Anomalous diffusion and self-propulsion of radioactive colloidal particles

by

Graham S. Wilson

B.S., Kansas State University, 2018

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Mechanical and Nuclear Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2019

Approved by:

Co-Major Professor
Dr. Amir A. Bahadori

Approved by:

Co-Major Professor
Dr. Hitesh Bindra

Copyright

Abstract

A novel concept of self-propelled, radioactively-driven colloidal particles is introduced. The focus of this work is on assessing the impact of alpha emissions on the colloidal kinematics of radioactive microparticles and radioactive Janus particles. Using Langevin dynamics and a random walk model, a theory has been developed to describe the motion of a radioactively-driven colloid. This theory shows a special case of anomalous diffusion. Numerical simulations have substantiated the theory. It is shown that alpha-particle emission can significantly affect the motion of a radioactive microparticle, although a short-lived radioisotope is required. Using Brownian dynamics, a second theory has been developed to describe the motion of a radioactive Janus particle. Non-Gaussian behavior is shown in addition to the special case of anomalous diffusion. The augmented motion of radioactive Janus particles is great enough to be experimentally observed for radionuclides with moderate half-lives. The results presented herein are important for the design of radioactive colloidal particle based radiotherapeutic cancer treatments.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The concept of radioactively-driven colloidal particles is the brainchild of Dr. Hitesh Bindra, to whom the author is deeply indebted for the idea itself, as well as his assistance and guidance throughout the process of giving a first breath of life to this nascent field.

Dr. Amir A. Bahadori has provided invaluable help and insight at every step during the course of this research, for which the author is especially grateful.

The author would also like to thank Dr. Steven J. Eckels for his role as a member of the author's supervisory committee.

The author acknowledges the support of the Nuclear Regulatory Commission graduate fellowship program.

Computing for this project was performed on the Beocat Research Cluster at Kansas State University, which is funded in part by NSF grants CNS-1006860, EPS-1006860, and EPS-0919443. CNAP Center Applications Developer Kevin Dice and Beocat Application Scientist Dave Turner provided valuable technical expertise.

Dedication

To my family,

without whom none of my success would have been possible.

Chapter 1

Introduction

In the summer of 1827, the botanist and microscopist Robert Brown observed the peculiar "rapid oscillatory motion" of pollen grains and proved that the particles were not animated by repeating his experiments with inorganic materials [1]. This new-found "Brownian motion" was passive, and did not require any special circumstances to occur. In recent decades, active forms of Brownian motion have been investigated. Particles have been constructed such that when exposed to a gradient electric, temperature, or concentration field, particle motion is induced [2, 3]. Just like air- and watercraft require a medium to propel themselves, so too do these phoretic particles. A microscopic analog to spacecraft that exhibits self-propulsion with or without the presence of a medium is a radiation emitting micro/nanoparticle. In this work, alpha-particle emission is considered, but the analysis can be easily applied to any type of radiation. The work described here is motivated by the lack of a fundamental understanding about radiation-induced movement of colloidal particles.

Although the use of alpha-particle ejection has been considered for the propulsion of macroscopic objects [4, 5], and the effect of electromagnetic radiation on hydrodynamics has been studied [6, 7], the effect of radiation emission by a microparticle on colloid dynamics has not been considered. This does not mean, however, that radioactive particles are currently unused. For instance, nanoparticles with radioactive constituents that decay via alpha-particle ejection have been developed for use in targeted alpha-particle therapy, which shows promise as a cancer treatment [8–10]. It is important to understand the role radiation emission plays in the movement of microparticles to ensure the proper use of alpha-emitting particles in cancer treatments, and to explore possibilities such as extending the settling time of suspensions or enhancing the diffusivity of colloids.

When a radioisotope undergoes alpha decay, it emits an alpha particle, and the resulting daughter isotope moves in the opposite direction in accordance with the conservation of momentum. A microparticle composed of radioisotopes moves via a momentum transfer from the daughter isotope when one of its constituent nuclei decays. Since the colloidal particle has a much greater mass than the daughter isotope, the movement is smaller in magnitude than that of an unconstrained nucleus. The movement of a microparticle due to the emission of an alpha particle is illustrated in Fig. 1.1, where the dowel pin symbol is the center of mass of the microparticle. Viscous drag causes the movement to be finite. A radioactive colloidal particle is comprised of many radioactive atoms, resulting in a center of mass motion like that in Fig. 1.2, due to the ejection of many alpha particles at different times.

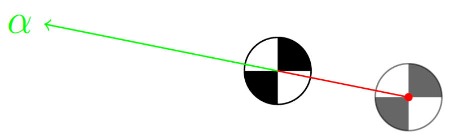

Figure 1.1: *The motion of a microparticle due to the ejection of a single alpha particle [11].*

Neither Fig. 1.1 nor Fig. 1.2 exhibit Brownian motion. The alpha-induced motion shown in these two figures would be in addition to Brownian motion. The exponential decay of the conglomeration of radionuclides within the microparticle has an ever-changing effect on the motion of the colloidal particle, resulting in a special case of anomalous diffusion.

Using conservation of momentum, the initial velocity of a microparticle, originally at rest, due to the emission of an alpha particle is on the order of 10^{-5} m/s for the cases considered herein. This results in a Reynolds number on the order of 10^{-5}; therefore, it is valid to analyze the problem as a Stokes flow. The drag on the microparticle and its terminal velocity due to gravity can then be characterized by Stokes' law [12].

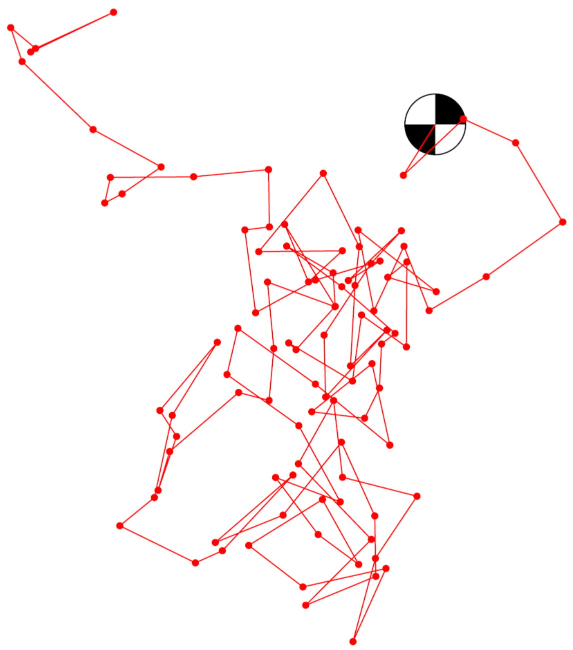

Figure 1.2: *The motion of a microparticle due to the ejection of many alpha particles [11].*

The remainder of this work is divided into three chapters. Radioactively-driven colloids and radioactive Janus particles are discussed in Chapters 2 and 3, respectively. In Chapter 2, a theoretical framework to define the motion of a radioactive colloidal particle is presented, followed by a numerical model that utilizes more realistic physics. In Chapter 3, an additional theory is disclosed to describe the motion of a radioactive Janus particle. Finally, Chapter 4 reiterates the main conclusions of this work along with suggestions for future study.

Chapter 2

Radioactively-Driven Colloids

2.1 Introduction

An effective way to quantify the motion of a particle is its mean square displacement (MSD). In his seminal 1905 paper on Brownian motion, Einstein showed that the MSD of a colloidal particle within a dilute dispersion has a linear relationship with time [13]. Since then, more detailed derivations have been completed, including those taking into account an additional force term and having short-time accuracy [14]. Furthermore, a phenomenon known as anomalous diffusion was discovered in 1926, where due to non-Brownian factors such as turbulence, the MSD is a non-linear function of time [15]. Another such factor is momentum imparted to a radioactive colloidal particle by discharge of radiation. This chapter presents the kinematics of colloidal particles due to alpha emissions and shows that the resulting motion is a special case of anomalous diffusion [16].

The rest of the chapter is organized as follows. Section 2.2 provides the methods, analytical and numerical, of determining the MSD of a radioactive colloidal particle. The results of both methods are compared in Section 2.3 and the numerical results are analyzed in Section 2.4. Final remarks and conclusions are made in Section 2.5.

2.2 Theory

In Langevin dynamics, particle motion is described with

$$m\frac{d^2\boldsymbol{x}}{dt^2} + \gamma\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}\left(t\right), \tag{2.1}$$

where m is the mass of the particle, t is time, $\boldsymbol{x}$ is the center-of-mass position vector (alternatively denoted $\boldsymbol{r}$), γ is the drag coefficient, and due to the equipartition theorem $\boldsymbol{f}\left(t\right)$ is a Gaussian white noise fluctuating force such that

$$\langle f_i\left(t\right)\rangle = 0, \tag{2.2}$$

$$\langle f_i\left(t\right) f_j\left(t+\tau\right)\rangle = 2\gamma k_B T \delta_{ij}\delta\left(\tau\right), \tag{2.3}$$

where k_B is the Boltzmann constant and T is the temperature [17]. For a liquid, the effective mass of the particle is $m = m_p + (1/2)m_f$ where m_p is the actual mass of the particle and m_f is the mass of the displaced fluid, assuming spherical particles from here on [18]. The drag coefficient is $\gamma = 6\pi\mu a$ where μ is the dynamic viscosity of the fluid and a is the particle radius [17].

Rewriting the Langevin equation (Eq. 2.1) as an Itô stochastic differential equation and including the effect of gravity results in an Ornstein-Uhlenbeck process

$$d\boldsymbol{u}_t = -\beta\left(\boldsymbol{u}_t - \boldsymbol{\mu}\right)dt + \sigma d\boldsymbol{W}_t, \tag{2.4}$$

where $\boldsymbol{u}_t$ is the particle velocity, $\beta = \gamma/m$, $|\boldsymbol{\mu}| = |2g(\rho_f - \rho_p)a^2/(9\mu)|$ is the Stokes velocity of a particle with density ρ_p in a fluid with density ρ_f, g is the acceleration due to gravity, $\sigma = \sqrt{2\gamma k_B T}/m$, and $\boldsymbol{W}_t$ is the Wiener process [17, 19].

A Wiener process is a continuous stochastic process, denoted W_t or $W(t)$, that satisfies the following three conditions in a single dimension on $t \in [0, t_{end}]$ [20].

1. $W(0) = 0$ with probability 1.

2. $W(t) - W(s) \sim \sqrt{t-s}\,\mathcal{N}(0,1)$ for $0 \le s < t \le t_{end}$, where $\mathcal{N}(0,1)$ denotes a normally distributed random variable with zero mean and unit variance. It should be noted that $k\,\mathcal{N}(\mu, \sigma^2) = \mathcal{N}(k\mu, k^2\sigma^2)$.

3. For $0 \le s < t < u < v \le t_{end}$ the increments $W(t) - W(s)$ and $W(v) - W(u)$ are independent.

Including the effect of alpha-particle emission in Eq. 2.4 results in a modified Ornstein-Uhlenbeck process

$$d\boldsymbol{u}_t = -\beta\left(\boldsymbol{u}_t - \boldsymbol{\mu}\right)dt + \frac{\boldsymbol{F}_t}{m}dt + \sigma d\boldsymbol{W}_t. \tag{2.5}$$

The integral of the product $\boldsymbol{F}_t dt$ is the momentum change of (i.e. linear impulse applied to) the colloidal particle due to radiation emission. The magnitude of the momentum change due to the ejection of an alpha particle is

$$M_0 = |\Delta\boldsymbol{p}| = \sqrt{\frac{E^2 + 2Em_\alpha c^2}{c^2}}, \tag{2.6}$$

where E is the kinetic energy of the alpha particle, m_α is the rest mass of an alpha particle, and c is the speed of light [21].

The colloidal particle here is treated as a point particle so that rotational motion due to Brownian forces and the angular impulse from the ejection of alpha particles can be neglected, as can energy loss of the alpha particle as it travels out of the colloidal particle. The escape of recoiling daughter isotopes from the colloidal particle, heating of the fluid by the alpha particles, and daughter product decays are ignored as well.

2.2.1 Analytical Methods

Solving Eq. 2.4 for velocity results in [14, 19]

$$\boldsymbol{u}_t = \boldsymbol{u}_0 e^{-\beta t} + \boldsymbol{\mu}\left(1 - e^{-\beta t}\right) + \frac{\sigma}{\sqrt{2\beta}}e^{-\beta t}\boldsymbol{W}\left(e^{2\beta t} - 1\right). \tag{2.7}$$

Integrating with respect to time yields [14, 19]

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \frac{\boldsymbol{u}_0}{\beta}\left(1 - e^{-\beta t}\right) + \frac{\boldsymbol{\mu}}{\beta}\left(\beta t - 1 + e^{-\beta t}\right) + \boldsymbol{W}\left(\frac{\sigma^2}{2\beta^3}\left(-3 + 4e^{-\beta t} - e^{-2\beta t} + 2\beta t\right)\right), \tag{2.8}$$

where $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ are the initial position and velocity of the particle, respectively. Eq. 2.8 describes only the Brownian component of motion.

The movement of a colloidal particle due solely to the discharge of radiation can be approximated as a random walk on a three-dimensional lattice. The step length is $\varepsilon = M_0/\gamma$, or the finite distance traveled by a sphere propelled by an impulse [17]. Successive steps occur at a frequency equal to the activity of the radioisotope $1/\delta t = A = \lambda N_0 e^{-\lambda t}$, where $\lambda = \ln(2)/T_{1/2}$ is the decay constant of the radionuclide, $T_{1/2}$ is the half-life of the radionuclide, $N_0 = m_p N_A/\mathcal{A}$ is the number of radioactive nuclei in the colloidal particle at $t = 0$, N_A is Avogadro's number, and $\mathcal{A}$ is the atomic mass of the radioisotope [21].

Define $P_N(m, n, o)$ as the probability to find the random walker colloidal particle at position $(x, y, z) = (m\varepsilon, n\varepsilon, o\varepsilon)$ at time $t = N\delta t$, where (m, n, o) are integers. Considering an equal probability of motion in all directions, we have

$$\begin{aligned} P_{N+1}(m,n,o) = {} & \frac{1}{6}P_N(m-1,n,o) + \frac{1}{6}P_N(m+1,n,o) + \frac{1}{6}P_N(m,n-1,o) \\ & + \frac{1}{6}P_N(m,n+1,o) + \frac{1}{6}P_N(m,n,o-1) + \frac{1}{6}P_N(m,n,o+1). \end{aligned} \tag{2.9}$$

In the limit of large N and small ε

$$\delta t \frac{\partial P}{\partial t} = \frac{\varepsilon^2}{6}\left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2}\right). \tag{2.10}$$

Thus as the step size approaches zero and the frequency approaches infinity, a random walk converges to the diffusion equation

$$\frac{\partial f}{\partial t} = D\nabla^2 f, \qquad D = \frac{\varepsilon^2}{6\delta t}, \tag{2.11}$$

where D is the diffusion coefficient and f is the time-dependent probability of the particle's location [22]. Since Eq. 2.11 results in a normal distribution for f, the total MSD of the particle over all three dimensions is given by $\langle (\boldsymbol{r}(t) - \boldsymbol{r}_0)^2 \rangle = 6Dt$. A random walk may then be described by the Wiener process $\boldsymbol{W}(\varepsilon^2 t/(3\delta t))$, the variance in one dimension being one third of the total variance.

In the case of a radioactively-driven colloid, the diffusion coefficient is dependent on time and thus the diffusion equation is solved with respect to t' instead of t, where $t' = \int_0^t e^{-\lambda s} ds = \left(1 - e^{-\lambda t}\right)/\lambda$. This substitution results in the Wiener process $\boldsymbol{W}(\varepsilon^2 t'/(3\delta t))$, where the diffusion coefficient is $D_{rad} = N_0 (M_0/\gamma)^2 \left(1 - e^{-\lambda t}\right)/(6t)$. Thus the stochastic process describing the position of the radioactive colloidal particle is

$$\begin{aligned}\boldsymbol{x}_t = \boldsymbol{x}_0 + \frac{\boldsymbol{u}_0}{\beta}\left(1 - e^{-\beta t}\right) + \frac{\boldsymbol{\mu}}{\beta}\left(\beta t - 1 + e^{-\beta t}\right) \\ + \boldsymbol{W}\left(\frac{\sigma^2}{2\beta^3}\left(-3 + 4e^{-\beta t} - e^{-2\beta t} + 2\beta t\right) + \frac{N_0}{3}\left(\frac{M_0}{\gamma}\right)^2 \left(1 - e^{-\lambda t}\right)\right). \quad (2.12)\end{aligned}$$

Letting t approach infinity results in $\boldsymbol{x}_t = \boldsymbol{\mu} t + (\sigma/\beta)\boldsymbol{W}_t$, from which it can be seen that $D_0 = k_B T/\gamma$, the Stokes-Einstein equation.

This is a special case of anomalous diffusion, where the MSD is essentially only a non-linear function of time from the initialization of the system until a timescale similar to that of the half-life. This aging process is unique in that it does not exhibit a traditional sub- or super-diffusive behavior, but rather transitions from anomalous diffusion to normal diffusion. The exponentially-decaying behavior is reminiscent of the transient terms in the Brownian MSD, but with a much greater magnitude and duration.

Due to the central limit theorem and the random walk approximation, Eq. 2.12 may not be converged to the true colloidal particle position at very short times because not enough steps will have been taken. Furthermore, the discharge of radiation was approximated to occur on a three-dimensional lattice instead of isotropically. These shortcomings suggest that a more detailed model is required.

2.2.2 Numerical Methods

In order to verify the accuracy of the stochastic process describing particle position (Eq. 2.12), and to provide a more accurate model for short times, the physics of individual alpha-particles were simulated to compute the $\boldsymbol{F}_t dt$ term in the Langevin model of Eq. 2.5. The velocity of the particle was found with Euler-Maruyama (EM) integration, and then the position was found via trapezoidal integration [20]. For stability of the solution, the time step size was chosen such that for a pure Ornstein-Uhlenbeck process the EM approximate solution would be stable in distribution ($\Delta t < 1/\beta$) [23].

To apply a numerical method to Eq. 2.5, the interval of simulation is discretized with time steps Δt_j. Let $\tau_j = \sum_{n=1}^{j} \Delta t_n$. The numerical approximation to $\boldsymbol{u}(\tau_j)$ is denoted $\boldsymbol{u}_j$. The EM method takes the form

$$\boldsymbol{u}_j = \boldsymbol{u}_{j-1} - \beta\left(\boldsymbol{u}_{j-1} - \boldsymbol{\mu}\right)\Delta t_j + \frac{\Delta \boldsymbol{p}_j}{m} + \sigma\left(\boldsymbol{W}\left(\tau_j\right) - \boldsymbol{W}\left(\tau_{j-1}\right)\right). \tag{2.13}$$

Trapezoidal integration takes the form

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + 0.5\Delta t_j\left(\boldsymbol{u}_{j-1} + \boldsymbol{u}_j\right). \tag{2.14}$$

The escape of recoiling daughter isotopes from the colloidal particle is still neglected, and the colloidal particle is still considered a point particle for rotational purposes, but not for interactions of alpha particles with matter. The $\Delta \boldsymbol{p}_j$ term in Eq. 2.13 is obtained via the following simulation steps:

1. Randomly choose the Cartesian coordinates of the alpha decay with a uniform distribution for location within the colloidal particle using ball point picking [24]

$$(x, y, z) = \frac{a\left(X_1, X_2, X_3\right)}{\sqrt{Y + X_1^2 + X_2^2 + X_3^2}}, \tag{2.15}$$

   where X_1, X_2, and X_3 are independent random variates from a standard normal distribution and Y is an independent random variate from an exponential distribution with mean $\lambda = 1$. The location of the alpha decay in spherical coordinates $\boldsymbol{r}$ is then

$$r_{loc} = |\,(x, y, z)\,|, \quad \theta_{loc} = \mathrm{acos}\left(\frac{X_3}{\sqrt{X_1^2 + X_2^2 + X_3^2}}\right), \quad \phi_{loc} = \mathrm{atan2}\,(X_2, X_1)\,, \tag{2.16}$$

where r is the radial direction, θ is the polar direction, and ϕ is the azimuthal direction.

2. Randomly choose the radiation emission direction $\boldsymbol{\Omega}$ with a uniform distribution for direction over 4π steradians using sphere point picking [25]

$$r_{dir} = 1, \qquad \theta_{dir} = \mathrm{acos}\,(2v - 1)\,, \qquad \phi_{dir} = 2\pi u, \tag{2.17}$$

where u and v are independent random variates from a uniform distribution on $(0, 1)$.

3. Calculate the distance Δx the alpha particle travels from the decay location until the point where it exits the colloidal particle.

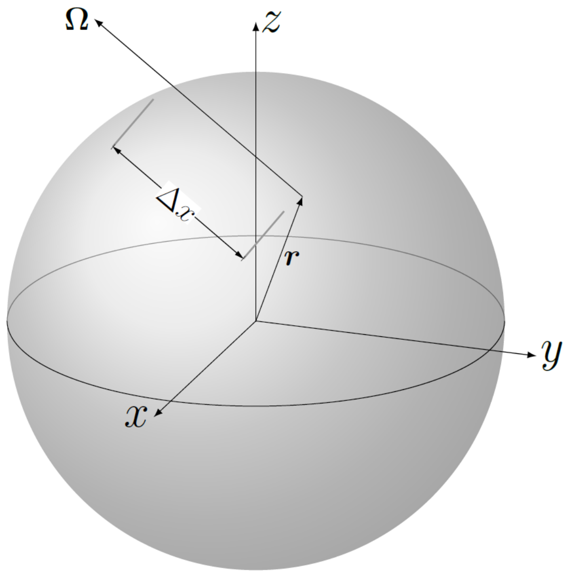

Figure 2.1: *An example geometry for a simulated alpha decay.*

The angle ω between $\boldsymbol{r}$ and $\boldsymbol{\Omega}$ is found using the dot product to be

$$\omega = \operatorname{acos}\left(\sin\theta_{loc}\sin\theta_{dir}\cos\left(\phi_{loc}-\phi_{dir}\right)+\cos\theta_{loc}\cos\theta_{dir}\right). \tag{2.18}$$

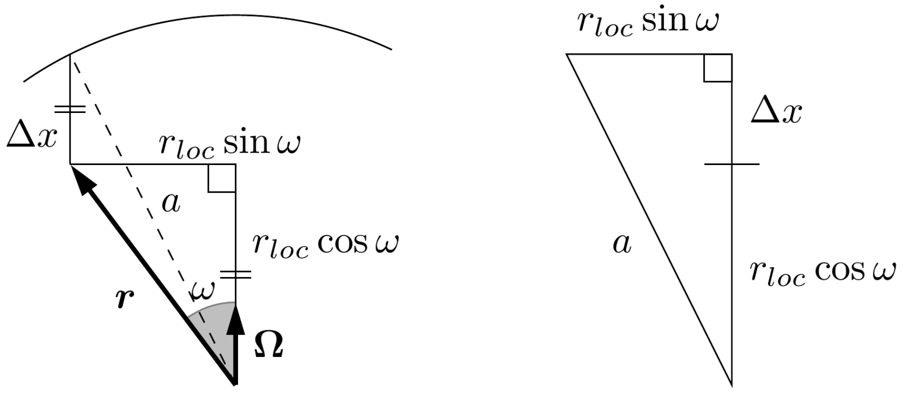

Figure 2.2: *The trigonometry in the $\boldsymbol{r}$ - $\boldsymbol{\Omega}$ plane used to find Δx.*

From Fig. 2.2 it can be shown that

$$\Delta x = \sqrt{a^2 - r_{loc}^2\sin^2\omega} - r_{loc}\cos\omega. \tag{2.19}$$

4. Calculate the energy of the alpha particle as it exits the colloidal particle with

$$E_{exit} = E_\alpha - \left.\frac{dE}{dx}\right|_{tot}\Delta x, \tag{2.20}$$

where E_α is the initial energy of the alpha particle and $-dE/dx|_{tot}$ is the stopping power of the colloidal particle. Since only a fraction of the alpha particle's energy would be lost if it traveled through the entire diameter of the colloidal particle, the stopping power is taken to be constant.

5. Use E_{exit} in Eq. 2.6 to determine the magnitude of the momentum imparted to the colloidal particle in the direction opposite of the radiation discharge.

6. The net momentum change from all alpha particles i emitted in a time step in Cartesian coordinates is

$$\Delta \boldsymbol{p}_j = -\sum_i |\Delta \boldsymbol{p}_i| \left(\sin\theta_{dir,i}\cos\phi_{dir,i}, \sin\theta_{dir,i}\sin\phi_{dir,i}, \cos\theta_{dir,i}\right). \tag{2.21}$$

The particles simulated had a radius of 0.5 μm and were suspended in water at 20°C. The motion of a larger particle is dominated by gravity, while for a smaller particle viscous drag ($\sim a$) would damp out the greatly decreased radiation-induced motion ($\sim a^3$). The correlation for temperature dependent water density in kg/m^3 for $T \in [0°\text{C}, 150°\text{C}]$ is [26]

$$\rho_f(T) = \big(999.83952 + 16.945176T - 7.9870401\times10^{-3}T^2 - 46.170461\times10^{-6}T^3 + 105.56302\times10^{-9}T^4 - 280.54253\times10^{-12}T^5\big) / \left(1 + 16.879850\times10^{-3}T\right), \tag{2.22}$$

where the temperature is in Celsius, which results in a density of 998.2041 kg/m^3 at 20°C. The Vogel equation provides the temperature dependent dynamic viscosity of water in Pa s for $T \in [273\text{ K}, 373\text{ K}]$ [27]

$$\mu(T) = \frac{1}{1000}e^{-3.7188+578.919/(-137.546+T)}, \tag{2.23}$$

where the temperature is in Kelvin, which results in a dynamic viscosity of 1.0017×10^{-3} Pa s at 20°C. Particles composed of the four materials listed in Table 2.1 were simulated to cover a wide range of half lives. To understand the effect of radiation discharge, "particle pairs" were simulated such that both particles in a pair had the same random Brownian component of motion, but only one took into account the effect of alpha-particle emission. Example trajectories from one particle pair are plotted in Fig. 2.3. To achieve low standard error with a reasonable computational time 10,000 particle pairs were simulated for each scenario.

In order to determine whether different computational timescales of alpha-particle ejection and Brownian motion would change the computed position of the colloidal particle, simulations of two different time step types were conducted. The first time step was set to the value of 5×10^{-7} seconds for ^{222}Th and ^{225}Pa and 1×10^{-7} seconds for ^{222}Ra and ^{218}Ac. Both time step values were near the stability limit for each particle to reduce com-

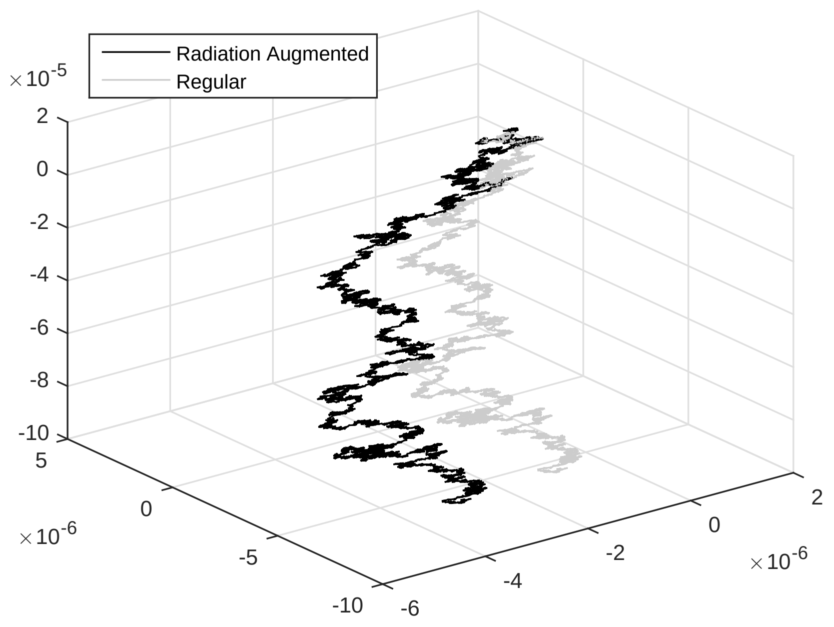

Figure 2.3: *The trajectories of a particle pair.*

putational expense. In this case, the number of nuclei that decayed in each time step was determined as a Poisson random variable with a mean of $\int_t^{t+\Delta t} A(s)ds = N_0 e^{-\lambda t}\left(1 - e^{-\lambda \Delta t}\right)$. The second time step was adaptive such that on average there would be one decay per time step. The decay constant is defined as $\lambda \equiv \lim_{\Delta t \to 0}(\Delta N/N)/\Delta t$, so the time step was set to $\Delta t = 1/\lambda N(t)$, where $N(t)$ is the number of undecayed nuclei [21]. In this case, the number of nuclei that decayed in each time step was determined as a Poisson random variable with a mean of 1. Once the adaptive time steps grew larger than the set time steps from the former case, the set time steps and mean number of decays from the former case were implemented. The C++ particle simulation code for each time step case is provided in Appendix A.

2.2.3 Timescale Analysis

The timescale of Brownian motion, or the Brownian relaxation time, is $t_b = 1/\beta$, and the timescale of vorticity diffusion, or the viscous relaxation time, is $t_v = \rho_f a^2/\mu$ [33]. For transient behavior of the fluid to play a negligible role in the colloidal particle dynamics, the viscous relaxation time must be less than the Brownian relaxation time, which is the case

Table 2.1: *Alpha-decaying radioisotopes used in the simulations [28–32].*

Case	1	2(a)	2(b)	3
Isotope	^{222}Ra	^{225}Pa	^{222}Th	^{218}Ac
ρ (g/cm^3)	5.0	15.37	11.72	10.07
$T_{1/2}$ (s)	36.2	1.7	2.24×10^{-3}	1.1×10^{-6}
E_α (MeV)	6.556	7.22	7.98	9.20
$\mathcal{A}$ (g/mol)	222	225	222	218
$-\frac{dE}{dx}\big\rvert_{tot}$ (MeV/mm)	101.8	296.5	214.4	176.5
$A(0)$ (decays/s)	1.36×10^{8}	8.78×10^{9}	5.15×10^{12}	9.18×10^{15}
M_0 (N s)	1.18×10^{-19}	1.24×10^{-19}	1.30×10^{-19}	1.40×10^{-19}
m_p (kg)	2.62×10^{-15}	8.05×10^{-15}	6.14×10^{-15}	5.27×10^{-15}

for the colloidal particles presented herein. The Langevin model utilized is thus valid for Brownian motion.

As the time-independent magnitudes of the Brownian and radiation-emission MSDs differ, it is difficult to define a timescale of radiation-emission motion. However, it is known that the average time between alpha decays within the colloidal particle is on the order of magnitude of the sonic timescale, $t_s = a/s$, where s is the speed of sound in the fluid. This means that the inertia of the fluid, transient hydrodynamic interactions between the fluid and the colloidal particle, and fluid compressibility effects may play a role in colloidal particle motion. The Langevin model ignores these physics to compute the colloidal particle motion to a first approximation.

2.2.4 Heat Transfer Analysis

Once the alpha particles exit the colloidal particle, they thermalize in the surrounding fluid, ultimately imparting their kinetic energy to the fluid as heat. As directly ionizing quanta, the alpha particles have a well-defined finite range R in the fluid [21]. Since the magnitude of the Brownian motion of the particle is governed by the temperature of the fluid, it is important to characterize heating of the fluid by the alpha particles. In the following, the colloidal particle is approximated as a point, and the alpha particle heating is assumed to be uniform volumetric heat generation within a sphere of radius R centered at the point colloidal particle. This system is one-dimensional in a semi-infinite spherical domain; thus, the heat equation becomes [34]

$$\frac{1}{r}\frac{\partial^2}{\partial r^2}(rT) + \frac{1}{k}g(r,t) = \frac{1}{\alpha}\frac{\partial T}{\partial t}, \tag{2.24}$$

with the conditions

$$T(r \to 0) \Rightarrow \text{finite}, \tag{2.25}$$

$$T(r, t = 0) = T_0, \tag{2.26}$$

where $\alpha = k/(\rho_f c_p)$ is the thermal diffusivity of the fluid, k is the thermal conductivity of the fluid, c_p is the specific heat capacity of the fluid, and T_0 is the initial fluid temperature. The volumetric heat generation is

$$g(r,t) = \frac{A(t)E}{\frac{4}{3}\pi R^3} = \frac{3\lambda N_0 E}{4\pi R^3}e^{-\lambda t}, \tag{2.27}$$

for $r \leq R$, and 0 for $r > R$. The solution of the homogeneous problem

$$\frac{1}{r}\frac{\partial^2}{\partial r^2}(r\Psi) = \frac{1}{\alpha}\frac{\partial \Psi}{\partial t}, \tag{2.28}$$

with the conditions

$$\Psi(r \to 0) \Rightarrow \text{finite}, \tag{2.29}$$

$$\Psi(r, t = 0) = T_0, \tag{2.30}$$

is given by [34]

$$\Psi(r,t) = \frac{1}{r\sqrt{4\pi\alpha t}}\int_0^\infty \left\{\exp\left[-\frac{(r-r')^2}{4\alpha t}\right] - \exp\left[-\frac{(r+r')^2}{4\alpha t}\right]\right\} T_0 r' dr' = T_0, \tag{2.31}$$

$$= \int_0^\infty G(r,t|r',\tau)\big|_{\tau=0} T_0 r'^2 dr', \tag{2.32}$$

where G is Green's function. The solution of Eq. 2.24 is then [34]

$$\begin{aligned}T(r,t) &= \Psi(r,t) + \frac{\alpha}{k}\int_0^t\int_0^\infty G\left(r,t|r',\tau\right)g(r',\tau)r'^2dr'd\tau \\ &= T_0 + \frac{3\lambda N_0 E}{4\pi R^3 r\sqrt{4\pi k\rho_f c_p}}\int_0^t \frac{e^{-\lambda\tau}}{\sqrt{t-\tau}}\int_0^R \left\{\exp\left[-\frac{(r-r')^2}{4\alpha(t-\tau)}\right] - \exp\left[-\frac{(r+r')^2}{4\alpha(t-\tau)}\right]\right\} r'dr'd\tau.\end{aligned} \tag{2.33}$$

The ^{225}Pa particle described in the previous section is used for the heat transfer analysis. This isotope was chosen since it is the most active colloidal particle simulated for which Fourier's law is still valid. Fourier's law does not account for the time lag needed to establish steady-state heat conduction from a sudden temperature gradient. In liquids, this thermal relaxation time is on the order of 10^{-10} s to 10^{-14} s, and the average time between alpha decays within a colloidal particle composed of ^{225}Pa is at the upper limit of this range [35].

The solution to Eq. 2.33 for the ^{225}Pa particle described in the previous section is plotted in Fig. 2.4 and Fig. 2.5 using the code in Appendix F. There is roughly a ten percent maximum increase in fluid temperature. Although fluid heating by the alpha particles is ignored in the MSD model and particle simulations, it would present an interesting multiphysics problem. Both the density and dynamic viscosity of water decrease as temperature increases, causing counteracting effects on β and σ, and, since $\rho_p > \rho_f$, an increase in the Stokes velocity $|\boldsymbol{\mu}|$ of the particle. The temperature increase also directly increases σ. Since the MSD of the Brownian motion is ultimately governed by D_0, its temperature dependence is proportional to T/μ. Therefore, in water, the MSD due to Brownian motion will increase as a result of alpha particle heating. The decrease in the dynamic viscosity of the water also decreases the drag coefficient γ, increasing the MSD due to radiation ejection. Overall, the increased ambient temperature would amplify the motion of the particle, in both the mean position and variance about that mean.

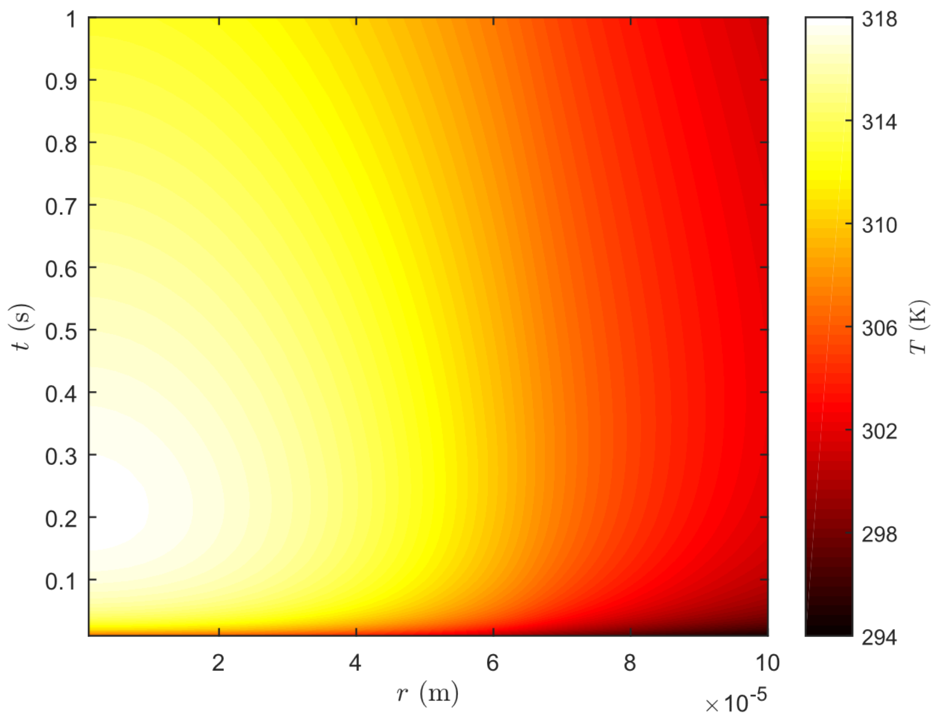

Figure 2.4: *The heat distribution near a ^{225}Pa particle.*

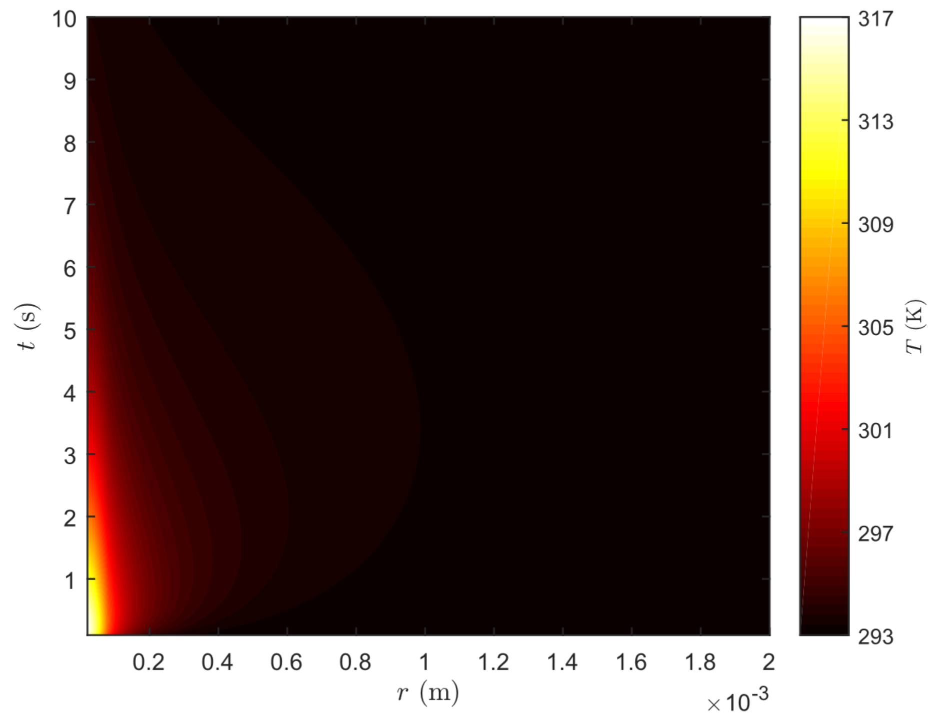

Figure 2.5: *The heat distribution far from a ^{225}Pa particle.*

2.3 Results

A single output file from the particle simulation codes of Appendix A was generated for each time step case and radionuclide combination as described in the first section of Appendix B. The second section of Appendix B provides the code that made the subsequent MSD plots. For the following plots, the MSD was computed about the displacement due to gravity so that the variance of the Wiener process is more readily observable. Plots of the MSD including the displacement due to gravity are included in Appendix G. The theoretical MSD $\langle (\boldsymbol{r}(t) - \boldsymbol{r}_0)^2 \rangle_{theor}$, the MSD computed from the numerical simulations $\langle (\boldsymbol{r}(t) - \boldsymbol{r}_0)^2 \rangle$, and the 1σ error band about the numerical MSD $\langle (\boldsymbol{r}(t) - \boldsymbol{r}_0)^2 \rangle \pm \sigma$ are all plotted with and without the effect of radioactivity.

To calculate the MSD, the displacement about the mean position due to gravity $\boldsymbol{x}_t - \boldsymbol{x}_0 - \frac{\boldsymbol{\mu}}{\beta}\left(\beta t - 1 + e^{-\beta t}\right)$, for $\boldsymbol{u}_0 = \boldsymbol{0}$, is squared for every particle and then averaged over all particles for each Cartesian direction. The overall MSD $\langle (\boldsymbol{r}(t) - \boldsymbol{r}_0)^2 \rangle$ is then simply the sum of the three directional MSDs. The standard error of the MSD is $\sigma = \sigma_i / \sqrt{N_{part}}$, where σ_i is the corrected sample standard deviation of the square displacements about the mean position due to gravity and N_{part} is the number of particles [36].

In Case 1, there is no significant difference in the motion of a colloidal particle with and without the effect of alpha-particle emission. Case 2 highlights the difference in motion between a radioactive and non-radioactive colloidal particle. In Case 3, the adaptive and set time step simulations give different results.

For Case 1, the MSDs for ^{222}Ra are plotted in Fig. 2.6, with the numerical MSD from the adaptive time step case. The MSD considering the effect of alpha-particle emission is only slightly larger than the MSD without the effect because the half-life of ^{222}Ra is relatively long. The inset plot is of the time derivative of the theoretical MSDs, and also shows a negligible difference. The derivatives highlight the Brownian exponential terms in Eq. 2.12. Both the Brownian and radioactive parts of the numerical simulations show good agreement with Eq. 2.12. The numerical MSD from the set time step case was virtually identical to that from the adaptive case.

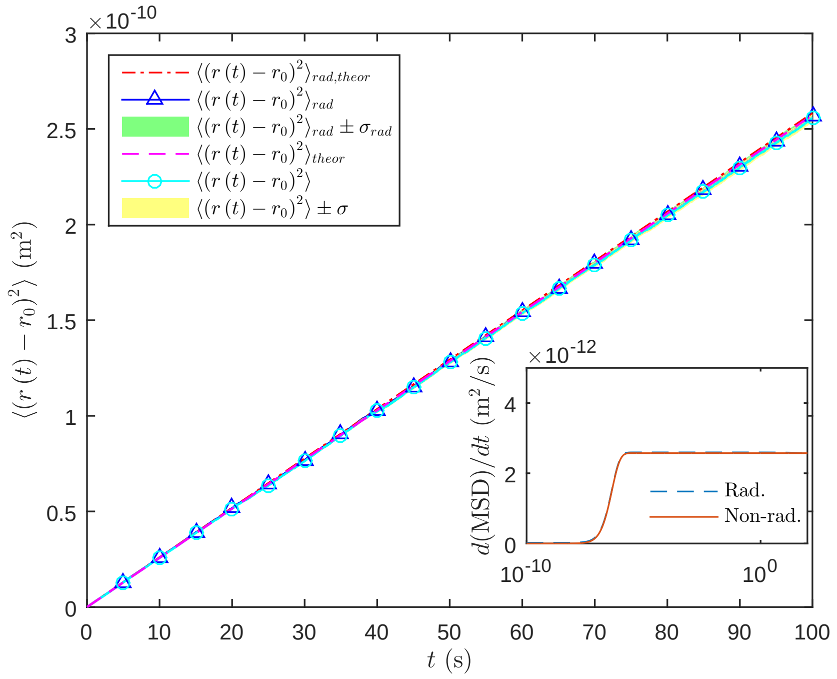

Figure 2.6: *The MSDs for ^{222}Ra.*

For Case 2(a), the MSDs for ^{225}Pa are plotted in Fig. 2.7, with the numerical MSD from the adaptive time step case. The anomalous diffusion is of an order of magnitude comparable to the regular diffusion. Both the Brownian and radioactive parts of the numerical simulations are in agreement with Eq. 2.12. The inset plot provides a useful comparison of the effects of the exponential terms in Eq. 2.12. At very short times, the MSDs behave as shown in Fig. 2.8. There is an offset between the theoretical and numerical MSDs when radiation is considered as allowed by the central limit theorem and the random walk approximation as discussed earlier. The Brownian theory and simulation results are still in accord. For the radioactive case, the numerical MSD converges fairly quickly to the theoretical MSD, as seen in Fig. 2.9.

For Case 2(b), the MSDs for ^{222}Th are plotted in Fig. 2.10, with the numerical MSD from the adaptive time step case. Anomalous diffusion is apparent in this case because of the short half-life of ^{222}Th. The slope of the MSD for the radiation ejection case is considerably larger than it is for ^{225}Pa. The Brownian and radioactive theory and simulation results are congruous. For both isotopes in Case 2, the numerical MSDs from the adaptive and set time

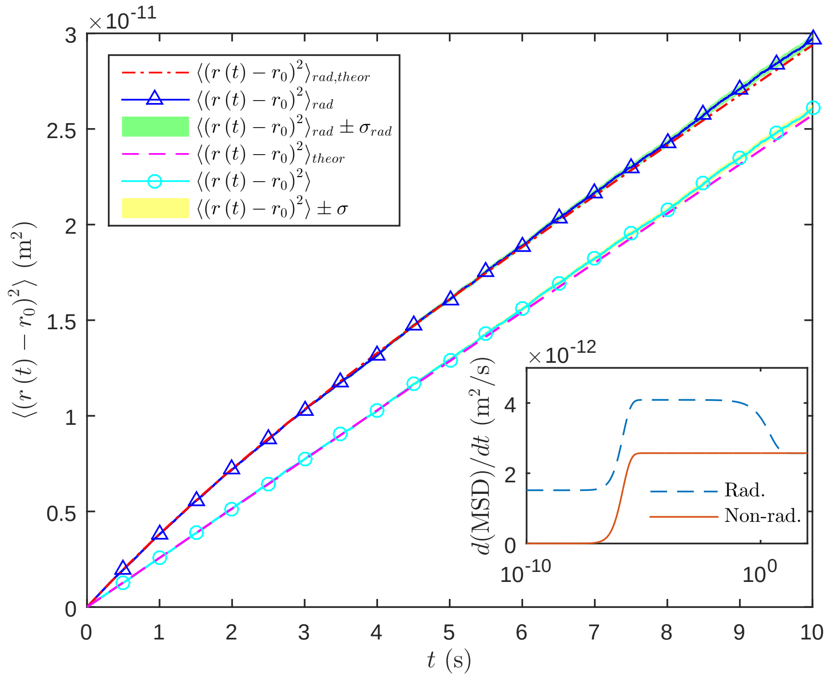

Figure 2.7: *The MSDs for* ^{225}*Pa.*

step cases were practically equal.

For Case 3, the MSDs for ^{218}Ac are plotted in Fig. 2.11, with the numerical MSD from the set time step case. The theoretical and numerical MSDs for the radiation emission case never match.

In Fig. 2.12 the MSDs are plotted with the numerical MSD from the adaptive time step case. The theoretical and numerical MSDs for the radiation emission case are not in accord at first, but the simulation results do eventually converge to theory. With such a short half-life, the computational time step size does play a role: the limiting simulation value for the adaptive case shows good agreement with theory, but the limiting simulation value for the set case does not. The offset at earlier times for both cases is allowed by the central limit theorem and is so pronounced for this isotope as compared to the others because of the short half-life. Anomalous diffusion is quite prominent because of the short half-life as well. The Brownian theory and simulation results still match, and the slope of the MSD considering radiation emission is significantly larger than it is for ^{222}Th.

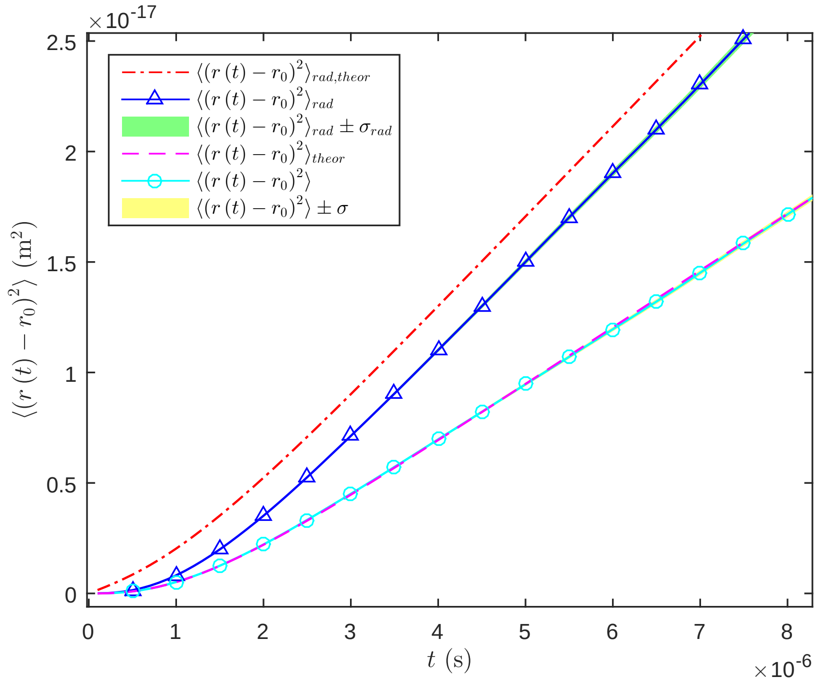

Figure 2.8: *The MSDs for ^{225}Pa at very short times.*

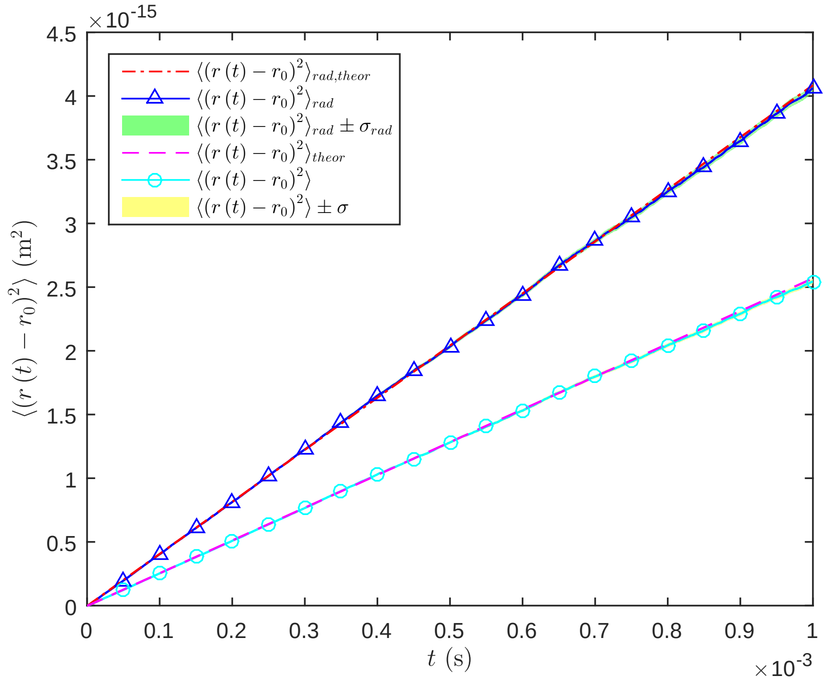

Figure 2.9: *The MSDs for ^{225}Pa at short times.*

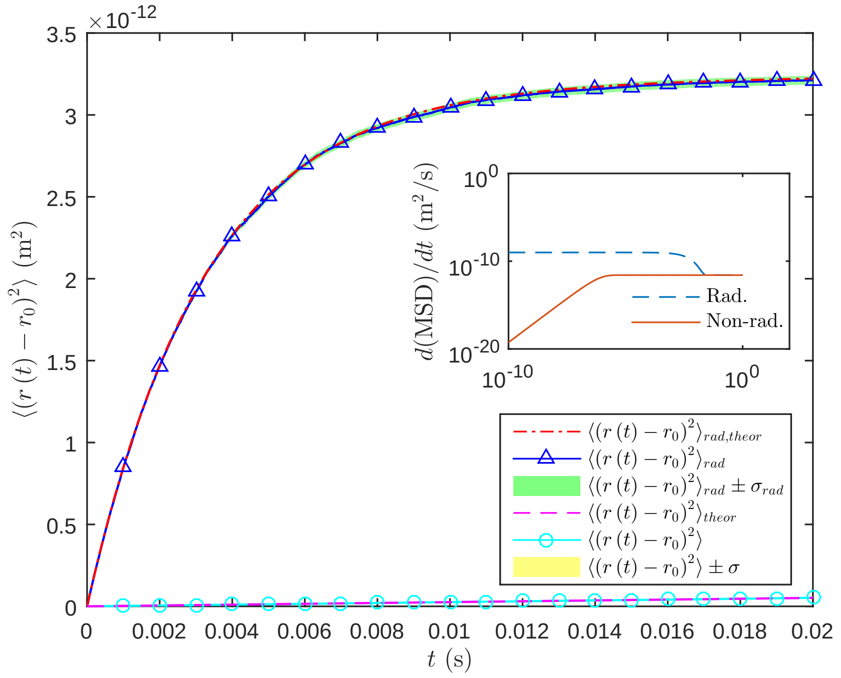

Figure 2.10: *The MSDs for ^{222}Th.*

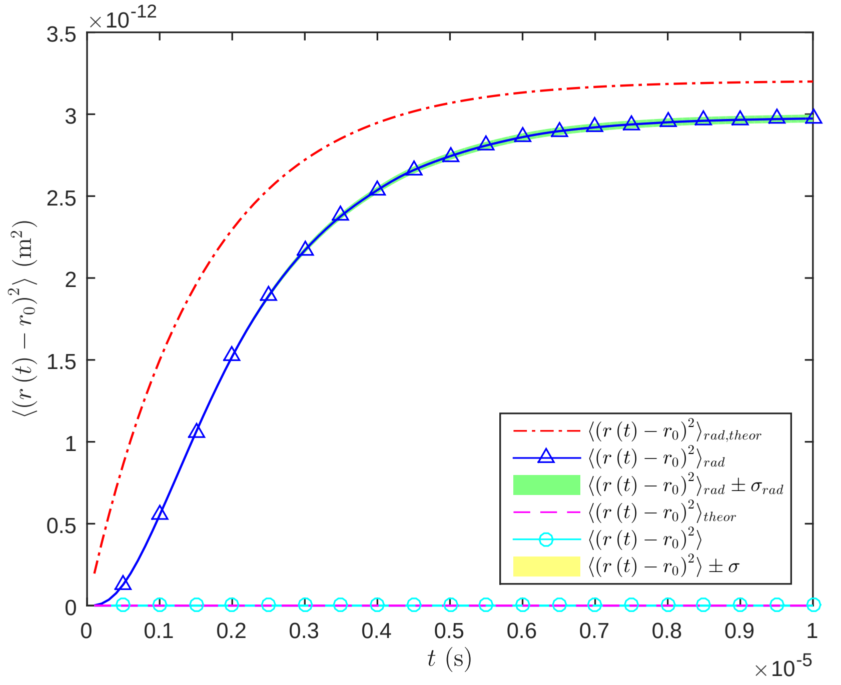

Figure 2.11: *The set time step MSDs for ^{218}Ac.*

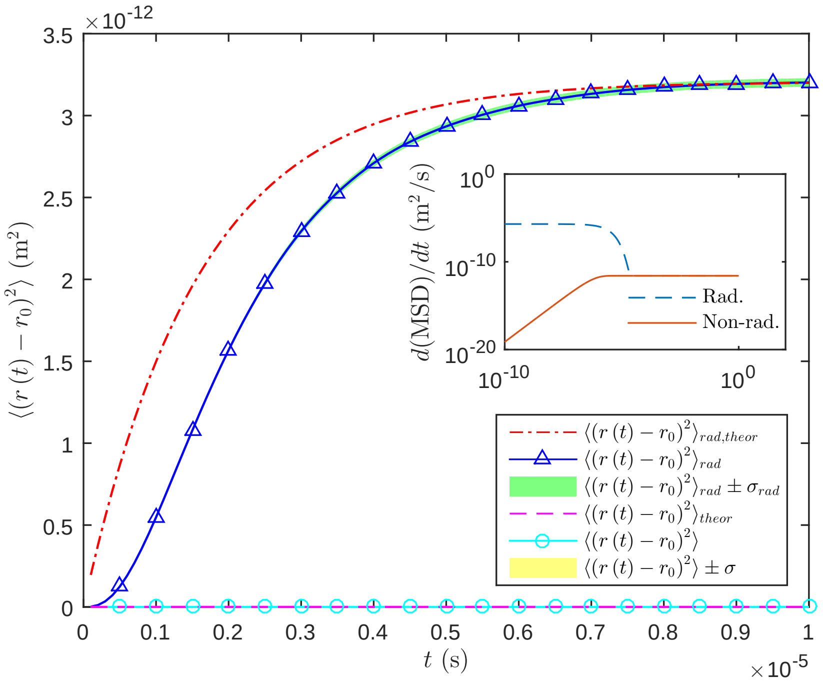

Figure 2.12: *The adaptive time step MSDs for ^{218}Ac.*

2.4 Analysis

To ensure effective random number generation within the numerical simulations, the autocorrelation function of velocity data and the excess kurtosis of the colloidal particle position were both computed. The autocorrelation functions for both the non-radioactive and radioactive adaptive time step cases with ^{225}Pa are plotted in Fig. 2.13 and Fig. 2.14, respectively. The script used to generate these figures is given in the third section of Appendix B. Values larger in magnitude than the solid/dashed lines mean that the null hypothesis that there is no autocorrelation at and beyond a given lag is rejected at a significance level of 95/99%. Thus, from the figures, the velocity is random with and without consideration of alpha-particle ejection.

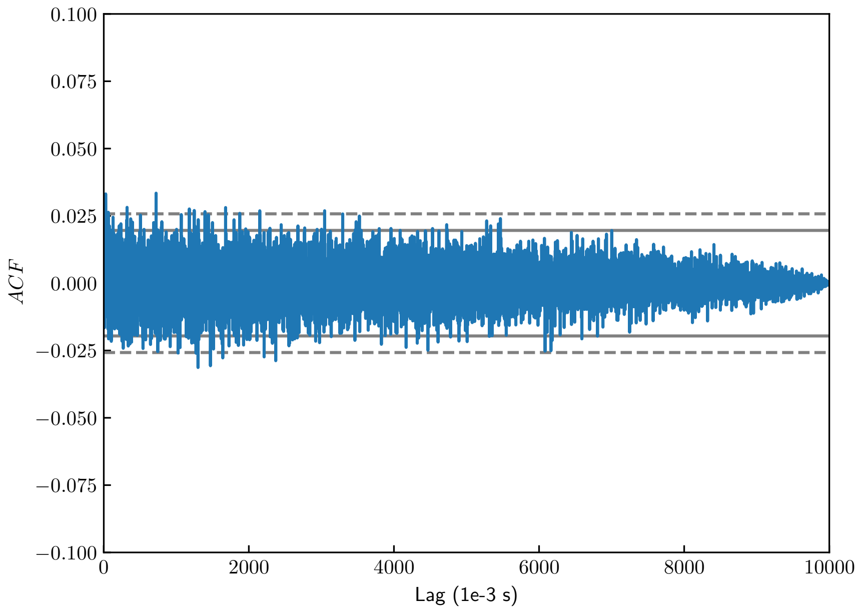

Figure 2.13: *The autocorrelation function of the u velocity data from the non-radioactive, adaptive time step case for ^{225}Pa.*

The sample kurtosis b_2 of the three-dimensional colloidal particle position was calculated as [37]

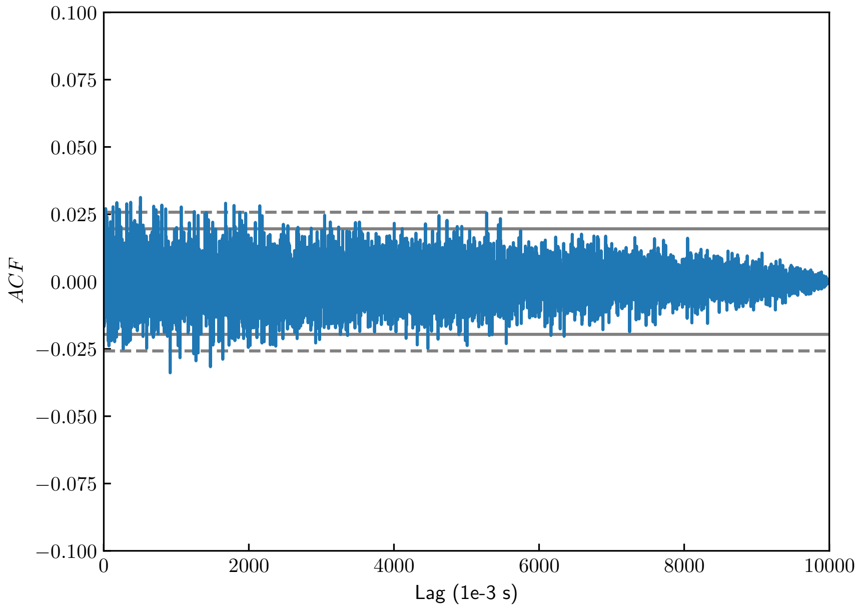

Figure 2.14: *The autocorrelation function of the u velocity data from the radioactive, adaptive time step case for* ^{225}*Pa.*

$$b_2 = \frac{1}{N_{part}} \sum_{i=1}^{N_{part}} \left[(\boldsymbol{x}_i - \bar{\boldsymbol{x}})' \boldsymbol{S}^{-1} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \right]^2 , \tag{2.34}$$

where $\boldsymbol{x}_i$ is the displacement about the mean position due to gravity of the i^{th} particle, $\bar{\boldsymbol{x}}$ is the average displacement about the mean position due to gravity, and $\boldsymbol{S}$ is the sample covariance matrix [38]

$$\boldsymbol{S} = \frac{1}{N_{part}} \sum_{i=1}^{N_{part}} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) (\boldsymbol{x}_i - \bar{\boldsymbol{x}})' . \tag{2.35}$$

The three-dimensional kurtosis of a Gaussian population is $\beta_2 = d(d+2) = 15$, where d is the number of dimensions, and the three-dimensional excess kurtosis of a Gaussian population is $\gamma_2 = \beta_2 - d(d+2) = 0$; therefore, the sample excess kurtosis of the three-dimensional colloidal particle position is $g_2 = b_2 - 15$ [37, 39]. In Fig. 2.15, the sample excess kurtosis of the three-dimensional colloidal particle position about the displacement due to gravity for the adaptive time step case with ^{225}Pa is plotted using the script in the third section of Appendix B. From the figure it can be seen that the particles are approximately normally distributed, as is expected from a Wiener process.

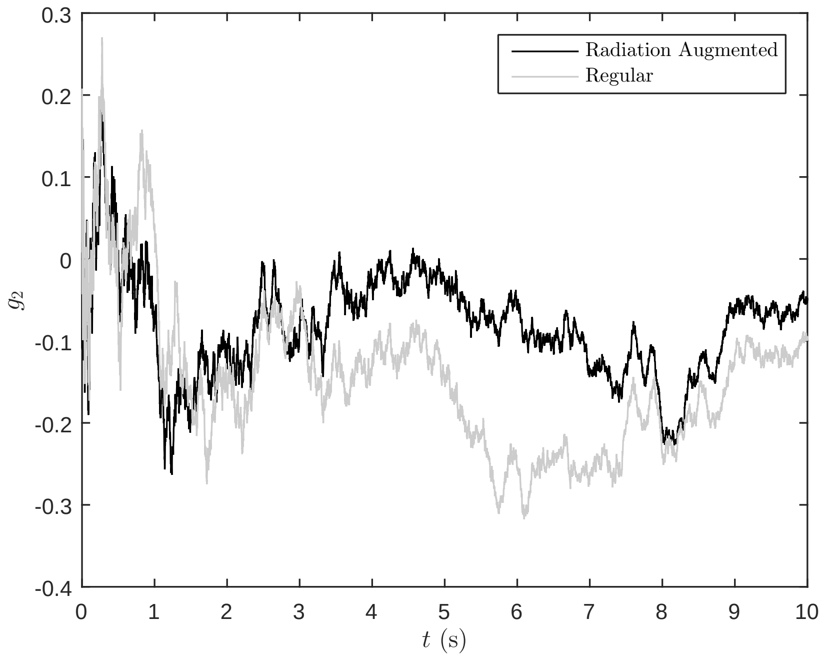

Figure 2.15: *Sample excess kurtosis of the colloidal particle position about the displacement due to gravity for the adaptive time step case with* ^{225}Pa.

The distribution of the radioactive particle about the non-radioactive particle for each particle pair is plotted with respect to the three Cartesian directions in Fig. 2.16, Fig. 2.17, and Fig. 2.18, and Cartesian direction pairs in Fig. 2.19, Fig. 2.20, and Fig. 2.21. This reinforces the excess kurtosis value of approximately zero at all times, providing a more intuitive way to visualize the randomness of the radiation ejection.

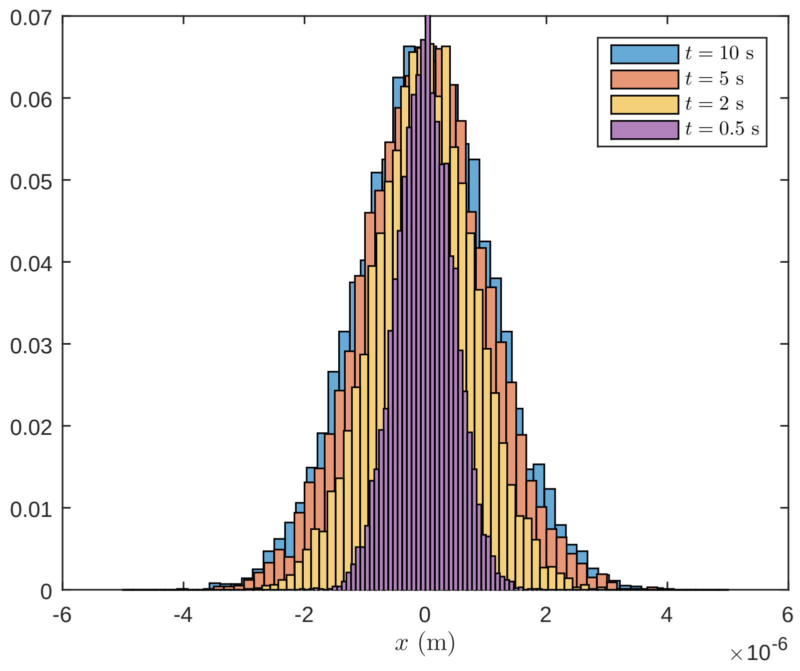

Figure 2.16: *A normalized histogram of the difference in the radioactive and non-radioactive particles' x-coordinates in a particle pair, for all particle pairs, at various times from the adaptive time step case with ^{225}Pa.*

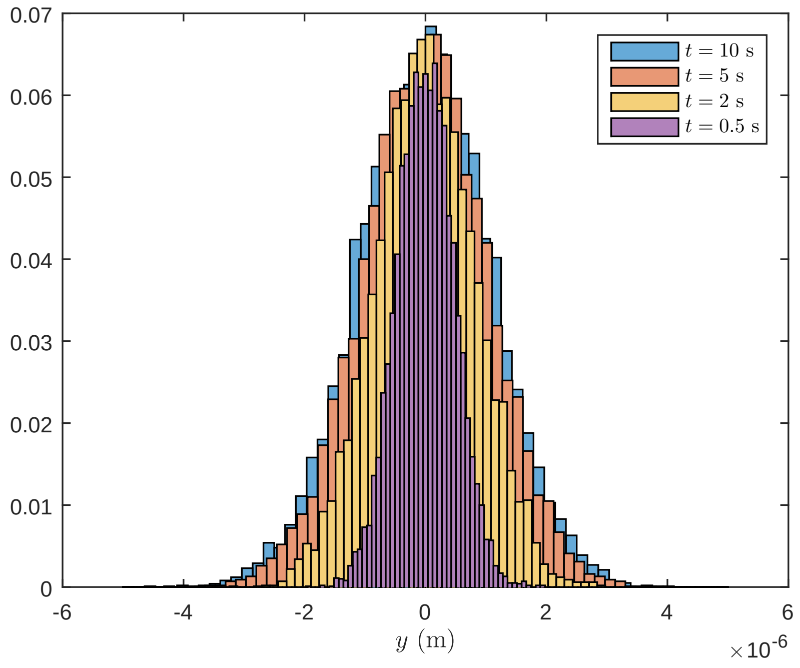

Figure 2.17: *A normalized histogram of the difference in the radioactive and non-radioactive particles' y-coordinates in a particle pair, for all particle pairs, at various times from the adaptive time step case with ^{225}Pa.*

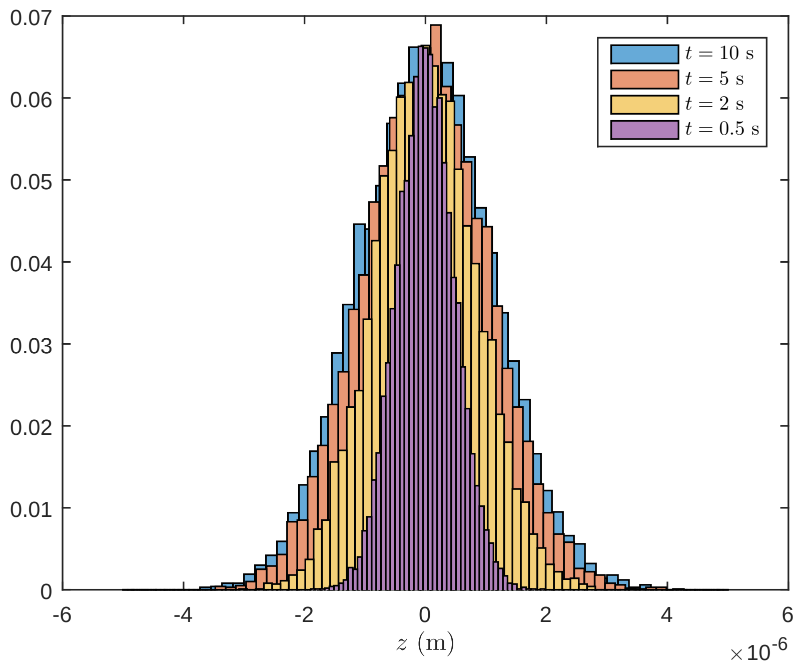

Figure 2.18: *A normalized histogram of the difference in the radioactive and non-radioactive particles' z-coordinates in a particle pair, for all particle pairs, at various times from the adaptive time step case with* ^{225}Pa.

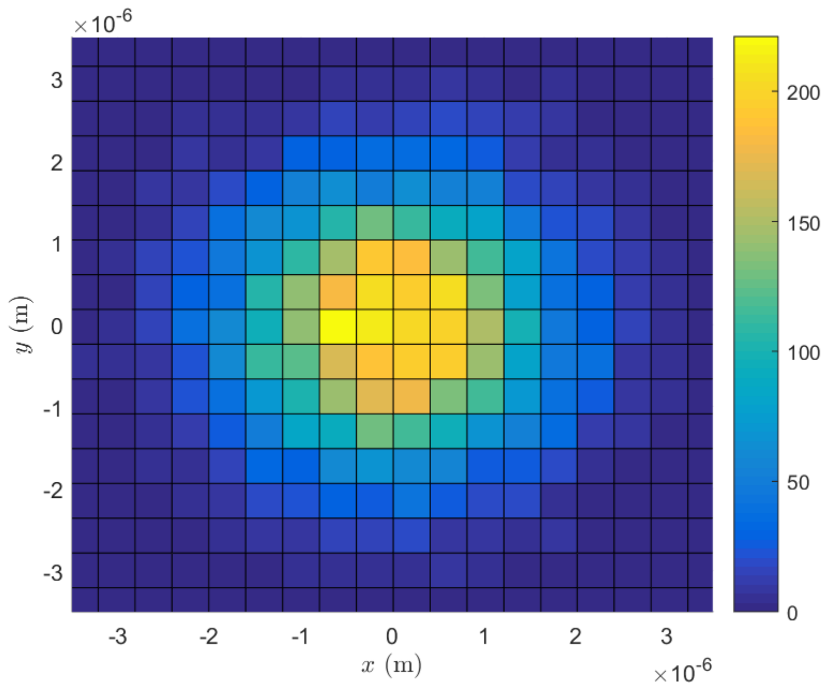

Figure 2.19: *A histogram of the difference in the radioactive and non-radioactive particles' (x, y)-coordinates in a particle pair, for all particle pairs, at $t = 10$ s from the adaptive time step case with* ^{225}Pa.

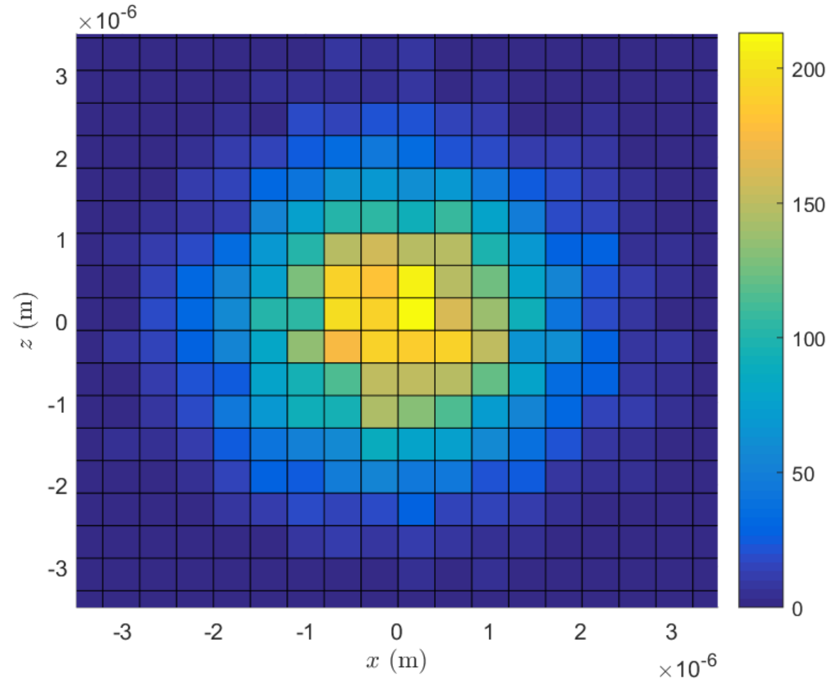

Figure 2.20: *A histogram of the difference in the radioactive and non-radioactive particles' (x, z)-coordinates in a particle pair, for all particle pairs, at $t = 10$ s from the adaptive time step case with ^{225}Pa.*

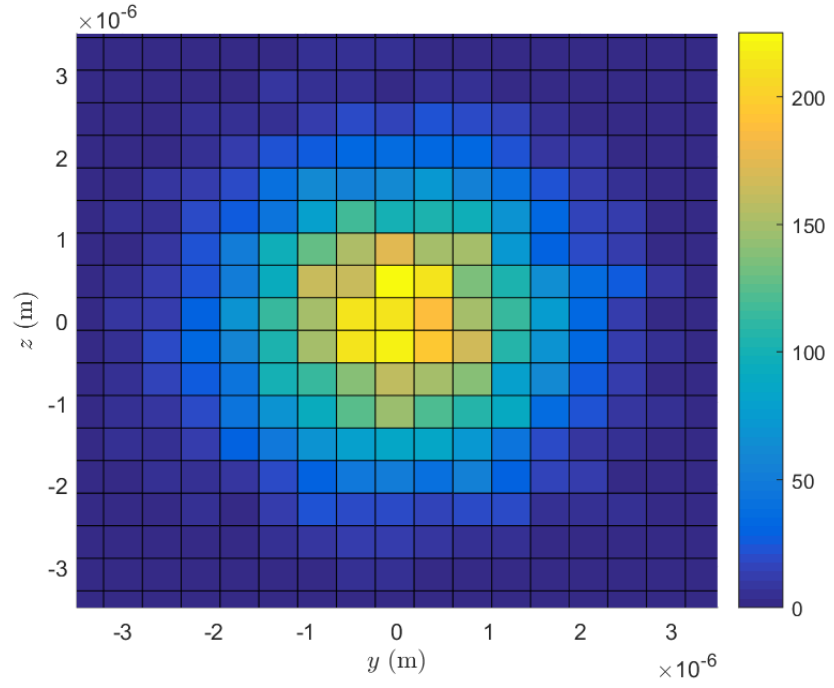

Figure 2.21: *A histogram of the difference in the radioactive and non-radioactive particles' (y, z)-coordinates in a particle pair, for all particle pairs, at $t = 10$ s from the adaptive time step case with ^{225}Pa.*

2.5 Conclusions

The motion of a colloidal particle due to the emission of radiation can be modeled as a random walk, resulting in an expression for the MSD of the particle, which exhibits a special case of anomalous diffusion. It was shown that the effect of alpha-particle ejection on the motion of a colloidal particle can be consequential, is most readily observable on timescales similar to that of the half-life, and will only be significant for short half-lives.

Results from numerical simulations have corroborated the theory. It was shown that computational time steps need to be on the order of the time between radioactive decays. Since the time between alpha decays within the colloidal particle is very short, further study with a numerical method such as molecular dynamics should be performed to understand the effect of the inertia of the fluid, transient hydrodynamic interactions between the fluid and the colloidal particle, and fluid compressibility on the motion of the colloidal particle.

There are two major challenges to be overcome for the production of a radioactive colloidal particle. The first is the very short half-life of the required radioisotopes. Gathering the radionuclides together and forming a microparticle in a fraction of a second is enormously difficult. It would still be possible to create the particle with isotopes that when exposed to certain radiation decay via alpha emission, but an extremely intense isotropic radiation field would be required. The second major challenge is preventing the disintegration of the particle as its constituent atoms decay, since the energy of the recoiling daughter isotopes is much greater than the energy of the chemical bonds holding the colloidal particle together [40]. Gold coated lanthanide phosphate particles could be created [41], or particles using any other methods of daughter isotope containment [40], but doing so would increase the mass of the particle, decreasing the magnitude of the motion induced by radiation ejection.

Chapter 3

Radioactive Janus Particles

3.1 Introduction

Named after the two-faced Roman god, Janus particles have two distinct surfaces with different properties, as shown in Fig. 3.1.

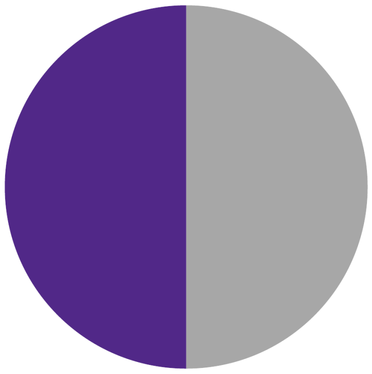

Figure 3.1: *A simplistic depiction of a Janus particle.*

The first Janus particles were created in the 1980s from glass beads, such that one side was hydrophillic, and the other hydrophobic. Since then, myriad types of Janus particles have been created with diverse applications such as electronic displays, optical probes for biological interactions or rheological measurements in small areas, and emulsifiers [42]. Self-phoretic Janus particles that propel themselves by generating gradient electric, temperature, or concentration fields have also been fabricated [2, 3]. For example, a gold-platinum Janus particle immersed in hydrogen peroxide undergoes self-electrophoresis. Peroxide oxidation occurs on the platinum end (anode) and peroxide reduction occurs on the gold end (cathode). As a result, an electrical body force within the fluid is generated along with electroosmotic slip around the particle, resulting in self-propulsion [2].

In a radioactive Janus particle, one side of the particle is radioactive and the other is not. Directed self-propulsion results if the non-radioactive side is composed of a material that is opaque the emitted radiation — a radiopaque material. Self-propelled particles are typically called "swimmer particles" or "microswimmers", and as the name implies, require the presence of a fluid medium to propel themselves. However, since radioactive Janus particles exhibit self-propulsion regardless of the medium, they are the first truly self-propelled particles.

Radioactive Janus particles hitherto have not been investigated. The purpose of the theoretical work herein is to establish, to a first approximation, the characteristics of motion of a radioactive Janus particle. A simplified mathematical model for the motion of a radioactive Janus particle is presented, and that model is utilized to derive analytical expressions for mean square displacement and excess kurtosis.

3.2 Model Description

With the inclusion of a driving force $\mathbf{F}(t)$, the Langevin equation becomes

$$m\frac{d^2\boldsymbol{r}}{dt^2} + \gamma\frac{d\boldsymbol{r}}{dt} = \mathbf{F}(t) + \boldsymbol{f}(t)\,. \tag{3.1}$$

A common assumption is to neglect the inertial term, leading to overdamped Langevin dynamics, also known as Brownian dynamics, which can be represented by [43]

$$\frac{d\boldsymbol{r}}{dt} = \frac{1}{\gamma}\left(\mathbf{F}(t) + \boldsymbol{f}(t)\right). \tag{3.2}$$

The rotational Brownian diffusion coefficient D_r is derived from the Euler-Langevin equation of motion

$$\frac{d\boldsymbol{L}}{dt} + \boldsymbol{\omega} \times \boldsymbol{L} + \zeta \boldsymbol{\omega} = \boldsymbol{g}\left(t\right), \tag{3.3}$$

where $\boldsymbol{L} = \boldsymbol{I}\boldsymbol{\omega}$ is the angular momentum, $\boldsymbol{I}$ is the inertia matrix, $\boldsymbol{\omega}$ is the angular velocity, $\zeta = 8\pi\mu a^3$ is the rotational drag coefficient for a spherical particle, and due to the equipartition theorem $\boldsymbol{g}\left(t\right)$ is a Gaussian white noise fluctuating torque such that [44, 45]

$$\langle g_i\left(t\right)\rangle = 0, \tag{3.4}$$

$$\langle g_i\left(t\right) g_j\left(t+\tau\right)\rangle = 2\zeta k_B T \delta_{ij} \delta\left(\tau\right). \tag{3.5}$$

For a spherical particle $\boldsymbol{\omega} \times \boldsymbol{L} = 0$. Thus, by analogy to the translational Langevin equation, $D_r = k_B T/\zeta$. The rotational Brownian diffusion coefficient is related to the translational Brownian diffusion coefficient by $D_0 = (4/3)a^2 D_r$ for a spherical particle.

To define the probability density function (pdf) P for the orientation of a Janus particle, which as shown later is the direction of the driving force, Fick's second law of diffusion is used [46]

$$\frac{\partial P}{\partial t} = D\nabla^2 P. \tag{3.6}$$

A rotational version of Fick's second law can be obtained by setting $r = 1$ and $\partial P/\partial r = 0$ [46]

$$\frac{1}{D_r}\frac{\partial P}{\partial t} = \frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial P}{\partial\theta}\right) + \frac{1}{\sin^2\theta}\frac{\partial^2 P}{\partial\phi^2}, \tag{3.7}$$

with $\theta = \theta(t)$ and $\phi = \phi(t)$ defined in Fig. 3.2.

The pdf can be solved for via a spherical harmonics expansion

$$P\left(\theta(t), \phi(t), t\right) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} A_l^m Y_l^m\left(\theta, \phi\right), \tag{3.8}$$

noting that Eq. 3.8 is arrived at since the solution should be non-singular over the entire

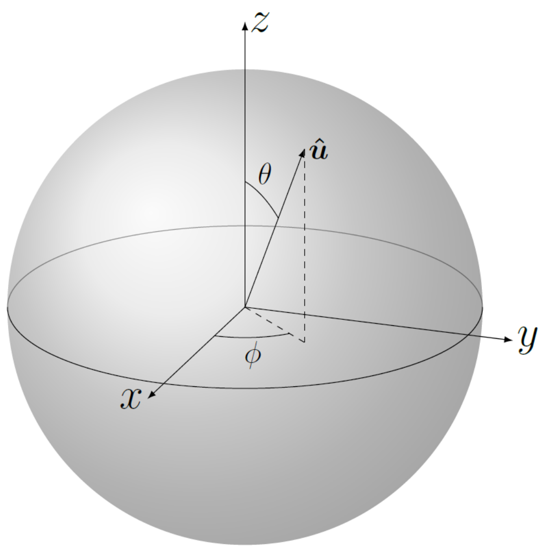

Figure 3.2: *The spherical coordinate system used in this work.*

spherical domain and 2π-periodic over the azimuthal angle. Using Eq. C.1 it can be shown that

$$\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{\partial}{\partial t}\left(A_l^m\right)Y_l^m\left(\theta,\phi\right)=-\sum_{l=0}^{\infty}\sum_{m=-l}^{l}D_r l\left(l+1\right)A_l^m Y_l^m\left(\theta,\phi\right). \tag{3.9}$$

Then, with the orthonormality condition given in Eq. C.2

$$\frac{\partial}{\partial t}\left(A_l^m\right)=-D_r l\left(l+1\right)A_l^m, \tag{3.10}$$

$$A_l^m=a_l^m e^{-l(l+1)D_r(t-t_0)}. \tag{3.11}$$

Considering an initial orientation Ω_0, the initial pdf is $P\left(\theta,\phi,t_0\right)=\delta\left(\Omega-\Omega_0\right)$. Using the closure relationship for spherical harmonics given in Eq. C.3

$$a_l^m=Y_l^{m*}\left(\theta_0,\phi_0\right), \tag{3.12}$$

which results in

$$P\left(\theta(t), \phi(t), t\right) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} e^{-l(l+1)D_r(t-t_0)} Y_l^{m*}\left(\theta_0, \phi_0\right) Y_l^m\left(\theta, \phi\right). \tag{3.13}$$

The driving force is given by $\mathbf{F}\left(t\right) = F\left(t\right)\hat{\boldsymbol{u}}$ where $\hat{\boldsymbol{u}} = \left(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta\right)$ and θ and ϕ are taken from Eq. 3.13. In the case of a radioactive colloidal particle, the force from the emission of one alpha particle is

$$F'\left(t\right) = \frac{dp}{dt} = M_0 A\left(t\right) = \lambda M_0 N_0 e^{-\lambda t}. \tag{3.14}$$

To simplify the following analysis, the force is averaged over all 4π steradians of possible emission directions, and only alpha-particle ejections in 2π steradians contribute any force. Approximating the radiation ejection to occur only in directions normal to the surface of the microparticle allows one to find the average force vector for a radioactive Janus particle using the center of mass of a hemispherical shell with a radius of $F'\left(t\right)$. Thus, the average force is aligned with the particle's axis of symmetry, in the direction of the radioactive side, with a magnitude of $F\left(t\right) = (1/2)\lambda M_0 N_0 e^{-\lambda t}$. If the radioactive spherical cap of the Janus particle is not a hemisphere, the factor $(1/2)$ in $F\left(t\right)$ should be replaced with $\cos^2\left(\theta_{rad}/2\right)$, where θ_{rad} is the polar extent of the radioactive spherical cap.

The actual average force magnitude would be slightly less, as the alpha particle emission direction does not have to be normal to the surface of the colloidal particle. Additionally, alpha-particle attenuation by the radioactive material is neglected, the escape of recoiling daughter isotopes is ignored, the angular impulse from the ejection of alpha particles is disregarded, and it is assumed that the non-radioactive material stops all incident alpha particles. Daughter product decays and heating of the fluid by the alpha particles are ignored as well. These simplifications lead to analytical expressions for the mean square displacement of a radioactive Janus particle and for the excess kurtosis of the probability distribution of a radioactive Janus particle's position.

3.3 Mean Square Displacement

Integrating Eq. 3.2, one obtains [47]

$$\boldsymbol{r}(t) - \boldsymbol{r}_0 = \frac{1}{\gamma}\left[\int_0^t F(t_1)\,\hat{\boldsymbol{u}}(t_1)\,dt_1 + \int_0^t \boldsymbol{f}(t_1)\,dt_1\right]. \tag{3.15}$$

Squaring and taking the ensemble average [47]

$$\begin{aligned}\langle(\boldsymbol{r}(t) - \boldsymbol{r}_0)^2\rangle = \frac{1}{\gamma^2}\Bigg[&\int_0^t\int_0^t \langle F(t_1)\,\hat{\boldsymbol{u}}(t_1)\,F(t_2)\,\hat{\boldsymbol{u}}(t_2)\rangle dt_1 dt_2 \\ &+ 2\int_0^t\int_0^t \langle F(t_1)\,\hat{\boldsymbol{u}}(t_1)\,\boldsymbol{f}(t_2)\rangle dt_1 dt_2 + \int_0^t\int_0^t \langle \boldsymbol{f}(t_1)\,\boldsymbol{f}(t_2)\rangle dt_1 dt_2\Bigg].\end{aligned} \tag{3.16}$$

For a non-probabilistic driving force magnitude, and using Eq. 2.2 and Eq. 2.3, the MSD becomes

$$\langle(\boldsymbol{r}(t) - \boldsymbol{r}_0)^2\rangle = \frac{1}{\gamma^2}\left[2!\int_0^t\int_0^{t_1} F(t_1)\,F(t_2)\,\langle\hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\rangle_{t_1>t_2} dt_2 dt_1 + 6\gamma k_B T t\right]. \tag{3.17}$$

The ensemble average of $\hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)$ can be expanded to obtain

$$\begin{aligned}\langle\hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\rangle_{t_1>t_2} &= \langle\sin\theta_1\cos\phi_1\sin\theta_2\cos\phi_2\rangle_{t_1>t_2} \\ &\quad + \langle\sin\theta_1\sin\phi_1\sin\theta_2\sin\phi_2\rangle_{t_1>t_2} + \langle\cos\theta_1\cos\theta_2\rangle_{t_1>t_2}.\end{aligned} \tag{3.18}$$

Finally, following the derivation in the first section of Appendix D, the mean square displacement is calculated using Eq. 3.17

$$\begin{aligned}&\langle(\boldsymbol{r}(t) - \boldsymbol{r}_0)^2\rangle \\ &= \frac{1}{\gamma^2}\left(\frac{(\lambda M_0 N_0)^2}{2(\lambda - 2D_r)}\left[\frac{1}{\lambda + 2D_r}\left(1 - e^{-(\lambda+2D_r)t}\right) - \frac{1}{2\lambda}\left(1 - e^{-2\lambda t}\right)\right] + 6\gamma k_B T t\right) \\ &= \left(\frac{\lambda M_0 N_0}{\gamma}\right)^2 \frac{1}{2(\lambda - 2D_r)}\left[\frac{1}{\lambda + 2D_r}\left(1 - e^{-(\lambda+2D_r)t}\right) - \frac{1}{2\lambda}\left(1 - e^{-2\lambda t}\right)\right] + 6D_0 t.\end{aligned} \tag{3.19}$$

In the following figures, the MSDs of a regular particle, a radioactive particle, and a radioactive Janus particle are compared. The plots were made with the script in Appendix E. For the regular particle none of the constituent atoms are considered radioactive, for the radioactive particle all of the constituent atoms are considered radioactive, and for the radioactive Janus particle half of the constituent atoms are considered radioactive (one hemisphere radioactive, the other radiopaque). The particles all have a radius of 0.5 μm. Particles composed of the four materials listed in Table 3.1 were simulated to cover a wide range of half lives. As shown, the radioactive Janus particle will move very far before approaching the same rate of motion as a regular particle.

Table 3.1: *Alpha-decaying radioisotopes used [28].*

Isotope	^{225}Pa	^{224}Ra	^{225}Ac	^{210}Po
ρ (g/cm^3)	15.37	5.0	10.07	9.32
$T_{1/2}$	1.7 s	3.63 d	10.0 d	138.38 d
E_α (MeV)	7.22	5.6855	5.829	5.3044

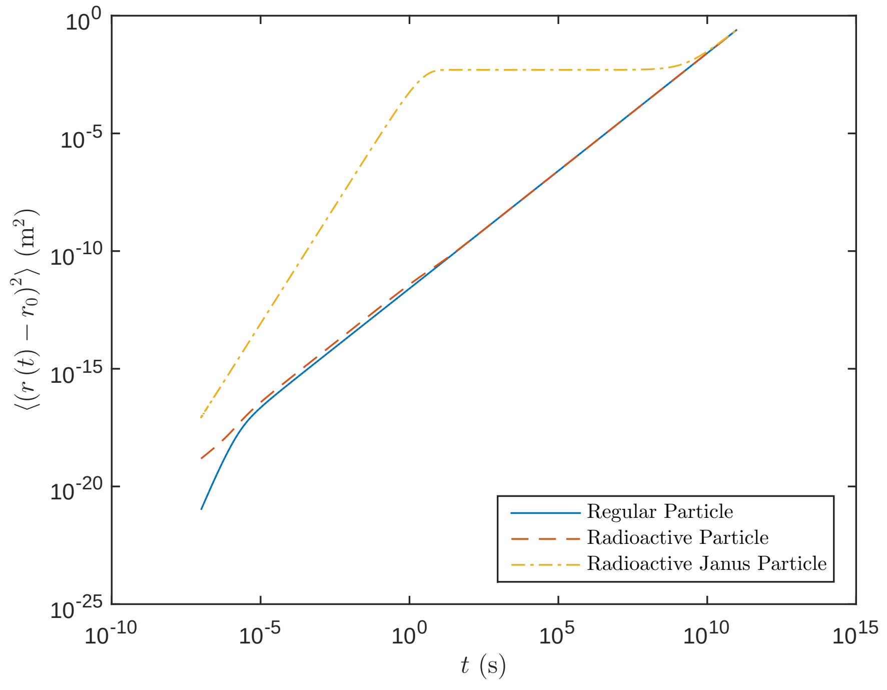

Figure 3.3: *Mean square displacements of three particle types with ^{225}Pa.*

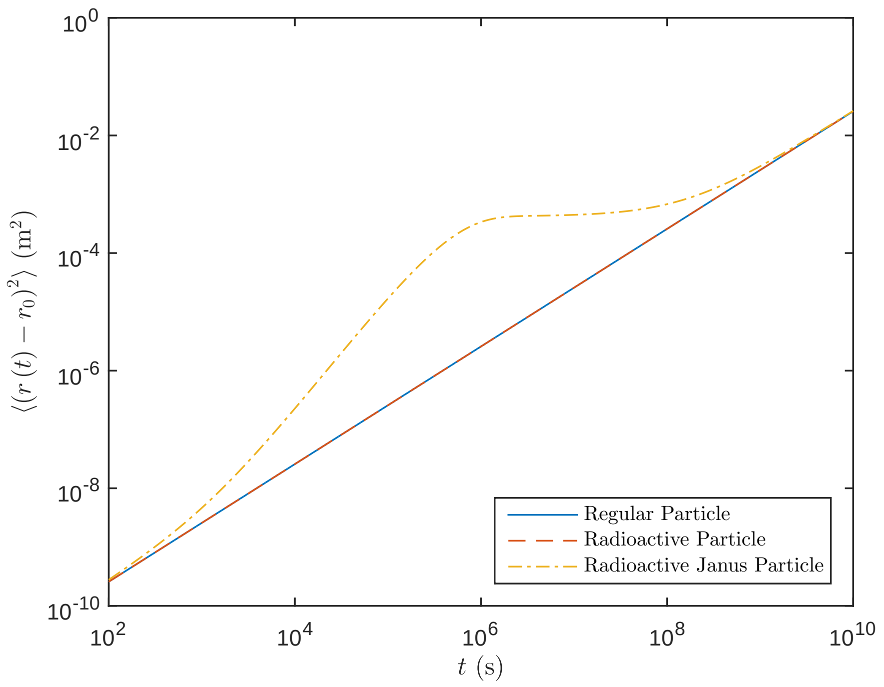

Figure 3.4: *Mean square displacements of three particle types with ^{224}Ra.*

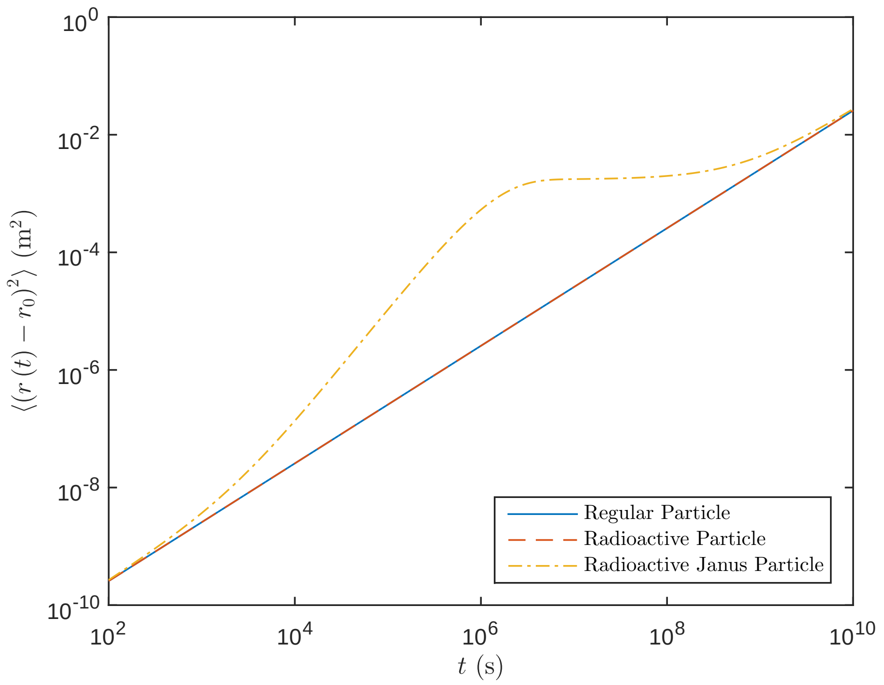

Figure 3.5: *Mean square displacements of three particle types with ^{225}Ac.*

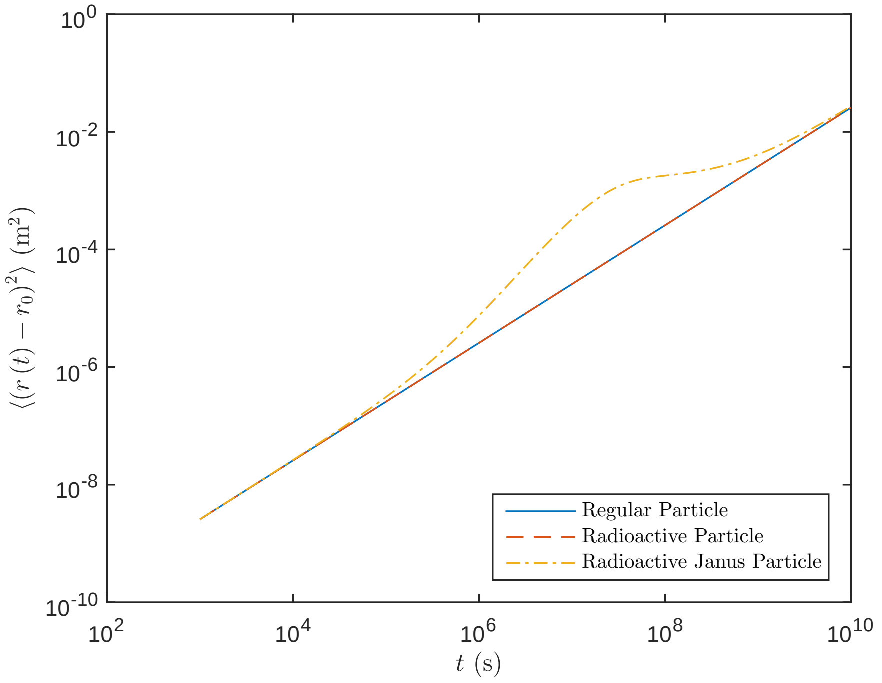

Figure 3.6: *Mean square displacements of three particle types with ^{210}Po.*

3.4 Excess Kurtosis

Non-Gaussian behavior is shown with nonzero values of skewness S and excess kurtosis γ_2, given by

$$S = \frac{\langle (\Delta \boldsymbol{r} - \langle \Delta \boldsymbol{r} \rangle)^3 \rangle}{\langle (\Delta \boldsymbol{r} - \langle \Delta \boldsymbol{r} \rangle)^2 \rangle^{3/2}}, \tag{3.20}$$

$$\gamma_2 = \frac{\langle (\Delta \boldsymbol{r} - \langle \Delta \boldsymbol{r} \rangle)^4 \rangle}{\langle (\Delta \boldsymbol{r} - \langle \Delta \boldsymbol{r} \rangle)^2 \rangle^2} - 15, \tag{3.21}$$

where $\Delta \boldsymbol{r} = \boldsymbol{r}(t) - \boldsymbol{r}_0$ is used for conciseness.

To make to following results readily applicable to laboratory experiments, additional averaging over the initial particle orientation $\hat{\boldsymbol{u}}_0$ is performed after the ensemble average, denoted by $\langle \cdots \rangle_{\hat{\boldsymbol{u}}_0}$. This results in $\langle \Delta \boldsymbol{r} \rangle_{\hat{\boldsymbol{u}}_0} = 0$, $\langle (\Delta \boldsymbol{r})^2 \rangle_{\hat{\boldsymbol{u}}_0} = \langle (\Delta \boldsymbol{r})^2 \rangle$, and $S = 0$. The excess kurtosis, however, is nonzero and is calculated hereafter with [48]

$$\gamma_2 = \frac{\langle (\Delta \boldsymbol{r})^4 \rangle_{\hat{\boldsymbol{u}}_0}}{\langle (\Delta \boldsymbol{r})^2 \rangle_{\hat{\boldsymbol{u}}_0}^2} - 15. \tag{3.22}$$

For an ensemble of particles starting at the same initial position with a random initial orientation, a probability density function describes the possible locations of the particles after a given time. The pdf is a normal distribution if the behavior of the particles is Gaussian. A distribution with positive excess kurtosis is called leptokurtic, and has fatter tails than a normal distribution. A distribution with negative excess kurtosis is called platykurtic, and has thinner tails than a normal distribution.

Taking Eq. 3.15 to the fourth power and then ensemble averaging results in the following nonzero terms [47]

$$\begin{aligned}\langle(\Delta\boldsymbol{r})^4\rangle_{\hat{\boldsymbol{u}}_0} \\ &= \frac{1}{\gamma^4}\int_0^t\int_0^t\int_0^t\int_0^t \Big[\langle F(t_1)\,\hat{\boldsymbol{u}}(t_1)\,F(t_2)\,\hat{\boldsymbol{u}}(t_2)\,F(t_3)\,\hat{\boldsymbol{u}}(t_3)\,F(t_4)\,\hat{\boldsymbol{u}}(t_4)\rangle_{\hat{\boldsymbol{u}}_0} \\ &\qquad + 6\langle F(t_1)\,\hat{\boldsymbol{u}}(t_1)\,F(t_2)\,\hat{\boldsymbol{u}}(t_2)\rangle\langle \boldsymbol{f}(t_3)\,\boldsymbol{f}(t_4)\rangle_{\hat{\boldsymbol{u}}_0} \\ &\qquad + \langle \boldsymbol{f}(t_1)\,\boldsymbol{f}(t_2)\,\boldsymbol{f}(t_3)\,\boldsymbol{f}(t_4)\rangle_{\hat{\boldsymbol{u}}_0}\Big]dt_1dt_2dt_3dt_4. \quad (3.23)\end{aligned}$$

For a non-probabilistic driving force magnitude, and using Eq. D.5 and Eq. 2.3, the excess kurtosis becomes

$$\begin{aligned}\langle(\Delta\boldsymbol{r})^4\rangle_{\hat{\boldsymbol{u}}_0} = \frac{15}{\gamma^4}\Bigg[&4!\int_0^t\int_0^{t_1}\int_0^{t_2}\int_0^{t_3} F(t_1)\,F(t_2)\,F(t_3)\,F(t_4) \\ &\times \langle \hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\,\hat{\boldsymbol{u}}(t_3)\,\hat{\boldsymbol{u}}(t_4)\rangle_{\hat{\boldsymbol{u}}_0,t_1>t_2>t_3>t_4}dt_4dt_3dt_2dt_1 + (\lambda M_0N_0)^2\,\frac{18\gamma k_BTt}{\lambda-2D_r} \\ &\times\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{2\lambda}\left(1-e^{-2\lambda t}\right)\right]+(6\gamma k_BTt)^2\Bigg], \quad (3.24)\end{aligned}$$

where the ensemble average can be expanded as in Eq. 3.18, and the factor of 15 comes about since this is a three-dimensional problem.

Finally, following the derivation in the second section of Appendix D, the excess kurtosis is calculated using Eq. 3.24

$$\begin{aligned}\gamma_2 =\Bigg\{ &\left(\frac{\lambda M_0N_0}{\gamma}\right)^4 \frac{15}{\lambda-2D_r}\Bigg[\frac{2}{5}\Bigg(\frac{1}{\lambda-4D_r}\Bigg\{\frac{1}{\lambda+4D_r}\Bigg[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right) \\ &-\frac{1}{2\lambda+6D_r}\left(1-e^{-(2\lambda+6D_r)t}\right)\Bigg]-\frac{1}{2\lambda}\Bigg[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right) \\ &-\frac{1}{3\lambda+2D_r}\left(1-e^{-(3\lambda+2D_r)t}\right)\Bigg]\Bigg\}-\frac{1}{2\lambda-6D_r}\Bigg\{\frac{1}{\lambda+4D_r} \\ &\times\Bigg[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{2\lambda+6D_r}\left(1-e^{-(2\lambda+6D_r)t}\right)\Bigg] \\ &-\frac{1}{3\lambda-2D_r}\Bigg[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{4\lambda}\left(1-e^{-4\lambda t}\right)\Bigg]\Bigg\}\Bigg)\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{2}\left(\frac{1}{\lambda+2D_r}\left\{\frac{1}{\lambda-2D_r}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{2\lambda}\left(1-e^{-2\lambda t}\right)\right]\right.\right.\\
&\left.-\frac{1}{2\lambda}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{3\lambda+2D_r}\left(1-e^{-(3\lambda+2D_r)t}\right)\right]\right\}\\
&-\frac{1}{2\lambda}\left\{\frac{1}{\lambda-2D_r}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{2\lambda}\left(1-e^{-2\lambda t}\right)\right]\right.\\
&\left.\left.\left.-\frac{1}{3\lambda-2D_r}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{4\lambda}\left(1-e^{-4\lambda t}\right)\right]\right\}\right)\right]\\
&+\left(\frac{\lambda M_0N_0}{\gamma}\right)^2\frac{18D_0t}{\lambda-2D_r}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{2\lambda}\left(1-e^{-2\lambda t}\right)\right]\\
&\left.+\left(6D_0t\right)^2\right\}\left\{\left(\frac{\lambda M_0N_0}{\gamma}\right)^2\frac{1}{2\left(\lambda-2D_r\right)}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)\right.\right.\\
&\left.\left.-\frac{1}{2\lambda}\left(1-e^{-2\lambda t}\right)\right]+6D_0t\right\}^{-2}-15. \qquad (3.25)
\end{aligned}
$$

In the following figures, the excess kurtosis of radioactive Janus particles are compared using the radionuclides from the previous section. The plots were made with the script in Appendix E. The excess kurtosis is the solid line, plotted against the left axis, and the time derivative of the MSD is the dashed line, plotted against the right axis. Although the excess kurtosis is always 0 at time $t = 0$, numerical instabilities prevent the excess kurtosis at such short times from being plotted for radioisotopes with shorter half-lives. The transitions away from and back to the Gaussian value of 0 occur at the same time that the MSD of a radioactive Janus particle diverges from and converges back to that of a regular particle. The distribution of the position of a radioactive Janus particle is seen to become leptokurtic during the transitions, and platykurtic in between them. The time derivative of the MSD peaks in the center of the platykurtic region. For a Janus particle composed of a radioisotope with a short half-life, the excess kurtosis approaches its lower bound at $\gamma_2 = \beta_2 - d(d+2) \geq d^2 - d(d+2) = -6$ [37, 39]. Although the excess kurtosis in the following figures peaks near 14, there is no upper bound on the excess kurtosis.

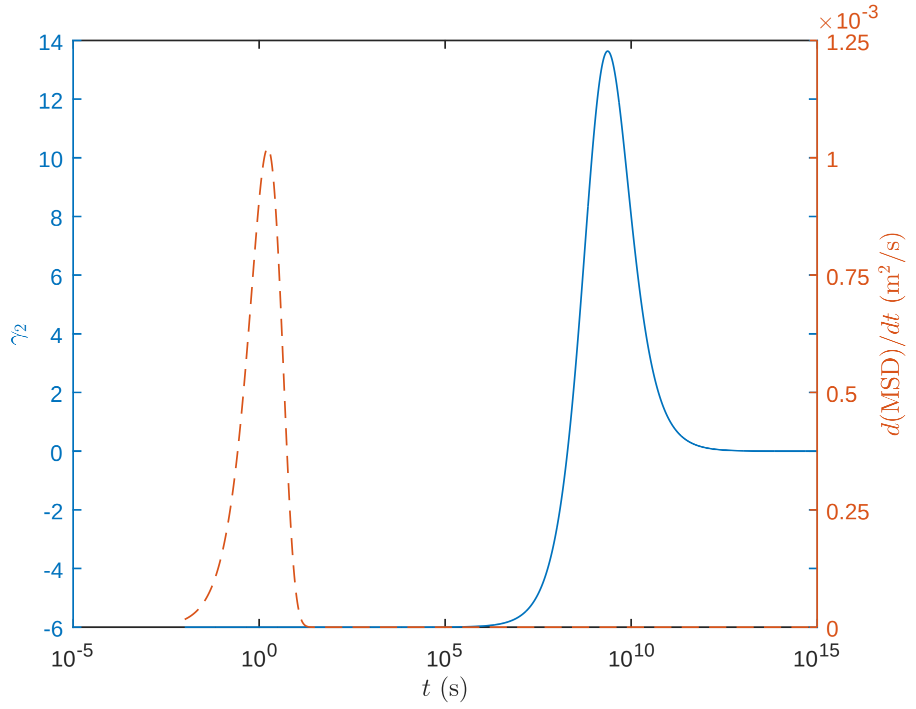

Figure 3.7: *Excess kurtosis (solid line, left axis) and MSD time derivative (dashed line, right axis) of a radioactive Janus particle composed of* ^{225}*Pa.*

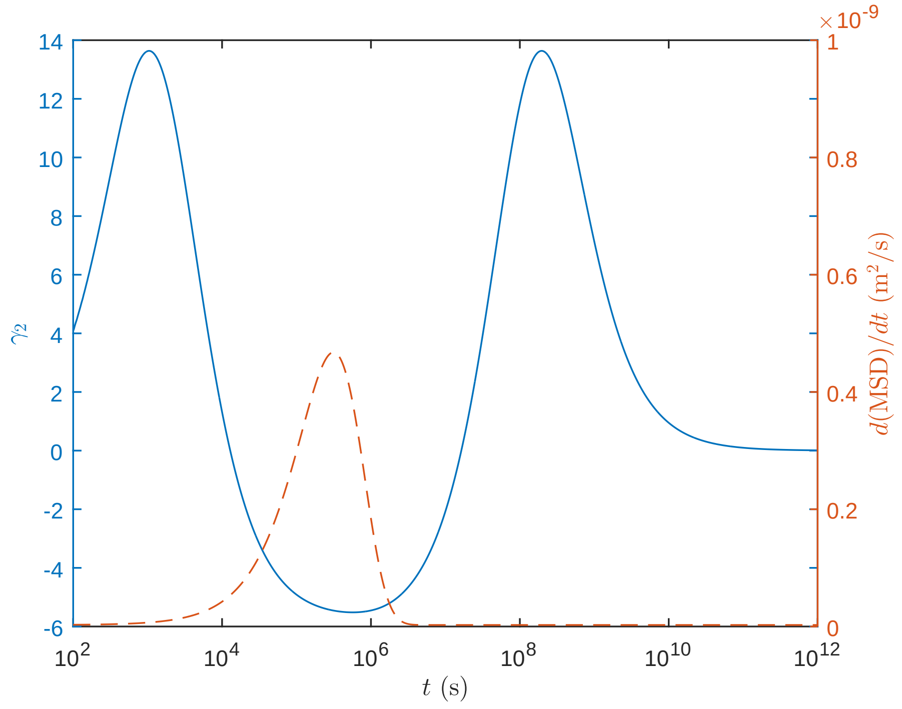

Figure 3.8: *Excess kurtosis (solid line, left axis) and MSD time derivative (dashed line, right axis) of a radioactive Janus particle composed of* ^{224}*Ra.*

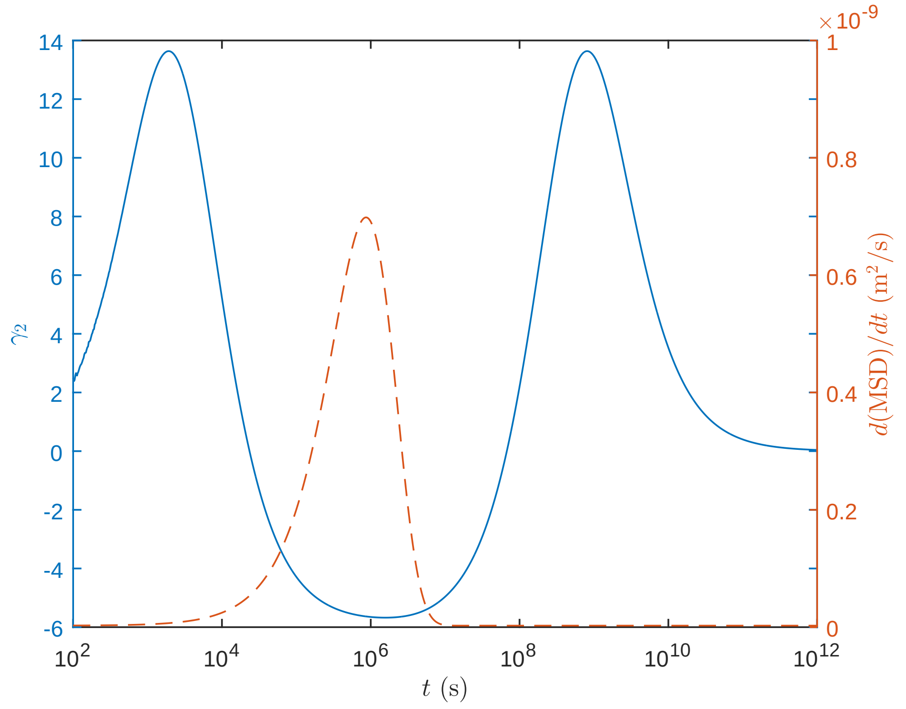

Figure 3.9: *Excess kurtosis (solid line, left axis) and MSD time derivative (dashed line, right axis) of a radioactive Janus particle composed of* ^{225}*Ac.*

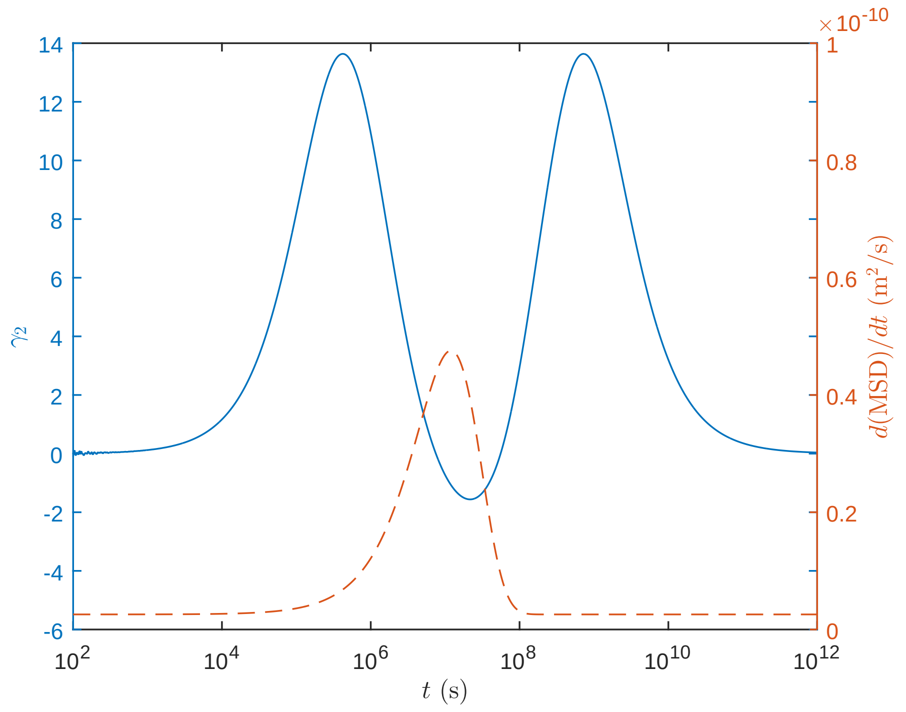

Figure 3.10: *Excess kurtosis (solid line, left axis) and MSD time derivative (dashed line, right axis) of a radioactive Janus particle composed of* ^{210}*Po.*

3.5 Conclusions

As in the case of a radioactive colloidal particle, radiation emission can substantially change the motion of a radioactive Janus particle, although radionuclides with short half-lives are required. In this case, however, half-lives on the order of one week are viable to experimentally observe an increased MSD (see Fig. 3.4 – ^{224}Ra and Fig. 3.5 – ^{225}Ac). The MSD increases greatly until around the half-life of the particle, at which point it stagnates as self-phoresis stops and through Brownian motion the particles diffuse back towards and farther from the initial location. Eventually, the particles evenly diffuse through all of the space between the start and end points of the self-phoretic motion and the MSD increases to align with that of a regular particle. The excess kurtosis becomes non-Gaussian when the MSD differs from that of a regular particle.

In addition to the two major challenges for production of a radioactive colloidal particle, a radioactive Janus particle has two additional complexities. The first is finding an extremely radiopaque material that can stop or significantly slow the decay radiation over a distance equal to the radius of the microparticle. The second is constructing the Janus particle, somehow bonding the two disparate hemispheres. Although much more development is required before a radioactive Janus particle can be made, it would be worthwhile to numerically simulate a radioactive Janus particle without the assumptions listed at the end of Section 3.2 in order to ascertain the propulsive capabilities of the radiation emission.

Chapter 4

Conclusions and Future Work

An overview of the motion of colloidal particles due to the emission of radiation has been presented. It was shown that the effect of alpha-particle ejection on the motion of a microparticle can be significant, although radionuclides with very short half-lives are needed. A theory was developed to describe the motion of a radioactive colloidal particle, and results from numerical simulations have corroborated the theory, showing a special case of anomalous diffusion in which the MSD exponentially decays towards a linear relationship with time. It was shown that computational time steps need to be on the order of the time between radioactive decays, as for radionuclides with very short half-lives the numerical solution does not converge to the analytical solution if the time steps are too large.

A second theory was developed to describe the motion of a radioactive Janus particle. This theory also showed a special case of anomalous diffusion like the first theory. However, the motion of radioactive colloidal particles is Gaussian with a time-dependent variance, whereas the motion of radioactive Janus particles is non-Gaussian. Additionally, unlike uniform radioactive particles, radioactive Janus particles can be constructed with radioisotopes with sufficiently long half-lives to produce an experimentally viable difference in MSD as compared to non-radioactive particles.

As work on the motion of radioactive colloidal particles is still in an embryonic stage, there is much future work that can be performed. To start, the numerical simulations should be improved to include rotational motion due to Brownian forces, the angular impulse from the ejection of alpha particles, recoiling daughter isotopes (including the reduction in daughter isotope energy as the chemical bonds holding the isotope in place are broken), the subsequent disintegration of the microparticle, heating of the fluid by the alpha particles, and daughter product decays.

To modify the existing particle simulation code in Appendix A to simulate Janus particles, the following changes should be executed. The particle mass `m_p`, initial number of radioactive atoms `N_0`, and total stopping power `dEdx` would need to be changed. Rotational Brownian motion can be included with the Euler-Langevin equation, which would need to be modified to include an additional radiation-induced angular impulse term obtained from simulating individual alpha particles. From the SDE form of the modified Euler-Langevin equation, the angular velocity and then orientation of the Janus particle can be found. Each Janus particle simulated would need a random initial orientation. The author suggests the use of quaternions for orientation representation since they are quick for chaining multiple rotations and yet do not suffer from gimbal lock. The orientation needs to be known when computing each alpha particle's path and energy loss. Finally, a different sampling method such as rejection sampling is required for the alpha decay location based on the geometry of the Janus particle.

In addition to the Langevin dynamics simulation, a study could be performed with a Stokesian dynamics code to quantify the effect of fluid inertia and transient hydrodynamic interactions between the fluid and the colloidal particle. A molecular dynamics code could be used to determine the effect of fluid compressibility. If the new simulation results differ greatly from the current theory, a refined theory would need to be developed. If the new simulation results predict motion that could be measurable in a laboratory setting, then the fabrication of radioactive microparticles would need to be investigated, overcoming the challenges described in previous chapters. Once radioactive colloidal particles can be made, experiments can be carried out to test the validity of the theory and simulation results.

An interesting application of radioactive colloidal particles is in complex or dusty plasmas, where already the microparticles in a nuclear-induced complex plasma can become radioactive [49]. Complex plasmas consist of highly charged microparticles within a weakly ionized gas (plasma) and exhibit undamped particle dynamics [50]. A complex plasma containing the radioactive microparticles discussed in this work could be very useful for experimentally observing alpha-induced motion, although a model of the motion of a microparticle in plasma would need to be developed.

Bibliography

[1] R. Brown and J. J. Bennett, *The miscellaneous botanical works of Robert Brown*, vol. 1 (Published for the Ray Society by R. Hardwicke, London, 1866).

[2] J. L. Moran and J. D. Posner, Annu. Rev. Fluid Mech. **49**, 511 (2017).

[3] K. Kroy, D. Chakraborty, and F. Cichos, Eur. Phys. J. Special Topics **225**, 2207 (2016).

[4] W. Zhang, Z. Liu, Y. Yang, and S. Du, Applied Radiation and Isotopes **114**, 14 (2016).

[5] J. Chapman, Tech. Rep. NF1676L-12188, NASA Langley Research Center, Hampton, VA (2011).

[6] J. I. Castor, *Radiation Hydrodynamics* (Lawrence Livermore National Laboratory, 2003), 17th ed.

[7] P. J. Coelho, Journal of Heat Transfer **134** (2012).

[8] *Advancing Nuclear Medicine Through Innovation* (The National Academies Press, 2007).

[9] J. Elgqvist, S. Frost, J. Pouget, and P. Albertsson, Frontiers in Oncology **3** (2014).

[10] M. Hoover, D. Myers, L. J. Cash, R. Guilmette, W. Kreyling, G. Oberdorster, and R. Smith, Tech. Rep. 176, National Council on Radiation Protection and Measurements, 7910 Woodmont Avenue, Suite 400 / Bethesda, MD 20814-3095 (2017).

[11] G. Wilson, A. A. Bahadori, and H. Bindra (2019), Unpublished manuscript.

[12] G. G. Stokes, *On the effect of the internal friction of fluids on the motion of pendulums*, vol. 9 of *Transactions of the Cambridge Philosophical Society* (Cambridge University Press, 1856).

[13] A. Einstein, Annalen der Physik **322**, 549 (1905).

[14] G. E. Uhlenbeck and L. S. Ornstein, Physical Review **36**, 823 (1930).

[15] R. Metzler, J. Jeon, A. G. Cherstvy, and E. Barkai, Phys. Chem. Chem. Phys. **16** (2014).

[16] G. Wilson, A. A. Bahadori, and H. Bindra, in *Bulletin of the American Physical Society* (2018).

[17] W. B. Russel, D. A. Saville, and W. R. Schowalter, *Colloidal Dispersions* (Cambridge University Press, 1999).

[18] C. E. Brennen, Tech. Rep. CR 82.010, Naval Civil Engineering Laboratory, Port Hueneme, CA 93043 (1982).

[19] M. Abundo, Scientiae Mathematicae Japonicae Online **e-2013**, 719 (2013).

[20] D. J. Higham, Society for Industrial and Applied Mathematics **43**, 525 (2001).

[21] J. K. Shultis and R. E. Faw, *Fundamentals of Nuclear Science and Engineering* (CRC Press, 2008), 2nd ed.

[22] R. Metzler and J. Klafter, Physics Reports **339**, 1 (2000).

[23] C. Yuan and X. Mao, Stochastic Analysis and Applications **22**, 1133 (2004).

[24] E. W. Weisstein, *Ball point picking* (2017), From MathWorld — A Wolfram Web Resource. http://mathworld.wolfram.com/BallPointPicking.html.

[25] E. W. Weisstein, *Sphere point picking* (2017), From MathWorld — A Wolfram Web Resource. http://mathworld.wolfram.com/SpherePointPicking.html.

[26] G. S. Kell, Journal of Chemical and Engineering Data **20**, 97 (1975).

[27] U. Onken, J. Rarey-Nies, and J. Gmehling, International Journal of Thermophysics **10** (1989).

[28] E. M. Baum, M. C. Ernesti, H. D. Knox, T. R. Miller, and A. M. Watson, *Nuclides and Isotopes Chart of the Nuclides* (Bechtel Marine Propulsion Corporation, 2009), 17th ed.

[29] S. C. Wu, Nuclear Data Sheets **110**, 681 (2009).

[30] A. K. Jain and B. Singh, Nuclear Data Sheets **107**, 1027 (2006).

[31] A. K. Jain, S. Singh, S. Kumar, and J. K. Tuli, Nuclear Data Sheets **108**, 883 (2007).

[32] J. F. Ziegler, M. D. Ziegler, and J. P. Biersack, Nuclear Instruments and Methods in Physics Research B **268**, 1818 (2010).

[33] X. Bian, C. Kim, and G. E. Karniadakis, Soft Matter **12**, 6331 (2016).

[34] D. W. Hahn and M. N. Özişik, *Heat Conduction* (John Wiley & Sons, Inc., 2012), 3rd ed.

[35] D. S. Chandrasekharaiah, Applied Mechanics Reviews **39**, 355 (1986).

[36] N. Tsoulfanidis and S. Landsberger, *Measurement and Detection of Radiation* (CRC Press, 2015), 4th ed.

[37] K. V. Mardia, Biometrika **57**, 519 (1970).

[38] K. Koizumi, T. Sumikawa, and T. Pavlenko, SUT Journal of Mathematics **50**, 483 (2014).

[39] K. V. Mardia, Sankhyā: The Indian Journal of Statistics **36**, 115 (1974).

[40] R. M. de Kruijff, H. T. Wolterbeek, and A. G. Denkova, Pharmaceuticals **8**, 321 (2015).

[41] M. F. McLaughlin, J. Woodward, R. A. Boll, J. S. Wall, A. J. Rondinone, S. J. Kennel, S. Mirzadeh, and J. D. Robertson, PLOS One **8** (2013).

[42] A. Walther and A. H. E. Müller, Soft Matter **4**, 663 (2008).

[43] B. ten Hagen, S. van Teeffelen, and H. Löwen, J. Phys.: Condens. Matter **23** (2011).

[44] G. W. Ford, J. T. Lewis, and J. McConnell, Physical Review A **19**, 907 (1979).

[45] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, vol. 6 of *Course of Theoretical Physics* (Pergamon Press, 1987), 2nd ed.

[46] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical Methods for Physics and Engineering* (Cambridge University Press, 2006), 3rd ed.

[47] B. ten Hagen, S. van Teeffelen, and H. Löwen, Condens. Matter Phys. **12** (2009).

[48] X. Zheng, B. ten Hagen, A. Kaiser, M. Wu, H. Cui, Z. Silber-Li, and H. Löwen, Phys. Rev. E **88** (2013).

[49] J. Winter, V. E. Fortov, and A. P. Nefedov, Journal of Nuclear Materials **290–293**, 509 (2001).

[50] M. Chaudhuri, A. V. Ivlev, S. A. Khrapak, H. M. Thomas, and G. E. Morfill, Soft Matter **7**, 1287 (2011).

Appendix A

Particle Simulation Code

The codes used to simulate particle movement are listed herein. There are two variants – the first using set time steps and the second using adaptive time steps.

A.1 Set Time Steps

Below is the C++ code used for simulating particles with set time steps. It is configured for ^{225}Pa, but by reading the comments can be easily changed to any of the other three isotopes used in this work.

Listing A.1: *Particle simulation code with set time steps - HPC2.cpp.*

```
#define _USE_MATH_DEFINES
#include <cmath>
#include <limits>
typedef std::numeric_limits< double > dbl;
#include <iostream>
#include <fstream>
#include <random>
#include <string>

int main(int argc, char *argv[])
{
    // Arguments
    if (argc != 3) // name of program is one argument
    {
        std::cout << "Two arguments required." << std::endl;
        return 1;
    }
    unsigned int number = atof(argv[1]);

    // RNG Configuration
    std::random_device rd; // random seed
    std::mt19937 generator(rd());
    std::normal_distribution<double> standardnormalrnd(0.0, 1.0);  // mean = 0.0,
       ↪ stddev = 1.0
    std::exponential_distribution<double> exponentialrnd(1.0); // mean = 1.0,
       ↪ lambda = 1 / mean
    std::uniform_real_distribution<double> uniformrnd(0.0, 1.0); // min (inclusive
       ↪ ) = 0.0, max (exclusive) = 1.0
    // Simulation parameters
    // Pa-225
    double t_tot = 10.0; // s, total time
    double freq = 1000.0; // Hz, sample rate
    double data[130000] = { 0.0 }; // buffer variable for output data, 13 * t_tot
        ↪ * freq
    /* Ac-218
    double t_tot = 1e-5; // s, total time
    double freq = 1e7; // Hz, sample rate
    double data[1300] = { 0.0 }; // buffer variable for output data, 13 * t_tot *
        ↪ freq
    */
    /* Ra-222
    double t_tot = 100.0; // s, total time
    double freq = 100.0; // Hz, sample rate
    double data[130000] = { 0.0 }; // buffer variable for output data, 13 * t_tot
        ↪ * freq
    */
    /* Th-222
    double t_tot = 2e-2; // s, total time
    double freq = 5e5; // Hz, sample rate
    double data[130000] = { 0.0 }; // buffer variable for output data, 13 * t_tot
        ↪ * freq
    */

    // Constants
    double m_a = 6.6447e-27; // kg
    double c = 299792458.0; // m/s
    double N_A = 6.02214179e23; // mol^-1
    double k_B = 1.38064852e-23; // J/K
    double g = 9.81; // m/s^2

    // Pa-225
    double rho_p = 15.37e3; // kg/m^3, particle density
    double am = (225.0 / 1000.0); // kg / mol
    double t_h = 1.7; // s, radioisotope half life
    double E_a = 7.22; // MeV, average initial energy of alpha particle
    double dEdx = 296.5e3;  // MeV/m, SRIM total stopping power
    /* Ac-218
    double rho_p = 10.07e3; // kg/m^3, particle density
    double am = (218.0 / 1000.0); // kg / mol
    double t_h = 1.1e-6; // s, radioisotope half life
    double E_a = 9.2; // MeV, average initial energy of alpha particle
    double dEdx = 176.5e3;  // MeV/m, SRIM total stopping power
    */
    /* Ra-222
    double rho_p = 5.0e3; // kg/m^3, particle density
    double am = (222.0 / 1000.0); // kg / mol
    double t_h = 36.2; // s, radioisotope half life
    double E_a = 6.556; // MeV, average initial energy of alpha particle
    double dEdx = 101.8e3;  // MeV/m, SRIM total stopping power
    */
    /* Th-222
    double rho_p = 11.72e3; // kg/m^3, particle density
    double am = (222.0 / 1000.0); // kg / mol
    double t_h = 2.24e-3; // s, radioisotope half life
    double E_a = 7.98; // MeV, average initial energy of alpha particle
    double dEdx = 214.4e3;  // MeV/m, SRIM total stopping power
    */

    // Material parameters
    double a = 0.5e-6; // m, particle radius
    double m_p = (4.0 / 3.0) * M_PI * pow(a, 3) * rho_p; // kg, particle mass
    double tm = 20.0; // C, temperature
    double rho_f = (999.83952 + 16.945176 * tm - 7.9870401e-3 * pow(tm, 2) -
        ↪ 46.170461e-6 * pow(tm, 3) + 105.56302e-9 * pow(tm, 4) - 280.54253e-12 *
        ↪ pow(tm, 5)) / (1.0 + 16.879850e-3 * tm); // kg/m^3, water density
    double m_f = (4.0 / 3.0) * M_PI * pow(a, 3) * rho_f; // kg, fluid mass
    double M_star = m_p + 0.5 * m_f; // kg, effective particle mass
    unsigned long long N_0 = (((4.0 / 3.0) * M_PI * pow(a, 3) * rho_p * N_A / am)
        ↪ + 0.5); // number of atoms, + 0.5 to account for truncation
    double lambda = log(2.0) / t_h; // s^-1, decay constant
    double T = tm + 273.15; // K, temperature
    double A = -3.7188;
    double B = 578.919;
    double C = -137.546;
    double mu = (1.0 / 1000.0) * exp(A + B / (C + T)); // Pa s, dynamic viscosity
        ↪ of water (Vogel equation)
    double beta = 6.0 * M_PI * mu * a / M_star; // s^-1

    // Choose nearly largest stable time step size that is a factor of 1
    double dt = 1.0; // s, time step size
    int i = 0;
    while (dt >= (1.0 / beta))
    {
        i = i + 1;
        dt = pow(10.0, -i); // s
    } // Ac-218: comment out the following if statement so that dt = 1e-7 to allow
        ↪ freq to be 1e7
    if ((5.0 * dt) < (1.0 / beta))
    {
        dt = 5.0 * dt; // s
    }

    // SDE parameters
    double sigma = sqrt(12.0 * M_PI * mu * a * k_B * T) / M_star; // m/s^2
    double mu_OU[3] = { 0.0, 0.0, (2.0 * g * (rho_f - rho_p) * pow(a, 2) / (9.0 *
        ↪ mu)) }; // m/s

    // Preallocation
    int k = 1;
    long long number_of_decays = 0; // number of nuclei that decay
    int N = int(t_tot / dt); // number of time steps
    double temprad = 0.0;
    double tempnonrad = 0.0;
    double dW = 0.0;
    double pnet[3] = { 0.0 };
    unsigned long long poissmean = 0;
    double x[12] = { 0.0 };

    // Initial conditions
    double rad[6] = { 0.0 };
    double nonrad[6] = { 0.0 };

    // C++ speed
    double invMstuff = (1.0 / (M_star * c)); // kg^-1
    double pstuff = 2.0 * m_a * pow(c, 2); // kg m^2/s^2
    double asq = pow(a, 2); // m^2

    // Time steps
    for (int i = 0; i < N; ++i)
    {
        // Number of decays
        poissmean = (N_0 * exp(-lambda * dt * (i + 1)) * (1 - exp(-lambda * dt)));
            ↪ // mean of poisson distribution
        if (poissmean != 0)
        {
            std::poisson_distribution<unsigned long long> poissrnd(poissmean);
            number_of_decays = poissrnd(generator); // number of nuclei that decay
        }
        else
        {
            number_of_decays = 0;
        }

        // Alpha induced impulse
        for (int j = 0; j < 3; ++j)
        {
            pnet[j] = 0.0;
        }
        if (number_of_decays != 0)
        {
            for (int j = 0; j < number_of_decays; ++j)
            {
                // Ball point picking for location of alpha decay
                x[0] = standardnormalrnd(generator); // X_1
                x[1] = standardnormalrnd(generator); // X_2
                x[2] = standardnormalrnd(generator); // X_3
                x[3] = exponentialrnd(generator); // Y
                x[4] = a / sqrt(x[3] + pow(x[0], 2) + pow(x[1], 2) + pow(x[2], 2))
                    ↪ ;
                x[5] = x[4] * sqrt(pow(x[0], 2) + pow(x[1], 2) + pow(x[2], 2)); //
                    ↪ m, r location
                x[6] = acos(x[4] * x[2] / x[5]); // rad, theta

                // Sphere point picking for direction of alpha emission
                x[7] = 2.0 * M_PI * uniformrnd(generator); // rad, phi
                x[8] = acos(2.0 * uniformrnd(generator) - 1.0); // rad, theta

                // Distance to sphere exit
                x[9] = acos((sin(x[8]) * sin(x[6]) * cos(x[7] - atan2(x[1], x[0]))
                    ↪ ) + (cos(x[8]) * cos(x[6]))); // rad

```

```cpp
                // Alpha exit energy
                x[10] = (E_a - (sqrt(asq - (pow(x[5], 2) * pow(sin(x[9]), 2))) - (x[5] * cos(x[9]))) * dEdx) * 1.60218e-13; // J

                // Particle momentum (opposite direction of alpha emission)
                x[11] = sqrt(pow(x[10], 2) + pstuff * x[10]); // kg m/s, r

                // Net particle momentum
                pnet[0] = pnet[0] - x[11] * cos(x[7]) * sin(x[8]); // kg m/s, x
                pnet[1] = pnet[1] - x[11] * sin(x[7]) * sin(x[8]); // kg m/s, y
                pnet[2] = pnet[2] - x[11] * cos(x[8]); // kg m/s, z
            }
        }

        for (int j = 0; j < 3; ++j)
        {
            // Ornstein-Uhlenbeck process, Euler-Maruyama integration
            temprad = rad[j + 3];
            tempnonrad = nonrad[j + 3];
            dW = sqrt(dt) * standardnormalrnd(generator); // s, stochastic term
            rad[j + 3] = rad[j + 3] - (beta * (rad[j + 3] - mu_OU[j]) * dt) + (pnet[j] * invMstuff) + (sigma * dW);
            nonrad[j + 3] = nonrad[j + 3] - (beta * (nonrad[j + 3] - mu_OU[j]) * dt) + (sigma * dW);

            // Solve for position via trapezoidal integration
            rad[j] = rad[j] + 0.5 * dt * (temprad + rad[j + 3]);
            nonrad[j] = nonrad[j] + 0.5 * dt * (tempnonrad + nonrad[j + 3]);
        }

        // Save data: [t (x y z u v w)_rad (x y z u v w)_nonrad]
        if ((freq * dt * (i + 1) - k) >= 0)
        {
            data[13 * (k - 1)] = 0.001 * std::round(1000 * dt * (i + 1)); // Pa-225 time stamp
            // data[13 * (k - 1)] = 1e-7 * std::round(1e7 * dt * (i + 1)); // Ac-218 time stamp
            // data[13 * (k - 1)] = 0.001 * std::round(1000 * dt * (i + 1)); // Ra-222 time stamp
            // data[13 * (k - 1)] = 1e-6 * std::round(1e6 * dt * (i + 1)); // Th-222 time stamp
            for (int j = 0; j < 6; ++j)
            {
                data[13 * (k - 1) + 1 + j] = rad[j]; // rad displacements and velocities
            }
            for (int j = 0; j < 6; ++j)
            {
                data[13 * (k - 1) + 7 + j] = nonrad[j]; // nonrad displacements and velocities
            }
            k = k + 1;
        }
    }

    // Create and open output file
    std::string path = argv[2];
    std::string filename = path + "SDall" + std::to_string(number) + ".csv";
    std::ofstream outputFile;
    outputFile.open(filename, std::ofstream::out);

    // Write data to output file: [t (x y z u v w)_rad (x y z u v w)_nonrad]
    for (int h = 0; h < std::round(freq * t_tot); ++h)
    {
        outputFile.precision(4); // Pa-225, Ac-218, Ra-222
        // outputFile.precision(5); // Th-222
        outputFile << data[13 * h] << ","; // time stamp
        outputFile.precision(dbl::max_digits10);
        for (int j = 1; j < 12; ++j)
        {
            outputFile << data[13 * h + j] << ","; // displacements and velocities
        }
        outputFile << data[13 * h + 12]; // no ending comma
        outputFile << "\n"; // newline
    }

    // Close output file
    outputFile.close();

    return 0;
}
```

A.2 Adaptive Time Steps

Below is the C++ code used for simulating particles with adaptive time steps. It is configured for ^{225}Pa, but by reading the comments can be easily changed to any of the other three isotopes used in this work.

Listing A.2: *Particle simulation code with adaptive time steps - HPC3.cpp.*

```cpp
#define _USE_MATH_DEFINES
#include <cmath>
#include <limits>
typedef std::numeric_limits< double > dbl;
#include <iostream>
#include <fstream>
#include <random>
#include <string>

int main(int argc, char *argv[])
{
    // Arguments
    if (argc != 3) // name of program is one argument
    {
        std::cout << "Two arguments required." << std::endl;
        return 1;
    }
    unsigned int number = atof(argv[1]);

    // RNG Configuration
    std::random_device rd; // random seed
    std::mt19937 generator(rd());
    std::normal_distribution<double> standardnormalrnd(0.0, 1.0);  // mean = 0.0, stddev = 1.0
    std::exponential_distribution<double> exponentialrnd(1.0); // mean = 1.0, lambda = 1 / mean
    std::uniform_real_distribution<double> uniformrnd(0.0, 1.0); // min (inclusive) = 0.0, max (exclusive) = 1.0

    // Simulation parameters
    // Pa-225
    double t_tot = 10.0; // s, total time
    double freq = 1000.0; // Hz, sample rate
    double data[130000] = { 0.0 }; // buffer variable for output data, 13 * t_tot * freq
    /* Ac-218
    double t_tot = 1e-5; // s, total time
    double freq = 1e7; // Hz, sample rate
    double data[1300] = { 0.0 }; // buffer variable for output data, 13 * t_tot * freq
    */
    /* Ra-222
    double t_tot = 100.0; // s, total time
    double freq = 100.0; // Hz, sample rate
    double data[130000] = { 0.0 }; // buffer variable for output data, 13 * t_tot * freq
    */
    /* Th-222
    double t_tot = 2e-2; // s, total time
    double freq = 5e5; // Hz, sample rate
    double data[130000] = { 0.0 }; // buffer variable for output data, 13 * t_tot * freq
    */

    // Constants
    double m_a = 6.6447e-27; // kg
    double c = 299792458.0; // m/s
    double N_A = 6.02214179e23; // mol^-1
    double k_B = 1.38064852e-23; // J/K
    double g = 9.81; // m/s^2

    // Pa-225
    double rho_p = 15.37e3; // kg/m^3, particle density
    double am = (225.0 / 1000.0); // kg / mol
    double t_h = 1.7; // s, radioisotope half life
    double E_a = 7.22; // MeV, average initial energy of alpha particle
    double dEdx = 296.5e3;  // MeV/m, SRIM total stopping power
    /* Ac-218
    double rho_p = 10.07e3; // kg/m^3, particle density
    double am = (218.0 / 1000.0); // kg / mol
    double t_h = 1.1e-6; // s, radioisotope half life
    double E_a = 9.2; // MeV, average initial energy of alpha particle
    double dEdx = 176.5e3;  // MeV/m, SRIM total stopping power
    */
    /* Ra-222
    double rho_p = 5.0e3; // kg/m^3, particle density
    double am = (222.0 / 1000.0); // kg / mol
    double t_h = 36.2; // s, radioisotope half life
    double E_a = 6.556; // MeV, average initial energy of alpha particle
    double dEdx = 101.8e3;  // MeV/m, SRIM total stopping power
    */
    /* Th-222
    double rho_p = 11.72e3; // kg/m^3, particle density
    double am = (222.0 / 1000.0); // kg / mol
    double t_h = 2.24e-3; // s, radioisotope half life
    double E_a = 7.98; // MeV, average initial energy of alpha particle
    double dEdx = 214.4e3;  // MeV/m, SRIM total stopping power
    */

    // Material parameters
    double a = 0.5e-6; // m, particle radius
    double m_p = (4.0 / 3.0) * M_PI * pow(a, 3) * rho_p; // kg, particle mass
    double tm = 20.0; // C, temperature
    double rho_f = (999.83952 + 16.945176 * tm - 7.9870401e-3 * pow(tm, 2) -
        ↪ 46.170461e-6 * pow(tm, 3) + 105.56302e-9 * pow(tm, 4) - 280.54253e-12 *
        ↪ pow(tm, 5)) / (1.0 + 16.879850e-3 * tm); // kg/m^3, water density
    double m_f = (4.0 / 3.0) * M_PI * pow(a, 3) * rho_f; // kg, fluid mass
    double M_star = m_p + 0.5 * m_f; // kg, effective particle mass
    unsigned long long N_0 = (((4.0 / 3.0) * M_PI * pow(a, 3) * rho_p * N_A / am)
        ↪ + 0.5); // number of atoms, + 0.5 to account for truncation
    double lambda = log(2.0) / t_h; // s^-1, decay constant
    double T = tm + 273.15; // K, temperature
    double A = -3.7188;
    double B = 578.919;
    double C = -137.546;
    double mu = (1.0 / 1000.0) * exp(A + B / (C + T)); // Pa s, dynamic viscosity
        ↪ of water (Vogel equation)
    double beta = 6.0 * M_PI * mu * a / M_star; // s^-1

    // Find nearly largest stable time step size that is a factor of 1
    double dtmax = 1.0; // s, maximum time step size
    int i = 0;
    while (dtmax >= (1.0 / beta))
    {
        i = i + 1;
        dtmax = pow(10.0, -i); // s
    } // Ac-218: comment out the following if statement so that dt = 1e-7 to allow
        ↪ freq to be 1e7
    if ((5.0 * dtmax) < (1.0 / beta))
    {
        dtmax = 5.0 * dtmax; // s
    }

    // SDE parameters
    double sigma = sqrt(12.0 * M_PI * mu * a * k_B * T) / M_star; // m/s^2
    double mu_OU[3] = {0.0, 0.0, (2.0 * g * (rho_f - rho_p) * pow(a, 2) / (9.0 *
        ↪ mu))}; // m/s

    // Preallocation
    int k = 1;
    long long number_of_decays = 0; // number of nuclei that decay
    double t = 0.0; // s, current time
    double dt = 0.0; // s, time step size
    unsigned long long N = N_0;
    double temprad = 0.0;
    double tempnonrad = 0.0;
    double dW = 0.0;
    double pnet[3] = { 0.0 };
    unsigned long long poissmean = 0;
    double x[12] = { 0.0 };

    // Initial conditions
    double rad[6] = {0.0};
    double nonrad[6] = {0.0};

    // C++ speed
    double invlambda = (1.0 / lambda); // s^-1
    double invMstuff = (1.0 / (M_star * c)); // kg^-1
    double pstuff = 2.0 * m_a * pow(c, 2); // kg m^2/s^2
    double asq = pow(a, 2); // m^2

    // Time steps when smaller than max size
    while((t_tot - t) >= 0)
    {
        // Number of decays
        N = N - number_of_decays; // number of undecayed nuclei
        dt = invlambda / N; // time step such that the average number of decays in
            ↪  the time step is 1
        if ((dt - dtmax) <= 0)
        {
            std::poisson_distribution<int> poissrnd(1); // The mean for this
                ↪ distribution is 1
            number_of_decays = poissrnd(generator); // number of nuclei that decay
        }
        else
        {
            break; // Move on to max size time step while loop
        }
        t = t + dt;

        // Alpha induced impulse
        for (int j = 0; j < 3; ++j)
        {
            pnet[j] = 0.0;
        }
        if (number_of_decays != 0)
        {
            for (int j = 0; j < number_of_decays; ++j)
            {
                // Ball point picking for location of alpha decay
                x[0] = standardnormalrnd(generator); // X_1
                x[1] = standardnormalrnd(generator); // X_2
                x[2] = standardnormalrnd(generator); // X_3
                x[3] = exponentialrnd(generator); // Y
                x[4] = a / sqrt(x[3] + pow(x[0], 2) + pow(x[1], 2) + pow(x[2], 2))
                    ↪ ;
                x[5] = x[4] * sqrt(pow(x[0], 2) + pow(x[1], 2) + pow(x[2], 2)); //
                    ↪  m, r location
                x[6] = acos(x[4] * x[2] / x[5]); // rad, theta

                // Sphere point picking for direction of alpha emission
                x[7] = 2.0 * M_PI * uniformrnd(generator); // rad, phi
                x[8] = acos(2.0 * uniformrnd(generator) - 1.0); // rad, theta

                // Distance to sphere exit
                x[9] = acos((sin(x[8]) * sin(x[6]) * cos(x[7] - atan2(x[1], x[0]))
                    ↪ ) + (cos(x[8]) * cos(x[6]))); // rad

                // Alpha exit energy
                x[10] = (E_a - (sqrt(asq - (pow(x[5], 2) * pow(sin(x[9]), 2))) - (
                    ↪ x[5] * cos(x[9]))) * dEdx) * 1.60218e-13; // J

                // Particle momentum (opposite direction of alpha emission)
                x[11] = sqrt(pow(x[10], 2) + pstuff * x[10]); // kg m/s, r

                // Net particle momentum
                pnet[0] = pnet[0] - x[11] * cos(x[7]) * sin(x[8]); // kg m/s, x
                pnet[1] = pnet[1] - x[11] * sin(x[7]) * sin(x[8]); // kg m/s, y
                pnet[2] = pnet[2] - x[11] * cos(x[8]); // kg m/s, z
            }
        }

        for (int j = 0; j < 3; ++j)
        {
            // Ornstein-Uhlenbeck process, Euler-Maruyama integration
            temprad = rad[j + 3];
            tempnonrad = nonrad[j + 3];
            dW = sqrt(dt) * standardnormalrnd(generator); // s, stochastic term
            rad[j + 3] = rad[j + 3] - (beta * (rad[j + 3] - mu_OU[j]) * dt) + (
                ↪ pnet[j] * invMstuff) + (sigma * dW);
            nonrad[j + 3] = nonrad[j + 3] - (beta * (nonrad[j + 3] - mu_OU[j]) *
                ↪ dt) + (sigma * dW);

            // Solve for position via trapezoidal integration
            rad[j] = rad[j] + 0.5 * dt * (temprad + rad[j + 3]);
            nonrad[j] = nonrad[j] + 0.5 * dt * (tempnonrad + nonrad[j + 3]);
        }

        // Save data: [t (x y z u v w)_rad (x y z u v w)_nonrad]
        if ((t * freq - k) >= 0)
        {
            data[13 * (k - 1)] = t; // time stamp
        for (int j = 0; j < 6; ++j)
        {
          data[13 * (k - 1) + 1 + j] = rad[j]; // rad displacements and
            ↪ velocities
        }
        for (int j = 0; j < 6; ++j)
        {
          data[13 * (k - 1) + 7 + j] = nonrad[j]; // nonrad displacements
            ↪ and velocities
        }
        k = k + 1;
      }
    }

    // Time steps of max size
    dt = dtmax;
    N = N + number_of_decays; // reset N
    while((t_tot - t) >= 0)
    {
      // Number of decays
      N = N - number_of_decays; // number of undecayed nuclei
      poissmean = (N * lambda * dt); // mean of poisson distribution
      if (poissmean != 0)
      {
        std::poisson_distribution<unsigned long long> poissrnd(poissmean);
        number_of_decays = poissrnd(generator); // number of nuclei that decay
      }
      else
      {
        number_of_decays = 0;
      }
      t = t + dt;

      // Alpha induced impulse
      for (int j = 0; j < 3; ++j)
      {
        pnet[j] = 0.0;
      }
      if (number_of_decays != 0)
      {
        for (int j = 0; j < number_of_decays; ++j)
        {
          // Ball point picking for location of alpha decay
          x[0] = standardnormalrnd(generator); // X_1
          x[1] = standardnormalrnd(generator); // X_2
          x[2] = standardnormalrnd(generator); // X_3
          x[3] = exponentialrnd(generator); // Y
          x[4] = a / sqrt(x[3] + pow(x[0], 2) + pow(x[1], 2) + pow(x[2], 2))
            ↪ ;
          x[5] = x[4] * sqrt(pow(x[0], 2) + pow(x[1], 2) + pow(x[2], 2)); //
            ↪  m, r location
          x[6] = acos(x[4] * x[2] / x[5]); // rad, theta

          // Sphere point picking for direction of alpha emission
          x[7] = 2.0 * M_PI * uniformrnd(generator); // rad, phi
          x[8] = acos(2.0 * uniformrnd(generator) - 1.0); // rad, theta

          // Distance to sphere exit
          x[9] = acos((sin(x[8]) * sin(x[6]) * cos(x[7] - atan2(x[1], x[0]))
            ↪ ) + (cos(x[8]) * cos(x[6]))); // rad

          // Alpha exit energy
          x[10] = (E_a - (sqrt(asq - (pow(x[5], 2) * pow(sin(x[9]), 2))) - (
            ↪ x[5] * cos(x[9]))) * dEdx) * 1.60218e-13; // J

          // Particle momentum (opposite direction of alpha emission)
          x[11] = sqrt(pow(x[10], 2) + pstuff * x[10]); // kg m/s, r

          // Net particle momentum
          pnet[0] = pnet[0] - x[11] * cos(x[7]) * sin(x[8]); // kg m/s, x
          pnet[1] = pnet[1] - x[11] * sin(x[7]) * sin(x[8]); // kg m/s, y
          pnet[2] = pnet[2] - x[11] * cos(x[8]); // kg m/s, z
        }
      }

      for (int j = 0; j < 3; ++j)
      {
        // Ornstein-Uhlenbeck process, Euler-Maruyama integration
        temprad = rad[j + 3];
        tempnonrad = nonrad[j + 3];
        dW = sqrt(dt) * standardnormalrnd(generator); // s, stochastic term
        rad[j + 3] = rad[j + 3] - (beta * (rad[j + 3] - mu_OU[j]) * dt) + (
          ↪ pnet[j] * invMstuff) + (sigma * dW);
```

```
            nonrad[j + 3] = nonrad[j + 3] - (beta * (nonrad[j + 3] - mu_OU[j]) *
                ↪ dt) + (sigma * dW);

            // Solve for position via trapezoidal integration
            rad[j] = rad[j] + 0.5 * dt * (temprad + rad[j + 3]);
            nonrad[j] = nonrad[j] + 0.5 * dt * (tempnonrad + nonrad[j + 3]);
        }

        // Save data: [t (x y z u v w)_rad (x y z u v w)_nonrad]
        if ((t * freq - k) >= 0)
        {
            data[13 * (k - 1)] = t; // time stamp
            for (int j = 0; j < 6; ++j)
            {
                data[13 * (k - 1) + 1 + j] = rad[j]; // rad displacements and
                    ↪ velocities
            }
            for (int j = 0; j < 6; ++j)
            {
                data[13 * (k - 1) + 7 + j] = nonrad[j]; // nonrad displacements
                    ↪ and velocities
            }
            k = k + 1;
        }
    }

    // Create and open output file
    std::string path = argv[2];
    std::string filename = path + "SDall" + std::to_string(number) + ".csv";
    std::ofstream outputFile;
    outputFile.open(filename, std::ofstream::out);
    outputFile.precision(dbl::max_digits10);

    // Write data to output file: [t (x y z u v w)_rad (x y z u v w)_nonrad]
    for (int h = 0; h < std::round(freq * t_tot); ++h)
    {
        outputFile << data[13 * h] << ","; // time stamp
        for (int j = 1; j < 12; ++j)
        {
            outputFile << data[13 * h + j] << ","; // displacements and velocities
        }
        outputFile << data[13 * h + 12]; // no ending comma
        outputFile << "\n"; // newline
    }

    // Close output file
    outputFile.close();

    return 0;
}
```

Appendix B

Methods

The methods of running the codes of Appendix A, and processing and analyzing the resulting data, are presented herein.

B.1 Generating an Output File

The particle simulation codes are compiled with the Makefile shown below, which is run with the command `make`.

Listing B.1: *Makefile for compiling the particle simulation code - Makefile.*

```
all: main

main: Main.cpp
	g++ -std=c++11 Main.cpp -o Main
```

To run the particle simulation executable in an embarrassingly parallel fashion on the Beocat Research Cluster at Kansas State University, the shell script below submits an array job to Slurm via the command `sbatch ./Beocat.sh`. The runtimes used for the combinations of isotope and time step are listed in Table B.1.

Listing B.2: *Shell script for high performance computing utilization - Beocat.sh.*

```
#!/bin/bash -l
#SBATCH --job-name=Pa-225   # job name displayed in kstat
#SBATCH --time=00-30:00:00   # dd-hh:mm:ss
#SBATCH --mem=1G   # Memory per node
#SBATCH --nodes=1   # number of nodes
#SBATCH --ntasks-per-node=1   # number of tasks per node
```

```
#SBATCH --partition=killable.q   # killable job can be killed, but can be run
    ↪ sooner
#SBATCH -x, --exclude=dwarf[29-32]    # exclude Bahadori's GPU nodes
#SBATCH --mail-type=ALL,TIME_LIMIT    # send me an email when job begins, ends,
    ↪ fails, is requeued, or runs out of time
#SBATCH --array=1-10000:1   # array job runs this shell script...
RUNSIZE=$SLURM_ARRAY_TASK_ID   # ...for each array value
TEMPORARY_DIRECTORY=/tmp/

./Main $RUNSIZE $TEMPORARY_DIRECTORY  # Run executable with argument from array
    ↪ job

cp ${TEMPORARY_DIRECTORY}SD* .; # copy output files from temporary directory
```

Table B.1: *Particle simulation executable runtimes in* ***hh:mm:ss*** *format.*

	^{222}Ra	^{225}Pa	^{222}Th	^{218}Ac
Set Time Step	06:00:00	15:00:00	15:00:00	12:00:00
Adaptive Time Step	14:00:00	30:00:00	24:00:00	21:00:00

The 10,000 CSV files are then concatenated together with the Python script shown below. This decreases MATLAB®'s data read-in time.

Listing B.3: *Output file concatenation script - Concatenate.py.*

```python
radioisotope = input('Which radioisotope?\n')
HPC = input('Which HPC number?\n')

file = "Z:\Graham Wilson\Radioactive Colloids\HPC" + HPC + "\\" + radioisotope + "
    ↪ \SDall"
with open(file + ".csv","a") as fout:
    for num in range(1,10001):
        with open(file + str(num) + ".csv") as f:
            for line in f:
                fout.write(line)
```

B.2 Determining the MSDs

The MSDs are calculated and plotted with the MATLAB® script shown below. Certain sections are run depending on the time step type.

Listing B.4: *MSD calculation and plotting script - Plots.m.*

```matlab
% Initialization
clear

% Constants
m_a = 6.6447e-27; % kg
c = 299792458; % m/s
N_A = 6.02214179e23; % mol^-1
k_B = 1.38064852e-23; % J/K
g = 9.81; % m/s^2
length = 10000; % number of data points

% Choose radioisotope
```

```
radioisotope = input('Which radioisotope?\n','s');
switch radioisotope
    case 'Ac-218'
        rho_p = 10.07e3; % kg/m^3, particle density
        am = (218 / 1000); % kg / mol
        t_h = 1.1e-6; % s, radioisotope half life
        E_a = 9.2; % MeV, average inital energy of alpha particle
        length = 100;
        t_tot = 1e-5; % s, total time
        freq = 1e7; % Hz, sample rate
        t_ddt = logspace(-14,0,1000);
    case 'Pa-225'
        rho_p = 15.37e3; % kg/m^3, particle density
        am = (225 / 1000); % kg / mol
        t_h = 1.7; % s, radioisotope half life
        E_a = 7.22; % MeV, average inital energy of alpha particle
        t_tot = 10; % s, total time
        freq = 1000; % Hz, sample rate
        t_ddt = logspace(-10,2,1000);
    case 'Th-222'
        rho_p = 11.72e3; % kg/m^3, particle density
        am = (222 / 1000); % kg / mol
        t_h = 2.24e-3; % s, radioisotope half life
        E_a = 7.98; % MeV, average inital energy of alpha particle
        t_tot = 2e-2; % s, total time
        freq = 5e5; % Hz, sample rate
        t_ddt = logspace(-14,0,1000);
    case 'Ra-222'
        rho_p = 5e3; % kg/m^3, particle density
        am = (222 / 1000); % kg / mol
        t_h = 36.2; % s, radioisotope half life
        E_a = 6.556; % MeV, average inital energy of alpha particle
        t_tot = 100; % s, total time
        freq = 100; % Hz, sample rate
        t_ddt = logspace(-10,2,1000);
end

% Material parameters
a = 0.5e-6; % m, particle radius
tm = 20; % C, temperature
rho_f = (999.83952 + 16.945176 * tm - 7.9870401e-3 * tm^2 - 46.170461e-6 * tm^3 + 105.56302e-9 * tm^4 - 280.54253e-12 * tm^5) / (1 + 16.879850e-3 * tm); % kg/m^3, water density
m_p = (4 / 3) * pi * a^3 * rho_p; % kg, particle mass
m_f = (4 / 3) * pi * a^3 * rho_f; % kg, fluid mass
M_star = m_p + 0.5 * m_f; % kg, effective particle mass
N_0 = round((4 / 3) * pi * a^3 * rho_p * N_A / am); % number of atoms
lambda = log(2) / t_h; % s^-1, decay constant
T = tm + 273.15; % K, temperature
A = -3.7188;
B = 578.919;
C = -137.546;
mu = (1 / 1000) * exp(A + B / (C + T)); % Pa s, dynamic viscosity of water (Vogel equation)

% Get data
file = ['Z:\Graham Wilson\Radioactive Colloids\HPC3\' radioisotope '\SDall.csv']; % python concatenated file
temp = csvread(file);
SD = zeros(length,13,10000);
for i = 1:10000
    SD(:,:,i) = temp((length * (i - 1) + 1):(length * i),:);
end
clear temp
SDrad = zeros(size(SD,1),7,size(SD,3));
SDrad(:,1,:) = SD(:,1,:); % time stamps
SDrad(:,2:end,:) = SD(:,2:7,:); % rad data
SDnonrad = zeros(size(SD,1),7,size(SD,3));
SDnonrad(:,1,:) = SD(:,1,:); % time stamps
SDnonrad(:,2:end,:) = SD(:,8:end,:); % nonrad data
clear SD
dt = 1 / freq; % s, time step size
t = dt:dt:t_tot; % s, time vector

% Calculate square displacements about mean due to gravity
SDrad(:,2:3,:) = SDrad(:,2:3,:).^2; % x and y
SDnonrad(:,2:3,:) = SDnonrad(:,2:3,:).^2; % x and y
SDrad(:,4,:) = (SDrad(:,4,:) - (2 * g * (rho_f - rho_p) * a^2 / (9 * mu)) * (SDrad(:,1,:) - (M_star / (6 * pi * mu * a)) * (ones(size(SDrad(:,1,:))) + exp(-6 * pi * mu * a * SDrad(:,1,:) / M_star)))).^2; % z
SDnonrad(:,4,:) = (SDnonrad(:,4,:) - (2 * g * (rho_f - rho_p) * a^2 / (9 * mu)) * (SDnonrad(:,1,:) - (M_star / (6 * pi * mu * a)) * (ones(size(SDnonrad(:,1,:)
    ↪ )) + exp(-6 * pi * mu * a * SDnonrad(:,1,:) / M_star)))).^2; % z

% Preallocate arrays
MSDrad = zeros(max(size(t)),6); % [t num(x) num(y) num(z) num theor]
MSDnonrad = zeros(max(size(t)),6); % [t num(x) num(y) num(z) num theor]

%% Set Time Steps
% Time stamps
MSDrad(:,1) = SDrad(:,1,1); % s
MSDnonrad(:,1) = SDnonrad(:,1,1); % s

% Calculate numerical mean square displacements
MSDrad(:,2:4) = mean(SDrad(:,2:4,:),3);
MSDnonrad(:,2:4) = mean(SDnonrad(:,2:4,:),3);
MSDrad(:,5) = sum(MSDrad(:,2:4),2);
MSDnonrad(:,5) = sum(MSDnonrad(:,2:4),2);

% Calculate standard deviations (of mean values)
STDrad = std(sum(SDrad(:,2:4,:),2),0,3) / sqrt(size(SDrad,3));
STDnonrad = std(sum(SDnonrad(:,2:4,:),2),0,3) / sqrt(size(SDnonrad,3));

%% Adaptive Time Steps
% Time stamps
MSDrad(:,1) = t; % s
MSDnonrad(:,1) = t; % s

% Calculate numerical mean square displacements
linrad = zeros(max(size(t)),3,size(SDrad,3)); % preallocation
linnonrad = zeros(max(size(t)),3,size(SDnonrad,3)); % preallocation
for i = 1:10000 % linear interpolation to specific times
    for j = 1:3
        linrad(:,j,i) = interp1([0;SDrad(:,1,i)],[0;SDrad(:,j+1,i)],t); % at
            ↪ origin at t = 0
        linnonrad(:,j,i) = interp1([0;SDnonrad(:,1,i)],[0;SDnonrad(:,j+1,i)],t); %
            ↪  at origin at t = 0
    end
end
clear SDrad SDnonrad
MSDrad(:,2:4) = mean(linrad(:,1:3,:),3);
MSDnonrad(:,2:4) = mean(linnonrad(:,1:3,:),3);
MSDrad(:,5) = sum(MSDrad(:,2:4),2);
MSDnonrad(:,5) = sum(MSDnonrad(:,2:4),2);

% Calculate standard deviations (of mean values)
STDrad = std(sum(linrad(:,1:3,:),2),0,3) / sqrt(size(linrad,3));
STDnonrad = std(sum(linnonrad(:,1:3,:),2),0,3) / sqrt(size(linnonrad,3));

%%
% Calculate theoretical mean square displacements
E = E_a * 1.60218e-13; % J
p = sqrt(E.^2 + 2 * E * m_a * c^2) / c; % kg m/s
MSDrad(:,6) = (3 * M_star * k_B * T / (6 * pi * mu * a)^2) * (-3 + 4 * exp(-6 * pi
    ↪  * mu * a * MSDrad(:,1) / M_star) - exp(-2 * 6 * pi * mu * a * MSDrad(:,1) /
    ↪  M_star) + 2 * 6 * pi * mu * a * MSDrad(:,1) / M_star) + (p / (6 * pi * mu *
    ↪  a))^2 * N_0 * (1 - exp(-lambda * MSDrad(:,1))); % m^2
MSDnonrad(:,6) = (3 * M_star * k_B * T / (6 * pi * mu * a)^2) * (-3 + 4 * exp(-6 *
    ↪  pi * mu * a * MSDrad(:,1) / M_star) - exp(-2 * 6 * pi * mu * a * MSDrad
    ↪ (:,1) / M_star) + 2 * 6 * pi * mu * a * MSDrad(:,1) / M_star); % m^2

% Calculate time derivative of mean square displacements
gamma = 6 * pi * mu * a;
rad = (6 * k_B * T / gamma) * (1 - 2 * exp(-gamma * t_ddt / M_star) + exp(-2 *
    ↪ gamma * t_ddt / M_star)) + N_0 * (p / gamma)^2 * lambda * exp(-lambda *
    ↪ t_ddt);
nonrad = (6 * k_B * T / gamma) * (1 - 2 * exp(-gamma * t_ddt / M_star) + exp(-2 *
    ↪ gamma * t_ddt / M_star));

% Save mean square displacements and standard deviations
save(['Z:\Graham Wilson\Radioactive Colloids\HPC3\' radioisotope '\raddata.mat'],'
    ↪ MSDrad','STDrad');
save(['Z:\Graham Wilson\Radioactive Colloids\HPC3\' radioisotope '\nonraddata.mat'
    ↪ ],'MSDnonrad','STDnonrad');

% Plot MSDs together
figure
LH(1) = fill([MSDrad(:,1); flipud(MSDrad(:,1))],[MSDrad(:,5) + STDrad; flipud(
    ↪ MSDrad(:,5) - STDrad)],'g','LineStyle','none');
alpha(0.5);hold all
plot(MSDrad(:,1),MSDrad(:,5),'-b',MSDrad(:,1),MSDrad(:,6),'-.r','LineWidth',1)
LH(2) = fill([MSDnonrad(:,1); flipud(MSDnonrad(:,1))],[MSDnonrad(:,5) + STDnonrad;
    ↪  flipud(MSDnonrad(:,5) - STDnonrad)],'y','LineStyle','none');
alpha(0.5);hold all
plot(MSDnonrad(:,1),MSDnonrad(:,5),'-c',MSDnonrad(:,1),MSDnonrad(:,6),'--m','
    ↪ LineWidth',1)
xlabel('$t$ (s)','interpreter','latex')
```

```
ylabel('$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle$ (m\
    ↪ textsuperscript{2})','interpreter','latex')
spacing = 500; % 5 for Ac-218
plot(MSDrad(spacing:spacing:end,1),MSDrad(spacing:spacing:end,5),'b^')
plot(MSDnonrad(spacing:spacing:end,1),MSDnonrad(spacing:spacing:end,5),'co')
LH(3) = plot(nan, nan, '-b^');LH(4) = plot(nan, nan, '-.r');
LH(5) = plot(nan, nan, '-co');LH(6) = plot(nan, nan, '--m');
L{1} = '$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle_{rad} \pm \
    ↪ sigma_{rad}$';
L{2} = '$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle \pm \sigma$';
L{3} = '$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle_{rad}$';
L{4} = '$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle_{rad,theor}$';
L{5} = '$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle$';
L{6} = '$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle_{theor}$';
lh = legend(LH, L,'interpreter','latex','Location','northwest');
lh.PlotChildren = lh.PlotChildren([4 3 1 6 5 2]);
axes('Position',[0.6 0.2 0.3 0.25]) % [0.58 0.475 0.3 0.25]
box on
semilogx(t_ddt,rad,'--',t_ddt,nonrad)
ylabel('$d (\textrm{MSD}) / dt$ (m\textsuperscript{2}/s)','interpreter','latex')
h = legend({'Rad.','Non-rad.'},'interpreter','latex','Location','southeast');
set(h, 'Position', [0.77, 0.225, .08, .0625]) % [0.75, 0.5, .08, .0625]
legend('boxoff');xlim([1e-10 100]);ylim([0 5e-12])
```

B.3 Analyzing the Data

The excess kurtosis of the data can be computed and plotted with the MATLAB® script shown below. The data must be read in with the first 77 lines of the script above, and certain sections of the script below are run depending on the time step type.

Listing B.5: *Excess kurtosis computation and plotting script - ExcessKurtosis.m.*

```
% Run the first 81 lines of Plots.m - stop on the line "t = ..."
%% Adaptive time step
linrad = zeros(max(size(t)),3,size(SDrad,3)); % preallocation
linnonrad = zeros(max(size(t)),3,size(SDnonrad,3)); % preallocation
for i = 1:10000 % linear interpolation to specific times
    for j = 1:3
        linrad(:,j,i) = interp1([0;SDrad(:,1,i)],[0;SDrad(:,j+1,i)],t); % at
            ↪ origin at t = 0
        linnonrad(:,j,i) = interp1([0;SDnonrad(:,1,i)],[0;SDnonrad(:,j+1,i)],t); %
            ↪  at origin at t = 0
    end
end
SDrad(:,2:4,:) = linrad;
SDnonrad(:,2:4,:) = linnonrad;
time = t; % s

%% Set time step
time = SDrad(:,1,1); % s

%%
SDrad(:,4,:) = SDrad(:,4,:) - (2 * g * (rho_f - rho_p) * a^2 / (9 * mu)) * (SDrad
    ↪ (:,1,:) - (M_star / (6 * pi * mu * a)) * (ones(size(SDrad(:,1,:))) + exp(-6
    ↪ * pi * mu * a * SDrad(:,1,:) / M_star))); % z about mean due to gravity
SDnonrad(:,4,:) = SDnonrad(:,4,:) - (2 * g * (rho_f - rho_p) * a^2 / (9 * mu)) * (
    ↪ SDnonrad(:,1,:) - (M_star / (6 * pi * mu * a)) * (ones(size(SDnonrad(:,1,:))
    ↪ ) + exp(-6 * pi * mu * a * SDnonrad(:,1,:) / M_star))); % z about mean due
    ↪ to gravity
xbarrad = mean(SDrad(:,2:4,:),3); % sample mean vector
xbarnonrad = mean(SDnonrad(:,2:4,:),3); % sample mean vector
Srad = zeros(3,3,size(SDrad,1)); % sample covariance matrix
Snonrad = zeros(3,3,size(SDnonrad,1)); % sample covariance matrix
for i = 1:size(SDrad,1) % for each time step
    for j = 1:size(SDrad,3) % for each particle
```

```
        Srad(:,:,i) = Srad(:,:,i) + (SDrad(i,2:4,j) - xbarrad(i,:))' * (SDrad(i
            ↪ ,2:4,j) - xbarrad(i,:));
        Snonrad(:,:,i) = Snonrad(:,:,i) + (SDnonrad(i,2:4,j) - xbarnonrad(i,:))' *
            ↪  (SDnonrad(i,2:4,j) - xbarnonrad(i,:));
    end
end
Srad = Srad / size(SDrad,3);
Snonrad = Snonrad / size(SDrad,3);
b2rad = zeros(1,size(SDrad,1)); % sample kurtosis
b2nonrad = zeros(1,size(SDnonrad,1)); % sample kurtosis
for i = 1:size(SDrad,1) % for each time step
    Sradinv = inv(Srad(:,:,i));
    Snonradinv = inv(Snonrad(:,:,i));
    for j = 1:size(SDrad,3) % for each particle
        b2rad(:,i) = b2rad(:,i) + ((SDrad(i,2:4,j) - xbarrad(i,:)) * Sradinv * (
            ↪ SDrad(i,2:4,j) - xbarrad(i,:))')^2;
        b2nonrad(:,i) = b2nonrad(:,i) + ((SDnonrad(i,2:4,j) - xbarnonrad(i,:)) *
            ↪ Snonradinv * (SDnonrad(i,2:4,j) - xbarnonrad(i,:))')^2;
    end
end
b2rad = b2rad / size(SDrad,3);
b2nonrad = b2nonrad / size(SDrad,3);

% Save statistical data
save('Z:\Graham Wilson\Radioactive Colloids\HPC3\Pa-225\radstatdata.mat','xbarrad'
    ↪ ,'Srad','b2rad');
save('Z:\Graham Wilson\Radioactive Colloids\HPC3\Pa-225\nonradstatdata.mat','
    ↪ xbarnonrad','Snonrad','b2nonrad');

% Plot excess kurtosis
figure
plot(time,b2rad - 15,'k')
hold on
plot(time,b2nonrad - 15,'Color',[0.8 0.8 0.8])
xlabel('$t$ (s)','interpreter','latex')
ylabel('$g_2$','interpreter','latex')
legend({'Radiation Augmented','Regular'},'interpreter','latex','Location','
    ↪ northeast')
```

The autocorrelation function of velocity data for individual particle pairs can be calculated and plotted with the Python script below. Data is read from a file produced by the particle simulation executable, not from a concatenated file.

Listing B.6: *Autocorrelation function calculation and plotting script - Autocorrelation.py.*

```python
import pandas as pd
import matplotlib.pyplot as plt
from pandas.plotting import autocorrelation_plot

plt.rc('text', usetex=True)
data = pd.read_csv('SDall1.csv',header=None)

# Radioactive u velocity
fig = plt.figure()
autocorrelation_plot(data[[4]])
plt.xlabel('Lag (1e-3 s)')
plt.ylabel(r"$ACF$")
ax = fig.gca()
ax.grid(False)
ax.tick_params(direction="in")
plt.ylim(-0.1,0.1)
plt.show()

# Regular u velocity
fig = plt.figure()
autocorrelation_plot(data[[10]])
plt.xlabel('Lag (1e-3 s)')
plt.ylabel(r"$ACF$")
ax = fig.gca()
ax.grid(False)
ax.tick_params(direction="in")
plt.ylim(-0.1,0.1)
plt.show()
```

Appendix C

Useful Relations

Spherical harmonics satisfy the angular part of Laplace's equation in spherical coordinates [46]

$$\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial Y_l^m}{\partial\theta}\right)+\frac{1}{\sin^2\theta}\frac{\partial^2 Y_l^m}{\partial\phi^2}=-l\left(l+1\right)Y_l^m. \tag{C.1}$$

Spherical harmonics form an orthonormal set [46]

$$\int_0^{2\pi}\int_0^{\pi}Y_l^{m*}\left(\theta,\phi\right)Y_{l'}^{m'}\left(\theta,\phi\right)\sin\theta d\theta d\phi=\int_{4\pi}Y_l^{m*}\left(\Omega\right)Y_{l'}^{m'}\left(\Omega\right)d\Omega=\delta_{ll'}\delta_{mm'}, \tag{C.2}$$

where the asterisk denotes complex conjugation.

The closure relationship for the spherical harmonics is [46]

$$\sum_{l=0}^{\infty}\sum_{m=-l}^{l}Y_l^m\left(\Omega\right)Y_l^{m*}\left(\Omega'\right)=\delta\left(\Omega-\Omega'\right). \tag{C.3}$$

The values of some key spherical harmonics are as follows [46]

$$Y_0^0=\sqrt{\frac{1}{4\pi}}, \tag{C.4}$$

$$Y_1^0 = \sqrt{\frac{3}{4\pi}} \cos\theta, \tag{C.5}$$

$$Y_1^{\pm 1} = \mp\sqrt{\frac{3}{8\pi}} \sin\theta e^{\pm i\phi}, \tag{C.6}$$

$$Y_2^0 = \sqrt{\frac{5}{16\pi}} \left(3\cos^2\theta - 1\right), \tag{C.7}$$

$$Y_2^{\pm 2} = \sqrt{\frac{15}{32\pi}} \sin^2\theta e^{\pm 2i\phi}, \tag{C.8}$$

$$Y_3^0 = \frac{1}{4}\sqrt{\frac{7}{\pi}} \left(5\cos^3\theta - 3\cos\theta\right), \tag{C.9}$$

$$Y_3^{\pm 1} = \mp\frac{1}{8}\sqrt{\frac{21}{\pi}} \sin\theta \left(5\cos^2\theta - 1\right) e^{\pm i\phi}, \tag{C.10}$$

$$Y_3^{\pm 3} = \mp\frac{1}{8}\sqrt{\frac{35}{\pi}} \sin^3\theta e^{\pm 3i\phi}, \tag{C.11}$$

$$Y_4^0 = \frac{3}{16}\sqrt{\frac{1}{\pi}} \left(35\cos^4\theta - 30\cos^2\theta + 3\right), \tag{C.12}$$

$$Y_4^{\pm 2} = \frac{3}{8}\sqrt{\frac{5}{2\pi}} \sin^2\theta \left(7\cos^2\theta - 1\right) e^{\pm 2i\phi}, \tag{C.13}$$

$$Y_4^{\pm 4} = \frac{3}{16}\sqrt{\frac{35}{2\pi}} \sin^4\theta e^{\pm 4i\phi}. \tag{C.14}$$

Using Eq. C.5

$$\cos\theta = \sqrt{\frac{4\pi}{3}} Y_1^{0*}. \tag{C.15}$$

Using Eq. C.4 and Eq. C.7

$$\cos^2\theta = \frac{1}{3}\left(\sqrt{\frac{16\pi}{5}}Y_2^{0*} + \sqrt{4\pi}Y_0^{0*}\right). \tag{C.16}$$

Using Eq. C.5 and Eq. C.9

$$\cos^3\theta = \frac{1}{5}\left[4\sqrt{\frac{\pi}{7}}Y_3^{0*} + 3\sqrt{\frac{4\pi}{3}}Y_1^{0*}\right]. \tag{C.17}$$

Using Eq. C.4, Eq. C.7, and Eq. C.12

$$\cos^4\theta = \frac{1}{35}\left[\frac{16}{3}\sqrt{\pi}Y_4^{0*} + 10\sqrt{\frac{16\pi}{5}}Y_2^{0*} + 7\sqrt{4\pi}Y_0^{0*}\right]. \tag{C.18}$$

Using Eq. C.6

$$\sin\theta\cos\phi = \frac{1}{2}\sqrt{\frac{8\pi}{3}}\left(Y_1^{-1*} - Y_1^{1*}\right), \tag{C.19}$$

$$\sin\theta\sin\phi = \frac{1}{2i}\sqrt{\frac{8\pi}{3}}\left(Y_1^{-1*} + Y_1^{1*}\right). \tag{C.20}$$

Using Eq. C.4, Eq. C.7, and Eq. C.8

$$\sin^2\theta\cos^2\phi = \frac{1}{4}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*} + Y_2^{-2*}\right) - \frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*} + \frac{1}{3}\sqrt{4\pi}Y_0^{0*}, \tag{C.21}$$

$$\sin^2\theta\sin^2\phi = -\frac{1}{4}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*} + Y_2^{-2*}\right) - \frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*} + \frac{1}{3}\sqrt{4\pi}Y_0^{0*}. \tag{C.22}$$

Using Eq. C.6, Eq. C.10, and Eq. C.11

$$\sin^3\theta\cos^3\phi = \sqrt{\frac{\pi}{35}}\left(Y_3^{-3*} - Y_3^{3*}\right) + \frac{3}{5}\sqrt{\frac{\pi}{21}}\left(Y_3^{1*} - Y_3^{-1*}\right) - \frac{3}{10}\sqrt{\frac{8\pi}{3}}\left(Y_1^{1*} - Y_1^{-1*}\right), \tag{C.23}$$

$$\sin^3\theta\sin^3\phi = \frac{-1}{i}\left[\sqrt{\frac{\pi}{35}}\left(Y_3^{-3*}+Y_3^{3*}\right)+\frac{3}{5}\sqrt{\frac{\pi}{21}}\left(Y_3^{1*}+Y_3^{-1*}\right)\right.$$
$$\left.-\frac{3}{10}\sqrt{\frac{8\pi}{3}}\left(Y_1^{1*}+Y_1^{-1*}\right)\right]. \quad (C.24)$$

Using Eq. C.4, Eq. C.7, Eq. C.8, Eq. C.12, Eq. C.13, and Eq. C.14

$$\sin^4\theta\cos^4\phi = \frac{1}{16}\left[\frac{16}{3}\sqrt{\frac{2\pi}{35}}\left(Y_4^{4*}+Y_4^{-4*}\right)-\frac{32}{21}\sqrt{\frac{2\pi}{5}}\left(Y_4^{2*}+Y_4^{-2*}\right)\right.$$
$$\left.+\frac{24}{7}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}+Y_2^{-2*}\right)+\frac{6}{35}\left(\frac{16}{3}\sqrt{\pi}Y_4^{0*}-\frac{40}{3}\sqrt{\frac{16\pi}{5}}Y_2^{0*}+\frac{56}{3}\sqrt{4\pi}Y_0^{0*}\right)\right], \quad (C.25)$$

$$\sin^4\theta\sin^4\phi = \frac{1}{16}\left[\frac{16}{3}\sqrt{\frac{2\pi}{35}}\left(Y_4^{4*}+Y_4^{-4*}\right)+\frac{32}{21}\sqrt{\frac{2\pi}{5}}\left(Y_4^{2*}+Y_4^{-2*}\right)\right.$$
$$\left.-\frac{24}{7}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}+Y_2^{-2*}\right)+\frac{6}{35}\left(\frac{16}{3}\sqrt{\pi}Y_4^{0*}-\frac{40}{3}\sqrt{\frac{16\pi}{5}}Y_2^{0*}+\frac{56}{3}\sqrt{4\pi}Y_0^{0*}\right)\right]. \quad (C.26)$$

Appendix D

Derivations

The tedious portions of the derivations of the mean square displacement and excess kurtosis for a radioactive Janus particle are provided in the following sections. The probability of arriving at Ω_1 from Ω_2 over time $t_1 - t_2$ is expressed as $P(\Omega_1 \leftarrow \Omega_2, t_1 - t_2)$.

D.1 Mean Square Displacement

Here $\langle \cos\theta_1 \cos\theta_2 \rangle_{t_1>t_2}$ is computed using Eq. 3.13 and the equations in Appendix C.

$$
\begin{aligned}
&\langle \cos\theta_1 \cos\theta_2 \rangle_{t_1>t_2} \\
&\quad = \int_{4\pi}\int_{4\pi} \cos\theta_1 \cos\theta_2 P(\Omega_1 \leftarrow \Omega_2, t_1 - t_2)\, P(\Omega_2 \leftarrow \Omega_0, t_2 - t_0)\, d\Omega_1 d\Omega_2 \\
&\quad = \sqrt{\frac{4\pi}{3}} \int_{4\pi}\int_{4\pi} Y_1^{0*}(\Omega_1) \left[\sum_{l=0}^{\infty} \sum_{m=-l}^{l} e^{-l(l+1)D_r(t_1-t_2)} Y_l^{m*}(\Omega_2)\, Y_l^{m}(\Omega_1) \right] d\Omega_1 \\
&\qquad \times \cos\theta_2 P(\Omega_2 \leftarrow \Omega_0, t_2 - t_0)\, d\Omega_2 \\
&\quad = \sqrt{\frac{4\pi}{3}} e^{-2D_r(t_1-t_2)} \int_{4\pi} Y_1^{0*}(\Omega_2) \cos\theta_2 P(\Omega_2 \leftarrow \Omega_0, t_2 - t_0)\, d\Omega_2 \\
&\quad = e^{-2D_r(t_1-t_2)} \int_{4\pi} \cos^2\theta_2 P(\Omega_2 \leftarrow \Omega_0, t_2 - t_0)\, d\Omega_2 \\
&\quad = e^{-2D_r(t_1-t_2)} \int_{4\pi} \frac{1}{3} \left(\sqrt{\frac{16\pi}{5}} Y_2^{0*}(\Omega_2) + \sqrt{4\pi} Y_0^{0*}(\Omega_2) \right)
\end{aligned}
$$

$$
\begin{aligned}
&\times \left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l} e^{-l(l+1)D_r(t_2-t_0)} Y_l^{m*}\left(\Omega_0\right) Y_l^m\left(\Omega_2\right)\right] d\Omega_2 \\
&= \frac{1}{3} e^{-2D_r(t_1-t_2)} \left[\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_2-t_0)} Y_2^{0*}\left(\Omega_0\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right)\right] \\
&= \frac{1}{3} e^{-2D_r(t_1-t_2)} \left[e^{-6D_r(t_2-t_0)} \left(3\cos^2\theta_0 - 1\right) + 1\right]. \qquad \text{(D.1)}
\end{aligned}
$$

Here $\langle \sin\theta_1 \cos\phi_1 \sin\theta_2 \cos\phi_2 \rangle_{t_1>t_2}$ is computed using Eq. 3.13 and the equations in Appendix C.

$$
\begin{aligned}
&\langle \sin\theta_1 \cos\phi_1 \sin\theta_2 \cos\phi_2 \rangle_{t_1>t_2} \\
&= \int_{4\pi}\int_{4\pi} \sin\theta_1 \cos\phi_1 \sin\theta_2 \cos\phi_2 P\left(\Omega_1 \leftarrow \Omega_2, t_1 - t_2\right) P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_1 d\Omega_2 \\
&= \frac{1}{2}\sqrt{\frac{8\pi}{3}} \int_{4\pi}\int_{4\pi} \left(Y_1^{-1*}\left(\Omega_1\right) - Y_1^{1*}\left(\Omega_1\right)\right) \left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l} e^{-l(l+1)D_r(t_1-t_2)} Y_l^{m*}\left(\Omega_2\right) Y_l^m\left(\Omega_1\right)\right] \\
&\quad \times d\Omega_1 \sin\theta_2 \cos\phi_2 P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_2 \\
&= \frac{1}{2}\sqrt{\frac{8\pi}{3}} e^{-2D_r(t_1-t_2)} \int_{4\pi} \left(Y_1^{-1*}\left(\Omega_2\right) - Y_1^{1*}\left(\Omega_2\right)\right) \sin\theta_2 \cos\phi_2 P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_2 \\
&= e^{-2D_r(t_1-t_2)} \int_{4\pi} \sin^2\theta_2 \cos^2\phi_2 P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_2 \\
&= e^{-2D_r(t_1-t_2)} \int_{4\pi} \left(\frac{1}{4}\sqrt{\frac{32\pi}{15}} \left(Y_2^{2*}\left(\Omega_2\right) + Y_2^{-2*}\left(\Omega_2\right)\right) - \frac{1}{6}\sqrt{\frac{16\pi}{5}} Y_2^{0*}\left(\Omega_2\right)\right. \\
&\quad \left. + \frac{1}{3}\sqrt{4\pi} Y_0^{0*}\left(\Omega_2\right)\right) \left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l} e^{-l(l+1)D_r(t_2-t_0)} Y_l^{m*}\left(\Omega_0\right) Y_l^m\left(\Omega_2\right)\right] d\Omega_2 \\
&= e^{-2D_r(t_1-t_2)} \left[\frac{1}{4}\sqrt{\frac{32\pi}{15}} e^{-6D_r(t_2-t_0)} \left(Y_2^{2*}\left(\Omega_0\right) + Y_2^{-2*}\left(\Omega_0\right)\right)\right. \\
&\quad \left. - \frac{1}{6}\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_2-t_0)} Y_2^{0*}\left(\Omega_0\right) + \frac{1}{3}\sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right)\right] \\
&= e^{-2D_r(t_1-t_2)} \left[\frac{1}{2} e^{-6D_r(t_2-t_0)} \sin^2\theta_0 \left(\cos^2\phi_0 - \sin^2\phi_0\right)\right. \\
&\quad \left. - \frac{1}{6} e^{-6D_r(t_2-t_0)} \left(3\cos^2\theta_0 - 1\right) + \frac{1}{3}\right]. \qquad \text{(D.2)}
\end{aligned}
$$

Here $\langle \sin\theta_1 \sin\phi_1 \sin\theta_2 \sin\phi_2 \rangle_{t_1>t_2}$ is computed using Eq. 3.13 and the equations in Appendix C.

$$
\begin{aligned}
&\langle \sin\theta_1 \sin\phi_1 \sin\theta_2 \sin\phi_2 \rangle_{t_1>t_2} \\
&\quad = \int_{4\pi}\int_{4\pi} \sin\theta_1 \sin\phi_1 \sin\theta_2 \sin\phi_2 P\left(\Omega_1 \leftarrow \Omega_2, t_1 - t_2\right) P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_1 d\Omega_2 \\
&\quad = \frac{1}{2i}\sqrt{\frac{8\pi}{3}} \int_{4\pi}\int_{4\pi} \left(Y_1^{-1*}\left(\Omega_1\right) + Y_1^{1*}\left(\Omega_1\right)\right) \left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l} e^{-l(l+1)D_r(t_1-t_2)} Y_l^{m*}\left(\Omega_2\right) Y_l^{m}\left(\Omega_1\right)\right] \\
&\qquad\quad \times d\Omega_1 \sin\theta_2 \sin\phi_2 P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_2 \\
&\quad = \frac{1}{2i}\sqrt{\frac{8\pi}{3}} e^{-2D_r(t_1-t_2)} \int_{4\pi} \left(Y_1^{-1*}\left(\Omega_2\right) + Y_1^{1*}\left(\Omega_2\right)\right) \sin\theta_2 \sin\phi_2 P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_2 \\
&\quad = e^{-2D_r(t_1-t_2)} \int_{4\pi} \sin^2\theta_2 \sin^2\phi_2 P\left(\Omega_2 \leftarrow \Omega_0, t_2 - t_0\right) d\Omega_2 \\
&\quad = e^{-2D_r(t_1-t_2)} \int_{4\pi} \Bigg(-\frac{1}{4}\sqrt{\frac{32\pi}{15}} \left(Y_2^{2*}\left(\Omega_2\right) + Y_2^{-2*}\left(\Omega_2\right)\right) - \frac{1}{6}\sqrt{\frac{16\pi}{5}} Y_2^{0*}\left(\Omega_2\right) \\
&\qquad\quad + \frac{1}{3}\sqrt{4\pi} Y_0^{0*}\left(\Omega_2\right) \Bigg) \left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l} e^{-l(l+1)D_r(t_2-t_0)} Y_l^{m*}\left(\Omega_0\right) Y_l^{m}\left(\Omega_2\right)\right] d\Omega_2 \\
&\quad = e^{-2D_r(t_1-t_2)} \Bigg[-\frac{1}{4}\sqrt{\frac{32\pi}{15}} e^{-6D_r(t_2-t_0)} \left(Y_2^{2*}\left(\Omega_0\right) + Y_2^{-2*}\left(\Omega_0\right)\right) \\
&\qquad\quad - \frac{1}{6}\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_2-t_0)} Y_2^{0*}\left(\Omega_0\right) + \frac{1}{3}\sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right) \Bigg] \\
&\quad = e^{-2D_r(t_1-t_2)} \Bigg[-\frac{1}{2} e^{-6D_r(t_2-t_0)} \sin^2\theta_0 \left(\cos^2\phi_0 - \sin^2\phi_0\right) \\
&\qquad\quad - \frac{1}{6} e^{-6D_r(t_2-t_0)} \left(3\cos^2\theta_0 - 1\right) + \frac{1}{3} \Bigg].
\end{aligned}
\tag{D.3}
$$

Using Eq. 3.18

$$
\langle \hat{\boldsymbol{u}}\left(t_1\right) \hat{\boldsymbol{u}}\left(t_2\right) \rangle_{t_1>t_2} = e^{-2D_r(t_1-t_2)}. \tag{D.4}
$$

The first summand in Eq. 3.17 is then

$$
\begin{aligned}
2!\int_0^t\int_0^{t_1} & F(t_1)\,F(t_2)\,\langle\hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\rangle_{t_1>t_2}dt_2dt_1\\
&= 2\int_0^t\int_0^{t_1}\left(\frac{\lambda M_0N_0}{2}\right)^2 e^{-\lambda(t_1+t_2)}e^{-2D_r(t_1-t_2)}dt_2dt_1\\
&= \frac{1}{2}(\lambda M_0N_0)^2\int_0^t e^{-(\lambda+2D_r)t_1}\int_0^{t_1}e^{-(\lambda-2D_r)t_2}dt_2dt_1\\
&= \frac{(\lambda M_0N_0)^2}{2(\lambda-2D_r)}\int_0^t e^{-(\lambda+2D_r)t_1}\left[1-e^{-(\lambda-2D_r)t_1}\right]dt_1\\
&= \frac{(\lambda M_0N_0)^2}{2(\lambda-2D_r)}\left[\frac{1}{\lambda+2D_r}\left(1-e^{-(\lambda+2D_r)t}\right)-\frac{1}{2\lambda}\left(1-e^{-2\lambda t}\right)\right].
\end{aligned}
\tag{D.5}
$$

D.2 Excess Kurtosis

Here $\langle\cos\theta_1\cos\theta_2\cos\theta_3\cos\theta_4\rangle_{t_1>t_2>t_3>t_4}$ is computed starting with Eq. D.1 with $(t_0,\theta_0)\to(t_3,\theta_3)$, and using Eq. 3.13 and the equations in Appendix C.

$$
\begin{aligned}
&\langle\cos\theta_1\cos\theta_2\cos\theta_3\cos\theta_4\rangle_{t_1>t_2>t_3>t_4}\\
&\quad=\frac{1}{3}e^{-2D_r(t_1-t_2)}\int_{4\pi}\int_{4\pi}\left[e^{-6D_r(t_2-t_3)}\left(3\cos^2\theta_3-1\right)+1\right]\\
&\qquad\times\cos\theta_3\cos\theta_4P(\Omega_3\leftarrow\Omega_4,t_3-t_4)\,P(\Omega_4\leftarrow\Omega_0,t_4-t_0)\,d\Omega_3d\Omega_4\\
&\quad=\frac{1}{3}e^{-2D_r(t_1-t_2)}\int_{4\pi}\int_{4\pi}\left[e^{-6D_r(t_2-t_3)}\left(\frac{1}{5}\left[4\sqrt{\frac{\pi}{7}}Y_3^{0*}(\Omega_3)+3\sqrt{\frac{4\pi}{3}}Y_1^{0*}(\Omega_3)\right]\right.\right.\\
&\qquad\left.\left.-\sqrt{\frac{4\pi}{3}}Y_1^{0*}(\Omega_3)\right)+\sqrt{\frac{4\pi}{3}}Y_1^{0*}(\Omega_3)\right]\left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l}e^{-l(l+1)D_r(t_3-t_4)}Y_l^{m*}(\Omega_4)\,Y_l^m(\Omega_3)\right]\\
&\qquad\times\cos\theta_4P(\Omega_4\leftarrow\Omega_0,t_4-t_0)\,d\Omega_3d\Omega_4\\
&\quad=\frac{1}{3}e^{-2D_r(t_1-t_2)}\int_{4\pi}\left[e^{-6D_r(t_2-t_3)}\left(\frac{12}{5}\sqrt{\frac{\pi}{7}}e^{-12D_r(t_3-t_4)}Y_3^{0*}(\Omega_4)\right.\right.\\
&\qquad\left.\left.+\frac{4}{5}\sqrt{\frac{4\pi}{3}}e^{-2D_r(t_3-t_4)}Y_1^{0*}(\Omega_4)\right)+\sqrt{\frac{4\pi}{3}}e^{-2D_r(t_3-t_4)}Y_1^{0*}(\Omega_4)\right]\\
&\qquad\times\cos\theta_4P(\Omega_4\leftarrow\Omega_0,t_4-t_0)\,d\Omega_4\\
&\quad=e^{-2D_r(t_1-t_2)}\int_{4\pi}\left[e^{-6D_r(t_2-t_3)}\left(e^{-12D_r(t_3-t_4)}\left\{\cos^3\theta_4-\frac{3}{5}\cos\theta_4\right\}\right.\right.
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{4}{15} e^{-2D_r(t_3-t_4)} \cos\theta_4 \Bigg) + \frac{1}{3} e^{-2D_r(t_3-t_4)} \cos\theta_4 \Bigg] \cos\theta_4 P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right) d\Omega_4 \\
= {} & e^{-2D_r(t_1-t_2)} \int_{4\pi} \Bigg[e^{-6D_r(t_2-t_3)} \Bigg(e^{-12D_r(t_3-t_4)} \Bigg\{ \frac{1}{35} \Bigg[\frac{16}{3}\sqrt{\pi} Y_4^{0*}\left(\Omega_4\right) + 10\sqrt{\frac{16\pi}{5}} Y_2^{0*}\left(\Omega_4\right) \\
&+ 7\sqrt{4\pi} Y_0^{0*}\left(\Omega_4\right) \Bigg] - \frac{1}{5} \left(\sqrt{\frac{16\pi}{5}} Y_2^{0*}\left(\Omega_4\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_4\right) \right) \Bigg\} \\
&+ \frac{4}{45} e^{-2D_r(t_3-t_4)} \left(\sqrt{\frac{16\pi}{5}} Y_2^{0*}\left(\Omega_4\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_4\right) \right) \Bigg) \\
&+ \frac{1}{9} e^{-2D_r(t_3-t_4)} \left(\sqrt{\frac{16\pi}{5}} Y_2^{0*}\left(\Omega_4\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_4\right) \right) \Bigg] \\
&\times \left[\sum_{l=0}^{\infty} \sum_{m=-l}^{l} e^{-l(l+1)D_r(t_4-t_0)} Y_l^{m*}\left(\Omega_0\right) Y_l^{m}\left(\Omega_4\right) \right] d\Omega_4 \\
= {} & e^{-2D_r(t_1-t_2)} \Bigg[e^{-6D_r(t_2-t_3)} \Bigg(e^{-12D_r(t_3-t_4)} \Bigg\{ \frac{1}{35} \Bigg[\frac{16}{3}\sqrt{\pi} e^{-20D_r(t_4-t_0)} Y_4^{0*}\left(\Omega_0\right) \\
&+ 10\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_4-t_0)} Y_2^{0*}\left(\Omega_0\right) + 7\sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right) \Bigg] \\
&- \frac{1}{5} \left(\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_4-t_0)} Y_2^{0*}\left(\Omega_0\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right) \right) \Bigg\} \\
&+ \frac{4}{45} e^{-2D_r(t_3-t_4)} \left(\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_4-t_0)} Y_2^{0*}\left(\Omega_0\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right) \right) \Bigg) \\
&+ \frac{1}{9} e^{-2D_r(t_3-t_4)} \left(\sqrt{\frac{16\pi}{5}} e^{-6D_r(t_4-t_0)} Y_2^{0*}\left(\Omega_0\right) + \sqrt{4\pi} Y_0^{0*}\left(\Omega_0\right) \right) \Bigg] \\
= {} & e^{-2D_r(t_1-t_2)} \Bigg[e^{-6D_r(t_2-t_3)} \Bigg(e^{-12D_r(t_3-t_4)} \Bigg\{ \frac{1}{35} \Bigg[e^{-20D_r(t_4-t_0)} \left(35\cos^4\theta_0 - 30\cos^2\theta_0 + 3\right) \\
&+ 10 e^{-6D_r(t_4-t_0)} \left(3\cos^2\theta_0 - 1\right) + 7 \Bigg] - \frac{1}{5} \left(e^{-6D_r(t_4-t_0)} \left(3\cos^2\theta_0 - 1\right) + 1 \right) \Bigg\} \\
&+ \frac{4}{45} e^{-2D_r(t_3-t_4)} \left(e^{-6D_r(t_4-t_0)} \left(3\cos^2\theta_0 - 1\right) + 1 \right) \Bigg) \\
&+ \frac{1}{9} e^{-2D_r(t_3-t_4)} \left(e^{-6D_r(t_4-t_0)} \left(3\cos^2\theta_0 - 1\right) + 1 \right) \Bigg] \\
= {} & e^{-2D_r(t_1-t_2)} \Bigg[e^{-6D_r(t_2-t_3)} \Bigg(e^{-12D_r(t_3-t_4)} \Bigg\{ e^{-20D_r(t_4-t_0)} \left(\cos^4\theta_0 - \frac{6}{7}\cos^2\theta_0 + \frac{3}{35} \right)
\end{aligned}
$$

$$+\frac{3}{35}e^{-6D_r(t_4-t_0)}\left(3\cos^2\theta_0-1\right)\Bigg\}+\frac{4}{45}e^{-2D_r(t_3-t_4)}\left(e^{-6D_r(t_4-t_0)}\left(3\cos^2\theta_0-1\right)+1\right)\Bigg)$$

$$+\frac{1}{9}e^{-2D_r(t_3-t_4)}\left(e^{-6D_r(t_4-t_0)}\left(3\cos^2\theta_0-1\right)+1\right)\Bigg]. \tag{D.6}$$

Here $\langle\sin\theta_1\cos\phi_1\sin\theta_2\cos\phi_2\sin\theta_3\cos\phi_3\sin\theta_4\cos\phi_4\rangle_{t_1>t_2>t_3>t_4}$ is computed starting with Eq. D.2 with $(t_0,\theta_0)\to(t_3,\theta_3)$, and using Eq. 3.13 and the equations in Appendix C.

$$\langle\sin\theta_1\cos\phi_1\sin\theta_2\cos\phi_2\sin\theta_3\cos\phi_3\sin\theta_4\cos\phi_4\rangle_{t_1>t_2>t_3>t_4}$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\int_{4\pi}\Bigg[\frac{1}{2}e^{-6D_r(t_2-t_3)}\sin^2\theta_3\left(\cos^2\phi_3-\sin^2\phi_3\right)$$

$$-\frac{1}{6}e^{-6D_r(t_2-t_3)}\left(3\cos^2\theta_3-1\right)+\frac{1}{3}\Bigg]\sin\theta_3\cos\phi_3\sin\theta_4\cos\phi_4$$

$$\times P\left(\Omega_3\leftarrow\Omega_4,t_3-t_4\right)P\left(\Omega_4\leftarrow\Omega_0,t_4-t_0\right)d\Omega_3d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\int_{4\pi}\left[e^{-6D_r(t_2-t_3)}\left(\sin^3\theta_3\cos^3\phi_3-\frac{1}{3}\sin\theta_3\cos\phi_3\right)+\frac{1}{3}\sin\theta_3\cos\phi_3\right]$$

$$\times\sin\theta_4\cos\phi_4P\left(\Omega_3\leftarrow\Omega_4,t_3-t_4\right)P\left(\Omega_4\leftarrow\Omega_0,t_4-t_0\right)d\Omega_3d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\int_{4\pi}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(\sqrt{\frac{\pi}{35}}\left(Y_3^{-3*}(\Omega_3)-Y_3^{3*}(\Omega_3)\right)$$

$$+\frac{3}{5}\sqrt{\frac{\pi}{21}}\left(Y_3^{1*}(\Omega_3)-Y_3^{-1*}(\Omega_3)\right)-\frac{3}{10}\sqrt{\frac{8\pi}{3}}\left(Y_1^{1*}(\Omega_3)-Y_1^{-1*}(\Omega_3)\right)$$

$$-\frac{1}{6}\sqrt{\frac{8\pi}{3}}\left(Y_1^{-1*}(\Omega_3)-Y_1^{1*}(\Omega_3)\right)\Bigg)+\frac{1}{6}\sqrt{\frac{8\pi}{3}}\left(Y_1^{-1*}(\Omega_3)-Y_1^{1*}(\Omega_3)\right)\Bigg]$$

$$\times\left[\sum_{l=0}^{\infty}\sum_{m=-l}^{l}e^{-l(l+1)D_r(t_3-t_4)}Y_l^{m*}(\Omega_4)Y_l^m(\Omega_3)\right]$$

$$\times\sin\theta_4\cos\phi_4P\left(\Omega_4\leftarrow\Omega_0,t_4-t_0\right)d\Omega_3d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\sqrt{\frac{\pi}{35}}\left(Y_3^{-3*}(\Omega_4)-Y_3^{3*}(\Omega_4)\right)$$

$$+\frac{3}{5}\sqrt{\frac{\pi}{21}}\left(Y_3^{1*}(\Omega_4)-Y_3^{-1*}(\Omega_4)\right)\Bigg\}+\frac{2}{15}\sqrt{\frac{8\pi}{3}}e^{-2D_r(t_3-t_4)}\left(Y_1^{-1*}(\Omega_4)-Y_1^{1*}(\Omega_4)\right)\Bigg)$$

$$+\frac{1}{6}\sqrt{\frac{8\pi}{3}}e^{-2D_r(t_3-t_4)}\left(Y_1^{-1*}\left(\Omega_4\right)-Y_1^{1*}\left(\Omega_4\right)\right)\Bigg]\sin\theta_4\cos\phi_4 P\left(\Omega_4\leftarrow\Omega_0,t_4-t_0\right)d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\frac{1}{4}\sin^3\theta_4\big(4\cos^3\phi_4-3\cos\phi_4\big)$$

$$-\frac{3}{20}\sin\theta_4\left(5\cos^2\theta_4-1\right)\cos\phi_4\Bigg\}+\frac{4}{15}e^{-2D_r(t_3-t_4)}\sin\theta_4\cos\phi_4\Bigg)$$

$$+\frac{1}{3}e^{-2D_r(t_3-t_4)}\sin\theta_4\cos\phi_4\Bigg]\sin\theta_4\cos\phi_4 P\left(\Omega_4\leftarrow\Omega_0,t_4-t_0\right)d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\sin^3\theta_4\cos^3\phi_4-\frac{3}{5}\sin\theta_4\cos\phi_4\Bigg\}$$

$$+\frac{4}{15}e^{-2D_r(t_3-t_4)}\sin\theta_4\cos\phi_4\Bigg)+\frac{1}{3}e^{-2D_r(t_3-t_4)}\sin\theta_4\cos\phi_4\Bigg]$$

$$\times\sin\theta_4\cos\phi_4 P\left(\Omega_4\leftarrow\Omega_0,t_4-t_0\right)d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\int_{4\pi}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\frac{1}{16}\Bigg[\frac{16}{3}\sqrt{\frac{2\pi}{35}}\big(Y_4^{4*}\left(\Omega_4\right)+Y_4^{-4*}\left(\Omega_4\right)\big)$$

$$-\frac{32}{21}\sqrt{\frac{2\pi}{5}}\left(Y_4^{2*}\left(\Omega_4\right)+Y_4^{-2*}\left(\Omega_4\right)\right)+\frac{24}{7}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)$$

$$+\frac{6}{35}\Bigg(\frac{16}{3}\sqrt{\pi}Y_4^{0*}\left(\Omega_4\right)-\frac{40}{3}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)+\frac{56}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg)\Bigg]-\frac{3}{5}\Bigg[\frac{1}{4}\sqrt{\frac{32\pi}{15}}$$

$$\times\big(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\big)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg]\Bigg\}+\frac{4}{15}e^{-2D_r(t_3-t_4)}$$

$$\times\Bigg[\frac{1}{4}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg]\Bigg)$$

$$+\frac{1}{3}e^{-2D_r(t_3-t_4)}\Bigg[\frac{1}{4}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)$$

$$+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg]\Bigg]\Bigg[\sum_{l=0}^{\infty}\sum_{m=-l}^{l}e^{-l(l+1)D_r(t_4-t_0)}Y_l^{m*}\left(\Omega_0\right)Y_l^m\left(\Omega_4\right)\Bigg]d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\frac{1}{16}\Bigg[e^{-20D_r(t_4-t_0)}\Bigg(\frac{16}{3}\sqrt{\frac{2\pi}{35}}\big(Y_4^{4*}\left(\Omega_0\right)$$

$$+Y_4^{-4*}\left(\Omega_0\right)\big)-\frac{32}{21}\sqrt{\frac{2\pi}{5}}\left(Y_4^{2*}\left(\Omega_0\right)+Y_4^{-2*}\left(\Omega_0\right)\right)\Bigg)+\frac{24}{7}\sqrt{\frac{32\pi}{15}}e^{-6D_r(t_4-t_0)}\big(Y_2^{2*}\left(\Omega_0\right)$$

$$
\begin{aligned}
&+ Y_2^{-2*}(\Omega_0)\big) + \frac{6}{35}\Bigg(\frac{16}{3}\sqrt{\pi}e^{-20D_r(t_4-t_0)}Y_4^{0*}(\Omega_0) - \frac{40}{3}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}(\Omega_0) \\
&+ \frac{56}{3}\sqrt{4\pi}Y_0^{0*}(\Omega_0)\Bigg)\Bigg] - \frac{3}{5}\Bigg[\frac{1}{4}\sqrt{\frac{32\pi}{15}}e^{-6D_r(t_4-t_0)}\big(Y_2^{2*}(\Omega_0) + Y_2^{-2*}(\Omega_0)\big) \\
&- \frac{1}{6}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}(\Omega_0) + \frac{1}{3}\sqrt{4\pi}Y_0^{0*}(\Omega_0)\Bigg]\Bigg\} + \frac{4}{15}e^{-2D_r(t_3-t_4)}\Bigg[\frac{1}{4}\sqrt{\frac{32\pi}{15}} \\
&\times e^{-6D_r(t_4-t_0)}\left(Y_2^{2*}(\Omega_0) + Y_2^{-2*}(\Omega_0)\right) - \frac{1}{6}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}(\Omega_0) \\
&+ \frac{1}{3}\sqrt{4\pi}Y_0^{0*}(\Omega_0)\Bigg]\Bigg) + \frac{1}{3}e^{-2D_r(t_3-t_4)}\Bigg[\frac{1}{4}\sqrt{\frac{32\pi}{15}}e^{-6D_r(t_4-t_0)}\left(Y_2^{2*}(\Omega_0) + Y_2^{-2*}(\Omega_0)\right) \\
&- \frac{1}{6}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}(\Omega_0) + \frac{1}{3}\sqrt{4\pi}Y_0^{0*}(\Omega_0)\Bigg]\Bigg] \\
= {} & e^{-2D_r(t_1-t_2)}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\frac{1}{16}\Bigg[e^{-20D_r(t_4-t_0)}\Bigg(2\sin^4\theta_0\cos(4\phi_0) \\
&- \frac{8}{7}\sin^2\theta_0\left(7\cos^2\theta_0 - 1\right)\cos(2\phi_0) + \frac{6}{35}\left(35\cos^4\theta_0 - 30\cos^2\theta_0 + 3\right)\Bigg)\Bigg] \\
&+ e^{-6D_r(t_4-t_0)}\Bigg(\frac{9}{70}\sin^2\theta_0\cos(2\phi_0) - \frac{3}{70}\left(3\cos^2\theta_0 - 1\right)\Bigg)\Bigg\} + \frac{4}{15}e^{-2D_r(t_3-t_4)} \\
&\times \Bigg[e^{-6D_r(t_4-t_0)}\Bigg(\frac{1}{2}\sin^2\theta_0\cos(2\phi_0) - \frac{1}{6}\left(3\cos^2\theta_0 - 1\right)\Bigg) + \frac{1}{3}\Bigg]\Bigg) + \frac{1}{3}e^{-2D_r(t_3-t_4)} \\
&\times \Bigg[e^{-6D_r(t_4-t_0)}\Bigg(\frac{1}{2}\sin^2\theta_0\cos(2\phi_0) - \frac{1}{6}\left(3\cos^2\theta_0 - 1\right)\Bigg) + \frac{1}{3}\Bigg]\Bigg]. \qquad \text{(D.7)}
\end{aligned}
$$

Here $\langle \sin\theta_1 \sin\phi_1 \sin\theta_2 \sin\phi_2 \sin\theta_3 \sin\phi_3 \sin\theta_4 \sin\phi_4 \rangle_{t_1>t_2>t_3>t_4}$ is computed starting with Eq. D.3 with $(t_0, \theta_0) \to (t_3, \theta_3)$, and using Eq. 3.13 and the equations in Appendix C.

$$
\begin{aligned}
&\langle \sin\theta_1 \sin\phi_1 \sin\theta_2 \sin\phi_2 \sin\theta_3 \sin\phi_3 \sin\theta_4 \sin\phi_4 \rangle_{t_1>t_2>t_3>t_4} \\
&\quad = e^{-2D_r(t_1-t_2)}\int_{4\pi}\int_{4\pi}\Bigg[-\frac{1}{2}e^{-6D_r(t_2-t_3)}\sin^2\theta_3\left(\cos^2\phi_3 - \sin^2\phi_3\right) \\
&\quad - \frac{1}{6}e^{-6D_r(t_2-t_3)}\left(3\cos^2\theta_3 - 1\right) + \frac{1}{3}\Bigg]\sin\theta_3\sin\phi_3\sin\theta_4\sin\phi_4 \\
&\quad \times P\left(\Omega_3 \leftarrow \Omega_4, t_3 - t_4\right)P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right)d\Omega_3 d\Omega_4
\end{aligned}
$$

$$= e^{-2D_r(t_1-t_2)} \int_{4\pi}\int_{4\pi} \left[e^{-6D_r(t_2-t_3)} \left(\sin^3\theta_3 \sin^3\phi_3 - \frac{1}{3}\sin\theta_3\sin\phi_3 \right) + \frac{1}{3}\sin\theta_3\sin\phi_3 \right]$$

$$\times \sin\theta_4 \sin\phi_4 P\left(\Omega_3 \leftarrow \Omega_4, t_3 - t_4\right) P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right) d\Omega_3 d\Omega_4$$

$$= e^{-2D_r(t_1-t_2)} \int_{4\pi}\int_{4\pi} \left[e^{-6D_r(t_2-t_3)} \left(\frac{-1}{i} \left[\sqrt{\frac{\pi}{35}} \left(Y_3^{-3*}(\Omega_3) + Y_3^{3*}(\Omega_3) \right) \right.\right.\right.$$

$$\left. + \frac{3}{5}\sqrt{\frac{\pi}{21}} \left(Y_3^{1*}(\Omega_3) + Y_3^{-1*}(\Omega_3) \right) - \frac{3}{10}\sqrt{\frac{8\pi}{3}} \left(Y_1^{1*}(\Omega_3) + Y_1^{-1*}(\Omega_3) \right) \right]$$

$$\left.\left. - \frac{1}{6i}\sqrt{\frac{8\pi}{3}} \left(Y_1^{-1*}(\Omega_3) + Y_1^{1*}(\Omega_3) \right) \right) + \frac{1}{6i}\sqrt{\frac{8\pi}{3}} \left(Y_1^{-1*}(\Omega_3) + Y_1^{1*}(\Omega_3) \right) \right]$$

$$\times \left[\sum_{l=0}^{\infty} \sum_{m=-l}^{l} e^{-l(l+1)D_r(t_3-t_4)} Y_l^{m*}(\Omega_4) Y_l^m(\Omega_3) \right]$$

$$\times \sin\theta_4 \sin\phi_4 P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right) d\Omega_3 d\Omega_4$$

$$= e^{-2D_r(t_1-t_2)} \int_{4\pi} \left[e^{-6D_r(t_2-t_3)} \left(e^{-12D_r(t_3-t_4)} \left\{ \frac{-1}{i} \left[\sqrt{\frac{\pi}{35}} \left(Y_3^{-3*}(\Omega_4) + Y_3^{3*}(\Omega_4) \right) + \frac{3}{5} \right.\right.\right.\right.$$

$$\left.\left. \times \sqrt{\frac{\pi}{21}} \left(Y_3^{1*}(\Omega_4) + Y_3^{-1*}(\Omega_4) \right) \right] \right\} + \frac{2}{15i}\sqrt{\frac{8\pi}{3}} e^{-2D_r(t_3-t_4)} \left(Y_1^{-1*}(\Omega_4) + Y_1^{1*}(\Omega_4) \right) \Bigg)$$

$$\left. + \frac{1}{6i}\sqrt{\frac{8\pi}{3}} e^{-2D_r(t_3-t_4)} \left(Y_1^{-1*}(\Omega_4) + Y_1^{1*}(\Omega_4) \right) \right] \sin\theta_4 \sin\phi_4 P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right) d\Omega_4$$

$$= e^{-2D_r(t_1-t_2)} \int_{4\pi} \left[e^{-6D_r(t_2-t_3)} \left(e^{-12D_r(t_3-t_4)} \left\{ \frac{1}{4}\sin^3\theta_4 \left(4\sin^3\phi_4 - 3\sin\phi_4 \right) \right.\right.\right.$$

$$\left. - \frac{3}{20}\sin\theta_4 \left(5\cos^2\theta_4 - 1 \right) \sin\phi_4 \right\} + \frac{4}{15} e^{-2D_r(t_3-t_4)} \sin\theta_4 \sin\phi_4 \Bigg)$$

$$\left. + \frac{1}{3} e^{-2D_r(t_3-t_4)} \sin\theta_4 \sin\phi_4 \right] \sin\theta_4 \sin\phi_4 P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right) d\Omega_4$$

$$= e^{-2D_r(t_1-t_2)} \int_{4\pi} \left[e^{-6D_r(t_2-t_3)} \left(e^{-12D_r(t_3-t_4)} \left\{ \sin^3\theta_4 \sin^3\phi_4 - \frac{3}{5}\sin\theta_4 \sin\phi_4 \right\} \right.\right.$$

$$\left.\left. + \frac{4}{15} e^{-2D_r(t_3-t_4)} \sin\theta_4 \sin\phi_4 \right) + \frac{1}{3} e^{-2D_r(t_3-t_4)} \sin\theta_4 \sin\phi_4 \right]$$

$$\times \sin\theta_4 \sin\phi_4 P\left(\Omega_4 \leftarrow \Omega_0, t_4 - t_0\right) d\Omega_4$$

$$= e^{-2D_r(t_1-t_2)} \int_{4\pi} \left[e^{-6D_r(t_2-t_3)} \left(e^{-12D_r(t_3-t_4)} \left\{ \frac{1}{16} \left[\frac{16}{3}\sqrt{\frac{2\pi}{35}} \left(Y_4^{4*}(\Omega_4) + Y_4^{-4*}(\Omega_4) \right) \right.\right.\right.\right.$$

$$+\frac{32}{21}\sqrt{\frac{2\pi}{5}}\left(Y_4^{2*}\left(\Omega_4\right)+Y_4^{-2*}\left(\Omega_4\right)\right)-\frac{24}{7}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)$$

$$+\frac{6}{35}\Bigg(\frac{16}{3}\sqrt{\pi}Y_4^{0*}\left(\Omega_4\right)-\frac{40}{3}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)+\frac{56}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg)\Bigg]-\frac{3}{5}\Bigg[\frac{-1}{4}\sqrt{\frac{32\pi}{15}}$$

$$\times\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg]\Bigg\}+\frac{4}{15}e^{-2D_r(t_3-t_4)}$$

$$\times\Bigg[\frac{-1}{4}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg]\Bigg)$$

$$+\frac{1}{3}e^{-2D_r(t_3-t_4)}\Bigg[\frac{-1}{4}\sqrt{\frac{32\pi}{15}}\left(Y_2^{2*}\left(\Omega_4\right)+Y_2^{-2*}\left(\Omega_4\right)\right)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}Y_2^{0*}\left(\Omega_4\right)$$

$$+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_4\right)\Bigg]\Bigg]\Bigg[\sum_{l=0}^{\infty}\sum_{m=-l}^{l}e^{-l(l+1)D_r(t_4-t_0)}Y_l^{m*}\left(\Omega_0\right)Y_l^{m}\left(\Omega_4\right)\Bigg]d\Omega_4$$

$$=e^{-2D_r(t_1-t_2)}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\frac{1}{16}\Bigg[e^{-20D_r(t_4-t_0)}\Bigg(\frac{16}{3}\sqrt{\frac{2\pi}{35}}\big(Y_4^{4*}\left(\Omega_0\right)$$

$$+Y_4^{-4*}\left(\Omega_0\right)\big)+\frac{32}{21}\sqrt{\frac{2\pi}{5}}\left(Y_4^{2*}\left(\Omega_0\right)+Y_4^{-2*}\left(\Omega_0\right)\right)\Bigg)-\frac{24}{7}\sqrt{\frac{32\pi}{15}}e^{-6D_r(t_4-t_0)}\big(Y_2^{2*}\left(\Omega_0\right)$$

$$+Y_2^{-2*}\left(\Omega_0\right)\big)+\frac{6}{35}\Bigg(\frac{16}{3}\sqrt{\pi}e^{-20D_r(t_4-t_0)}Y_4^{0*}\left(\Omega_0\right)-\frac{40}{3}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}\left(\Omega_0\right)$$

$$+\frac{56}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_0\right)\Bigg)\Bigg]-\frac{3}{5}\Bigg[\frac{-1}{4}\sqrt{\frac{32\pi}{15}}e^{-6D_r(t_4-t_0)}\big(Y_2^{2*}\left(\Omega_0\right)+Y_2^{-2*}\left(\Omega_0\right)\big)$$

$$-\frac{1}{6}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}\left(\Omega_0\right)+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_0\right)\Bigg]\Bigg\}+\frac{4}{15}e^{-2D_r(t_3-t_4)}\Bigg[\frac{-1}{4}\sqrt{\frac{32\pi}{15}}$$

$$\times e^{-6D_r(t_4-t_0)}\left(Y_2^{2*}\left(\Omega_0\right)+Y_2^{-2*}\left(\Omega_0\right)\right)-\frac{1}{6}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}\left(\Omega_0\right)$$

$$+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_0\right)\Bigg]\Bigg)+\frac{1}{3}e^{-2D_r(t_3-t_4)}\Bigg[\frac{-1}{4}\sqrt{\frac{32\pi}{15}}e^{-6D_r(t_4-t_0)}\left(Y_2^{2*}\left(\Omega_0\right)+Y_2^{-2*}\left(\Omega_0\right)\right)$$

$$-\frac{1}{6}\sqrt{\frac{16\pi}{5}}e^{-6D_r(t_4-t_0)}Y_2^{0*}\left(\Omega_0\right)+\frac{1}{3}\sqrt{4\pi}Y_0^{0*}\left(\Omega_0\right)\Bigg]\Bigg]$$

$$=e^{-2D_r(t_1-t_2)}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{\frac{1}{16}e^{-20D_r(t_4-t_0)}\Bigg(2\sin^4\theta_0\cos\left(4\phi_0\right)$$

$$+\frac{8}{7}\sin^2\theta_0\left(7\cos^2\theta_0-1\right)\cos\left(2\phi_0\right)+\frac{6}{35}\left(35\cos^4\theta_0-30\cos^2\theta_0+3\right)\Bigg)$$

$$
\begin{aligned}
&+ e^{-6D_r(t_4-t_0)}\left(\frac{-9}{70}\sin^2\theta_0\cos(2\phi_0) - \frac{3}{70}\left(3\cos^2\theta_0 - 1\right)\right)\Bigg\} + \frac{4}{15}e^{-2D_r(t_3-t_4)} \\
&\times \left[e^{-6D_r(t_4-t_0)}\left(\frac{-1}{2}\sin^2\theta_0\cos(2\phi_0) - \frac{1}{6}\left(3\cos^2\theta_0 - 1\right)\right) + \frac{1}{3}\right]\Bigg) + \frac{1}{3}e^{-2D_r(t_3-t_4)} \\
&\times \left[e^{-6D_r(t_4-t_0)}\left(\frac{-1}{2}\sin^2\theta_0\cos(2\phi_0) - \frac{1}{6}\left(3\cos^2\theta_0 - 1\right)\right) + \frac{1}{3}\right]\Bigg]. \qquad \text{(D.8)}
\end{aligned}
$$

Then the total ensemble average is

$$
\begin{aligned}
&\langle \hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\,\hat{\boldsymbol{u}}(t_3)\,\hat{\boldsymbol{u}}(t_4)\rangle_{t_1>t_2>t_3>t_4} \\
&= e^{-2D_r(t_1-t_2)}\Bigg[e^{-6D_r(t_2-t_3)}\Bigg(e^{-12D_r(t_3-t_4)}\Bigg\{e^{-20D_r(t_4-t_0)}\Bigg(\frac{1}{4}\sin^4\theta_0\cos(4\phi_0) \qquad \text{(D.9)} \\
&\quad + \frac{7}{4}\cos^4\theta_0 - \frac{3}{2}\cos^2\theta_0 + \frac{3}{20}\Bigg)\Bigg\} + \frac{4}{15}e^{-2D_r(t_3-t_4)}\Bigg) + \frac{1}{3}e^{-2D_r(t_3-t_4)}\Bigg].
\end{aligned}
$$

Averaging over the initial particle orientation results in

$$
\begin{aligned}
&\langle \hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\,\hat{\boldsymbol{u}}(t_3)\,\hat{\boldsymbol{u}}(t_4)\rangle_{\hat{\boldsymbol{u}}_0,t_1>t_2>t_3>t_4} \\
&= \frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi}\langle \hat{\boldsymbol{u}}(t_1)\,\hat{\boldsymbol{u}}(t_2)\,\hat{\boldsymbol{u}}(t_3)\,\hat{\boldsymbol{u}}(t_4)\rangle_{t_1>t_2>t_3>t_4}\sin\theta_0 d\theta_0 d\phi_0 \qquad \text{(D.10)} \\
&= e^{-2D_r(t_1-t_2+t_3-t_4)}\left[\frac{4}{15}e^{-6D_r(t_2-t_3)} + \frac{1}{3}\right].
\end{aligned}
$$

The first summand in Eq. 3.24 is

$$
\begin{aligned}
&24\int_0^t\int_0^{t_1}\int_0^{t_2}\int_0^{t_3}\left(\frac{\lambda M_0 N_0}{2}\right)^4 e^{-\lambda(t_1+t_2+t_3+t_4)}e^{-2D_r(t_1-t_2+t_3-t_4)} \\
&\qquad \times \left[\frac{4}{15}e^{-6D_r(t_2-t_3)} + \frac{1}{3}\right]dt_4dt_3dt_2dt_1 \\
&= (\lambda M_0 N_0)^4\int_0^t e^{-(\lambda+2D_r)t_1}\int_0^{t_1} e^{-(\lambda-2D_r)t_2}\int_0^{t_2}\left[\frac{2}{5}e^{-6D_r t_2}e^{-(\lambda-4D_r)t_3} + \frac{1}{2}e^{-(\lambda+2D_r)t_3}\right] \\
&\qquad \times \int_0^{t_3} e^{-(\lambda-2D_r)t_4}dt_4dt_3dt_2dt_1 \\
&= \frac{(\lambda M_0 N_0)^4}{\lambda - 2D_r}\int_0^t e^{-(\lambda+2D_r)t_1}\int_0^{t_1} e^{-(\lambda-2D_r)t_2}\int_0^{t_2}\Bigg[\frac{2}{5}e^{-6D_r t_2}e^{-(\lambda-4D_r)t_3}
\end{aligned}
$$

$$
\begin{aligned}
&\times \left(1 - e^{-(\lambda-2D_r)t_3}\right) + \frac{1}{2} e^{-(\lambda+2D_r)t_3} \left(1 - e^{-(\lambda-2D_r)t_3}\right) \Bigg] dt_3 dt_2 dt_1 \\
&= \frac{(\lambda M_0 N_0)^4}{\lambda - 2D_r} \int_0^t e^{-(\lambda+2D_r)t_1} \int_0^{t_1} e^{-(\lambda-2D_r)t_2} \Bigg[\frac{2}{5} e^{-6D_r t_2} \Bigg(\frac{1}{\lambda - 4D_r} \\
&\quad \times \left\{ 1 - e^{-(\lambda-4D_r)t_2} \right\} - \frac{1}{2\lambda - 6D_r} \left\{ 1 - e^{-(2\lambda-6D_r)t_2} \right\} \Bigg) \\
&\quad + \frac{1}{2} \Bigg(\frac{1}{\lambda + 2D_r} \left\{ 1 - e^{-(\lambda+2D_r)t_2} \right\} - \frac{1}{2\lambda} \left\{ 1 - e^{-2\lambda t_2} \right\} \Bigg) \Bigg] dt_2 dt_1 \\
&= \frac{(\lambda M_0 N_0)^4}{\lambda - 2D_r} \int_0^t e^{-(\lambda+2D_r)t_1} \Bigg[\frac{2}{5} \Bigg(\frac{1}{\lambda - 4D_r} \Bigg\{ \frac{1}{\lambda + 4D_r} \left[1 - e^{-(\lambda+4D_r)t_1} \right] \\
&\quad - \frac{1}{2\lambda} \left[1 - e^{-2\lambda t_1} \right] \Bigg\} - \frac{1}{2\lambda - 6D_r} \Bigg\{ \frac{1}{\lambda + 4D_r} \left[1 - e^{-(\lambda+4D_r)t_1} \right] \\
&\quad - \frac{1}{3\lambda - 2D_r} \left[1 - e^{-(3\lambda-2D_r)t_1} \right] \Bigg\} \Bigg) + \frac{1}{2} \Bigg(\frac{1}{\lambda + 2D_r} \Bigg\{ \frac{1}{\lambda - 2D_r} \\
&\quad \times \left[1 - e^{-(\lambda-2D_r)t_1} \right] - \frac{1}{2\lambda} \left[1 - e^{-2\lambda t_1} \right] \Bigg\} - \frac{1}{2\lambda} \Bigg\{ \frac{1}{\lambda - 2D_r} \\
&\quad \times \left[1 - e^{-(\lambda-2D_r)t_1} \right] - \frac{1}{3\lambda - 2D_r} \left[1 - e^{-(3\lambda-2D_r)t_1} \right] \Bigg\} \Bigg) \Bigg] dt_1 \\
&= \frac{(\lambda M_0 N_0)^4}{\lambda - 2D_r} \Bigg[\frac{2}{5} \Bigg(\frac{1}{\lambda - 4D_r} \Bigg\{ \frac{1}{\lambda + 4D_r} \Bigg[\frac{1}{\lambda + 2D_r} \left(1 - e^{-(\lambda+2D_r)t} \right) \\
&\quad - \frac{1}{2\lambda + 6D_r} \left(1 - e^{-(2\lambda+6D_r)t} \right) \Bigg] - \frac{1}{2\lambda} \Bigg[\frac{1}{\lambda + 2D_r} \left(1 - e^{-(\lambda+2D_r)t} \right) \\
&\quad - \frac{1}{3\lambda + 2D_r} \left(1 - e^{-(3\lambda+2D_r)t} \right) \Bigg] \Bigg\} - \frac{1}{2\lambda - 6D_r} \Bigg\{ \frac{1}{\lambda + 4D_r} \\
&\quad \times \Bigg[\frac{1}{\lambda + 2D_r} \left(1 - e^{-(\lambda+2D_r)t} \right) - \frac{1}{2\lambda + 6D_r} \left(1 - e^{-(2\lambda+6D_r)t} \right) \Bigg] \\
&\quad - \frac{1}{3\lambda - 2D_r} \Bigg[\frac{1}{\lambda + 2D_r} \left(1 - e^{-(\lambda+2D_r)t} \right) - \frac{1}{4\lambda} \left(1 - e^{-4\lambda t} \right) \Bigg] \Bigg\} \Bigg) \\
&\quad + \frac{1}{2} \Bigg(\frac{1}{\lambda + 2D_r} \Bigg\{ \frac{1}{\lambda - 2D_r} \Bigg[\frac{1}{\lambda + 2D_r} \left(1 - e^{-(\lambda+2D_r)t} \right) - \frac{1}{2\lambda} \left(1 - e^{-2\lambda t} \right) \Bigg] \\
&\quad - \frac{1}{2\lambda} \Bigg[\frac{1}{\lambda + 2D_r} \left(1 - e^{-(\lambda+2D_r)t} \right) - \frac{1}{3\lambda + 2D_r} \left(1 - e^{-(3\lambda+2D_r)t} \right) \Bigg] \Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
&- \frac{1}{2\lambda}\Bigg\{ \frac{1}{\lambda - 2D_r}\Bigg[\frac{1}{\lambda + 2D_r}\Bigg(1 - e^{-(\lambda+2D_r)t}\Bigg) - \frac{1}{2\lambda}\Bigg(1 - e^{-2\lambda t}\Bigg)\Bigg] \\
&- \frac{1}{3\lambda - 2D_r}\Bigg[\frac{1}{\lambda + 2D_r}\Bigg(1 - e^{-(\lambda+2D_r)t}\Bigg) - \frac{1}{4\lambda}\Bigg(1 - e^{-4\lambda t}\Bigg)\Bigg]\Bigg\}\Bigg)\Bigg].
\end{aligned}
\tag{D.11}
$$

Appendix E

Janus Particle Plots

The MSD and excess kurtosis for the regular-radioactive-Janus particle comparisons are plotted with the MATLAB® script shown below.

Listing E.1: *MSD and excess kurtosis plotting script - Janus.m.*

```
% Initialization
clear

% Constants
m_a = 6.6447E-27; % kg
c = 299792458; % m/s
N_A = 6.02214179E23; % mol^-1
k_B = 1.38064852E-23; % J/K

% Choose radioisotope
radioisotope = input('Which radioisotope?\n','s');
switch radioisotope
    case 'Pa-225'
        rho_p = 15.37E3; % kg/m^3, particle density
        am = (225 / 1000); % kg / mol
        t_h = 1.7; % s, radioisotope half life
        E_a = 7.22; % MeV, average inital energy of alpha particle
        t_MSD = logspace(-7,11,10000); % s, time vector for MSD
        t = logspace(5,15,1000); % s, time vector for excess kurtosis
    case 'Ra-224'
        rho_p = 5E3; % kg/m^3, particle density
        am = (224 / 1000); % kg / mol
        t_h = 3.63 * 24 * 3600; % s, radioisotope half life
        E_a = 5.6855; % MeV, average inital energy of alpha particle
        t_MSD = logspace(2,10,10000); % s, time vector for MSD
        t = logspace(2,12,1000); % s, time vector for excess kurtosis
    case 'Ac-225'
        rho_p = 10.07E3; % kg/m^3, particle density
        am = (225 / 1000); % kg / mol
        t_h = 10.0 * 24 * 3600; % s, radioisotope half life
        E_a = 5.829; % MeV, average inital energy of alpha particle
        t_MSD = logspace(2,10,10000); % s, time vector for MSD
        t = logspace(2,12,1000); % s, time vector for excess kurtosis
    case 'Po-210'
        rho_p = 9.32E3; % kg/m^3, particle density
        am = (210 / 1000); % kg / mol
        t_h = 138.38 * 24 * 3600; % s, radioisotope half life
        E_a = 5.3044; % MeV, average inital energy of alpha particle
        t_MSD = logspace(5,10,10000); % s, time vector for MSD
        t = logspace(2,12,1000); % s, time vector for excess kurtosis
end

% Material parameters
a = 0.5E-6; % m, particle radius
```

```matlab
tm = 20; % C, temperature
rho_f = (999.83952 + 16.945176 * tm - 7.9870401E-3 * tm^2 - 46.170461E-6 * tm^3 + 105.56302E-9 * tm^4 - 280.54253E-12 * tm^5) / (1 + 16.879850E-3 * tm); % kg/m^3, water density
m_p = (4 / 3) * pi * a^3 * rho_p; % kg, particle mass
m_f = (4 / 3) * pi * a^3 * rho_f; % kg, fluid mass
M_star = m_p + 0.5 * m_f; % kg, effective particle mass
N_0 = round((4 / 3) * pi * a^3 * rho_p * N_A / am); % number of atoms
lambda = log(2) / t_h; % s^-1, decay constant
T = tm + 273.15; % K, temperature
A = -3.7188;
B = 578.919;
C = -137.546;
mu = (1 / 1000) * exp(A + B / (C + T)); % Pa s, dyn. visc. of water (Vogel eq.)
E = E_a * 1.60218E-13; % J
M_0 = sqrt(E.^2 + 2 * E * m_a * c^2)/c; % kg m/s
gamma = 6 * pi * mu * a; % kg/s
D_0 = k_B * T / gamma; % m^2/s
D_r = (3 / 4 * a^2) * D_0; % rad/s

% MSD regular particle
MSD_reg = (3 * M_star * D_0 / gamma) * (-3 + 4 * exp(-gamma * t_MSD / M_star) - exp(-2 * gamma * t_MSD / M_star) + 2 * gamma * t_MSD / M_star); % m^2

% MSD radioactive particle
MSD_rad = MSD_reg + (M_0 / gamma)^2 * N_0 * (1 - exp(-lambda * t_MSD)); % m^2

% MSD radioactive Janus particle
N_0 = N_0 / 2;
MSD_jan = ((lambda * M_0 * N_0 / gamma)^2 / (2 * (lambda - 2 * D_r))) * ((1 - exp(-(lambda + 2 * D_r) * t_MSD)) / (lambda + 2 * D_r) - (1 - exp(-2 * lambda * t_MSD)) / (2 * lambda)) + 6 * D_0 * t_MSD; % m^2
dMSD_jan = ((lambda * M_0 * N_0 / gamma)^2 / (2 * (lambda - 2 * D_r))) * (exp(-(lambda + 2 * D_r) * t) - exp(-2 * lambda * t)) + 6 * D_0; % m^2

% Plot MSDs
figure
loglog(t_MSD,MSD_reg,'-',t_MSD,MSD_rad,'--',t_MSD,MSD_jan,'-.')
legend({'Regular Particle','Radioactive Particle','Radioactive Janus Particle'},'interpreter','latex','Location','southeast');
xlabel('$t$ (s)','interpreter','latex')
ylabel('$\langle \left( r \left( t \right) - r_0 \right)^2 \rangle$ (m\textsuperscript{2})','interpreter','latex')

% Excess kurtosis radioactive Janus paticle
gamma_2 = (((lambda .* M_0 .* N_0 ./ gamma).^4 ./ (lambda - 2 .* D_r)) .* ((2 ./ 5) .* ((1 ./ (lambda - 4 .* D_r)) .* ((1 ./ (lambda + 4 .* D_r)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (2 .* lambda + 6 .* D_r)) .* (1 - exp(-(2 .* lambda + 6 .* D_r) .* t))) - (1 ./ (2 .* lambda)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (3 .* lambda + 2 .* D_r)) .* (1 - exp(-(3 .* lambda + 2 .* D_r) .* t)))) - (1 ./ (2 .* lambda - 6 .* D_r)) .* ((1 ./ (lambda + 4 .* D_r)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (2 .* lambda + 6 .* D_r)) .* (1 - exp(-(2 .* lambda + 6 .* D_r) .* t))) - (1 ./ (3 .* lambda - 2 .* D_r)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (4 .* lambda)) .* (1 - exp(-4 .* lambda .* t))))) + (1 ./ 2) .* ((1 ./ (lambda + 2 .* D_r)) .* ((1 ./ (lambda - 2 .* D_r)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (2 .* lambda)) .* (1 - exp(-2 .* lambda .* t))) - (1 ./ (2 .* lambda)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (3 .* lambda + 2 .* D_r)) .* (1 - exp(-(3 .* lambda + 2 .* D_r) .* t)))) - (1 ./ (2 .* lambda)) .* ((1 ./ (lambda - 2 .* D_r)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (2 .* lambda)) .* (1 - exp(-2 .* lambda .* t))) - (1 ./ (3 .* lambda - 2 .* D_r)) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (4 .* lambda)) .* (1 - exp(-4 .* lambda .* t)))))) + ((lambda .* M_0 .* N_0) ./ gamma).^2 .* (18 .* D_0 .* t) ./ (lambda - 2 .* D_r) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (2 .* lambda)) .* (1 - exp(-2 .* lambda .* t))) + 540 .* (D_0 .* t).^2) .* (((lambda .* M_0 .* N_0) ./ gamma).^2 .* (1 ./ (2 .* (lambda - 2 .* D_r))) .* ((1 ./ (lambda + 2 .* D_r)) .* (1 - exp(-(lambda + 2 .* D_r) .* t)) - (1 ./ (2 .* lambda)) .* (1 - exp(-2 .* lambda .* t))) + 6 .* D_0 .* t).^(-2) - 15;

% Plot excess kurtosis
figure
[hAx,hLine1,hLine2] = plotyy(t,gamma_2,t,dMSD_jan,'semilogx','semilogx');
hLine2.LineStyle = '--';set(hAx(1),'YLim',[-6 14]);set(hAx(2),'YLim',[0 1.25e-3])
set(hAx(1),'YTick',-6:2:14);set(hAx(2),'YTick',0:25e-5:1.25e-3)
xlabel('$t$ (s)','interpreter','latex')
ylabel(hAx(1),'$\gamma_2$','interpreter','latex')
ylabel(hAx(2),'$d (\textrm{MSD}) / dt$ (m\textsuperscript{2}/s)','interpreter','latex')
```

Appendix F

Heat Equation Solution

The heat equation is solved and plotted with the MATLAB® script shown below.

Listing F.1: *Heat equation calculation and plotting script - Heat.m.*

```
% Pa-225
r_0 = 65.2786e-6; % m, alpha particle range in water from SRIM
E_a = 7.22; % MeV, average inital energy of alpha particle
t_h = 1.7; % s, radioisotope half life
rho_p = 15.37E3; % kg/m^3, particle density
am = (225 / 1000); % kg / mol
r = 1e-6:1e-6:100e-6; %2e-5:2e-5:2e-3; % m, solution domain
t = 1e-2:1e-2:1; %1e-1:1e-1:10; % s, solution domain

% Material parameters
E_a = E_a * 1.60218E-13; % J
N_A = 6.02214179E23; % mol^-1
a = 0.5E-6; % m, particle radius
tm = 20; % C, temperature
rho_f = (999.83952 + 16.945176 * tm - 7.9870401E-3 * tm^2 - 46.170461E-6 * tm^3 + 105.56302E-9 * tm^4 - 280.54253E-12 * tm^5) / (1 + 16.879850E-3 * tm); % kg/m^3, water density
c_p = 4186; % J/kg K, water specific heat
k = 0.6; % W/m K
alpha = k / (rho_f * c_p); % m^2/s, thermal diffusivity of water
N_0 = round((4 / 3) * pi * a^3 * rho_p * N_A / am); % number of atoms
lambda = log(2) / t_h; % s^-1, decay constant
T_0 = tm + 273.15; % K, initial temperature

% Solve for temperature profile
T = zeros(length(t),length(r)); % K
for i = 1:length(t)
    for j = 1:length(r)
        integrand = @(rp,tau) exp(-lambda * tau) .* (exp(-(r(j) - rp).^2 ./ (4 * alpha * (t(i) - tau))) - exp(-(r(j) + rp).^2 ./ (4 * alpha * (t(i) - tau)))) ./ sqrt(t(i) - tau) .* rp;
        T(i,j) = T_0 + 3 * lambda * N_0 * E_a / (4 * pi * r_0^3 * r(j) * sqrt(4 * pi * k * rho_f * c_p)) * integral2(integrand,0,r_0,0,t(i),'Method','iterated');
    end
end

% Plot temperature profile
figure
contourf(r,t,T,1000,'LineColor','none');xlabel('$r$ (m)','interpreter','latex')
ylabel('$t$ (s)','interpreter','latex');colormap(hot)
c = colorbar; c.Label.Interpreter = 'latex'; c.Label.String = '$T$ (K)';
ticks = get(c,'Limits'); caxis([floor(ticks(1)) ceil(ticks(2))]); set(c,'Ticks', linspace(floor(ticks(1)),ceil(ticks(2)),7))
L = cellfun(@(x)sprintf('%i',x),num2cell(get(c,'xtick')),'Un',0); set(c,'xticklabel',L)
```

Appendix G

MSDs Including Gravity

The MSDs of the particles discussed in Chapter 2 are herein plotted including the displacement due to gravity. The full theoretical MSD can be calculated as

$$\begin{aligned}\left\langle \left(\boldsymbol{r}\left(t\right)-\boldsymbol{r}_0\right)^2\right\rangle = &\left|\frac{\boldsymbol{u}_0}{\beta}\left(1-e^{-\beta t}\right)+\frac{\boldsymbol{\mu}}{\beta}\left(\beta t-1+e^{-\beta t}\right)\right|^2 \\ &+\frac{3\sigma^2}{2\beta^3}\left(-3+4e^{-\beta t}-e^{-2\beta t}+2\beta t\right)+N_0\left(\frac{M_0}{\gamma}\right)^2\left(1-e^{-\lambda t}\right), \quad \text{(G.1)}\end{aligned}$$

since the multiple of a Wiener process still has a mean of 0.

The MSDs for ^{222}Ra, ^{225}Pa, ^{222}Th, and ^{218}Ac are plotted in Fig. G.1, Fig. G.2, Fig. G.3, and Fig. G.4, respectively, with the numerical MSD from the adaptive time step case. The MSDs of ^{222}Th and ^{218}Ac look almost identical to those in Chapter 2, while the MSDs of ^{222}Ra and ^{225}Pa are completely dominated by gravity. Gravity quickly plays a large role in the motion of these microparticles since they are much denser than water.

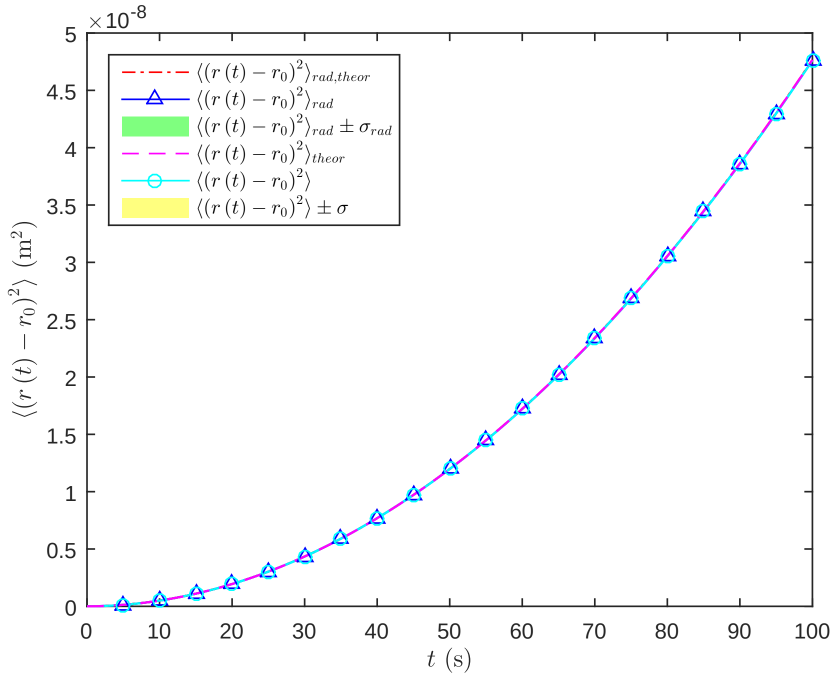

Figure G.1: *The MSDs for ^{222}Ra including the displacement due to gravity.*

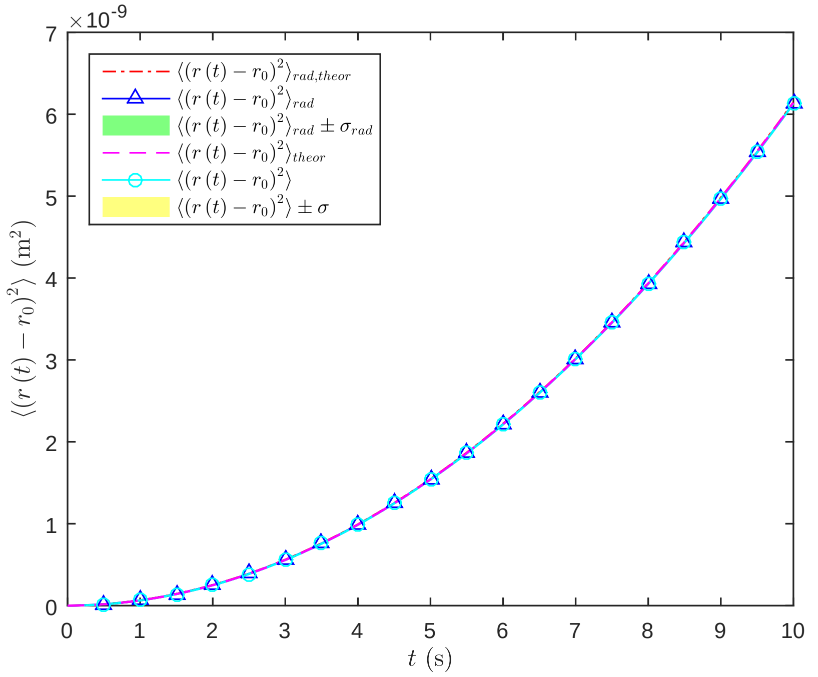

Figure G.2: *The MSDs for ^{225}Pa including the displacement due to gravity.*

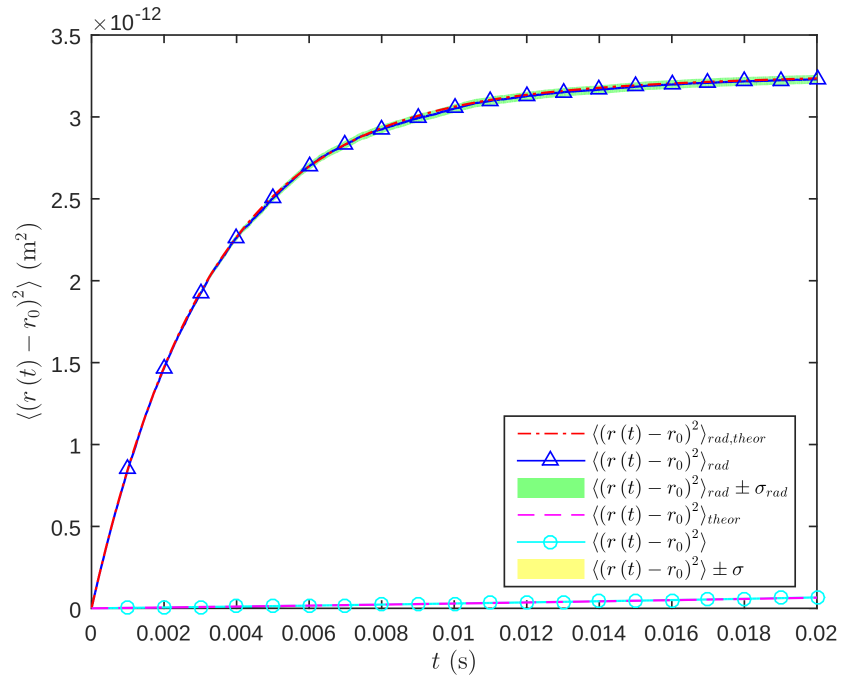

Figure G.3: *The MSDs for ^{222}Th including the displacement due to gravity.*

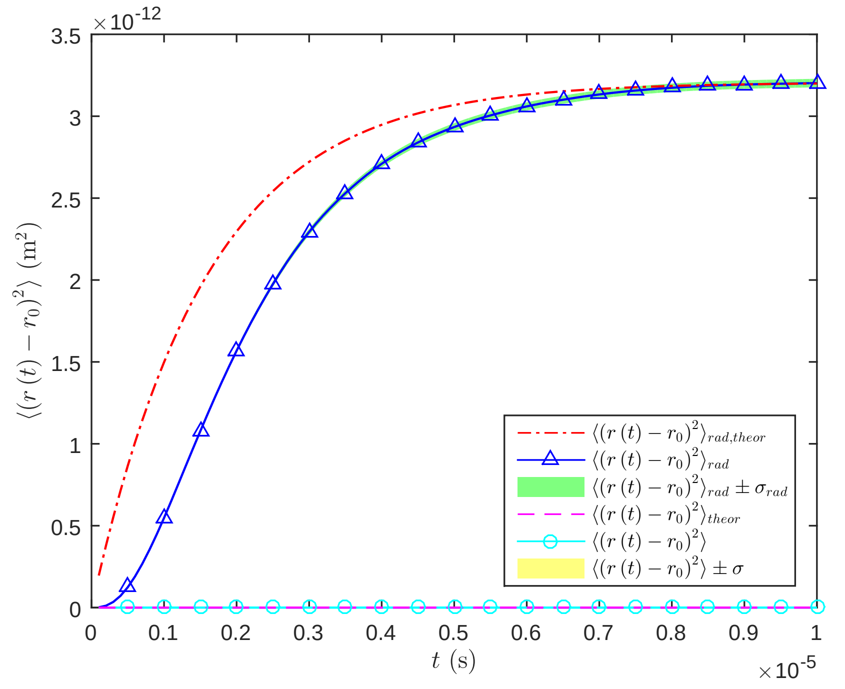

Figure G.4: *The MSDs for ^{218}Ac including the displacement due to gravity.*