen more clearly in fig. 3.5

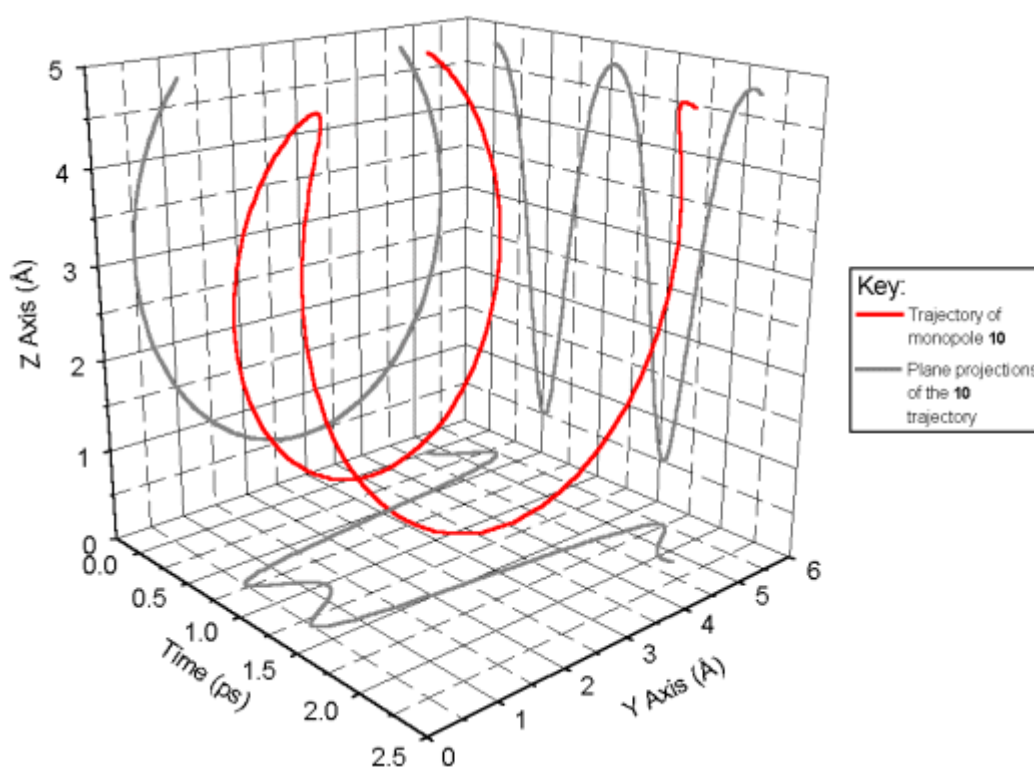


Figure 3.5. Path of monopole **10** in the YZ plane through time.

Figure 3.5 shows the path of monopole **10** in the YZ plane through time. Projections of this path are made on each plane (in grey) to allow better visualization of the movement. This is essentially the same as the trajectory in fig. 3.4, but the x-axis has been replaced with time. This is demonstrated by comparing fig. 3.4 with the YZ plane in fig. 3.5.

Finally, when the program terminates, it returns information about the simulation. The initial and final total energies are shown for comparison and the duration of the simulation is given. This is shown in the table 3.2.

Initial total system energy (eV)	0.770025
Final total system energy (eV)	0.770027
Energy drift (% of initial)	+0.0003%
Simulation duration (Pico-seconds)	2.1

Table 3.2. System energy and duration of simulation.

3.2. Discussion

The results shown in the previous section highlight the key features of this simulation. With respect to the dynamics of the simulation it has been shown how the dipole moves and that the motion is oscillatory. With respect to the system energies, it has been shown that energy is conserved well and which relative positions of dipoles give a potential energy minimum.

3.2.1. Energetics

Figure 3.1 shows that the change in kinetic and potential energy over time were well balanced. Fig. 3.1 also shows that total energy remained (approximately) constant over time, which is to be expected given that the total energy is the sum of kinetic and potential energy. As shown in table 3.2 the actual difference between the initial and final total energy was only 0.0003%. This is clearly sufficient accuracy. Also, it suggests that for future simulations the time step could be allowed to be longer than it was in this

simulation (see table 3.1). A longer time step would increase the 'energy drift'. However, given the value 0.0003%, an increase on this could be tolerated.

Figures 3.2 and 3.3 show the variation between time, energy and relative angular separation. Note, since only dipole-1 moves, the ϕ_1 SP coordinate is effectively the relative angular separation. It is of interest to know which relative positions give us a potential energy minimum. This will tell us the configuration a system of many bodies would tend to relax into over time. Consider the potential energy minima in fig. 3.2, it can be seen that these correspond to a value just greater than 3 radians. Upon closer examination of the data file itself, this value is approximately 3.14 (approximately π) radians. This strongly suggests that **the energy minimum position for two dipoles is anti-parallel**. The angle, ϕ , is always a measure clockwise from the Z-axis to the positive monopole. Therefore an angle of π would position the dipole parallel to the Z-axis with the negative monopole pointing in the positive Z direction from the dipole COM. In this case ϕ_1 is the effective angle between dipoles because ϕ_0 has a constant value of 0 radians. Therefore, since dipole-0 is parallel to the Z-axis with its positive monopole pointing in the positive Z direction from its COM, then dipole-1 must also be parallel to the Z-axis with its negative monopole pointing in the positive Z direction. Obviously, this description only applies to the coordinate system used in this particular case, but for two dipoles with fixed COM's, these results suggest that anti-parallel would give an energy minimum position. If both dipoles could freely rotate about their COM's, this particular configuration may only be a local potential energy minimum, since there could be more complicated configurations that would yield an even lower 'potential well'.

3.2.2. Dynamics

Figure 3.4 shows that the movement of dipole-1 is circular in the YZ plane. This is to be expected given that dipole-1 has a fixed COM and that it has already been shown that it would only feel a torque about the X-axis (see section **2.1.4.3a**). Looking at fig. 3.4 one may carelessly assume that it is an incomplete circle because the simulation did not run long enough. Upon examination of the ϕ_1 -time curves on either fig. 3.2 or 3.3, it can be seen that the dipole rotates from its initial position to a maximum angle, and then rotates back to approximately where it started. This would seem to indicate that the trajectory in fig. 3.4 is actually overlapping itself with a *return journey*. Figure 3.5 takes fig. 3.4 and extends it in time. In fig. 3.5 it can be seen that the dipole rotates, stops, and rotates in the opposite direction. This would explain the double potential energy minima in fig. 3.2. Dipole-1 passes through the energy minima position in a clockwise rotation, so it must pass back through this position on the anti-clockwise *return journey*. Conversely, this also explains the double kinetic energy peaks in fig. 3.3, since at a point of minimum potential energy maximum kinetic energy must also be achieved.

This can be understood when the energies involved in this motion are considered. Put simply, dipole-1 is stuck in a potential energy well. The potential well may be visualised by viewing the variation of potential energy with ϕ_1 in fig. 3.2. The initial orientation of dipole-1 *bestows* upon it a certain amount of potential energy, which it will use to move under the influence of the dipolar electric field. The dipole cannot have more or less energy than it initially starts with. As it rotates through this field, it will be stopped when it reaches a field potential equal to that of where it started. Using the potential well analogy, once it climbs to a point that is equal to where it started, it has climbed as high up the potential barrier as possible. The potential barrier arises due to the electric field emanating from each monopole. The potential barrier rises as like-charged monopoles

approach each other (c.f. monopole **10** approaching monopole **00** as it rotates). If the dipole had enough initial kinetic energy, it may be able to surmount this potential barrier (using up it's kinetic energy) and *fall down* the other side (hence building up kinetic energy again), but in this case it starts from rest, so the only way it could ever have the energy to surmount this barrier would be if the simulation started with the dipole at the top of the potential barrier. For a simulation starting from rest, the higher the dipole is initially placed on the potential barrier determines how far it may rotate. Conversely, if the dipole was placed at the bottom of the potential well to start with and had no kinetic energy, it would not rotate at all.

3.3. Review of discussions

This simulation satisfies logical expectations. It is modelled on well-known laws of physics, and does indeed obey the law of conservation of energy (which was not explicitly programmed into the simulation!). Given the mutual repulsion and attraction of monopoles, it is also logical to expect there to be potential (and kinetic) energy maxima and minima, which would form a potential well. Furthermore, lacking sufficient energy, the dipole would be confined to this potential well. The oscillatory motion of the dipole proves that it is indeed confined to a potential well.

A noteworthy point is that the oscillatory period of dipole-1 is approximately **2 Pico-seconds**. Some MD simulations utilize Pico-second to Femto-second time steps [21]. While Femto-second time steps may be suitable for this model (subject to testing), it is clear that such a large time step as a Pico-second is clearly not acceptable for this model given the period of oscillation of the dipole.

Chapter 4. Summary

4.1. Results

This simulation displayed several key features. One dipole was not permitted to move in any way. Therefore, the simulation can be regarded as the interaction of a dipole (which has a non-moving COM) with a fixed dipolar electric field. Because the dipole had a fixed COM, it became trapped in the potential well of the fixed field. This gave rise to an oscillatory rotational motion.

Given an initial potential energy of **0.77 eV** with **zero kinetic energy** the period of oscillation was approximately **2 Pico-seconds**.

It was also shown that when the dipoles became anti-parallel, the minimum potential energy was reached. From fig. 3.1 the minimum potential energy was approximately **-1.1 eV**.

Throughout the simulation it was demonstrated that there was an equal exchange of kinetic and potential energy, which lead to a constant total system energy. Conservation of energy was good; there was only a +0.0003% drift from the initial energy.

4.2. Limitations and solutions

The principal limitation in this simulation was the coordinate system used to calculate rotational motion. Spherical Polar coordinates were used to measure the orientation of the dipole in space. However, SP coordinates are insufficient to calculate non-planar rotations. The problem was avoided in this case by choosing an initial configuration such that planar rotation was the only possible motion. However for rotational motion about any Cartesian axis, Euler angles must be employed to evaluate such angular

displacements. The mathematics for this are more complex than for SP coordinates, but are obviously more useful.

The other limitation of the simulation was that even though there was a dipole that could rotate, it could only rotate, its COM could not move through space. To allow free translation of the COM, the mathematical analysis would have to be extended to consider the resultant translational forces on the COM. This would only mean a modest modification to the computer program. Once dipoles can be translated through space an extra factor has to be considered. If the dipoles can move towards and away from each other, they could collide or even be so close that their monopole bonds could *cross swords* as they rotate. They could also drift apart so far that their mutual fields have little effect on each other. So, it would have to be considered whether or not to impose a *simulation box* with *periodic boundary conditions* [23]. This would limit the volume of space in which the simulation took place. Periodic boundary conditions operate by taking a particle that was passing beyond the walls of the simulation box and replacing it on the opposite side of the box, as if particles were flowing through the simulation box. Also, use could be made of the Lennard-Jones potential, which imposes a potential on particles that creates an equilibrium separation between particles. This would perhaps remove the need for a simulation box for a small number of particles because the LJ potential would keep them all at a suitable separation.

This simulation has also shown that when considering rotational motion, the time step cannot be as large as for spherically symmetric particles if a continuous motion is to be achieved. Continuous motion is important for energy conservation.

4.3. Future Work

Future work on this model will consist of applications of the current model and extensions of the model (and applications there of!). The most important next step should be to replace the SP coordinate system with Euler angles so that non-planar rotations can be simulated. Investigations of the interactions between more than two dipoles could then be carried out. Also, tolerance to longer time steps could be experimented with. Diatomic lattice structures could be simulated using the fixed COM dipoles by having a large number of regularly arranged dipoles in a simulation. The monopole charges and masses could be altered to reflect the constituent atoms of the lattice.

An extension to the model could be made to allow the dipole completely free movement in space by allowing translation of the COM. Having achieved this; actual diatomic molecular dynamics can be performed, by setting the monopole charges and masses to appropriate values for particular molecules.

Once this can be done, it would be logical to try and incorporate thermodynamic quantities into the simulation to account for volume, pressure, temperature, etc.

Once the above extensions to the original model have been completed, an attempt should be made to model more complicated structures (than linear molecules). Then true molecular dynamics of polyatomic molecules can be performed. Since there is so much interest and difficulty in MD simulations of water, it would be of interest to try modelling the H₂O molecule.

Chapter 5. References

- [1] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E. *Equation of state calculations by fast computing machines*, J. Chem. Phys. **21**, 1087-1092. 1953.
- [2] Alder B J and Wainwright T E. *Phase transition for a hard sphere system*, J. Chem. Phys. **27**, 1208-1209. 1957.
- [3] Wood W W and Parker F R. *Monte Carlo equation of state of molecules interacting with the Lennard-Jones potential. I. A super critical isotherm at about twice the critical temperature*, J. Chem. Phys. **27**, 720-732. 1959.
- [4] Alder B J and Wainwright T E. *Phase transition for hard sphere system*, J. Chem. Phys. **27**, 1208-1209.
- [5] Alder B J and Wainwright T E. *Studies of molecular dynamics. I. General Method*, J. Chem. Phys. **31**, 459-466. 1959.
- [6] Rahman, A H. *Correlations in the motion of atoms in liquid Argon*, Phys. Rev. **136A**, 405-411. 1964
- [7] Verlet. *Computer 'Experiments' on classical Fluids. I. Thermodynamic properties of Lennard-Jones molecules*, Phys. Rev. **159**. 98-103, 1967
- [8] Harp G D and Berne B J. *Linear and angular momentum autocorrelation functions in diatomic liquids*, J. Chem. Phys. **49**, 1249-1254. 1968.
- [9] Barker and Watts. *Structure of water: a Monte Carlo calculation*, Chem. Phys. Lett. **3**, 144-145. 1969.
- [10] Marion Jerry Baskerville, *Classical dynamics of particles and systems*, 2nd edition, N.Y.: Academic Press, 1970.
- [11] Berne B J and Harp G D. *On the calculation of time correlation functions*, Adv. Chem. Phys. **17**, 63-227. 1970.

- [12] Hockney R W. *The Potential Calculation and some applications. Methods comput, Phys.* **9**, 136-211. 1970
- [13] Gear C W. *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, Englewood Cliffs, NJ. 1971
- [14] Rahman A H and Stillinger F H. *Molecular Dynamics Study of Liquid Water*, J Chem. Phys **7**, v55, 1971 (may 6th).
- [15] Potter. *Computational Physics*, Wiley New York. 1972.
- [16] Barojas J, Levesque D and Quenterz B. *Simulation of diatomic homonuclear liquids*, Phys. Rev. **A7**, 1092-1105. 1973.
- [17] Dahlquist G and Björk A. *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ. 1974
- [18] Evans D J and Murad. *On the representation of orientational space*, Mol. Physics, **34**, 317-325. 1977.
- [19] Fincham D and Heyes D M. *Integration algorithms in molecular dynamics*, CCP5 Quarterly **6**, 4-10. 1982.
- [20] Heyes D M and Singer K. *A very accurate molecular dynamics algorithm*, CCP5 Quarterly **6**, 11-23. 1982.
- [21] Allen M P and Tildesly D J. *Computer Simulations of Liquids*, Oxford University Press. 1987. ISBN 9-780198-556459
- [22] Ritchie, Dennis M., Kernighan, Brian W., *The C Programming Language*, 2nd edition, Prentice Hall. 1988. ISBN 0131103739, ISBN (PBK) 0131103628
- [23] Leach, Andrew R., *Molecular modelling: principles and applications*, Harlow: Addison Wesley, 1996. ISBN/ISSN 0582239338.
- [24] Williams James H, *Fundamentals of applied dynamics*, New York : John Wiley, 1996. ISBN 0471109371.

- [25] Jeffrey, George A., *An introduction to hydrogen bonding*, New York: Oxford University Press, 1997. ISBN/ISSN 0195095499/0195095480.
- [26] Duffin, W.J. (William John), *Electricity and Magnetism – 4th ed*, McGraw Hill 1998. ISBN 007707209-X
- [27] Steven W. Rick and B. J. Berne, *The Aqueous Solvation of Water: A Comparison of Continuum Methods with Molecular Dynamics*, Department of Chemistry and Centre for Biomolecular Simulation
- [28] *Electrostatic Potentials of free energies of solvation of polar and charged molecules*, J. Phys Chem. **B**, v101 3017
- [29] *Constructing ab-initio molecular dynamics simulations*, J Chem. Phys, v108, 4739