
Chapter 2

Methodology

This chapter presents an extensive survey of methodologies used to investigate and analyze dusty plasma systems. These approaches include both theoretical and simulation methods, offering a comprehensive understanding of plasmas and their properties. In our study, we utilized fluid theory as an analytical tool to derive expressions for the interaction potential in complex plasmas. Additionally, we have provided a comprehensive report on the simulation technique used for studying dusty plasmas, specifically focusing on molecular dynamics (MD) simulations. The detailed explanation of the principles and procedures involved in conducting MD simulations allows us to effectively simulate the behavior and dynamics of dusty plasmas at the individual particle level. Furthermore, this chapter explores the theoretical background of the Green-Kubo formalism, which serves as the basis for calculating transport coefficients in molecular systems. The chapter offers a comprehensive exploration of the fundamental principles and concepts underlying this formalism, including the core ideas of the fluctuation-dissipation theorem and its connection to equilibrium fluctuations and transport properties.

2.1 Fluid equations:

In our study, we have used fluid equations as an analytical tool to derive expressions for the interaction potential in complex plasmas. Understanding and analyzing these interaction mechanisms is important for comprehending the dynamics and properties of complex plasmas.

Fluid mechanics plays a pivotal role in plasma physics and is useful for investigating both static and dynamic properties of plasma systems. The fluid theory provides a macroscopic approach to studying plasmas by considering them as a continuous medium rather than a discrete collection of particles [11]. In fluid theory, plasma is divided into small fluid elements, each containing a large number of particles. These fluid elements are characterized by averaged properties such as density, pressure, and velocity [11]. The theory then uses a set of equations, known as fluid equations, to describe how these properties change over time and space. These equations are derived from the basic laws of physics, such as the laws of conservation of mass, momentum, and energy. In the context of plasma physics, these fluid equations may include additional terms to account for electromagnetic forces, as governed by Maxwell's equations. The fundamental equations of fluid theory can be described by the following equations:

$$m_s n_s \left[\frac{\partial v_s}{\partial t} + (v_s \cdot \nabla) v_s \right] = Q_s n_s [E + v_s \times B] - \nabla P + F_{other} \quad (2.1)$$

This is the momentum equation. Where, m_s : mass of a species s , n_s : number density of a species s , v_s : velocity of a species s , Q_s : charge of a species s , E : Electric field, B : Magnetic field and P : pressure. Another important equation that ensures the conservation of mass of the respective fluids is the equation of continuity.

$$\frac{\partial n_s}{\partial t} + \nabla \cdot (n_s \mathbf{v}_s) = 0 \quad (2.2)$$

Here, n_s : Density of a specific species 's' in the plasma. \mathbf{v}_s : Velocity vector of species 's'. The third equation is Poisson's equation given by

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0} \quad (2.3)$$

By solving this differential equation, we can determine the electric potential in a plasma, which, in turn, provides insights into the electric field and its impact on charged particle motion. The charge density (ρ) arises from the presence of ions, dust, and electrons in the plasma, and its non-uniform distribution leads to variations in the electric potential throughout the plasma. To solve Poisson's equation, various numerical and analytical methods are used, including techniques like Fourier analysis [12]. The solution is often coupled with other plasma equations, such as the continuity equation and momentum equation, to comprehensively study plasma dynamics.

2.2 Molecular Dynamics (MD) Simulation:

In many-body systems, where the interactions between individual particles or components are intricate and interdependent, finding analytical solutions to describe the system's behavior becomes extremely complex. As the number of particles increases, the mathematical equations governing the system become highly non-linear and difficult to solve explicitly. This is where numerical solutions, such as Molecular Dynamics (MD) simulation, and other numerical solutions play an important role. By employing numerical methods, we can approximate the behavior of the system over time by discretizing the equations and solving them iteratively. MD simulation allows to simulate the interactions between each particle based on known physical principles, yielding valuable insights into the system's collective behavior and dynamics. Thus, MD simulation becomes a powerful and indispensable tool in handling the complexities of many-body systems, offering a practical and computationally feasible approach to understanding their intricate behavior.

2.2.1 Background:

Molecular Dynamics (MD) simulation is a powerful computational technique used to study the behavior of atomic and molecular systems over time [123]. It allows us to investigate the dynamic evolution of a system by numerically solving the equations of motion for each individual particle, taking into account their interactions and forces. The objective of this chapter is to provide a comprehen-

sive overview of MD simulation, covering its principles and applications. We will explore the fundamental concepts underlying MD simulation, including Newton's equations of motion, force fields, integration algorithms, and ensemble methods. We will also discuss various aspects of setting up and initializing simulations, such as system definition, and equilibration techniques. Furthermore, we will explore different simulation protocols and techniques, including energy minimization, production runs, and the use of thermostats and barostats to control temperature and pressure. We will also discuss handling boundaries, long-range interactions, and constraints within MD simulations. An essential aspect of MD simulation is the analysis and post-processing of simulation data. We will explore various analysis techniques to extract meaningful information from simulation trajectories, including radial distribution functions, pair correlation functions, mean squared displacement, diffusion coefficients, and structural properties. Additionally, we will discuss visualization techniques to gain insights into the dynamics and behavior of the simulated system.

2.2.2 Theoretical foundations of Molecular Dynamics Simulation:

Molecular Dynamics (MD) simulation is based on the fundamental principles of classical mechanics and statistical mechanics [124]. The technique involves numerically solving the equations of motion for a system of interacting particles. Understanding the principles of MD simulation is essential for accurately capturing the dynamics and properties of the simulated system. In this section, we will discuss the key principles of MD simulation.

Particle Interactions and Force Fields

In molecular dynamics (MD) simulations, the precise representation of particle interactions is vital for obtaining reliable and meaningful results [123]. Particle interactions can be categorized into two types: bonded and non-bonded interactions. These interactions govern the behavior and properties of particles within a system. Bonded interactions include covalent bonds, angles, and dihedrals (also known as torsional or rotational bonds) [123]. Covalent bonds represent the shar-

ing of electrons between atoms and determine the connectivity of atoms within a molecule. Angles refer to the geometric arrangement of three consecutive atoms, while dihedrals involve the torsional rotation of a group of atoms around a bond. Non-bonded interactions, on the other hand, are relatively weaker and occur between atoms or molecules that are not directly bonded [123]. They can be further divided into two categories: van der Waals interactions and electrostatic interactions. (a) Van der Waals interactions arise from fluctuating dipoles in atoms or molecules [124]. These interactions play a significant role in determining the stability, structure, and physical properties of molecular systems. (b) Electrostatic interactions result from the presence of charged particles, such as electrons and ions, within a molecular system. These interactions can be either attractive (between oppositely charged particles) or repulsive (between particles of the same charge).

To model these molecular interactions in MD simulations, force fields are employed. A force field is a mathematical model that describes the potential energy and forces acting between the particles in the system [125]. It consists of a set of parameters, including equilibrium bond lengths, bond angles, dihedral angles, and atom types, along with corresponding force constants and interaction parameters [124]. Several widely used force fields in molecular dynamics simulations include CHARMM (Chemistry at Harvard Macromolecular Mechanics) [126], AMBER (Assisted Model Building with Energy Refinement) [127], and OPLS (Optimized Potentials for Liquid Simulations) [128]. These force fields are designed to capture the behavior of specific types of molecules and have been extensively parameterized and validated against experimental data.

The selection of an appropriate force field depends on the system under investigation and the level of accuracy required. Careful parameterization of force fields is crucial to accurately represent molecular interactions. The accuracy of the force field directly impacts the simulation results and their agreement with experimental observations. In summary, understanding and properly representing molecular interactions through force fields are essential for conducting accurate molecular dynamics simulations. Bonded and nonbonded interactions, along with their corresponding force field parameters, play a vital role in determining the be-

havior and properties of molecules within the simulated system. The selection and parameterization of appropriate force fields are essential steps to ensure reliable simulation results.

Integration Algorithms

In molecular dynamics (MD) simulations, integration algorithms are employed to numerically solve the equations of motion for the particles in a system [125]. The basic algorithm of MD simulation involves solving Newton's equations of motion to calculate the positions of particles at different time steps.

$$m_i \ddot{r}_i(t) = - \sum_{i \neq j} \nabla \phi + F_{ext} \quad (2.4)$$

Here, $r_i(t)$ is the position of the particle at time t , ϕ is the interaction potential, and F_{ext} is the external force. An integration algorithm governs the evolution of particle positions and velocities over time, enabling the simulation to accurately capture the dynamic behavior of the system. The Verlet algorithm, leapfrog algorithm, and velocity-Verlet algorithm are commonly used integration schemes in molecular dynamics simulations [129; 130; 131; 132; 133]. They are numerical methods that calculate the positions and velocities of particles at each time step based on the forces acting on them [129; 130].

In the context of the current thesis, our Molecular Dynamics (MD) code implements the Velocity Verlet algorithm. The velocity Verlet algorithm is an excellent integrator that approximates the trajectory of particles based on their current positions, velocities, and accelerations [125]. It is known for its simplicity, numerical stability, and conservation of energy properties. The algorithm follows a step-by-step procedure to update the positions and velocities of particles at each time step. The main steps of the Velocity Verlet algorithm are as follows:

1. **Initialization:** In this step, the initial positions and velocities of particles are assigned. This involves specifying the coordinates and velocities of each particle in the system at the beginning of the simulation. The positions can be defined based on a given initial configuration or randomly generated within the simulation box. Similarly, the velocities can be set to specific

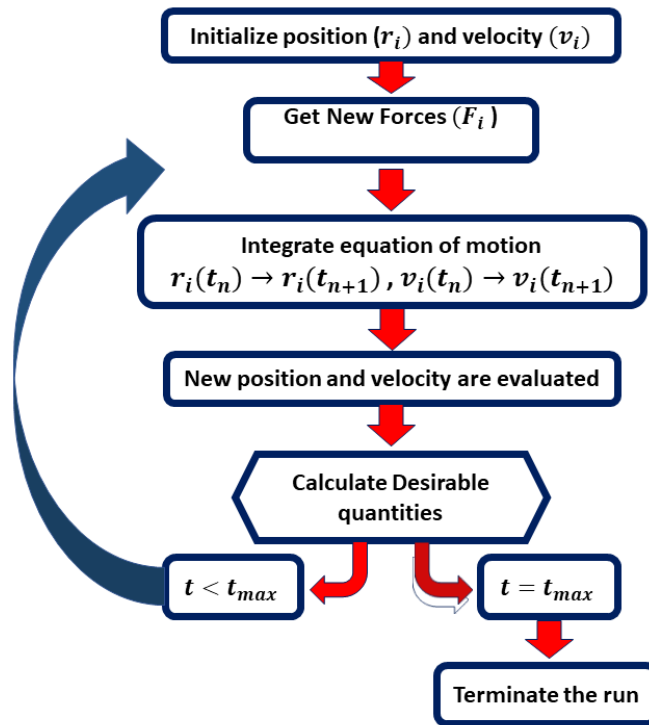


Figure 2.1: The basic algorithm of MD simulation.

values or assigned randomly according to a desired temperature or energy distribution. Proper initialization is crucial to ensure a realistic starting state for the simulation and to accurately capture the subsequent dynamics of the system.

2. **Calculation of Forces:** In this step, the forces acting on each particle are determined by computing the gradients of inter-particle potentials and force field parameters [125]. The forces are computed based on the spatial arrangement of neighboring particles and their associated potential energy functions. Efficient algorithms and techniques, such as neighbor lists and cutoff distances, are utilized to optimize force calculations [125]. Accurate force calculations are vital for precisely simulating the system's dynamics and capturing the interactions between particles.
3. **Prediction of Half-Step Velocities:** In this step, the velocities of particles are updated halfway through the time step using their current velocities and accelerations [125]. By updating the velocities at this intermediate point,

the simulation can better capture the motion of the particles and maintain numerical stability and energy conservation. This prediction of half-step velocities is an integral part of many integration algorithms and plays an important role in accurately simulating the dynamics of the system.

4. **Prediction of Positions:** In this step, the positions of particles are updated using the predicted velocities obtained from the previous step. This step is for maintaining the consistency between particle positions and velocities throughout the simulation and plays a fundamental role in accurately simulating the dynamics of the system. Careful consideration of the time step size is necessary to ensure numerical stability and preserve the simulation's accuracy. The position update equation used in the Velocity Verlet algorithm is as follows:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{1}{2}\mathbf{a}_i(t)\Delta t^2 \quad (2.5)$$

- $\mathbf{r}_i(t)$: Position vector of the i th particle at time t .
 - $\mathbf{v}_i(t)$: Velocity vector of the i th particle at time t .
 - $\mathbf{a}_i(t)$: Acceleration vector of the i th particle at time t .
 - Δt : Time step or time increment.
 - $\mathbf{r}_i(t + \Delta t)$: Updated position vector of the i th particle at time $t + \Delta t$.
5. **Calculation of New Forces:** In this step, the forces acting on particles are recalculated based on their updated positions. It step involves reassessing the interactions between particles, considering their new spatial arrangement and distances. By re-evaluating the inter-particle potentials and force field parameters, the forces between particles are determined. This recalculation ensures that the forces accurately represent the current state of the system and account for any changes in particle positions. Efficient algorithms and techniques, such as pairwise summation methods and cutoff distances, are employed to optimize the force calculations and minimize computational costs. The accuracy and reliability of the simulation results heavily depend on the accuracy of the force calculations, making it essential to choose ap-

appropriate force field parameters and employ validated methods. By recalculating the forces, the simulation accurately captures the evolving interactions and dynamics of the system, enabling a realistic representation of its behavior over time.

6. Prediction of Full-Step Velocities: In this step, the velocities of particles are updated by taking the average of the predicted velocities from the half-step and the newly calculated forces [125]. This process combines the information from the previous half-step velocities with the current forces to obtain more accurate velocity predictions for the full-time step. By incorporating both the particle velocities and the forces acting on them, this step enhances the accuracy and stability of the simulation. The predicted full-step velocities are used in subsequent steps to advance the simulation and accurately model the system's dynamic behavior. The choice of integration algorithm and the specific method for updating velocities may vary depending on the system being simulated and the desired level of accuracy, with the goal of achieving reliable and meaningful simulation results. The velocity update equation used in the Velocity Verlet algorithm is as follows:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{2} [\mathbf{r}_i''(t) + \mathbf{r}_i''(t + \Delta t)] \quad (2.6)$$

- $\mathbf{v}_i(t)$: Velocity vector of the i th particle at time t .
- Δt : Time step or time increment.
- $\mathbf{r}_i''(t)$: Second derivative of the position vector $\mathbf{r}_i(t)$ with respect to time at time t .
- $\mathbf{r}_i''(t + \Delta t)$: Second derivative of the position vector $\mathbf{r}_i(t)$ with respect to time at time $t + \Delta t$.
- $\mathbf{v}_i(t + \Delta t)$: Updated velocity vector of the i th particle at time $t + \Delta t$.

The Velocity Verlet algorithm ensures energy conservation by updating both positions and velocities in separate steps. This separation avoids issues related to numerical errors and provides better long-term stability compared to algorithms

that update positions and velocities simultaneously [125]. The algorithm's accuracy is second-order, meaning that the error in the predicted positions and velocities is proportional to the square of the time step [123]. Thus, reducing the time step leads to increased accuracy but also increases computational cost. The Velocity Verlet algorithm is widely used in molecular dynamics simulations due to its simplicity, numerical stability, and conservation properties [134; 135]. It accurately captures the dynamics of molecular systems while efficiently preserving important physical quantities, such as total energy, during the simulation. Its effectiveness has been demonstrated in various applications, ranging from studying small molecules to simulating complex bio-molecular systems and materials [134; 135].

Periodic Boundary Conditions

In molecular dynamics (MD) simulations, periodic boundary conditions (PBC) are applied to mimic an infinite system by creating replicas of the simulation box [124]. PBC allows for the simulation of small systems while incorporating the effects of long-range interactions, ensuring that particles interact as if they were in an extended system. Under PBC, when a particle moves beyond one face of the simulation box, it reappears on the opposite face, creating a continuous and repeating lattice structure as shown in Fig.2.2 [124]. This approach effectively eliminates surface effects and ensures that particles interact with their periodic images. To accurately account for long-range electrostatic interactions, Ewald summation is commonly employed in MD simulations [124].

Ensemble Methods

Ensemble methods play a crucial role in molecular dynamics (MD) simulations by defining the type of thermodynamic ensemble in which the system is simulated [123]. An ensemble represents a collection of possible states that a system can occupy, characterized by specific macroscopic properties such as temperature, pressure, and particle number. The choice of ensemble method in MD simulations depends on the specific thermodynamic conditions under investigation and the properties of interest. The three most commonly used ensembles in MD simula-

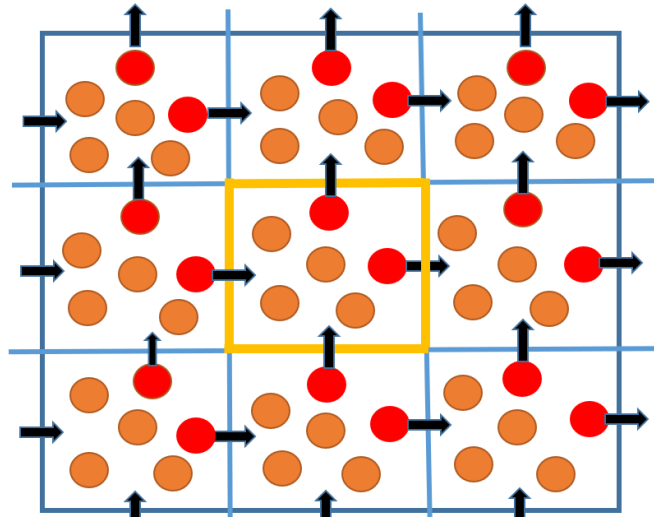


Figure 2.2: Visualization of Periodic Boundary Conditions (PBC) in Molecular Dynamics (MD) Simulation.

tions are the canonical ensemble (NVT), the isothermal-isobaric ensemble (NPT), and the microcanonical ensemble (NVE).

Canonical Ensemble (NVT):

The canonical ensemble maintains a constant number of particles (N), volume (V), and temperature (T). In NVT simulations, the system is typically coupled to a heat bath, which allows for temperature control and ensures that the system exchanges energy with the surroundings [123; 136]. Thermostats are used to maintain a constant temperature by re-scaling the velocities of particles or by modifying the forces acting on them [123].

Isothermal-Isobaric Ensemble (NPT):

The isothermal-isobaric ensemble maintains a constant number of particles (N), pressure (P), and temperature (T). In NPT simulations, the system is coupled to a heat bath and a pressure bath, allowing for both temperature and pressure control [123; 137]. Barostats are used to maintain a constant pressure by adjusting the volume of the simulation cell, ensuring that the system can expand or contract to accommodate changes in pressure [123].

Micro-canonical Ensemble (NVE):

The micro-canonical ensemble maintains a constant number of particles (N), volume (V), and total energy (E). In NVE simulations, the system is isolated from external heat baths or pressure baths, and the total energy of the system is conserved throughout the simulation [124]. NVE simulations are often used for studying energy conservation, long-time dynamics, and equilibrium properties.

The choice of ensemble depends on the specific properties of interest and the physical conditions being simulated. For instance, the canonical ensemble is often employed for studying equilibrium properties at a constant temperature, such as thermodynamics, phase transitions, and structural properties [124]. The isothermal-isobaric ensemble is useful for simulating systems at a constant temperature and pressure, which is relevant for studying processes occurring under specific environmental conditions or in the presence of solvent molecules [125]. The micro-canonical ensemble is employed when investigating energy conservation, long-term stability, and non-equilibrium processes [124; 125; 135].

Thermostat and Barostat Techniques

Thermostat and barostat techniques are essential components of molecular dynamics (MD) simulations that enable the control of temperature and pressure, respectively [124]. These techniques ensure that the simulated system accurately represents the desired thermodynamic conditions and allow for the study of equilibrium properties and dynamic behavior under specific environmental conditions.

(i) Thermostat Techniques:

Thermostat techniques are used to regulate the temperature of the system during MD simulations [138; 139; 140]. One commonly used thermostat is the Berendsen thermostat, which works by scaling the velocities of particles to achieve the desired temperature. The Berendsen thermostat operates by coupling the system to an external heat bath, allowing energy exchange between the system and the bath. Another commonly used thermostat is the Nosé-Hoover thermostat, which uses a set of additional degrees of freedom to control the temperature by dynamically

adjusting particle velocities. The Nosé-Hoover thermostat achieves temperature control by introducing a heat reservoir that interacts with the system, ensuring that the system reaches and maintains the desired temperature. Other thermostat techniques, such as the Andersen thermostat and the Langevin thermostat, are also available and provide different mechanisms for temperature control. The choice of thermostat technique depends on factors such as the desired accuracy, the nature of the system being simulated, and the computational resources available.

(ii) Barostat Techniques:

Barostat techniques are used to regulate the pressure of the system during MD simulations [139; 140; 141]. One commonly used barostat is the Berendsen barostat, which scales the simulation cell size to achieve the desired pressure. The Berendsen barostat weakly couples the system to an external pressure bath, allowing for the expansion or contraction of the simulation cell based on the pressure. The Parrinello-Rahman barostat is another widely used technique that employs a fictitious mass-spring system to control the pressure by dynamically adjusting the simulation cell dimensions. The Parrinello-Rahman barostat achieves pressure control by allowing the simulation cell to fluctuate while maintaining the desired average pressure. Like thermostat techniques, there are other barostat techniques available, such as the Martyna-Tobias-Klein barostat and the Monte Carlo barostat, which offer different approaches to pressure control.

The selection of thermostat and barostat techniques is important for accurately reproducing the desired temperature and pressure conditions in MD simulations. It is important to ensure that the chosen techniques are appropriate for the system under investigation and the specific research objectives.

Simulation Length and Time Step Considerations

In molecular dynamics (MD) simulations, the choice of simulation length and time step is critical for obtaining meaningful and reliable results. The simulation length determines the duration of the simulated dynamics, while the time step defines the increment at which the equations of motion are numerically integrated.

1. Simulation Length:

The simulation length refers to the total time span of the MD simulation [123; 125]. The choice of simulation length depends on the specific objectives of the study. Shorter simulations are suitable for exploring fast processes or obtaining preliminary insights, while longer simulations are necessary for studying slower dynamics or achieving converged statistical averages. The simulation length should be sufficient to allow the system to equilibrate and reach a steady state, ensuring that the observed properties are representative of the desired thermodynamic conditions. For systems undergoing transitions, longer simulation lengths may be required to capture the relevant dynamics adequately. It is important to note that longer simulation lengths come with an increased computational cost, and a balance between the desired level of accuracy and available computational resources should be considered.

2. Time Step: The time step determines the interval at which the equations of motion are numerically integrated [123; 125]. The selection of the precise time step is essential for obtaining accurate simulation results. A time step that is too large can lead to inaccuracies in the integration and cause the simulation to violate energy conservation and produce unphysical behavior. On the other hand, an excessively small time step can unnecessarily increase computational cost without significant improvement in accuracy. The appropriate time step depends on the dynamics of the system and the frequency of the fastest motions. As a general guideline, the time step should be small enough to capture the fastest motions in the system while maintaining numerical stability [123; 124].

In summary, the choice of simulation length and time step in MD simulations plays a critical role in obtaining accurate and reliable results. The simulation length should be appropriate to allow for equilibration and convergence of statistical properties. The time step should be selected carefully to balance accuracy and computational efficiency, considering the system dynamics and the fastest motions of particles. Validating the chosen simulation length and time step through energy conservation and other physical properties is essential for ensuring the integrity of the simulation results.

2.2.3 Data Analysis:

Data analysis in MD simulations involves extracting meaningful information from the simulation data to understand the structural, dynamical, thermodynamic, and specific properties relevant to the system under investigation. This analysis provides valuable insights into the behavior and properties of the simulated system and helps in comparing the simulation results with experimental observations or theoretical predictions. In our Molecular Dynamics (MD) simulation, the following quantities are being used to analyze and characterize the behavior of particles:

1. Radial Distribution Function (RDF): The Radial Distribution Function, denoted as $g(r)$, provides information about the average density variation of particles as a function of distance [142; 143]. It quantifies the structural correlations between particles and indicates whether they exhibit solid or fluid-like arrangements. The quantity $\rho g(r)$, where ρ represents the number density of particles, is proportional to the probability of finding a particle at a distance ranging from r to $r + dr$ from a reference particle. In MD simulation, the RDF is computed by placing particles into histogram bins, which can be visualized as concentric shells around a reference particle. Each bin has a width of dr and covers a specific range of distances up to a finite distance r_d in space. The resolution of the RDF is determined by the number of bins used in the calculation. By analyzing the RDF, we can find the spatial arrangement and organization of particles within the simulated system. It helps in characterizing the nature of interactions and the presence of ordering or clustering phenomena.

In a three-dimensional (3D) homogeneous and isotropic system, the pair correlation function, denoted as $g(r)$, is computed by averaging over a spherical region surrounding a central particle. The mathematical expression for $g(r)$ is defined as follows:

$$g(r) = \frac{V N(r, dr)}{N 4\pi r^2 dr} \quad (2.7)$$

Here, V represents the volume of the simulation box, and $N(r, dr)$ denotes the number of particles within a shell of infinitesimal thickness dr at a distance r from

the central particle. N represents the total number of particles in the system. For a two-dimensional (2D) system, the pair correlation function expression is modified to account for the area A of the simulation box. The modified expression is given by:

$$g(r) = \frac{A}{N} \frac{N(r, dr)}{2\pi r dr} \quad (2.8)$$

In this equation, $N(r, dr)$ represents the number of particles within a circular shell of infinitesimal thickness dr at a distance r from the central particle. These pair correlation functions provide insights into the spatial arrangement and density fluctuations of particles in the system. They describe the probability of finding a particle at a specific distance from a reference particle and are useful for understanding the structural properties of the system under investigation.

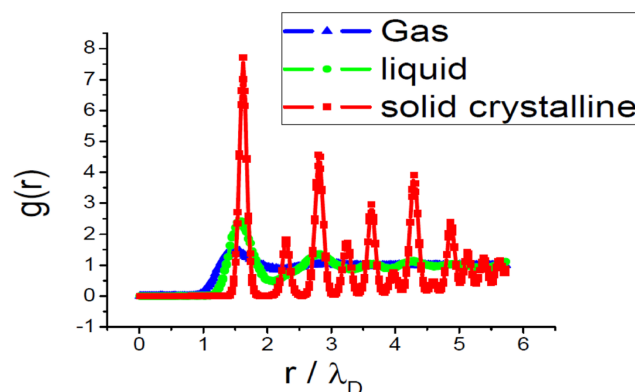


Figure 2.3: A typical plot of the Radial Distribution Function (RDF) depicting the differences between solid, liquid, and gaseous phases.

2. Lattice Correlation Factor (LCF): The Lattice Correlation Factor (LCF) is an important diagnostic tool used to assess the presence of long-range order in a system [125]. It provides a measure of the degree of order and can be computed based on the positions of particles in the system. The LCF can also be experimentally determined using techniques such as X-ray scattering. The LCF serves as an order parameter and is obtained by evaluating the local density at a finite distance from a reference particle. The local density, denoted as $\rho(r)$, is calculated by summing the Dirac delta functions over all particles in the system:

$$\rho(r) = \sum_{j=1}^N \delta(r - r_j) \quad (2.9)$$

To analyze the LCF in terms of reciprocal space, the expression is Fourier-transformed, resulting in:

$$\rho(k) = \frac{1}{N} \sum_{j=1}^N \exp(-ik \cdot r_j) \quad (2.10)$$

Here, k represents the reciprocal lattice vector, which is determined by the type of lattice being considered. For a perfectly crystalline state, $|\rho(k)|$ approaches unity. As the system transitions from an ordered to a disordered state, the value of LCF decreases from unity. In gaseous states, The typical magnitude of LCF is < 0.1 , indicating increased disorder in the system. The LCF provides valuable information about the transition from ordered to disordered states in the system. By analyzing the magnitude of $\rho(k)$, we can assess the level of long-range order and structural coherence present in the system.

3. Mean Square Displacement (MSD) and Diffusion coefficient: Mean Square Displacement (MSD) is a commonly used quantity in Molecular Dynamics (MD) simulations to characterize the diffusive behavior of particles [144; 145]. It measures the average displacement of particles over time. The MSD is calculated by computing the squared Euclidean distance between the initial position of a particle and its position at a later time and then averaging this value over all particles in the system. Mathematically, it is expressed as:

$$\text{MSD}(t) = \left\langle \frac{1}{N} \sum_{i=1}^N |\mathbf{r}_i(t) - \mathbf{r}_i(0)|^2 \right\rangle \quad (2.11)$$

Where $\text{MSD}(t)$ represents the MSD at time t , N is the total number of particles, $\mathbf{r}_i(t)$ is the position vector of the i th particle at time t , and $\mathbf{r}_i(0)$ is its initial position. By analyzing the time-dependent behavior of the MSD, it is possible to gain insights into the diffusion process in the system. In particular, the slope of the MSD plot at long timescales can provide information about the diffusivity

of particles and their transport properties. This can be used to characterize the nature of diffusion, such as Brownian motion or other types of transport mechanisms. In general, $MSD(t) \propto t^\alpha$ where $\alpha = 1$ for normal diffusion, $\alpha < 1$ is for sub-diffusion and $\alpha > 1$ is for super-diffusion.

The MSD is a versatile tool that can be applied to study diffusive behavior in a wide range of systems, including liquids, gases, solids, and complex materials. It is often used in combination with other analysis techniques to explore the dynamics and properties of particles in MD simulations. The diffusion coefficient can be estimated from the Mean Square Displacement (MSD) in MD simulations using the Einstein relation, which relates the MSD to the diffusion coefficient. The equation for this relationship is:

$$D = \frac{1}{2dt} \text{MSD} \quad (2.12)$$

where $MSD(t)$ is the MSD at time t and D is the diffusion coefficient. By plotting the MSD as a function of time and fitting the resulting curve to a linear function, one can extract the slope, which is equal to $6D$. Dividing the slope by 6 yields the diffusion coefficient D . Additionally, it is essential to consider the appropriate time range for fitting the MSD data. Typically, the linear regime of the MSD plot at longer timescales, where the displacement is proportional to time, is used for estimating the diffusion coefficient.

By estimating the diffusion coefficient, researchers can gain insights into the diffusive behavior and mobility of particles in the simulated system. In the specific case of dusty plasma, it is possible to relate the diffusion coefficient, which represents a macroscopic property, with the specific properties of dust particles through Einstein's theory of Brownian motion. According to Einstein's theory, the diffusion coefficient (D) can be expressed as:

$$D = \frac{K_B T_d}{6\pi\eta r_d} \quad (2.13)$$

Here, T_d represents the dust temperature and r_d represents the dust radius. The equation shows that the diffusion coefficient is inversely proportional to the dust

radius (r_d) and directly proportional to the dust temperature (T_d). The Boltzmann constant (K_B) and the dynamic viscosity of the medium (η) are constants that are also involved in the relationship. This relationship allows us to connect the macroscopic behavior of diffusion with the specific properties of individual dust particles in the system. It provides a valuable link between the microscopic and macroscopic aspects of Brownian motion in the context of complex plasma dynamics and can aid in understanding and predicting diffusion processes in dusty plasmas.

4. The Lindemann parameter (L)

The Lindemann parameter (L) is an important diagnostic for measuring the structural deformations of a lattice formed by strongly correlated particles [146; 147]. It quantifies the degree of particle displacement from their equilibrium positions in the lattice and is computed as the ratio of the root mean square displacement (MSD) of the particles to the average inter-particle distance (r_{av}). Mathematically, the Lindemann parameter is given by:

$$L = \frac{\sqrt{\text{MSD}}}{r_{av}} \tag{2.14}$$

Here, MSD represents the root mean square displacement of the particles and r_{av} represents the average inter-particle distance. The Lindemann parameter provides valuable insights into the structural changes and disorder in the lattice due to particle movements. It serves as a measure of lattice stability and can be used to characterize phase transitions and melting phenomena in various systems.

2.3 Brownian Dynamics Simulation:

2.3.1 Introduction to Brownian Dynamics Simulations (BD):

Brownian Dynamics simulation is a powerful computational tool used to model the behavior of particles in a fluid as they undergo random, fluctuating motion - a phenomenon known as Brownian motion [148]. This motion is named after the

botanist Robert Brown, who first observed it in pollen grains suspended in water in 1827 [149]. Brownian Dynamics simulations play a pivotal role in a wide array of disciplines, most notably within biological and physical systems. By enabling us to predict the trajectory of microscopic particles in a fluid, Brownian Dynamics simulations provide key insights into processes such as protein folding, molecular binding, diffusion of particles, and the dynamics of colloidal suspensions [148; 150; 149]. The power of Brownian Dynamics lies in its ability to capture these important physical and biological processes, often at scales that are challenging for direct experimental observations.

2.3.2 The Langevin equation:

The Langevin equation plays a central role in the simulation of Brownian Dynamics (BD). Named after French physicist Paul Langevin, the Langevin equation is a stochastic differential equation that describes the time evolution of a physical system subject to both deterministic and random forces [151; 152]. In the context of BD, it describes the motion of a microscopic particle suspended in a fluid medium.

$$\frac{dv(t)}{dt} = \frac{F_{int}}{m} - \gamma v(t) + \xi(t) \quad (2.15)$$

- The force F_{int} accounts for the interaction potential between particles.
- $-\gamma v(t)$ is the damping force proportional to the velocity of the particle that opposes its motion.
- $\xi(t)$ is a Gaussian random variable representing random thermal forces.

In the specific context of dusty plasmas, the frictional force, represented as $-\gamma v(t)$ in the Langevin equation, arises due to the motion of dust particles through the surrounding buffer plasma. The friction coefficient, γ , is associated with the collision frequency between the dust particles and the plasma particles. These collisions transfer momentum from the dust particles to the plasma, thereby slowing down the dust particles. This interaction results in a force that opposes the motion of the dust particles, acting as a damping force. Therefore, it is an important effect to account for in simulations of dusty plasmas, as it can significantly in-

fluence the dynamics of dust particles. In the realm of dusty plasmas, $\xi(t)$ is accountable for thermal fluctuations. These fluctuations are random movements and collisions of particles incited by the system's temperature. This randomness is typically modeled by presuming that $\xi(t)$ is Gaussian white noise with a zero mean and a variance decided by the temperature of the system and the friction coefficient γ [125].

The Langevin equation can be extended to incorporate additional forces acting on the particles in certain scenarios. For instance, we might consider the presence of magnetic fields and the influence of other forces as given by the following equation.

$$\frac{dv(t)}{dt} = \frac{F_{int}}{m} - \gamma v(t) + \xi(t) + \frac{Q_d}{m_d}(v_d \times B) + F_{other} \quad (2.16)$$

2.4 Green-Kubo Formalism: Underlying Principles and Applications in Molecular Systems:

The Green-Kubo formalism is a pivotal tool in computing transport coefficients and exploring the complex dynamics of molecular systems. This section explores the principles of this formalism, including the fluctuation-dissipation theorem, and how it ties together the concepts of equilibrium fluctuations and transport properties. We highlight the versatile applicability of the Green-Kubo formalism across diverse molecular systems.

The Green-Kubo formalism is a vital tool in statistical physics, playing a pivotal role in the exploration of the dynamics of many-body systems, particularly in the realm of transport properties. It is named after the physicists Ryogo Kubo and Melville S. Green, who independently developed this approach. At its core, the Green-Kubo formalism is a theoretical framework that uses time correlation functions to calculate macroscopic transport properties from microscopic dynamics [153; 154; 155]. These transport properties include but are not limited to, thermal conductivity, electrical conductivity, and viscosity. The relevance of the Green-Kubo formalism is its ability to bridge the microscopic and macroscopic worlds. On one side, it allows us to investigate the microscopic dynamics of

individual particles within a system. On the other side, it allows us to derive the macroscopic, bulk properties of the system that we can measure in an experiment. A key reason for the prominence of the Green-Kubo formalism in statistical physics is its universality. Regardless of the details of the system under investigation—whether it be a simple gas or a complex, many-body, quantum system—the formalism holds. This universality makes the Green-Kubo formalism an incredibly powerful tool for theoretical predictions and computations.

2.4.1 The Fluctuation-Dissipation Theorem:

The fluctuation-dissipation theorem is an essential cornerstone of the Green-Kubo formalism. This theorem elegantly interconnects two seemingly unrelated phenomena: equilibrium fluctuations and dissipative processes in a system [156; 157; 158; 159].

To put it in simpler terms, an ‘equilibrium fluctuation’ refers to the spontaneous deviations from the mean behavior of a system when it’s at equilibrium. On the other hand, ‘dissipative properties’ relate to how a system responds to perturbations and moves towards restoring equilibrium. The fluctuation-dissipation theorem states that the same microscopic processes that lead to the dissipation of energy (or in other words, the restoration of equilibrium) are the ones that give rise to fluctuations when the system is in equilibrium. In essence, the theorem provides a quantitative relationship between the fluctuations observed in a system at thermal equilibrium and the system’s response to small external perturbations.

Mathematically, this theorem provides the foundation for the Green-Kubo relations. It lays the groundwork for expressing macroscopic transport coefficients, such as viscosity, thermal conductivity, and electrical conductivity, as time integrals of correlation functions of spontaneous fluctuations in the system.

2.4.2 Viscosity Measurement Using Green-Kubo Formalism:

The viscosity of a fluid, which quantifies its resistance to flow, can be another key transport coefficient calculated using the Green-Kubo formalism [160; 161]. This approach offers a fundamental understanding of fluid behavior by exploring its microscopic dynamics. Mathematically, the Green-Kubo relation for shear viscosity (η) can be expressed as:

$$\eta = \frac{V}{k_B T} \int \langle P_{ij}(t) P_{ij}(0) \rangle dt \quad (2.17)$$

Here,

- η is the shear viscosity,
- V is the volume of the system,
- k_B is Boltzmann's constant,
- T is the absolute temperature,
- $P_{ij}(t)$ is the off-diagonal component of the pressure tensor (or stress tensor) at time t , and
- $\langle P_{ij}(t) P_{ij}(0) \rangle$ is the autocorrelation function of the off-diagonal component of the pressure tensor.

Essentially, the shear viscosity can be calculated by integrating the autocorrelation function of the off-diagonal components of the pressure tensor. This relation enables us to capture the fluid's response to shear stress and hence measure its viscosity. The Green-Kubo approach offers a microscopic route to understand the viscosity of complex fluids such as colloids, polymers, and biological fluids. For instance, it can help uncover how molecular interactions, particle shapes, or thermal fluctuations contribute to the viscosity in these systems.

