ENTRAINMENT DOMINATED EFFECTS IN THE LONG RESIDENCE TIMES OF SOLID SPHERES SETTLING IN SHARPLY STRATIFIED MISCIBLE VISCOUS FLUIDS

Claudia Falcon

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics.

Chapel Hill
2016

Approved by:

Roberto Camassa

Richard McLaughlin

David Adalsteinsson

Gregory Forest

Laura Miller

ABSTRACT

Claudia Falcon: Entrainment dominated effects in the long residence times of solid spheres settling in sharply stratified miscible viscous fluids
(Under the direction of Roberto Camassa and Richard McLaughlin)

This dissertation presents results on the effects of sharp density variations in the dynamics of settling spheres in viscous – dominated regimes by a combination of experimental, analytical, and numerical tools. Particles settling through naturally – stratified fluids, such as the ocean and the atmosphere, affect many aspects of life, from air quality and pollution clearing times to the formation of thin aggregate layers in the upper ocean. In this thesis, we develop an understanding of the dynamics that affect these problems by studying the behavior of a sphere falling under gravity through a two-layer fluid.

We have found that the sphere slows down dramatically as it passes through the density transition. In this system, we demonstrate the importance of the entrained fluid to the delayed settling of the particle due to its added buoyancy force. In particular, we compare long residence times at the interface rivaling the ones observed for porous spheres and marine snow – aggregates that occur naturally in the ocean – in similar configurations. We call these cases the entrainment dominated regimes, where diffusion of salt could play an active role and it is therefore needed in the modeling.

The developed first principle model is a highly coupled system that captures the most significant aspects of settling in a sharp two - layer fluid. We discuss previously implemented approximations and new experimental regimes where the approximation is no longer valid. The asymptotic approaches and exact solutions for the sphere exterior problem of the Stokes equations will be compared in a parametric study of relevance for experiments. The region of validity for the approximations, the full theory agreement, and the possible need to include

diffusion in the entrainment dominated regimes will be discussed and explained.

The single particle theory further sheds light in the settling rates of marine aggregates falling through sharp density transitions, and how this ultimately affects marine carbon cycling. In addition to solid spheres, we have examined porous, and drilled spheres , obtaining range of parameters that enhance the residence time at the interface. We have also investigated this phenomenon extensively through experiments by observing clouds of solid particles as they settle through a sharply stratified water column.

A mis padres -

El amor todo lo sufre,
todo lo cree, todo lo espera,
todo lo soporta.

-1 Corintios 13:7

ACKNOWLEDGEMENTS

I would like to thank my committee members for their recommendations and my advisors for the countless devoted hours of research, teaching, and mentorship. They have been instrumental to my career since I joined their lab as an undergraduate. Roberto Camassa is known for his unique guidance through difficult questions and making them seem easy through simplified examples. I am thankful to Richard McLaughlin for his intuition, excitement for interesting results, and challenging questions. Thank you for always pushing me to look deeper after going over long pages of mathematics or sharing the results of an experiment. Every meeting with them was a boost of energy to continue looking for solutions. Thank you to both for your support through all these years.

I would also like to thank Joyce Lin. Her training introduced me to the field of fluid dynamics and ultimately encouraged me to continue researching Mathematics at the graduate level. I look up to her as a Mathematician and a friend and hope to continue collaborations for years to come. A special mention must go to Kathryn Valchar for her immeasurable hours of work and endless discussions that helped us improve our experimental techniques. Her understanding of our goals made it easy to obtain good quality results. In addition, thanks to Dylan Bruney, Sabina Iftikhar, and Gabriella Stein, for being the best team of undergraduate researchers. Dylan's innovative ideas and long hours in the lab until the experiment was correctly completed, Sabina's creativity and no-problem-is-too-big attitude, and Gabi's dedication to all aspects of the project helped us obtain the data shown in this thesis. I would also like to thank Mike Baker, Mathew Chancey, Victoria Hughes, Dylan Owen, Pierre-Yves Passaggia, Bailey Watson, and Arthur Wood for their contributions. To all my colleagues in the Joint Fluids Lab and the Mathematics Department, thank you. In addition, David Adalsteinsson, the developer of DataTank and committee member, always

willing to advise on many numerical questions. Without his software and training, the analysis would have been more difficult and less visually appealing .

A special mention must go to the UNC IME program coordinator, Kathy Wood, for her hard work and support since my undergraduate years in AGEP. I am grateful for her encouragement to pursuit challenging projects, her career development advice, and friendship. Obvious acknowledgments go to the AGEP, Gates Millennium, NSF GRFP, and UNC Graduate School Dissertation Completion fellowships. The National Science Foundation has been a significant supporter of all the laboratory work. Specifically, credit is due to the NSF-CMG and NSF RTG DMS-0502266 grants.

Last but not least, I am thankful for my family and friends for providing understanding and many happy moments. I am thankful for my husband, who is a constant support when deadlines require all my attention. His pride for my work and continuous encouragement is a fuel like no other. There are no words to thank my parents for all their sacrifices. It gives me great pleasure to talk about my work at a mathematical level with great thinkers like them. My mom is my true confidant in moments of pressure and her words are always the ones I need to hear. My dad was the first who taught me to explore creativity in a mathematical way. I thank him for his willingness to always listen to my messy mathematical thoughts. Thank you both for being my role models as mathematicians but also for making me want to grow better everyday so that someday I get to be a teacher to my children the same way you have been to me.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Stratified environments frequently occur in nature. Examples include haloclines and thermoclines in the ocean and atmosphere. These are often sharp density transitions formed by density and temperature variations. Understanding how these sharp density differences affect an immersed falling particle has applications from effectively budgeting pollution in the ocean to testing the efficiency of the ocean's carbon pump. Thin layers of particulate matter have been measured to accumulate at the density transitions found in coastal waters. Marine snow particles are aggregates of organic and inorganic matter that are constantly raining in the ocean, and are a major component of the ocean's carbon cycle. When marine snow accumulates at a density interface, the layers of aggregates become hotspots for bacteria remineralization, preventing the particles from settling to the bottom of the ocean[31]. Understanding the residence time of these particles by studying the fluid mechanics behind their settling in variable density fluids, can provide insight into the evolution of aggregates and their fate as part of the carbon cycle.

There have been many studies of stratification and particle settling. Most of these studies, however, involve homogenous or linear stratified fluid, and very few account for the importance of entrainment in sharp density transitions. Some of the research investigating sharply-stratified environments is restricted to immiscible fluids where surface tension is dominant [3, 26]. Previous studies have also considered the settling of marine aggregates and porous spheres in sharp stratifications [11, 31].

The behavior of a single sphere falling in sharp stratification has been been studied by [1, 8, 9, 33], but many questions remain. Interesting phenomena have been discovered after it was found that a sphere undergoes levitation– sharp acceleration to near zero speed,

and sometimes even a directional (velocity) reversal, when falling under specific conditions through a sharply stratified fluid [1]. As the sphere passes through the interface, the entrained lower density fluid around the particle adds an extra buoyancy force that causes the delayed settling. Solving this problem is a simplification of the settling of multiple aggregates but it captures some of its most important dynamics.

The first–principle model that describes this behavior in viscous fluids is a highly coupled system, derived by Camassa et. al. [8, 9]. Focusing on the low-Reynolds-number regime simplifies the physics by making the fluid inertia negligible, which is reflected mathematically by the linearization of the governing equations of motion. This makes them more accessible to analysis and numerics. We use this model to study the dynamics of the long residence times exhibited by solid spheres at density transition layers. In particular, we study the cases with long residence times comparable to the ones observed for porous spheres and marine aggregates in similar configurations. These cases fall into what we call the entrainment dominated regime. This regime occurs when the difference between the sphere density and the bottom layer density ($\rho_s - \rho_b$) approaches the difference between the bottom layer density and the top layer density ($\rho_b - \rho_t$).When the residence times are comparable to diffusion time scales, salt diffusion through the entrainment shell can play a significant role in the sphere settling, making it important to understand its contribution in the modeling.

In order to compare the first-principle model solutions with the experimental data, we need to numerically implement the final equations of motion, an integro-differential equation of the density of the fluid ρ. The way in which we go about solving the resulting three dimensional integral that defines the flow of the fluid determines the speed and accuracy of the code. In this paper, we discuss previously implemented approximations and new experimental parameters where these approximation are no longer valid. To remedy this, we implement new ways to solve for the flow, including different integration techniques to account for the integrable and removable singularities, analytical integration of one of the integrals, as well a matched asymptotics of near and far field. In addition, we discuss properties of the

integrand- the Oseen tensor provided by [29]. The asymptotic approaches and exact solutions for the sphere exterior problem of Stokes equations will be compared in a parametric study of relevance for experiments. The region of validity for the approximations and the need to implement matched asymptotic with the full solution will be discussed and explained.

Theoretically, there is much work to be done to understand the settling of marine aggregates through sharp density transitions, and how this ultimately affects marine carbon cycling. To develop a mechanistic understanding of the behavior of these particles in the ocean,we investigated this phenomenon in detail through experiments by observing clouds of solid particles as they settle through a sharply stratified saltwater column. Chapter 5 describes the experimental work and its Section 5.4 focuses on the multiple particle experiments in sharply stratified fluids. The applications of our work can be instrumental for estimating pollution-clearing times and the effectiveness of the ocean as a pump in driving carbon excess. Our study's predictive tool gives us a better understanding on time scales and the attributes of of delayed settling.

The set up of the single particle problem consists of a sphere of radius A settling under gravity in an infinite cylinder of radius R_0 that contains a stable two-layer stratification of miscible fluid, with top layer density ρ_t and bottom layer density of ρ_b. We are interested in obtaining the velocity $\boldsymbol{V}(t)$ and center position $\boldsymbol{Y}(t)$ of the sphere, as well as the fluid flow denoted as $\boldsymbol{v}(\boldsymbol{x}, t)$. In cylindrical coordinates, the observation point $\boldsymbol{x} = (R, \phi, Z)$

CHAPTER 2

FIRST PRINCIPLE MODEL

In this chapter, we set up the equations of motion for a solid sphere of radius A and velocity $\boldsymbol{V}(t) = (0, 0, V(t))$ settling in sharp stratification and discuss the solutions as given by [9]. The Navier-Stokes equations for incompressible, Newtonian fluid of velocity $\boldsymbol{v}(\boldsymbol{x}, t)$ and variable density $\rho(\boldsymbol{x}, t)$ and at an observation point $\boldsymbol{x}$ in the fluid domain inside a cylinder of radius R_0 are given by:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \boldsymbol{v} \cdot \nabla \rho &= 0, \qquad \nabla \cdot \boldsymbol{v} = 0, && (2.1)\\
\rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} \right) &= \rho \hat{g} - \nabla p + \mu \nabla^2 \boldsymbol{v}, && (2.2)\\
\boldsymbol{v} &= \boldsymbol{V}(t) \text{ for } |\boldsymbol{x} - Y_3(t)| = A, && (2.3)\\
\boldsymbol{v} &= 0 \text{ for } \sqrt{x_1^2 + x_2^2} = R_0, \ -\infty < x_3 < \infty, && (2.4)\\
\boldsymbol{v} &\to 0 \text{ for } |x_3| \to \infty, && (2.5)
\end{aligned}$$

where $\boldsymbol{Y}(t)$ is the position of the center of the sphere in the laboratory frame of reference, so that $\dot{Y}_3(t) = V(t)$, $\hat{g} = (0, 0, g)$ is the gravity acceleration vector with magnitude $g = 981\text{cm/s}^2$.

By nondimensionalizing the equations of motion, important dimensionless parameters arise. The Reynolds, Strouhal, and Froude numbers are defined as $Re = AU/\nu$, $St = A/UT$, and $Fr = U/\sqrt{gA}$, respectively, where U is the terminal velocity of the sphere in a homogeneous fluid of density ρ_{ref}, and T is the deceleration time. The momentum equation becomes

$$Re\, St\, \tilde{\rho} \frac{\partial \tilde{\boldsymbol{v}}}{\partial \tilde{t}} + Re\, \tilde{\rho} \tilde{\boldsymbol{v}} \cdot \tilde{\nabla} \tilde{\boldsymbol{v}} = \frac{Re}{Fr^2}\, \tilde{\rho} \hat{z} - \tilde{\nabla} \tilde{p} + \tilde{\nabla}^2 \tilde{\boldsymbol{v}}, \tag{2.6}$$

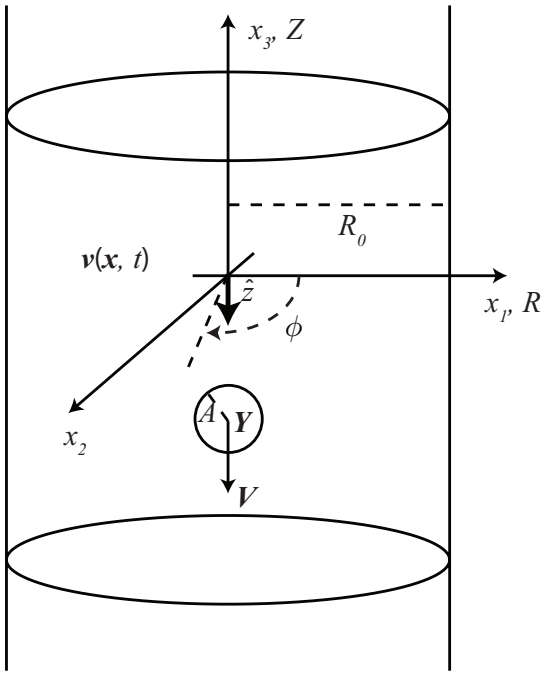


Figure 2.1: Schematic of the theoretical setup and notation.

with the pressure scaled by $\mu U/A$. Our experiments are consistent with low Re number regime making it possible to scale out the inertial terms, simplifying (2.6) to the Stokes equation with variable density:

$$\tilde{\nabla}^2 \tilde{\boldsymbol{v}} = \tilde{\nabla}\tilde{p} - \frac{Re}{Fr^2}\,\tilde{\rho}\hat{z}. \tag{2.7}$$

In dimensional form, the Equation (2.7), the incompressibility and boundary conditions, for the Stoke's approximation.

$$\begin{aligned}
\mu\nabla^2\boldsymbol{v} &= \nabla p - \rho\hat{g}, && (2.8)\\
\nabla\cdot\boldsymbol{v} &= 0, && (2.9)\\
\boldsymbol{v} &= \boldsymbol{V}(t) \text{ for } |\boldsymbol{x} - Y_3(t)| = A, && (2.10)\\
\boldsymbol{v} &= 0 \text{ for } \sqrt{x_1^2 + x_2^2} = R_0,\ -\infty < x_3 < \infty, && (2.11)\\
\boldsymbol{v} &\to 0 \text{ for } |x_3| \to \infty && (2.12)
\end{aligned}$$

$$\frac{\partial \rho}{\partial t} + \boldsymbol{v} \cdot \nabla \rho = 0. \tag{2.13}$$

It is important to note that in equation (2.13) we have ignored diffusion of salt, present in the bottom layer fluid that serves as a stratifying agent . Some of our experiments – the Entrainment regime experiments – slow down the settling rates and exhibit prolonged residence times that are comparable with diffusion time scales. By comparing those regimes with this model, we can study the need for diffusion to be included in the theory. In the Stokes dominated regimes, we see the persistence of sharp interfaces between the upper and lower fluids as well as along the entrainment around the sphere for the entire duration of the experiment. These observations indicate that diffusion is negligible in the Stokes regimes while it may not be the case for the Entrainment Regimes. The diffusivity of the salts used in our fluids was measured and discussed in Appendix D. More evidence that no diffusion effects are present in Stokes regime is given by the good agreement between experiments and this non-diffusive model. On the contrary, the Entrainment regimes show some discrepancy when diffusion time scales are of significance. These comparisons will be discussed in later chapters.

The equation of motion for the sphere can be written as

$$m_s \frac{dV(t)}{dt} = m_s \hat{g} + \oint_S \sigma \cdot \hat{n}\, dS, \tag{2.14}$$

where m_s is the mass of the sphere, σ is the stress tensor, S is the surface of the sphere, and $\hat{n}$ is the outward normal unit vector to this surface.

The fluid flow can be written as

$$\boldsymbol{v}(\boldsymbol{x}, t) = \boldsymbol{u}(\boldsymbol{x}, t) + \boldsymbol{w}(\boldsymbol{x}, t)\,. \tag{2.15}$$

and we can solve for each part by taking advantage of the linearity of the Stokes equations. The first part, $\boldsymbol{u}(\boldsymbol{x}, t)$, is a Stokes flow in a cylinder with static initial density distribution

$\rho_0(x_3) = \rho(\boldsymbol{x}, 0)$ while the second part, $\boldsymbol{w}(\boldsymbol{x}, t)$, is what we call the perturbation velocity, as it has homogeneous boundary conditions and a forcing term $\rho(\boldsymbol{x}, t) - \rho_0(x_3)$. The equation of motion for the sphere becomes

$$m_s \frac{dV(t)}{dt} = m_s \hat{g} + \oint_S \sigma_u \cdot \hat{n} dS + \oint_S \sigma_w \cdot \hat{n} dS, \tag{2.16}$$

where σ_u and σ_w are the stress tensors for $\boldsymbol{u}$ and $\boldsymbol{w}$, respectively. The latter stress tensor σ_w originates solely from the advection of the density field, and gives rise to an effective buoyancy-like force, which we refer to as the anomalous density force to distinguish it from the usual Archimedean buoyancy.

2.1 Stokes flow in a cylinder

The equations of motion for the velocity component $\boldsymbol{u}(\boldsymbol{x}, t)$ in equation (2.15), in the frame of reference of the sphere are

$$\begin{aligned}
\mu \nabla^2 \boldsymbol{u} &= \nabla p_s - \rho_0(x_3 + Y_3(t))\hat{g}, && (2.17)\\
\nabla \cdot \boldsymbol{u} &= 0, && (2.18)\\
\boldsymbol{u} &= 0 \text{ for } |\boldsymbol{x}| = A, && (2.19)\\
\boldsymbol{u} &= -\boldsymbol{V}(t) \text{ for } \sqrt{x_1^2 + x_2^2} = R_0, \ -\infty < x_3 < \infty && (2.20)\\
\boldsymbol{u} &\to -\boldsymbol{V}(t) \text{ for } |x_3| \to \infty\,. && (2.21)
\end{aligned}$$

The solutions are given by the method of reflections used in [18], found by decomposing the flow into a series

$$\boldsymbol{u} = (\boldsymbol{u}^{(0)} + \boldsymbol{u}^{(1)}) + (\boldsymbol{u}^{(2)} + \boldsymbol{u}^{(3)}) + \ldots \tag{2.22}$$

where

$$\boldsymbol{u}^{(0)} = -\boldsymbol{V}(t) \tag{2.23}$$

$$\boldsymbol{u}^{(1)} = \begin{cases} -\boldsymbol{u}^{(0)} & r = A \\ 0 & r \to \pm\infty \end{cases} \tag{2.24}$$

$$\boldsymbol{u}^{(2)} = \begin{cases} -\boldsymbol{u}^{(1)} & R = R_0 \\ 0 & Z \to \pm\infty \end{cases} \tag{2.25}$$

$$\boldsymbol{u}^{(3)} = \begin{cases} -\boldsymbol{u}^{(2)} & r = A \\ 0 & r \to \pm\infty \end{cases} \tag{2.26}$$

$$\vdots$$

The flow is expressed by this infinite sum and can be truncated to the *odd-labeled* terms, making the flow satisfy boundary conditions on the sphere. On the other hand, when the sum is truncated at the *even-labeled* terms the boundary conditions on the cylinder are satisfied. The first term $\boldsymbol{u}^{(0)}$ is a constant flow (in space), whose inclusion changes the frame of reference from the lab frame to a frame of reference moving with the sphere. The second term $\boldsymbol{u}^{(1)}$ satisfies the boundary conditions using a sphere in Stokes flow in free space. [18] provide the full solution for the third term $\boldsymbol{u}^{(2)}$, which cancels out the contribution of $\boldsymbol{u}^{(1)}$ on the boundary of the cylinder. The magnitudes of $\boldsymbol{u}^{(0)}$ and $\boldsymbol{u}^{(1)}$ are $O(1)$ and the sum of the two velocities satisfies the boundary conditions on the sphere. The error incurred on the cylinder walls has magnitude $O(A/R_0)$. The next reflection $\boldsymbol{u}^{(2)}$ has magnitude of order $O(A/R_0)$, and the sum $\boldsymbol{u}^{(0)} + \boldsymbol{u}^{(1)} + \boldsymbol{u}^{(2)}$ satisfies the boundary conditions on the cylinder so that this sum incurs an error on the sphere of order $O(A/R_0)$.

Happel and Byrne do not study the convergence nature of the series from this method of reflections, but notice that each reflection contributes a multiplicative (small) factor A/R_0. To correct for the error made on the sphere by $\boldsymbol{u}^{(2)}$, the next reflection $\boldsymbol{u}^{(3)}$ must be of order $O(A/R_0)$. Thus, $\boldsymbol{u}^{(2)}$ and $\boldsymbol{u}^{(3)}$ have the same order of magnitude. The rest of the terms are

not computed explicitly, but we can expect that this pattern be repeated, so that the series expansion $\boldsymbol{u} = (\boldsymbol{u}^{(0)} + \boldsymbol{u}^{(1)}) + (\boldsymbol{u}^{(2)} + \boldsymbol{u}^{(3)}) + \ldots$ decreases by an order of magnitude in pairs, as indicated explicitly by the parenthetical grouping.

In the asymptotic expansion for $\boldsymbol{u} = \boldsymbol{u}^{(0)} + \boldsymbol{u}^{(1)} + \boldsymbol{u}^{(2)} + \boldsymbol{u}^{(3)} \ldots$, as formulated in [18] in cylindrical coordinate (R, θ, Z) with $r = \sqrt{R^2 + Z^2}$ is:

$$u_Z^{(0)} = -V(t), \tag{2.27}$$

$$u_R^{(0)} = 0, \tag{2.28}$$

$$u_Z^{(1)} = -V(t)\left[\frac{-3A}{4r} - \frac{3AZ^2}{4r^3} - \frac{A^3}{4r^3} + \frac{3Z^2A^3}{4r^5}\right], \tag{2.29}$$

$$u_R^{(1)} = -V(t)\left[\frac{-3ARZ}{4r^3} + \frac{3A^3RZ}{4r^5}\right], \tag{2.30}$$

$$u_Z^{(2)} = \frac{1}{2\pi}\int_0^\infty \hat{u}_Z(R,\lambda)\cos(\lambda Z)d\lambda, \tag{2.31}$$

$$u_R^{(2)} = \frac{1}{2\pi}\int_0^\infty \hat{u}_R(R,\lambda)\sin(\lambda Z)d\lambda, \tag{2.32}$$

$$\frac{P^{(2)}}{\mu} = \frac{1}{2\pi}\int_0^\infty \hat{P}(R,\lambda)\sin(\lambda Z)d\lambda, \tag{2.33}$$

where

$$\hat{u}_Z(R,\lambda) = \frac{\lambda R}{2}(H(\lambda)+G(\lambda))I_1(\lambda R) + H(\lambda)I_0(\lambda R), \tag{2.34}$$

$$\hat{u}_R(R,\lambda) = \frac{\lambda R}{2}(H(\lambda)+G(\lambda))I_0(\lambda R) - G(\lambda)I_1(\lambda R), \tag{2.35}$$

$$\hat{P}(R,\lambda) = \lambda(H(\lambda)+G(\lambda))I_0(\lambda R), \tag{2.36}$$

and

$$H(\lambda) = \frac{AV\left\{3-(6+A^2\lambda^2)\left(K_0(\lambda R_0)I_2(\lambda R_0)+K_1(\lambda R_0)I_1(\lambda R_0)\right)\right\}}{I_0(\lambda R_0)I_2(\lambda R_0)-I_1(\lambda R_0)^2}, \tag{2.37}$$

$$G(\lambda) = \frac{AV\left\{-3+A^2(\lambda R_0)^2\left(K_1(\lambda R_0)I_1(\lambda R_0)+K_2(\lambda R_0)I_0(\lambda R_0)\right)\right\}}{I_0(\lambda R_0)I_2(\lambda R_0)-I_1(\lambda R_0)^2}, \tag{2.38}$$

with I_j and K_j are modified Bessel functions of the first and second kind.

For our purposes, we cannot truncate at the second reflection $\boldsymbol{u}^{(2)}$, which is the last term of the series provided explicitly by [18]. Truncating at the second reflection would imply satisfying boundary conditions on the wall but not on the sphere, therefore the interface would pass through the sphere when $\boldsymbol{u}^{(3)}$ is neglected, as this term is of the same order as $\boldsymbol{u}^{(2)}$. In order to compute $\boldsymbol{u}^{(3)}$, we must have $\boldsymbol{u}^{(2)}$ on the surface of the sphere for boundary conditions. An expansion around small values R/R_0 and Z/R_0, can be made due to the convergence properties of the infinity λ integral of the series expansion of the integrand (see Section 3.2). Therefore, evaluated at the surface of the sphere, the second reflection $\boldsymbol{u}^{(2)}$, and thus the third reflection $\boldsymbol{u}^{(3)}$ is

$$\begin{aligned} u_Z^{(3)}\Big|_{r=A} &= -u_Z^{(2)}\Big|_{r=A} = 2.10444\frac{A}{R_0}V - 2.18004\frac{A^3}{R_0^3}V - 0.140011\frac{A}{R_0^3}R^2V + \dots \\ u_R^{(3)}\Big|_{r=A} &= -u_R^{(2)}\Big|_{r=A} = 1.13669\frac{A}{R_0^3}\,R\,Z\,V + \dots \end{aligned} \tag{2.39}$$

The solution of the third reflection problem using the expansion from Equation (2.39) for the second reflection $\boldsymbol{u}^{(2)}$ as boundary conditions is derived in Chapter 3. The third reflection $\boldsymbol{u}^{(3)}$ was found by stream function solution of the Stokes problem with prescribed flow as boundary conditions on the sphere.

$$
\begin{aligned}
u_R^{(3)}(R,Z) = & \frac{5A^8RVZ((-1.13669)-(0.140011))\left(3R^2-4Z^2\right)}{8R_0^3r^9} \\
& - \frac{A^6RVZ\left((-1.13669)\left(23R^2-12Z^2\right)-6(2.18004)r^2\right)}{8R_0^3r^7} \\
& - \frac{A^6RVZ\left(-11(0.140011)R^2+24(0.140011)Z^2\right)}{8R_0^3r^7} \\
& - \frac{A^4RVZ\left(-3(-2.10444)R_0^2+3(2.18004)r^2-2(0.140011)r^2\right)}{4R_0^3r^5} \\
& - \frac{3A^2(-2.10444)RVZ}{4R_0r^3}
\end{aligned}
\tag{2.40}
$$

$$
\begin{aligned}
u_Z^{(3)}(R,Z) = & -\frac{A^8V((-1.13669)-(0.140011))\left(3R^4-24R^2Z^2+8Z^4\right)}{8R_0^3r^9} \\
& - \frac{A^6V\left(-R^2-Z^2\right)(-1.13669)\left(3R^4-24R^2Z^2+8Z^4\right)}{8R_0^3r^9} \\
& - \frac{A^6V\left((2.18004)\left(-2R^4+2R^2Z^2+4Z^4\right)+(0.140011)\left(R^4+20R^2Z^2-16Z^4\right)\right)}{8R_0^3r^9} \\
& - \frac{A^4V\left((-2.10444)R_0^2\left(R^2-2Z^2\right)+3(2.18004)\left(R^4+3R^2Z^2+2Z^4\right)\right)}{4R_0^3r^5} \\
& - \frac{A^4V\left(-2(0.140011)\left(R^4+3R^2Z^2+2Z^4\right)\right)}{4R_0^3r^5} \\
& - \frac{3A^2(-2.10444)V\left(R^2+2Z^2\right)}{4R_0r^3}
\end{aligned}
\tag{2.41}
$$

As mentioned in [9], the asymptotic properties of the reflection series as $A/R_0 \to 0$ are not discussed by [18]. The asymptotic ordering of terms in equation (2.22) would fail near the cylinder boundary, which would require techniques from matched asymptotics to address this nonuniformity in a region near the cylinder's boundary. By keeping the expansion up to the third reflection, we no longer satisfy the boundary conditions on the cylinder exactly, but the violation is consistent with the overall asymptotic error of the retained terms as $A/R_0 \to 0$.

All computations shown in this section use the above Happel and Byrne's formulation. The approximate solution for the homogeneous fluid Stokes flow in a cylinder can now be

used with variable density and time dependent sphere's velocity to find the drag force due to this flow,

$$\oint_S \sigma_s \cdot \hat{n} dS = -g \int_{\Omega_s} \rho_0(x_3 + Y_3(t)) d\Omega_s - 6\pi A \mu V(t) K, \tag{2.42}$$

where Ω_s is the sphere domain and $K = (1 - 2.10444(A/R_0) + 2.08877(A/R_0)^3 + ...)^{-1}$ is the drag coefficient.

2.2 Perturbation velocity

For the stratification-induced flow, we define $G(\boldsymbol{x}, t) = (\rho(\boldsymbol{x}, t) - \rho_0(x_3 + Y_3(t)))$ and write the governing equations in a moving frame of reference,

$$\begin{aligned}
\mu \nabla^2 \boldsymbol{w} &= \nabla p_w - G(\boldsymbol{x}, t)\hat{g}, && (2.43)\\
\nabla \cdot \boldsymbol{w} &= 0, && (2.44)\\
\boldsymbol{w} &= 0 \text{ for } |\boldsymbol{x}| = A, && (2.45)\\
\boldsymbol{w} &= 0 \text{ for } \sqrt{x_1^2 + x_2^2} = R_0, \ -\infty < x_3 < \infty && (2.46)\\
\boldsymbol{w} &\to 0 \text{ for } |x_3| \to \infty. && (2.47)
\end{aligned}$$

Thus, the boundary conditions for $\boldsymbol{w}$ are homogeneous, and we can find an approximate solution for $\boldsymbol{w}(\boldsymbol{x}, t)$ using the free space Green's function due to [29]. This is the solution of the equations

$$\begin{aligned}
\mu \nabla^2 \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{y}) &= \nabla P(\boldsymbol{x}, \boldsymbol{y}) - \hat{g}\delta(\boldsymbol{x} - \boldsymbol{y}), && (2.48)\\
\nabla \cdot \boldsymbol{W} &= 0, && (2.49)\\
\boldsymbol{W} &= 0 \text{ for } |\boldsymbol{x}| = A, && (2.50)\\
\boldsymbol{W} &\to 0 \text{ as } |\boldsymbol{x}| \to \infty, && (2.51)
\end{aligned}$$

for a Stokeslet of strength $\hat{g}$ located at the point $\boldsymbol{y}$ outside a rigid sphere of radius A surrounded by an infinite Stokes fluid. The resultant force on the sphere can be computed using the Reciprocal Theorem [9].

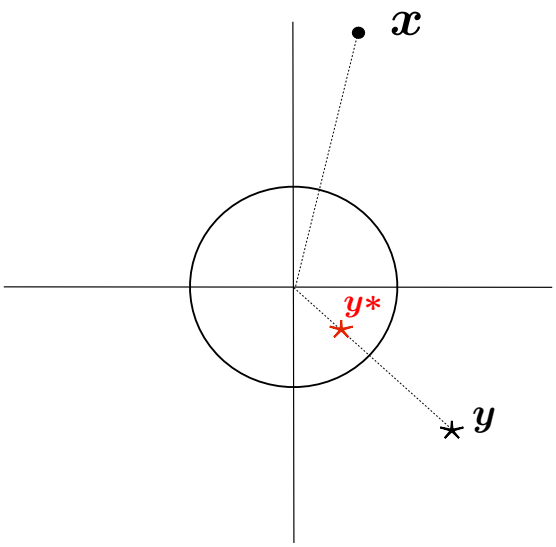


Figure 2.2: Diagram of the Stokeslets located at $\boldsymbol{y}$ and $\boldsymbol{y}^*$.

We look for an asymptotic expansion $\boldsymbol{w} = \boldsymbol{w}^{(0)} + \boldsymbol{w}^{(1)} + \boldsymbol{w}^{(2)} + \boldsymbol{w}^{(3)} + ...$ for small $(\rho(\boldsymbol{x}, t) - \rho_0)$. The first term of order $O\left(A/R_0\right)$ can be written as the convolution

$$\boldsymbol{w}^{(0)}(\boldsymbol{x}, t) = \int_{\Omega_f} G(\boldsymbol{y}, t)\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{y})d\Omega_f \,, \tag{2.52}$$

$$p_w^{(0)}(\boldsymbol{x}, t) = \int_{\Omega_f} G(\boldsymbol{y}, t)P(\boldsymbol{x}, \boldsymbol{y})d\Omega_f \,, \tag{2.53}$$

where Ω_f is the fluid domain.

By interchanging the order of integration (which is allowed because of the convergence properties determined by the integrands), the force calculation for the first order approximation

becomes

$$\begin{aligned}
\oint_S {\sigma_w^{(0)}}_{ij} n_j dS &= \oint_S \left(-p_w^{(0)} \delta_{ij} + \mu \left(\frac{\partial w_i^{(0)}}{\partial x_j} + \frac{\partial w_j^{(0)}}{\partial x_i} \right) \right) n_j dS \\
&= \int_{\Omega_f} G(\boldsymbol{y}, t) \oint_S \left(-P(\boldsymbol{x}, \boldsymbol{y}) \delta_{ij} + \mu \left(\frac{\partial W_i(\boldsymbol{x}, \boldsymbol{y})}{\partial x_j} + \frac{\partial W_j(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i} \right) \right) n_j dS \, d\Omega_f \\
&= -\int_{\Omega_f} G(\boldsymbol{y}, t) \frac{A\hat{g}}{4} \left\{ -\frac{3\left(r^2 + y_3^2\right)}{r^3} - \frac{A^2\left(r^2 - 3y_3^2\right)}{r^5} \right\} d\Omega_f \,, \qquad (2.54)
\end{aligned}$$

where $r = |\boldsymbol{y}|$.

Equations (2.52) and (2.54) determine the first order approximation to the fluid flow and the resultant force on the sphere due to the density variation.

2.3 Final Equations of motion

Combining the results from Sections 2.1 and 2.2, we have the equation for the vertical component of the velocity of the sphere and the advection of the fluid,

$$\begin{aligned}
m_s \frac{dV(t)}{dt} &= m_s g - g \int_{\Omega_s} \rho_0 d\Omega_s \\
&\quad -6\pi A \mu V(t) \left(1 - 2.10444(A/R_0) + 2.08877(A/R_0)^3 + ...\right)^{-1} \\
&\quad - \int_{\Omega_f} G(\boldsymbol{y}, t) \frac{A\hat{g}}{4} \left\{ -\frac{3\left(r^2 + y_3^2\right)}{r^3} - \frac{A^2\left(r^2 - 3y_3^2\right)}{r^5} \right\} d\Omega_f, \qquad (2.55) \\
\frac{\partial \rho}{\partial t}(\boldsymbol{x}, t) &+ (\boldsymbol{u}(\boldsymbol{x}, t) + \boldsymbol{w}(\boldsymbol{x}, t)) \cdot \nabla \rho(\boldsymbol{x}, t) = 0. \qquad (2.56)
\end{aligned}$$

This can be further reduced when written in nondimensional form,

$$\begin{aligned}
Re\, St\, \frac{4\pi}{3} \frac{\rho_s}{\rho_{ref}} \frac{d\tilde{V}(t)}{dt} &= \frac{Re}{Fr^2} \left(\frac{4\pi}{3} \frac{\rho_s}{\rho_{ref}} - \int_{\tilde{\Omega}_s} \tilde{\rho}_0 d\tilde{\Omega}_s \right) \\
&\quad -6\pi \tilde{V}(t) \left(1 - 2.10444(A/R_0) + 2.08877(A/R_0)^3 + ...\right)^{-1} \\
&\quad + \frac{Re}{Fr^2} \int_{\tilde{\Omega}_f} \frac{G(\boldsymbol{y}, t)}{4} \left\{ \frac{3\left(\tilde{r}^2 + \tilde{y}_3^2\right)}{\tilde{r}^3} + \frac{\left(\tilde{r}^2 - 3\tilde{y}_3^2\right)}{\tilde{r}^5} \right\} d\tilde{\Omega}_f \,, \qquad (2.57)
\end{aligned}$$

which shows that the sphere's acceleration term dV/dt can be scaled out. We are left with

$$\tilde{V}(t) = \frac{Re}{Fr^2}\left(\frac{4\pi}{3}\frac{\rho_s}{\rho_{ref}} - \int_{\tilde{\Omega}_s}\tilde{\rho}_0 d\tilde{\Omega}_s + \int_{\tilde{\Omega}_f}\frac{G(\boldsymbol{y},t)}{4}\left\{\frac{3\left(\tilde{r}^2+\tilde{y}_3^2\right)}{\tilde{r}^3} + \frac{\left(\tilde{r}^2-3\tilde{y}_3^2\right)}{\tilde{r}^5}\right\}d\tilde{\Omega}_f\right)/(6\pi K), \tag{2.58}$$

$$St\frac{\partial\tilde{\rho}}{\partial\tilde{t}}(\boldsymbol{x},t) + (\tilde{\boldsymbol{u}}(\boldsymbol{x},t) + \tilde{\boldsymbol{w}}(\boldsymbol{x},t))\cdot\tilde{\nabla}\tilde{\rho}(\boldsymbol{x},t) = 0, \tag{2.59}$$

where $K = (1 - 2.10444(A/R_0) + 2.08877(A/R_0)^3 + ...)^{-1}$.

In dimensional form, we have our final equations of motion for the sphere and fluid velocities:

$$\begin{aligned}\frac{dY_3}{dt}(t;\rho) = V(t;\rho) = (6\pi A\mu K)^{-1}&\left(m_s g - g\int_{\Omega_s}\rho_0(x_3+Y_3(t;\rho))d\Omega_s + \right.\\ &\left. + \int_{\Omega_f}G(\boldsymbol{y},t)\frac{A\hat{g}}{4}\left\{\frac{3\left(r^2+y_3^2\right)}{r^3} + \frac{A^2\left(r^2-3y_3^2\right)}{r^5}\right\}d\Omega_f\right),\\ \frac{\partial\rho}{\partial t}(\boldsymbol{x},t) + (\boldsymbol{u}(\boldsymbol{x},t;V) + \boldsymbol{w}(\boldsymbol{x},t;\rho))\cdot\nabla\rho(\boldsymbol{x},t) &= 0.\end{aligned} \tag{2.60}$$

The detailed formulations for the fluid velocitiy $\boldsymbol{w}$ can be found in Chapter 4. The formulas (2.60) in addition to the equations (2.27)–(2.30) and (2.52) for the approximation to the velocity field compose our final model for the settling of the sphere with the specified density stratification.

Mathematically, this model is a coupled pair of integro-differential equations in both $\rho(\boldsymbol{x},t)$ and $Y_3(t;\rho)$, where the speed of the sphere $V(t)$ determines $\boldsymbol{u}$, and the density field $\rho(\boldsymbol{x},t)$ determines the domain of integration for $\boldsymbol{w}$. The initial data completely determine the future evolution of the density field by advection through the fluid velocities, whose combination satisfies the rigid boundary conditions (to within the accuracy of the analytic approximations based on Stokes flow theory). The equations (2.60), show the highly coupled system by emphasizing the dependence on ρ and V by writing them as arguments of the functions.

CHAPTER 3

STOKES SOLUTION THIRD REFLECTION

3.1 Solution by Stream Function

The third reflection $\boldsymbol{u}^{(3)} = \left(u_R^{(3)}, 0, u_Z^{(3)}\right)$ solves the Stokes equations with a prescribed flow past the sphere. In cylindrical coordinates, the system of equations reads

$$\begin{aligned}
\mu\nabla^2\boldsymbol{u}^{(3)} &= \nabla p^{(3)} && (3.1)\\
\nabla\cdot\boldsymbol{u}^{(3)} &= 0 && (3.2)\\
u_Z^{(3)}\Big|_{r=A} &= -u_Z^{(2)}\Big|_{r=A} = 2.10444\frac{A}{R_0}V - 2.18004\frac{A^3}{R_0^3}V - 0.140011\frac{A}{R_0^3}R^2V && (3.3)\\
u_R^{(3)}\Big|_{r=A} &= -u_R^{(2)}\Big|_{r=A} = 1.13669\frac{A}{R_0^3}\,R\,Z\,V && (3.4)\\
\boldsymbol{u}^{(3)} &\to 0 \text{ as } r\to\infty. && (3.5)
\end{aligned}$$

In spherical coordinates, axial symmetry acts the same as above and $\boldsymbol{u}^{(3)} = \left(u_r^{(3)}, u_\theta^{(3)}, 0\right)$We define the stream function ψ to satisfy incompressibility. For the spherical coordinate system (r,θ,ϕ) , where θ is the polar angle and ϕ the aximuzal angle, the stream function is defined as

$$\begin{aligned}
u_r &= \frac{1}{r^2\sin\theta}\frac{\partial\psi}{\partial\theta} && (3.6)\\
u_\theta &= -\frac{1}{r\sin\theta}\frac{\partial\psi}{\partial r}. && (3.7)
\end{aligned}$$

The momentum equation and boundary conditions for ψ becomes

$$\left(\frac{\partial^2}{\partial r^2}+\frac{\sin\theta}{r^2}\frac{\partial}{\partial\theta}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)\right)^2\psi=0, \tag{3.8}$$

with boundary conditions

$$u_r\Big|_{r=A} = \frac{1}{r^2\sin\theta}\frac{\partial\psi}{\partial\theta}\Big|_{r=A}=B_1\cos\theta+B_2\cos 3\theta, \tag{3.9}$$

$$u_\theta\Big|_{r=A} = -\frac{1}{r\sin\theta}\frac{\partial\psi}{\partial r}\Big|_{r=A}=B_3\sin\theta+B_4\sin 3\theta, \tag{3.10}$$

where

$$B_1 = \frac{AV\left(2.10444R_0^2-1.86086A^2\right)}{R_0^3} \tag{3.11}$$

$$B_2 = -\frac{0.319175A^3V}{R_0^3} \tag{3.12}$$

$$B_3 = \frac{AV\left(2.3592A^2-2.10444R_0^2\right)}{R_0^3} \tag{3.13}$$

$$B_4 = \frac{0.319175A^3V}{R_0^3} \tag{3.14}$$

and the behavior at infinity $r\to\infty$,

$$\frac{1}{r^2\sin\theta}\frac{\partial\psi}{\partial\theta} \underset{r\to\infty}{\to} 0 \tag{3.15}$$

$$-\frac{1}{r\sin\theta}\frac{\partial\psi}{\partial r} \underset{r\to\infty}{\to} 0. \tag{3.16}$$

The above restrictions leads us to find a solution of the form

$$\psi=f(r)+g(r)\cos 2\theta+h(r)\cos 4\theta. \tag{3.17}$$

From the boundary conditions, we obtain the following restrictions on f, g and h

$$
\begin{aligned}
-\frac{4\left(g(A)+2h(A)\right)}{r^2} &= B_1 && (3.18)\\
-\frac{8h(A)}{r^2} &= B_2 && (3.19)\\
g(A) &= -f(A)-h(A) && (3.20)\\
-\frac{2f(A)}{r} &= B_3 && (3.21)\\
\frac{2h(A)}{r} &= B_4. && (3.22)
\end{aligned}
$$

The above together with the conditions at infinity, we get that the stream function is

$$
\begin{aligned}
\psi(r,\theta) = &-\left(20r^3\right)^{-1}\left(A\sin^2(\theta)\left(3A^4(B_2-4B_4)\right.\right.\\
&-A^2r^2(5B_1+6B_2+10B_3-14B_4)+5A^2\cos(2\theta)\left(A^2(B_2-4B_4)\right. \qquad (3.23)\\
&\left.\left.+\ r^2(4B_4-3B_2)\right)+r^4(-5B_1+3B_2+10B_3-2B_4)\right)\right),
\end{aligned}
$$

and the flow in spherical coordinate is,

$$\begin{aligned} u_r &= \frac{A^2V}{4r^5R_0^3}\cos(\theta)\left(A^6((-0.14001)-(1.13669))\right. \\ &+ A^4r^2(3(1.13669)-2(-2.18001)+(-0.14001)) \\ &+ A^2\left(-2(2.10444)r^2R_0^2+6(-2.18001)r^4\right. \\ &- \left.4(-0.14001)r^4\right)+A^4\left(5A^2-7r^2\right)((1.13669) \\ &- \left.(-0.14001))\cos(2\theta)+6(2.10444)r^4R_0^2\right), \end{aligned} \tag{3.24}$$

$$\begin{aligned} u_\theta &= \frac{A^2V}{16r^5R_0^3}\sin(\theta)\left(9A^6((1.13669)-(-0.14001))-A^4r^2((1.13669)\right. \\ &+ 4(-2.18001)-9(-0.14001)) \\ &- 4A^2\left((2.10444)r^2R_0^2+3(-2.18001)r^4-2(-0.14001)r^4\right) \\ &+ A^4\left(15A^2-7r^2\right)((1.13669) \\ &- \left.(-0.14001))\cos(2\theta)-12(2.10444)r^4R_0^2\right). \end{aligned} \tag{3.25}$$

Changing to cylindrical coordinates, we obtain the expression for the third reflection of the stokes velocity.

$$\begin{aligned} u_R^{(3)} &= \frac{5A^8RVZ((-1.13669)-(0.140011))\left(3R^2-4Z^2\right)}{8R_0^3r^9} \\ &- \frac{A^6RVZ\left((-1.13669)\left(23R^2-12Z^2\right)-6(2.18004)r^2\right)}{8R_0^3r^7} \\ &- \frac{A^6RVZ\left(-11(0.140011)R^2+24(0.140011)Z^2\right)}{8R_0^3r^7} \\ &- \frac{A^4RVZ\left(-3(-2.10444)R_0^2+3(2.18004)r^2-2(0.140011)r^2\right)}{4R_0^3r^5} \\ &- \frac{3A^2(-2.10444)RVZ}{4R_0r^3}, \end{aligned} \tag{3.26}$$

$$
\begin{aligned}
u_Z^{(3)} = & -\frac{A^8 V((-1.13669)-(0.140011))\left(3R^4-24R^2Z^2+8Z^4\right)}{8R_0^3 r^9} \\
& -\frac{A^6 V\left(-R^2-Z^2\right)(-1.13669)\left(3R^4-24R^2Z^2+8Z^4\right)}{8R_0^3 r^9} \\
& -\frac{A^6 V\left((2.18004)\left(-2R^4+2R^2Z^2+4Z^4\right)+(0.140011)\left(R^4+20R^2Z^2-16Z^4\right)\right)}{8R_0^3 r^9} \\
& -\frac{A^4 V\left((-2.10444)R_0^2\left(R^2-2Z^2\right)+3(2.18004)\left(R^4+3R^2Z^2+2Z^4\right)\right)}{4R_0^3 r^5} \\
& -\frac{A^4 V\left(-2(0.140011)\left(R^4+3R^2Z^2+2Z^4\right)\right)}{4R_0^3 r^5} \\
& -\frac{3A^2(-2.10444)V\left(R^2+2Z^2\right)}{4R_0 r^3}.
\end{aligned}
\tag{3.27}
$$

3.2 Second Reflection Approximation

Performing the change of variables $\alpha = \lambda R_0$, the second reflection from Section (2.1), $\boldsymbol{u}^{(2)} = \left(u_R^{(2)}, 0, u_Z^{(2)}\right)$ in cylindrical coordinates, is expressed as

$$\begin{aligned} u_R^{(2)}(R,Z) &= \frac{1}{2\pi R_0}\int_0^\infty \hat{u}_R(R/R_0,\alpha)\sin(\alpha Z/R_0)d\alpha, \\ u_Z^{(2)}(R,Z) &= \frac{1}{2\pi R_0}\int_0^\infty \hat{u}_Z(R/R_0,\alpha)\cos(\alpha Z/R_0)d\alpha, \end{aligned} \tag{3.28}$$

where

$$\hat{u}_R(R/R_0,\alpha) = \frac{\alpha R}{2R_0}\Big(H(\alpha)+G(\alpha)\Big)\, I_0\left(\alpha\frac{R}{R_0}\right) - G(\alpha) I_1\left(\alpha\frac{R}{R_0}\right), \tag{3.29}$$

$$\hat{u}_Z(R/R_0,\alpha) = \frac{\alpha R}{2R_0}\Big(H(\alpha)+G(\alpha)\Big)\, I_1\left(\alpha\frac{R}{R_0}\right) + H(\alpha) I_0\left(\alpha\frac{R}{R_0}\right), \tag{3.30}$$

and

$$H(\alpha) = \frac{AV\left\{3 - \left(6+\frac{A^2}{R_0^2}\alpha^2\right)\Big(K_0(\alpha)I_2(\alpha)+K_1(\alpha)I_1(\alpha)\Big)\right\}}{I_0(\alpha)I_2(\alpha) - I_1(\alpha)^2}, \tag{3.31}$$

$$G(\alpha) = \frac{AV\left\{-3 + \frac{A^2}{R_0^2}\alpha^2\Big(K_1(\alpha)I_1(\alpha)+K_2(\alpha)I_0(\alpha)\Big)\right\}}{I_0(\alpha)I_2(\alpha) - I_1(\alpha)^2}, \tag{3.32}$$

where I_j and K_j are modified Bessel functions of the first and second kind.

The integrals in equation (3.28) are not easily solved analytically, therefore to evaluate the flow on the sphere we can perform an expansion far from the tank walls as $R/R_0 \to 0$ and $Z/R_0 \to 0$.

To expand the integrand like in [18] by their infinite series representation, we use the

following series expansions which converge in the whole complex plane.

$$\cos(\lambda Z) = \cos\left(\frac{\alpha Z}{R_0}\right) = \sum_{n=0}^{\infty} \frac{(-1)^n (\alpha Z)^{2n}}{R_0^{2n}(2n)!} = 1 - \frac{\alpha^2 Z^2}{2R_0^2} + \dots \tag{3.33}$$

$$\sin(\lambda Z) = \sin\left(\frac{\alpha Z}{R_0}\right) = \sum_{n=0}^{\infty} \frac{(-1)^n (\alpha Z)^{2n+1}}{R_0^{2n+1}(2n+1)!} = \frac{\alpha Z}{R_0} + \dots \tag{3.34}$$

$$I_0(\lambda R) = I_0\left(\frac{\alpha Z}{R_0}\right) = \sum_{n=0}^{\infty} \frac{(1/4)^n (\alpha R)^{2n}}{n!\Gamma(n+1)R_0^{2n}} = 1 - \frac{\alpha^2 Z^2}{2R_0^2} + \dots \tag{3.35}$$

$$I_1(\lambda R) = I_1\left(\frac{\alpha Z}{R_0}\right) = \frac{\alpha R}{2R_0} \sum_{n=0}^{\infty} \frac{(1/4)^n (\alpha R)^{2n}}{n!\Gamma(n+2)R_0^{2n}} = 1 - \frac{\alpha^2 Z^2}{2R_0^2} + \dots \tag{3.36}$$

The second reflection then becomes,

$$\begin{aligned} u_R^{(2)} &= \frac{1}{2\pi R_0} \int_0^{\infty} d\lambda \left(\frac{\alpha}{2} \frac{R}{R_0} \Big(H(\alpha) + G(\alpha) \Big) \sum_{n=0}^{\infty} \frac{(1/4)^n (\alpha R)^{2n}}{n!\Gamma(n+1)R_0^{2n}} \right. \\ &- \left. G(\alpha) \frac{\alpha R}{2R_0} \sum_{n=0}^{\infty} \frac{(1/4)^n (\alpha R)^{2n}}{n!\Gamma(n+2)R_0^{2n}} \right) \left(\sum_{n=0}^{\infty} \frac{(-1)^n (\alpha Z)^{2n+1}}{R_0^{2n+1}(2n+1)!} \right), \end{aligned} \tag{3.37}$$

$$\begin{aligned} u_Z^{(2)} &= \frac{1}{2\pi R_0} \int_0^{\infty} d\lambda \left(\frac{\alpha}{2} \frac{R}{R_0} \Big(H(\alpha) + G(\alpha) \Big) \frac{\alpha R}{2R_0} \sum_{n=0}^{\infty} \frac{(1/4)^n (\alpha R)^{2n}}{n!\Gamma(n+2)R_0^{2n}} \right. \\ &+ \left. H(\alpha) \sum_{n=0}^{\infty} \frac{(1/4)^n (\alpha R)^{2n}}{n!\Gamma(n+1)R_0^{2n}} \right) \left(\sum_{n=0}^{\infty} \frac{(-1)^n (\alpha Z)^{2n}}{R_0^{2n}(2n)!} \right). \end{aligned} \tag{3.38}$$

Here, we justify Happel and Byrne expansion as $R/R_0 \to 0$ and $Z/R_0 \to 0$ inside the infinite α integral. Using the asymptotic expansions as $\alpha \to \infty$ provided in [2] and the series representation, we show the convergence of the integrals of the partial sums due to their exponentially decaying behavior as $\alpha \to \infty$. Using,

$$I_j(\alpha) \underset{\alpha\to\infty}{\sim} \frac{e^{\alpha}}{\sqrt{2\pi\lambda R_0}} \left(1 - \frac{4j^2 - 1}{8\lambda R_0} + \dots \right) \tag{3.39}$$

$$K_j(\alpha) \underset{\alpha\to\infty}{\sim} \sqrt{\frac{\pi}{2\alpha}} e^{-\alpha} \left(1 + \frac{4j^2 - 1}{8\alpha} + \dots \right), \tag{3.40}$$

we get that,

$$H(\alpha) \underset{\alpha\to\infty}{\sim} \frac{2\pi A^3 V}{R_0^2}\alpha^3 e^{-2\alpha}. \tag{3.41}$$

and,

$$H(\alpha) = \hat{H}(\alpha)\frac{2\pi A^3 V}{R_0^2}\alpha^3 e^{-2\alpha} \tag{3.42}$$

$$\text{where } \hat{H}(\alpha) \to 1 \text{ as } \alpha \to \infty. \tag{3.43}$$

For the purpose of being brief, we focus on the last portion of the $u_Z^{(2)}$ integral:

$$\int_0^\infty F d\alpha = \int_0^\infty d\alpha H(\alpha) I_0(\alpha R/R_0)\cos(\alpha Z/R_0) \tag{3.44}$$

$$F = \left(H(\alpha)\sum_{n=0}^{\infty}\frac{(1/4)^n(\alpha R)^{2n}}{n!\Gamma(n+1)R_0^{2n}}\right)\cos\left(\alpha Z/R_0\right). \tag{3.45}$$

Let $F = \sum F_n$, and F_n given by:

$$F_n = H(\alpha)\frac{(1/4)^n(\alpha R)^{2n}}{n!^2 R_0^{2n}}\cos(\alpha Z/R_0) \tag{3.46}$$

We know that,

$$\int_0^\infty \left|F_n\right| d\alpha \;\leq\; C_1\frac{(1/4)^n}{n!^2}\int_0^\infty \left(\alpha^{2n+3}e^{-2\alpha}\right) d\alpha, \tag{3.47}$$

Therefore solving the integral we get,

$$\int_0^\infty \left|F_n\right| d\alpha \leq C_1\frac{(1/4)^n}{n!^2}\left(\frac{(2n+3)!}{2^{2n+4}}\right) \tag{3.48}$$

taking the infinite sum

$$\begin{aligned}\sum_{n=0}^{\infty}\int_0^{\infty}\left|F_n\right|d\alpha &\leq \sum_{n=0}^{\infty}C_1\frac{(1/4)^n}{n!^2}\left(\frac{2n+3}{2^{2n+4}}\right) &(3.49)\\ &\leq C_1\frac{22}{9\sqrt{3}} &(3.50)\\ &< \infty. &(3.51)\end{aligned}$$

And so, since

$$\sum_{n=0}^{\infty}\int_0^{\infty}\left|F_n\right|d\alpha<\infty \tag{3.52}$$

Then by the Lebesgue Dominated Theorem,

$$\int_0^{\infty}F d\alpha=\sum_{n=0}^{\infty}\int_0^{\infty}F_n d\alpha, \tag{3.53}$$

and

$$\int_0^{\infty}F d\alpha\sim\int_0^{\infty}\sum_{n=0}^{N}F_n d\alpha. \tag{3.54}$$

Therefore,

$$\begin{aligned}&\int_0^{\infty}d\alpha H(\alpha)I_0(\alpha R/R_0)\cos(\alpha Z/R_0) &(3.55)\\ =\ &\int_0^{\infty}d\alpha H(\alpha)\sum_{n=0}^{\infty}\frac{(1/4)^n(\alpha R)^{2n}}{n!\Gamma(n+1)R_0^n}\sum_{n=0}^{\infty}\frac{(-1)^n(\alpha Z)^{2n}}{R_0^{2n}(2n)!} &(3.56)\\ \sim\ &\int_0^{\infty}d\alpha\, H(\alpha)+\int_0^{\infty}d\alpha\, H(\alpha)\frac{\alpha^2R^2}{4R_0^2}+\ldots &(3.57)\end{aligned}$$

The α integrals can be computed numerically to arbitrary accuracy providing a way to find the boundary conditions near the sphere. Similarly, it can be shown that other integrals involved in the computation of $\boldsymbol{u}^{(2)}$ satisfy the same property. In Figure 3.1, we compare $\boldsymbol{u}^{(2)}$

with its approximation .

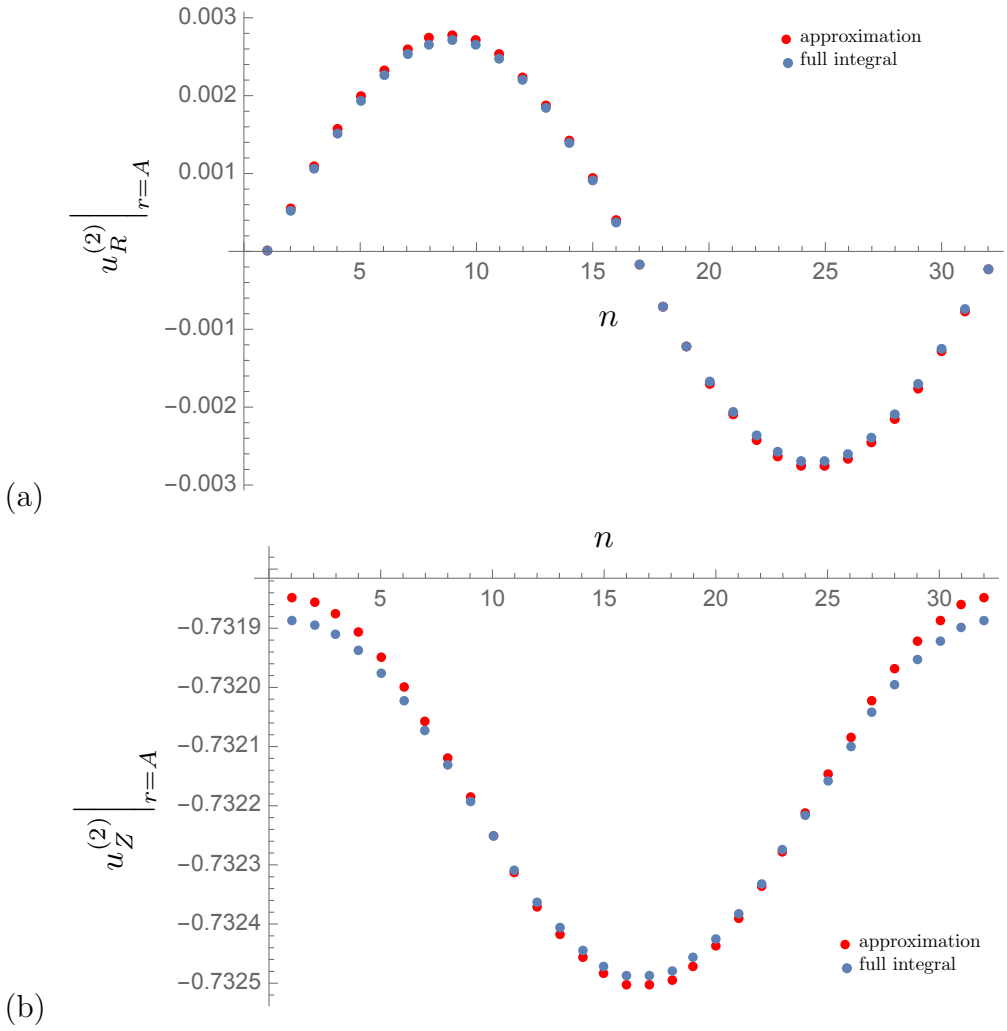


Figure 3.1: Comparison of $\boldsymbol{u}^{(2)}$ with its expansion as $R/R0 \to 0$ and $Z/R_0 \to 0$, evaluating on the surface of the sphere $r = \sqrt{R^2 + Z^2} = A$, along $\theta = [0, \pi]$ discretized by n.Blue dots correspond to numerical integration and red dots to the approximation. Values are for $A = 0.635\,cm$, $R_0 = 5.4\,cm$, $V = 1$.

CHAPTER 4

THE OSEEN TENSOR

4.1 Definition of Perturbation Velocity

The perturbation velocity is written as a convolution with the Green's function $W_j(\boldsymbol{x},\boldsymbol{y})$,

$$w_j^{(0)}(\boldsymbol{x},t) = \int_{\Omega_f} \epsilon G(\boldsymbol{y},t) W_j(\boldsymbol{x},\boldsymbol{y}) d\Omega_f \,, \tag{4.1}$$

where the Green's function is written explicitly in terms of the Oseen tensor [29] T_{jk} as $W_j(\boldsymbol{x},\boldsymbol{y}) = gT_{j3}/8\pi\mu$ and

$$\begin{aligned} T_{jk} = \frac{\delta_{jk}}{r} + \frac{(x_j-y_j)(x_k-y_k)}{r^3} - \frac{a}{|\boldsymbol{y}|}\frac{\delta_{jk}}{r^*} - \frac{a^3}{|\boldsymbol{y}|^3}\frac{(x_j-y_j^*)(x_k-y_k^*)}{r^{*3}} \\ - \frac{|\boldsymbol{y}|^2-a^2}{|\boldsymbol{y}|}\left\{\frac{y_j^*y_k^*}{a^3r^*} - \frac{a}{|\boldsymbol{y}|^2 r^{*3}}\left[y_j^*(x_k-y_k^*)+y_k^*(x_j-y_j^*)\right] + \frac{2y_j^*y_k^*}{a^3}\frac{y_l^*(x_l-y_l^*)}{r^{*3}}\right\} \\ - (|\boldsymbol{x}|^2-a^2)\frac{\partial \Phi_k}{\partial x_j}, \end{aligned}$$

$$\Phi_k = \frac{|\boldsymbol{y}|^2-a^2}{2|\boldsymbol{y}|^3}\left(\frac{3y_k}{ar^*} + \frac{a(x_k-y_k^*)}{r^{*3}} + \frac{2y_k}{a}y_j^*\frac{\partial}{\partial x_j}\frac{1}{r^*} + \frac{3a}{|\boldsymbol{y}^*|}\frac{\partial}{\partial y_k^*}\log\frac{|\boldsymbol{y}^*|r^* + x_jy_j^* - |\boldsymbol{y}^*|^2}{|\boldsymbol{x}||\boldsymbol{y}^*| + x_jy_j^*}\right),$$

where a is the radius of the sphere, $\boldsymbol{y}^* = \frac{a^2}{|\boldsymbol{y}|^2}\boldsymbol{y}$, $r = |\boldsymbol{y}-\boldsymbol{x}|$, $r^* = |\boldsymbol{y}^*-\boldsymbol{x}|$, and the convention of sum over repeated indexes is used. Thus, the Green's function can be viewed as resulting from the superposition of appropriate singularities inside the radius-a sphere at the reflection point corresponding to the position $\boldsymbol{y}$ of the Stokeslet outside the sphere

4.2 Integrable Singularities

The integrand W_j has a singularities at $\boldsymbol{x} = \boldsymbol{y}$ coming from the terms

$$\frac{\delta_{jk}}{r} + \frac{(x_j - y_j)(x_k - y_k)}{r^3} - \frac{a}{|\boldsymbol{y}|}\frac{\delta_{jk}}{r^*} \tag{4.2}$$

For simplicity, let us focus on the third component of the velocity W_3, defining 12 terms that add up to the total Green's function vertical component. Each term is analyzed for singularities separately.

$$W_3 = \sum_{n=1}^{n=12} I_n \tag{4.3}$$

This first integrand term I_1 is defined as

$$I_1 = \frac{1}{r} + \frac{(x_3 - y_3)^2}{r^3}. \tag{4.4}$$

The singularities of this term happen when

$$r = |\boldsymbol{x} - \boldsymbol{y}| = 0 \tag{4.5}$$

$$\boldsymbol{x} = \boldsymbol{y}$$

These singularities are integrable. In fact, in cylindrical coordinates

$$
\begin{aligned}
&\int_{\Omega_f} I_1 d\Omega_f \\
= &\int_{\Omega_f} \left(\frac{1}{r} + \frac{(x_3-y_3)^2}{r^3}\right) d\Omega_f \\
= &\int_{(\rho,\zeta)} \rho\, d\rho\, d\zeta \int_0^{2\pi} d\theta \Big(\frac{1}{\sqrt{R^2+\rho^2-2R\rho\cos(\phi-\theta)+(Z-\zeta)^2}} \\
&+ \frac{(Z-\zeta)^3}{(R^2+\rho^2-2R\rho\cos(\phi-\theta)+(Z-\zeta)^2)^{\frac{3}{2}}} \Big) \\
= &\int_{(\rho,\zeta)} \rho\, d\rho\, d\zeta \frac{4K\left(-\frac{4R\rho}{R^2-2\rho R+\rho^2+(\zeta-Z)^2}\right)}{\sqrt{(\rho-R)^2+(\zeta-Z)^2}} + \frac{4(Z-\zeta)^2 E\left(-\frac{4R\rho}{R^2-2\rho R+\rho^2+(\zeta-Z)^2}\right)}{((\rho-R)^2+(\zeta-Z)^2)^{\frac{3}{2}}}
\end{aligned}
\tag{4.6}
$$

where K and E are elliptic integrals of the first and second kind respectively. To proof integrability we multiply the integrand by an area element and bound it. If

$$
\lim_{\substack{\rho\to R \\ \zeta\to Z}} \left(\sqrt{(\rho-R)^2+(\zeta-Z)^2}\right)^{\alpha} \Big|I(\rho,\zeta)\Big| \le M, \tag{4.7}
$$

then,

$$
\Big|I(\rho,\zeta)\Big| \le \frac{M}{\left(\sqrt{(\rho-R)^2+(\zeta-Z)^2}\right)^{\alpha}} \tag{4.8}
$$

for $\alpha < 2$ because in a neighborhood D of (R,Z)

$$
\begin{aligned}
&\int_D \frac{M}{\left(\sqrt{(\rho-R)^2+(\zeta-Z)^2}\right)^{\alpha}} d\rho d\zeta < \infty \\
&\Rightarrow \int_D \Big|I(\rho,\zeta)\Big| d\rho d\zeta < \infty \\
&\Rightarrow \int_D I(\rho,\zeta) d\rho d\zeta < \infty
\end{aligned}
\tag{4.9}
$$

The integrand has one singularity when $R=\rho$ and $Z=\zeta$. Around this point $(R,Z) = (\rho,\zeta)$, the elliptic integral of the first kind K approaches zero and $E \sim i\sqrt{\frac{4R\rho}{R^2-2\rho R+\rho^2+(\zeta-Z)^2}}$

$$I(\rho,\zeta) = \rho\left(\frac{4K\left(-\frac{4R\rho}{R^2-2\rho R+\rho^2+(\zeta-Z)^2}\right)}{\sqrt{(\rho-R)^2+(\zeta-Z)^2}} + \frac{4(Z-\zeta)^2E\left(-\frac{4R\rho}{R^2-2\rho R+\rho^2+(\zeta-Z)^2}\right)}{((\rho-R)^2+(\zeta-Z)^2)^{\frac{3}{2}}}\right)$$

$$I \leq \rho\left(\frac{4}{\sqrt{(\rho-R)^2+(\zeta-Z)^2}} + \frac{4(Z-\zeta)^2E\left(-\frac{4R\rho}{R^2-2\rho R+\rho^2+(\zeta-Z)^2}\right)}{((\rho-R)^2+(\zeta-Z)^2)^{\frac{3}{2}}}\right)$$

$$\lim_{\substack{\rho\to R\\ \zeta\to Z}}\left(\sqrt{(\rho-R)^2+(\zeta-Z)^2}\right)^{\alpha}\left|\rho\left(\frac{4}{\sqrt{(\rho-R)^2+(\zeta-Z)^2}} + \frac{16R\rho(Z-\zeta)^2}{((\rho-R)^2+(\zeta-Z)^2)^2}\right)\right| \leq M$$

This remaining integrands with denominator r^* have no singularities because $r^* \neq 0$ since $\boldsymbol{y}^*$ is always located inside the sphere, exterior to the fluid domain.

4.3 Line of Removable Singularities

Before we integrate the Green' s function over the fluid domain, it is convenient to combine alike singularities and recognize cancelations. The line of removable singularities appears when the observation point $\boldsymbol{x}$ is co-linear with the evaluation point $\boldsymbol{y}$ in the opposite direction. They come from the log term I_L in the Φ expression and cancel out leaving no problems in the physical domain.

$$\begin{aligned}
I_L &= I_9 + I_{10} + I_{11} + I_{12} \\
I_L &= \frac{-|\boldsymbol{y}^*|\left(\frac{1}{|\boldsymbol{x}-\boldsymbol{y}^*|} - \frac{(x_3-y_3^*)^2}{|\boldsymbol{x}-\boldsymbol{y}^*|^3}\right) + \frac{y_3^*(x_3-y_3^*)}{|\boldsymbol{y}^*||\boldsymbol{x}-\boldsymbol{y}^*|} + 1}{-|\boldsymbol{y}^*|^2 + |\boldsymbol{y}^*||\boldsymbol{x}-\boldsymbol{y}^*| + x_1y_1^* + x_2y_2^* + x_3y_3^*} \\
&- \frac{\left(\frac{|\boldsymbol{y}^*|(x_3-y_3^*)}{|\boldsymbol{x}-\boldsymbol{y}^*|} + y_3^*\right)\left(-\frac{|\boldsymbol{y}^*|(x_3-y_3^*)}{|\boldsymbol{x}-\boldsymbol{y}^*|} + \frac{|\boldsymbol{x}-\boldsymbol{y}^*|y_3^*}{|\boldsymbol{y}^*|} + x_3 - 2y_3^*\right)}{\left(-|\boldsymbol{y}^*|^2 + |\boldsymbol{y}^*||\boldsymbol{x}-\boldsymbol{y}^*| + x_1y_1^* + x_2y_2^* + x_3y_3^*\right)^2} \\
&- \frac{\frac{x_3y_3^*}{|\boldsymbol{y}^*||\boldsymbol{x}|} + 1}{|\boldsymbol{y}^*||\boldsymbol{x}| + x_1y_1^* + x_2y_2^* + x_3y_3^*} + \frac{\left(\frac{|\boldsymbol{y}^*|x_3}{|\boldsymbol{x}|} + y_3^*\right)\left(\frac{|\boldsymbol{x}|y_3^*}{|\boldsymbol{y}^*|} + x_3\right)}{(|\boldsymbol{y}^*||\boldsymbol{x}| + x_1y_1^* + x_2y_2^* + x_3y_3^*)^2}
\end{aligned}$$

As an example, lets take a closer look at the term I_9,

$$I_9 = \frac{3A\left((x_3 - y_3^*)\left(|\boldsymbol{y^*}|^2(x_3 - y_3^*) + r^{*2}y_3^*\right) - r^{*2}|\boldsymbol{y^*}|(|\boldsymbol{y^*}| - r^*)\right)}{r^{*3}|\boldsymbol{y^*}|^2\left(r^*|\boldsymbol{y^*}| - |\boldsymbol{y^*}|^2 + x_1y_1^* + x_2y_2^* + x_3y_3^*\right)}.$$

The singularities of this term I_9 happen when its denominator vanishers at

$$\begin{aligned}
r^*|\boldsymbol{y^*}| - |\boldsymbol{y^*}|^2 + x_1y_1^* + x_2y_2^* + x_3y_3^* &= 0 \\
|\boldsymbol{y^*}|\left(r^* - |\boldsymbol{y^*}| + \boldsymbol{x}\cdot\frac{\boldsymbol{y}^*}{|\boldsymbol{y^*}|}\right) &= 0 \\
|\boldsymbol{y^*}|\left(r^* - |\boldsymbol{y^*}| + (\boldsymbol{x} - \boldsymbol{y^*} + \boldsymbol{y^*})\cdot\frac{\boldsymbol{y}^*}{|\boldsymbol{y^*}|}\right) &= 0 \\
|\boldsymbol{y^*}|\left(r^* + (\boldsymbol{x} - \boldsymbol{y^*})\cdot\boldsymbol{n}^*\right) &= 0 \\
|\boldsymbol{x} - \boldsymbol{y^*}| + (\boldsymbol{x} - \boldsymbol{y^*})\cdot\boldsymbol{n}^* &= 0 \\
\iff \boldsymbol{x} - \boldsymbol{y^*} &= \alpha\boldsymbol{y^*} \\
|\alpha||\boldsymbol{y^*}| &= \alpha\boldsymbol{y^*}\cdot\frac{\boldsymbol{y^*}}{|\boldsymbol{y^*}|} \\
-|\alpha| &= \alpha \\
\boldsymbol{x} &= \boldsymbol{y^*} + \alpha\cdot\boldsymbol{y^*} \\
\alpha &< 0
\end{aligned}$$

We can see the denominator of each term in I_L vanishes along an entire line, and it is not surprising that each term alone is divergent, since we know we need the addition of all the log terms $I_9 + I_{10} + I_{11} + I_{12}$ for the cancelation of singularities to occur.Plugging in the co-linear terms to $\boldsymbol{y} = \alpha\boldsymbol{x}$, for $\alpha < 0$ we get

$$\log \frac{|\boldsymbol{y}^*|r^* + x_j y_j^* - |\boldsymbol{y}^*|^2}{|\boldsymbol{x}||\boldsymbol{y}^*| + x_j y_j^*}\Big|_{x_j=\alpha y_j} \tag{4.10}$$

$$= \log \frac{|\boldsymbol{y}^*||(\alpha-1)y^*| + \alpha y_j^* y_j^* - |\boldsymbol{y}^*|^2}{|\alpha\boldsymbol{y}^*||\boldsymbol{y}^*| + \alpha y_j^* y_j^*} \tag{4.11}$$

$$= \log \frac{|\boldsymbol{y}^*||(\alpha-1)\boldsymbol{y}^*| + \alpha\boldsymbol{y}^* \cdot \boldsymbol{y}^* - |\boldsymbol{y}^*|^2}{|\alpha\boldsymbol{y}^*||\boldsymbol{y}^*| + \alpha\boldsymbol{y}^* \cdot \boldsymbol{y}^*} \tag{4.12}$$

$$= \log \frac{|\alpha-1| + (\alpha-1)}{|\alpha| + \alpha} \tag{4.13}$$

which is independent of the variable of differentiation y_3^*.

4.4 Reducing volume integral to 2D integral

The analytical integration in θ of each of the I_j terms is a long calculation (see Appendix A). Most of the integrals can be computed by finding the integral of the form

$$F = \int_0^{2\pi} \frac{d\theta}{\sqrt{B + C\cos\theta}} = \frac{4K}{\sqrt{B+C}} \tag{4.14}$$

and its derivatives with respect to B and C. In this expression, B and C are functions of the observation point $\boldsymbol{x}$ and the remaining integration variables ρ and ζ.

CHAPTER 5

EXPERIMENTAL WORK

In this chapter we discuss experiments involving solid and porous spheres settling in the low Reynolds regime, as well as particle clouds falling in homogenous and stratified finite Reynolds fluids. The stratified density profiles include linear and sharp stratification. The set up consists of miscible, often stably sharp stratifications of two layers and settling spheres that are more dense than the ambient fluid densities.

We enhance the effects of stratification and entrainment by studying a sphere in different environments. We compare these results to that of a porous sphere, to make it a more similar configuration to that of the ocean. Finally, experiments of multiple particles are performed to show the extent of entrainment in the delayed settling of aggregates. All experiments shed light to many issues of bio- and geo-physical applications. Similar to the higher Reynolds number experiments in [1], we find that a single sphere exhibits a prolonged settling rate and slows down substantially through a sharply stratified fluid. In addition, we find that there are significant differences between the Stokes dominated regime and the Entrainment regime, where diffusion of salt can play a role.

5.1 Methods

Clear, cylindrical plexiglass tanks with diameters ranging from 6.2 cm to 18.9 cm and heights of either 31.8 cm or 52.1 cm are stratified with a top layer of lower density corn syrup poured over a higher density (achieved by adding a chosen type of salt, often NaCl or KI) corn syrup solution. We start with enough fluid for both layers that is mixed together until homogenous. Then, we separate each layer and adjust with salt and water to attain desired density and viscosity. The layers are left to degas overnight if needed. Evaporation is

minimized by covering free surfaces with airtight membranes and thermal convection can be controlled by a thermal bath.

The settling of multiple particles thorough density transitions is performed in a fish tank with dimensions of 61.6 cm by 61.6 cm by 34.3 cm. Room temperature, DI water is mixed with room-temperature salt water to create two layers of the desired densities. To obtain a sharp stratification and minimize the density transition layer, we use a diffuser, composed of a sponge surrounded by buoyant styrofoam. The bottom poured into the tank, we place the diffuser carefully, and pour the top layer though the diffuser at a low speed. Conductivity meters can measure the tank at discrete heights that helps us obtain a density profile. The beads are then placed in a funnel closed with a plunger until release time to minimize initial speed.

In the case of sharp stratifications, the method vary depending on the viscosity fluid. For saltwater solutions, we use the same method as [1] - the bottom more dense layer is poured first, then a floating diffuser is placed on the bottom layer and slowly pour the top layer less dense fluid. The density profile is measured with an Orion conductivity probe. In the case of corn syrup, we slowly pour the top layer sliding it down the inclined wall of the tank.

To obtain a linear stratification, we use the two-bucket method from [16], where the top layer fluid from the first bucket is pumped into the mixing bucket at one half the flow rate as the mixed fluid is pumped into the tank where the experiment is to be conducted.

5.1.1 Data Analysis

The data obtained from experiments is recorded by taking a video or pictures at uniform intervals in time of the experiments. We then use DataTank - a graphical, object oriented programming work environment- to track the position of the particle and its velocity is obtained from numerical differentiation. We process the images by subtracting off the background and then taking a contour of the sphere to find its center of mass in time.

5.1.2 Matching viscosities

In order to compare our experiments with the theory, the viscosities between top and layer fluid μ_t and μ_b must be matched within 2% of each other. The matching process requires an extra level of careful measurements and adding water of to adjust while maintaining the desired density. We have collected data to understand the effect of water concentration in the dynamic viscosity of corn syrup. Starting with pure corn syrup at 20^oC of viscosity $\approx 40P$, we add different amounts of water obtaining the plot in Figure 5.1.

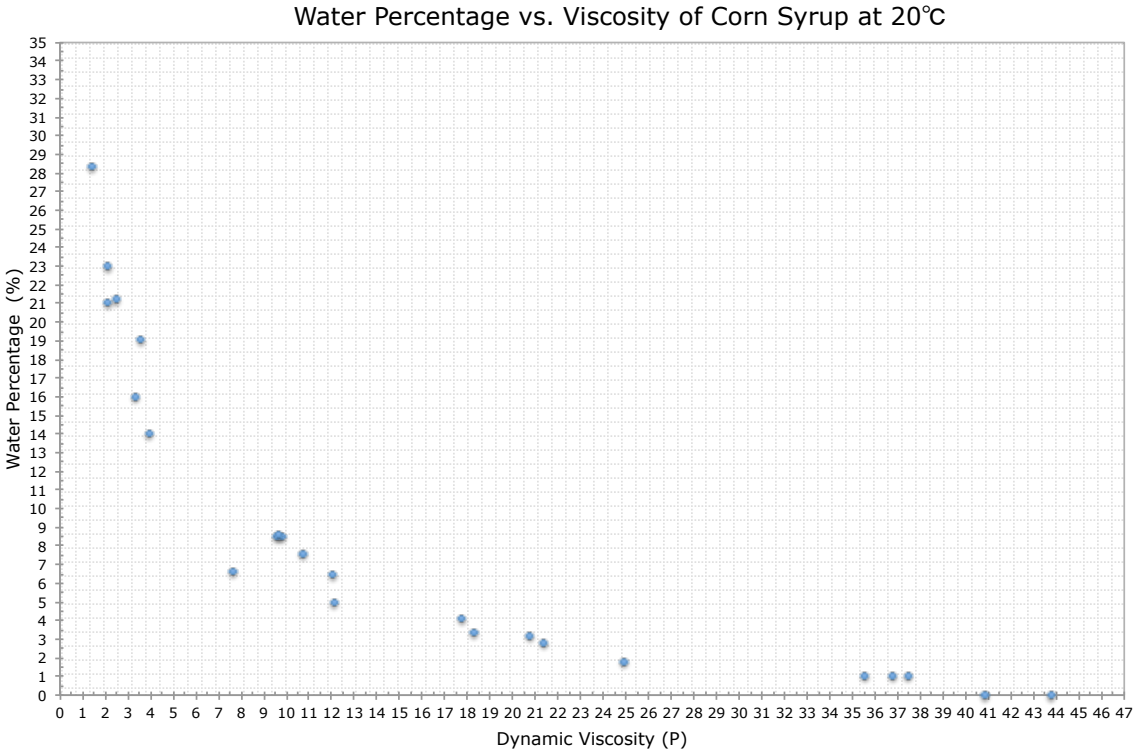


Figure 5.1: Plot of water percentage vs. dynamic viscosity (P). This data aids the viscosity matching process between the fluid layers of corn syrup.

If the layers are different viscosities, we obtain different outcomes depending on which viscosity is higher. We have performed experiments when the top layer viscosity μ_t is higher than the bottom layer viscosity μ_b and vicerversa. So far, the observations show that if $\mu_t > \mu_b$, then the sphere takes longer to achieve terminal velocity in the bottom layer. Further experiments should be conducted to compare these with the matched viscosities cases.

5.2 Solid Sphere Experiments

In this section, we focus on the single solid sphere experiments settling in two-layer stratifications. The solutions for a sphere settling in homogenous environment are well known. Experimentally, we have homogenous runs in fluid of density equal to that of the bottom layer density ρ_b as a bench mark to compare with the behavior of the sphere when settling in the bottom layer of a two-layer stratified fluid. IN each layer, the sphere should approach the theoretical terminal velocity.

Theoretically the terminal velocity in the bottom layer in the absence of entrainment is given by :

$$V_b = \frac{2K}{9} \frac{gA^2(\rho_s - \rho_b)}{\mu_b} \tag{5.1}$$

where $K = 2.10444(A/R_0) - 2.08877(A/R_0)^3$ is the enhanced wall drag coefficient [18].

In two layers, the sphere slows down beyond its terminal velocity to the extra buoyancy force provided by the entrainment fluid. In Figures (5.2) and (5.3), we show two Stokes dominated experiments for different radius spheres. This effect is magnified when the bottom layer density approaches the density of the sphere, the sphere velocity approaches zero, and the residence time increases. Because we are interested in testing the accuracy of the model as we approach the entrainment regimes, we perform a series of experiments increasing the bottom layer density. Examples of these experiments are portrayed in Figure (5.4). We note that if viscosities are not very closely matched, the results would be not agree perfectly with the theory, as seen in Figure (5.5)

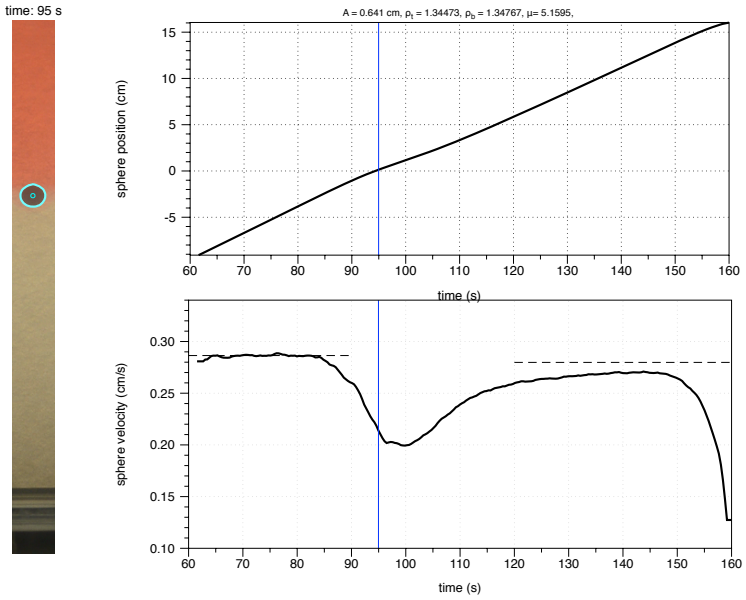


Figure 5.2: Velocity profiles of sphere of radius $A = 0.641cm$ and density $\rho_s = 1.36712 g/cm^3$ settling in a two-layer stratification with densities $\rho_t = 1.34473\, g/cm^3$and $\rho_b = 1.34767\, g/cm^3$ and average viscosity $\mu = 5.1595$ Poise. The black line represents the experiment tracking while, the dashed lines correspond to the theoretical terminal velocities and the vertical blue line shows the time at which the sphere is shown.

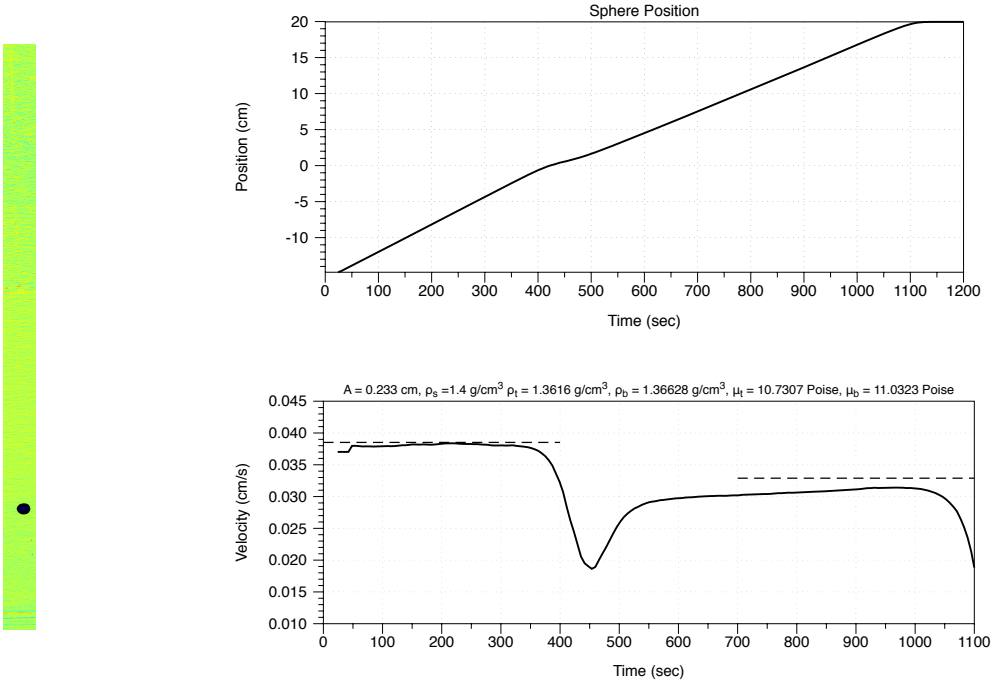


Figure 5.3: Velocity profiles of sphere of radius $A = 0.233\,cm$ and density $\rho_s = 1.4\,g/cm^3$ settling in a two-layer stratification with densities $\rho_t = 1.3616\,g/cm^3$and $\rho_b = 1.36628\,g/cm^3$ and viscosities $\mu_t = 10.7307$ Poise and $\mu_b = 11.0323$ Poise. The black line represents the experiment tracking while, the dashed lines correspond to the theoretical terminal velocities.

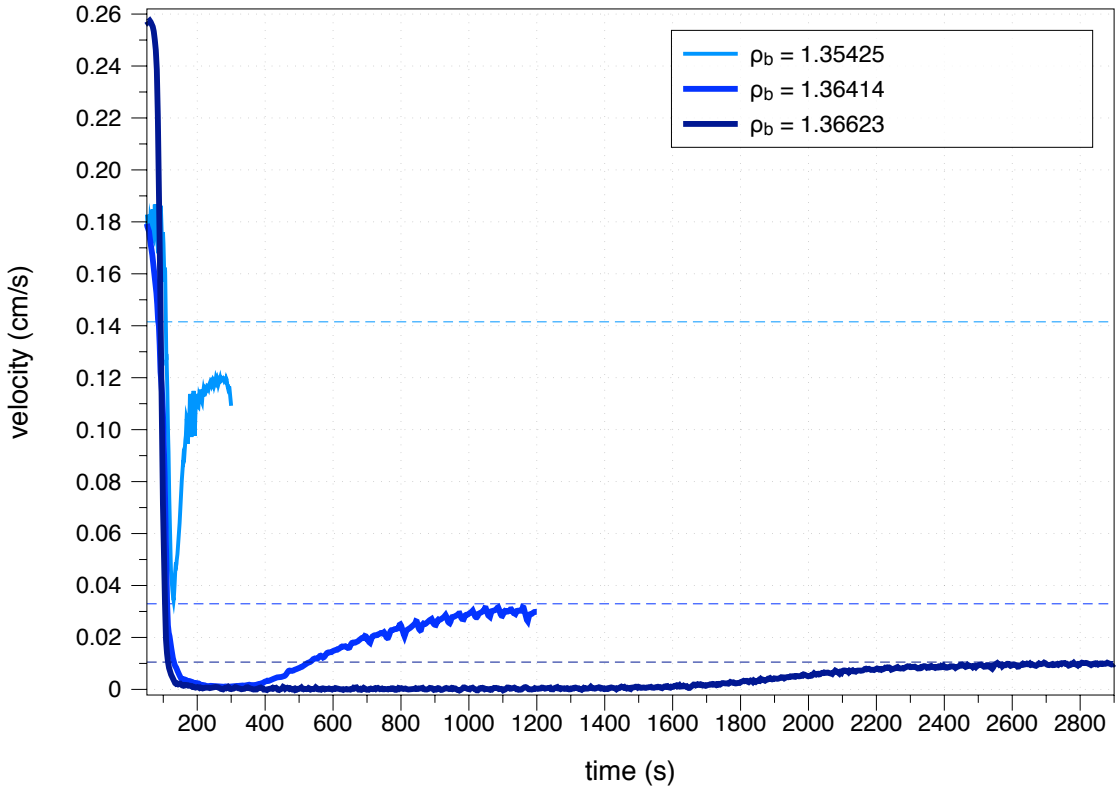


Figure 5.4: Series of experiments from Stokes dominated regimes to Entrainment dominated regimes. Experimentally measured velocity profiles of sphere of density $\rho_s = 1.36712\, g/cm^3$ and radius $A = 0.641\, cm$ settling in a two-layer stratification with similar top layer density and sequentially increasing bottom layer densities $\rho_b = 1.35425, 1.36414, 1.36623\, g/cm^3$

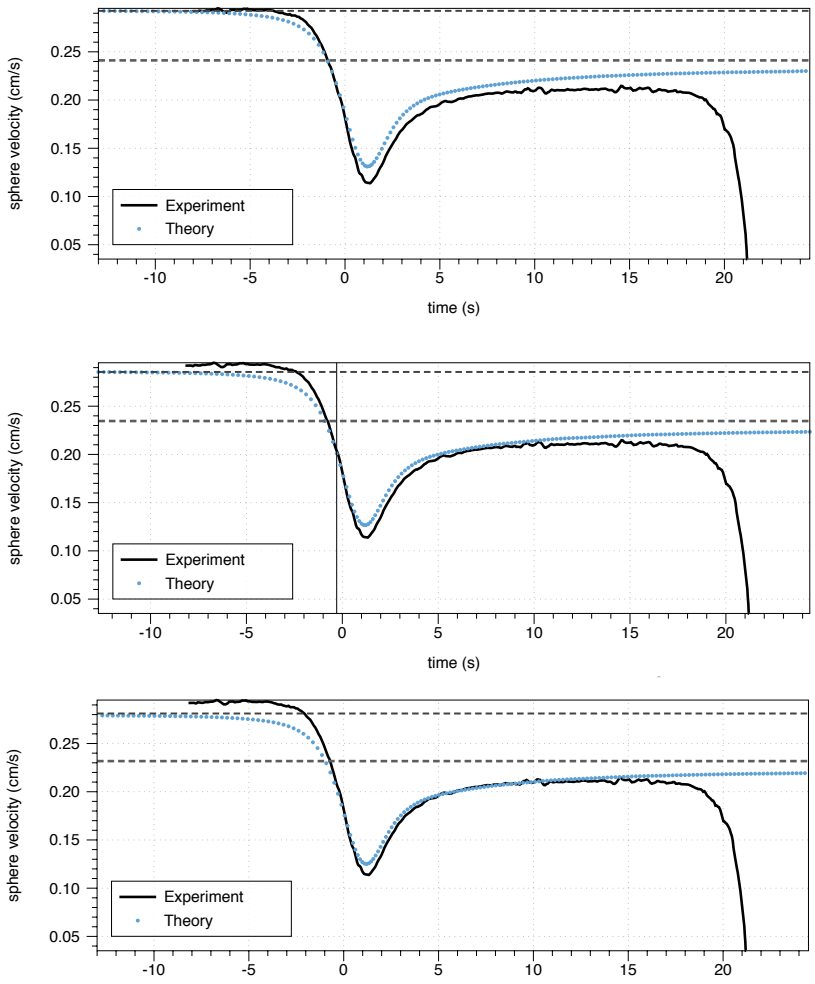


Figure 5.5: Velocity profiles of sphere of radius $A = 0.641\,cm$ and density $\rho_s = 1.36712 g/cm^3$ settling in a two-layer stratification with densities $\rho_t = 1.34647 g/cm^3$and $\rho_b = 1.35000 g/cm^3$ and viscosities $\mu_t = 5.06980$ Poise and $\mu_b = 5.27610$ Poise. The black line represents the experiment tracking while the blue line indicates the full theory using the top layer viscosity in the first upper panel, an average viscosity in the second panel, and bottom layer viscosity in the third lower panel. The dashed lines correspond to their respective theoretical terminal velocities using the code paremeters

5.3 Porous Sphere Experiments

A porous sphere assimilates better highly porous marine aggregates in the ocean. Understanding porous sphere settling and its comparison to solid sphere helps us enhance the effect of diffusion at the interface and understand its significance. Porous spheres that can be controlled the porosity and solid part density sphere is by drilling a solid sphere.

A sphere of mass $m_s = 1.4492 \pm 0.0010\ g$ with radius $A = 0.635 \pm 0.003\ cm$, and density $\rho_s = 1.3651\ g/cm^3$ was drilled to obtain a $m_d = 0.8986 \pm 0.0010\ g$ and porosity $P = 1 - m_d/m_s \approx 0.38$. The size of the drill bit was approximately $0.125\ cm$ and about 102 concentric holes were drilled, which made the surface area of the holes to be about 1/4 of the original sphere's surface area .This drilled sphere was dropped in a tank of $R_0 = 5.4\ cm$ of homogenous fluid consisting of a salt,water, and corn syrup mixture.

Figure 5.6: Picture of a plastic drilled sphere used in this sections experiments with radius $A = 0.635\,cm$, mass $m_s = 0.8986g$, and porosity $P = 0.38$.

In a sharply stratified environment, a solid sphere and a porous sphere were dropped into the same tank and their velocities compared. The velocity profiles are similar, showing the importance of entrainment in the long residence time of solid spheres, making it comparable with residence time of porous spheres that were initially dense than the bottom layer thus requiring diffusion of salt in order to fall through the bottom layer.

Homogenous experiments in salty corn syrup were also performed to understand the need

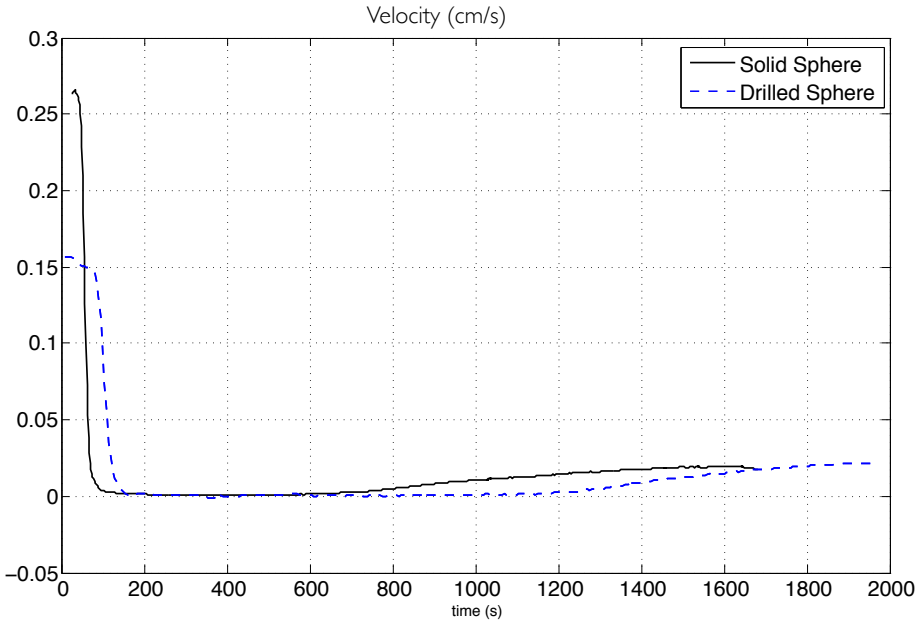


Figure 5.7: Velocity comparison between black solid line for the solid sphere and blue dashed line for the drilled sphere.

for diffusion. In the first homogenous experiment, the fluid inside the drilled sphere was of density ρ_f, same salt-water-corn syrup mixture the sphere was dropped in. Therefore, there should be no evidence of diffusion. The experiment was performed inside a temperature bath to avoid the effects of convection. The parameters of the experiment are listed below:

$$
\begin{aligned}
A &= 0.635\ cm \text{ - sphere radius} \\
\rho_s &= 1.3651\ g/cm^3 \text{ - density of sphere's solid part} \\
P &= 0.3799 \text{ - porosity of sphere} \\
R_0 &= 5.4 \text{ - radius of the tank} \\
\rho_f &= 1.36455\ g/cm^3 \text{ - density of the homogenous fluid in tank} \\
\rho_{sf} &= (1-P)\rho_s + P\rho_f = 1.36456\ g/cm^3 \text{ - density of the sphere} \\
\mu_f &= 14.763783 \text{ Poise - dynamic viscosity of the homogenous fluid}
\end{aligned}
$$

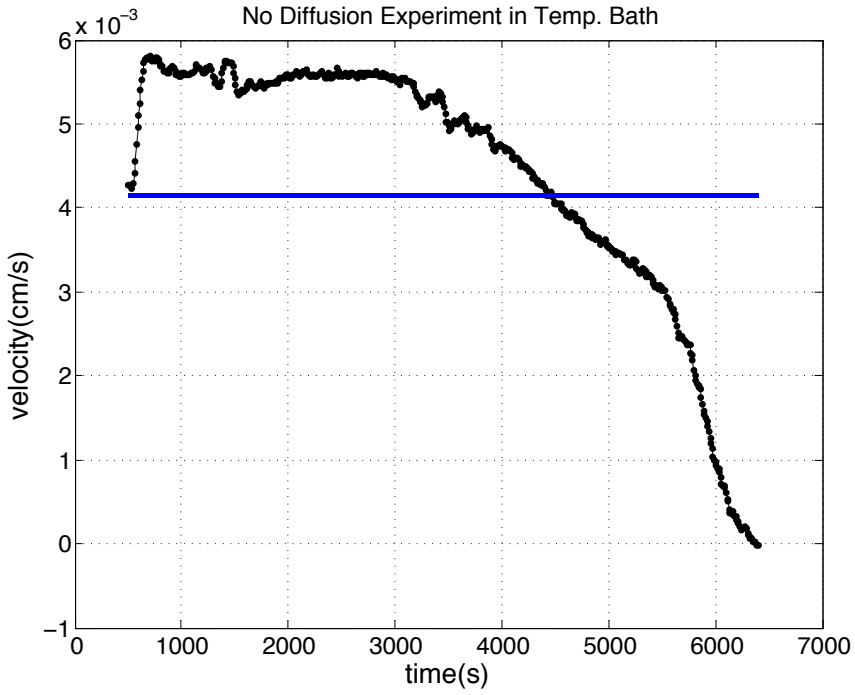


Figure 5.8: Velocity profile of drilled sphere with effective density $\rho_{sf} = 1.36455\ g/cm^3$, computed terminal velocity for ρ_{sf} is shown in blue

The sphere behaves like a solid sphere would, reaching and staying at terminal velocity

until it starts to slow down 10 cm from the bottom of the tank. For the values given above, the theoretical terminal velocity for a solid sphere in a cylinder is computed to be $vt = 0.00407 \pm 4 \times 10^{-}5\ cm/s$ shown as the blue line. However, the measured terminal velocity is higher, possibly a sign of different drag force on the surface of the drilled sphere. Another experiment in homogenous fluid in order to see evidence of diffusion. This time, we repeated the experiment dropping the drilled sphere in homogenous salty corn syrup fluid of density ρ_f but with the important difference that we pre-soaked the drilled sphere with fluid of 0% salinity, making the effective density of the drilled sphere $\rho_{si} < \rho_f$. We submerged the drilled sphere inside the fluid and hold it with a cap at the surface of tank. Since the effective density of the drilled sphere is initially less than the surrounding fluid, the porous sphere must stay at trapped at the top of fluid until diffusion of salt increases its effective density helping it settle. The experiment was performed inside a temperature bath to avoid the effects of convection. The parameters of the experiment are listed below:

$$
\begin{aligned}
A &= 0.635\ cm \text{ - sphere radius} \\
\rho_s &= 1.3651\ g/cm^3 \text{ - density of sphere's solid part} \\
P &= 0.37 \text{ - porosity of sphere} \\
R_0 &= 5.4 \text{ - radius of the tank} \\
\rho_i &= 1.35218\ g/cm^3 \text{ - density of the fluid initially inside the sphere} \\
\rho_f &= 1.36364\ g/cm^3 \text{ - density of the homogenous fluid in tank} \\
\rho_{si} &= (1-P)\rho_s + P\rho_i = 1.36032\ g/cm^3 \text{ - density of the sphere initially} \\
\rho_{sf} &= (1-P)\rho_s + P\rho_f = 1.36456\ g/cm^3 \text{ - density of the sphere after diffusion} \\
\mu_f &= 14.763783 \text{ Poise - dynamic viscosity of the homogenous fluid} \\
\mu_i &= 12.488899 \text{ Poise - dynamic viscosity of the fluid inside the sphere}
\end{aligned}
$$

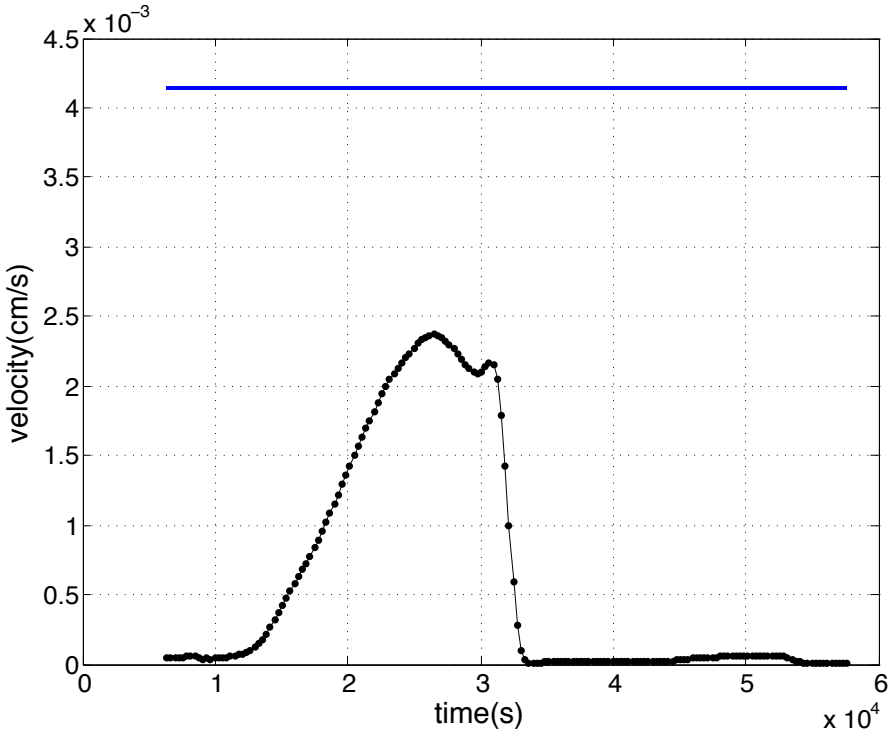


Figure 5.9: Velocity profile of drilled sphere initially with effective density $\rho_{si} = 1.36032$ g/cm^3 sinking in a homogenous surrounding fluid of density $\rho_f = 1.36364$. The drilled sphere begins to fall when its effective density becomes larger than the ambient fluid due to diffusion of salt. The theoretical terminal velocity for a solid sphere of density ρ_{sf} is shown in blue.

The velocity profile shown in below shows the velocity profile of the drilled sphere, which behaves as expected, having zero velocity for a long time until eventually increasing its velocity as it exchanges fluid with the surrounding salty corn syrup, thus becoming denser while settling. For the values given above, the theoretical terminal velocity for a solid sphere of density ρ_{sf} in a cylinder is shown as the blue line above. However, the highest measured velocity is a smaller, possible because the drilled sphere did not have time to exchange all the fluid inside before reaching the bottom of the tank. Approximately around $t = 2.6 \times 10^4$ seconds, the sphere is 10 cm from the bottom of the tank and its velocity begins to slow down as it approaches the rigid surface.

5.3.1 Further homogenous experiments

More experiments were conducted with the drilled sphere to understand the evidence of diffusion. When the fluid inside the drilled sphere was the same salt-water-corn syrup mixture the sphere was dropped in, there should be no evidence of diffusion. All experiments were performed inside a temperature bath to avoid the effects of convection.

Experiment where diffusion does not play a role

ρ_f = 1.35883 g/cm^3 - density of the homogenous fluid

ρ_{sf} = $(1-P)\rho_s + P\rho_f = 1.36278$ g/cm^3 - density of the sphere and salt-water -corn syrup

μ = 11.03830 Poise - dynamic viscosity

Figure 5.10) shows the velocity profile of the sphere, which behaves like a solid sphere would, reaching and staying at terminal velocity. For the values given above, the terminal velocity should be $vt = 0.02378$ cm/s shown as the red line. However, as seen before, the measured terminal velocity $vt = 0.02753$ cm/s is a bit higher, shown in blue.

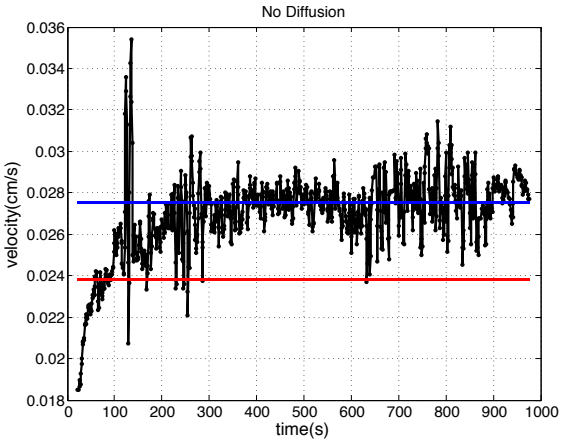


Figure 5.10: Velocity profile of drilled sphere with fluid $rho_f = 1.35883$ g/cm^3 inside. Red line indicates theoretical terminal velocity for a solid sphere of equivalent density ρ_{sf}

Note that the experiment was stopped before the sphere reached the bottom of the tank, therefore the velocity profile above does not show the slow-down that happens when the

sphere feels the effects of the bottom of the tank.

Experiment with evidence of diffusion In order to see evidence of diffusion, we pre-soaked the drilled sphere with fluid of 0% salinity.

$$\begin{aligned}
\rho_i &= 1.35050\ g/cm^3 \text{ - density of the fluid inside the sphere} \\
\rho_f &= 1.35883\ g/cm^3 \text{ - density of the homogenous fluid} \\
\rho_{si} &= (1-P)\rho_s + P\rho_i = 1.35970\ g/cm^3 \text{ - density of the sphere initially} \\
\rho_{sf} &= (1-P)\rho_s + P\rho_f = 1.36278\ g/cm^3 \text{ - density of the sphere after diffusion} \\
\mu_f &= 11.03830 \text{ Poise - dynamic viscosity of the homogenous fluid} \\
\mu_i &= 10.49592 \text{ Poise - dynamic viscosity of the fluid inside the sphere}
\end{aligned}$$

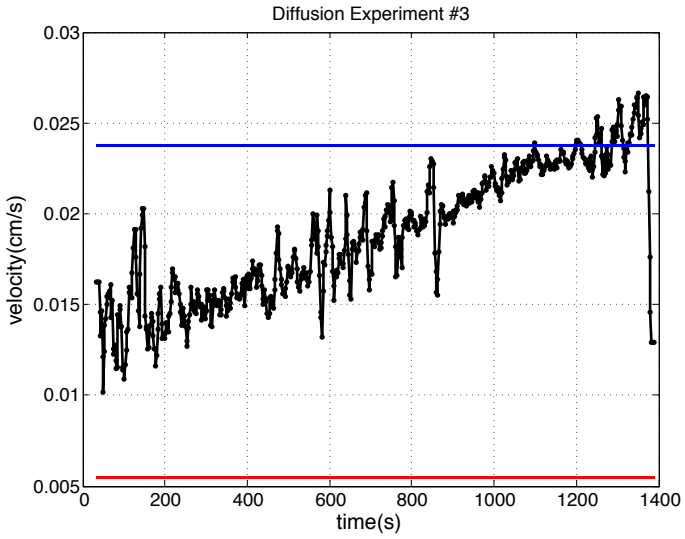


Figure 5.11: Velocity profile of drilled sphere initially with fluid $\rho_i = 1.35050\ g/cm^3$ inside, computed terminal velocities for ρ_{si} and ρ_{sf} are shown in red and blue respectively

Experiment where diffusion does not play a role

ρ_f = 1.35976 g/cm^3 - density of the homogenous fluid

ρ_{sf} = $(1-P)\rho_s + P\rho_f = 1.3631$ g/cm^3 - density of the sphere and salt-water corn syrup

μ = 9.37222 Poise - dynamic viscosity

Figure 5.12 shows the velocity profile of the sphere, which behaves like a solid sphere would, reaching and staying at terminal velocity until it feels the effects of the bottom of the tank.

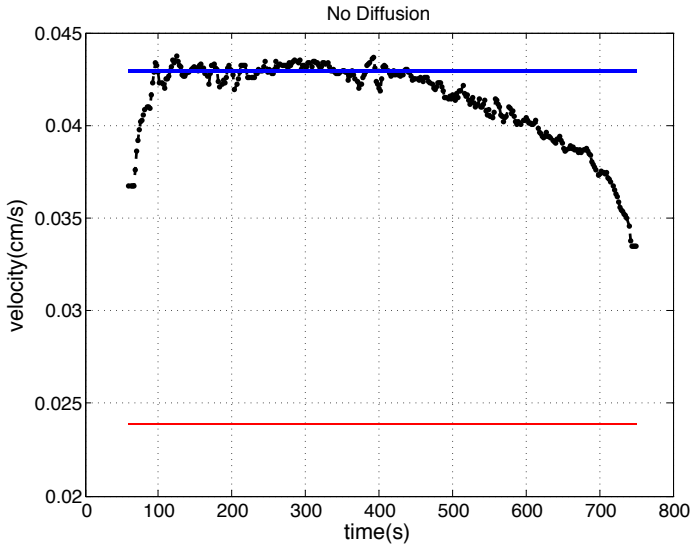


Figure 5.12: Velocity profile of drilled sphere with fluid $rho_f = 1.35976$ g/cm^3 inside

Note: For the values given above, the terminal velocity should be $vt = 0.0238$ cm/s shown as the red line in the figure above. However, the actual terminal velocity $vt = 0.0429$ cm/s is higher, as shown in blue, which outputs a $\rho_s = 1.36581$ g/cm^3, higher than the density of the solid part of the sphere, which is only possible if the fluid inside the sphere was heavier due to salt residues.

5.4 Particle Cloud Experiments

Experimental observations of cloud of particles settling in sharp and linear stratifications give important insight into the settling and formation of thin marine aggregate layers in the ocean. For this section, we performed a range of experiments, varying the systems parameters to develop an understanding of how these properties enhance the delayed settling of the particles, making it possible for the layers to form.

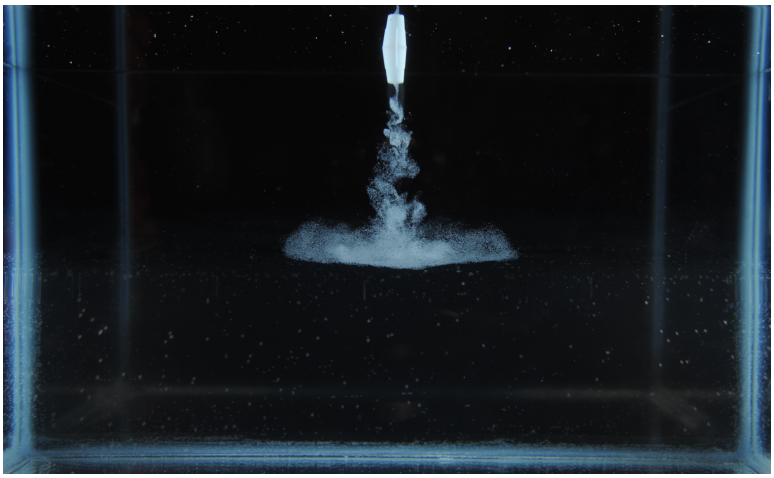


Figure 5.13: Particle cloud settling in two-layer stratified fluid, the cloud sharpens as it reaches the interface. The stratification consists of a top layer with density $\rho_t = 0.9987 g/cm^3$ and a bottom layer of density $\rho_b = 1.045 g/cm^3$. The particles are polystyrene beads of density $\rho_s = 1.05 g/cm^3$ and average radius $A = 0.02 cm$.

Homogeneous, sharply stratified, and linearly stratified tanks of salt water are prepared in a glass fish tank of dimension 61cm x 31cm x 63cm (W X D X H). The sharply stratified tank is prepared as described in the Methods Section, with chosen bottom density and fresh water on top (density 0.997 g/cm^3). The linear stratified tank is prepared using the two-bucket methods, and results in a density transition from chosen density at the tank bottom to fresh water at the top. Particle laden fluid is prepared in a separate container in which 10 grams of 1.05 g/cm^3 polystyrene spheres of chosen size ranging from 0.02 to 0.17 cm radius is mixed with 15 ml salty water. This results in an effective density for the particle laden cloud, exceeding the top density and the bottom ambient fluid density. A funnel is inserted

into the top of the tank, and plugged with a cylinder. A volume of 10ml of particle laden fluid is poured into the funnel and then the plug is removed. We note that even though the effective density of the mixture exceeds the bottom layer, the particulate pancakes on the transition layer, and descends at a very slow rate. The cases with the particulate cloud mixed with upper layer fluid are more resembling of falling marine snow . We also note that for the homogenous, sharply stratified, or linearly stratified cases, the descent rate of the cloud in the upper layer fluid greatly exceeds the settling speed of a single particle in the upper fluid by orders of magnitude. However, as the cloud is trapped upon the layer, the rate of descent of the cloud is dramatically reduced to a speed comparable with the single particle settling speed in the bottom layer. Ultimately, the trapped cloud in the sharply stratified case begins to rain out in more finger like structures which evolve into descending clouds again falling much faster than the single particle settling speed in the lower fluid.

In order to measure the changes of the particle cloud and track its position, we compute the horizontally average concentration and the location of its centroid. With these quantities, we can form definitions of residence time at the transition layer and compare velocity profiles. The horizontally averaged concentration of the particle is given by

$$C_R(Z,t) = \int_{-R_0}^{R_0} C(R,Z,t)dR. \tag{5.2}$$

We can are interested if two values that will help us define the residence time. One is the peak value of the concentration $C_P(t) = \max(C_R(Z,t))$ and the other is the average concentration around the interface

$$C_I(t) = \int_{I-Z_\epsilon}^{I+Z\epsilon} C_R(Z,t)dZ \tag{5.3}$$

These quantities can the provide the residence time T by selecting the time it takes for these measures of concentration to stay above a prescribed amount, which we set to 35% of

the maximum concentration value.

$$C_I(t_1) = C_I(t_2) = 0.35\,(\max[C_I(t)]) \tag{5.4}$$

$$T = t_2 - t_1 \tag{5.5}$$

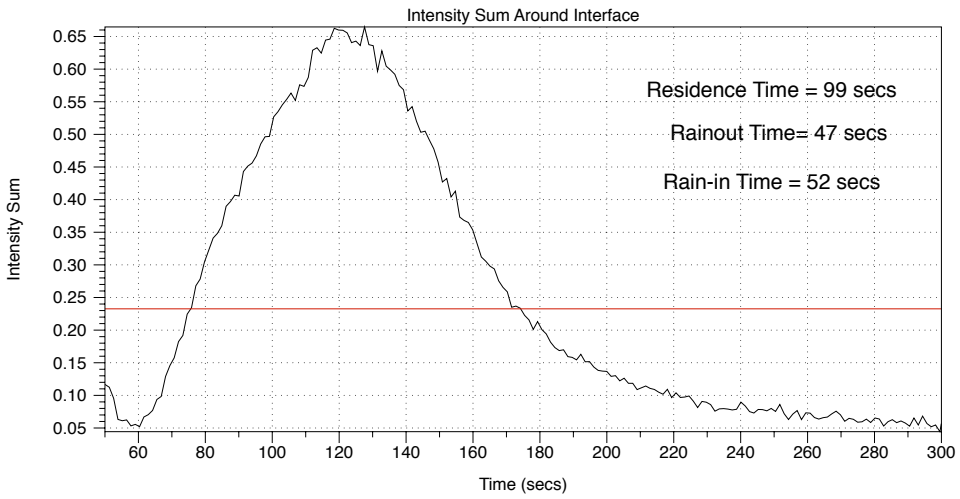
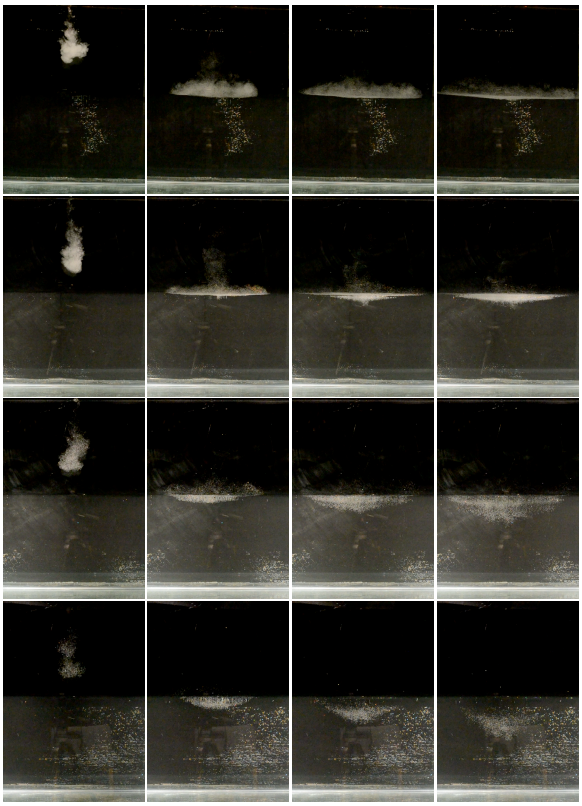


Figure 5.14: Concentration average around interface $C_I(t)$ versus time. The horizontal red line denotes the 35% of the maximum value providing a residence time $T = 99s$.

If we keep the effective density of the cloud constant, but change the size of the particles in it, we observe that the smaller beads means longer delayed settling while larger beads exhibit a shorter residence time.

When looking at porous aggregates, the behavior of the delayed settling is affected by the porosity. Less porous aggregates behave like its solid counterparts in that the clouds of larger particles exhibit shorter residence time. On the other hand, the more porous aggregates switch this relation.

As the bottom layer density increases, the particle cloud increases its residence time at the interface.



(a) (b) (c) (d)

Figure 5.15: Time-lapsed snapshots at (a) 2 s (b) 10 s (c) 15 s (d) 20 s of the particle clouds settling in a two-layer of sharply stratified fluid, varying the size of the particles. The size of the beads increase from the first row to the last.

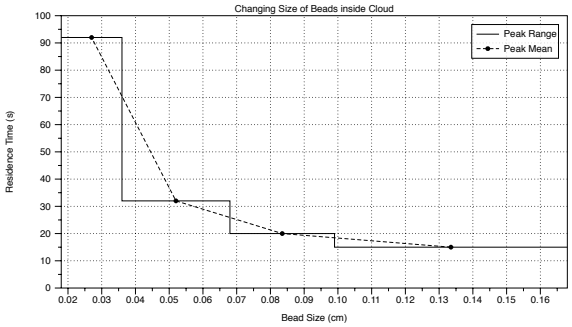


Figure 5.16: Residence time versus particle radius, the range of particle sizes inside a cloud are denoted by the solid line, while the dots represent the mean size inside the cloud.

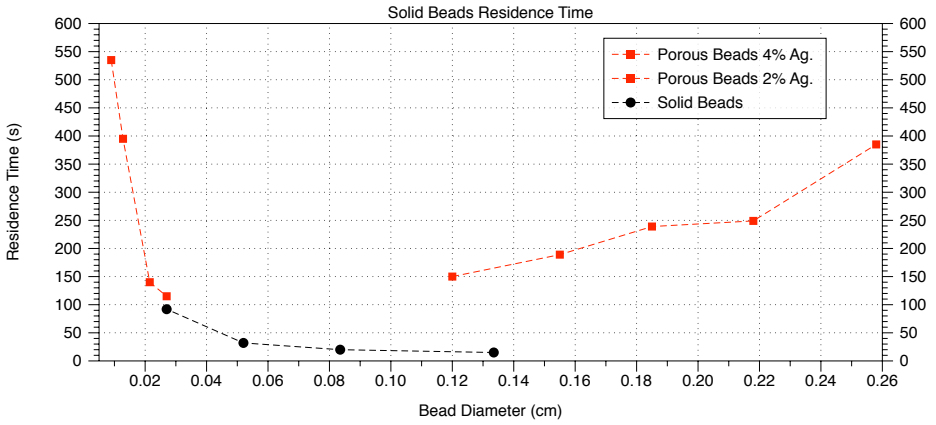


Figure 5.17: Residence time versus particle radius, read markings indicate porous particles while black markings indicate solid beads.

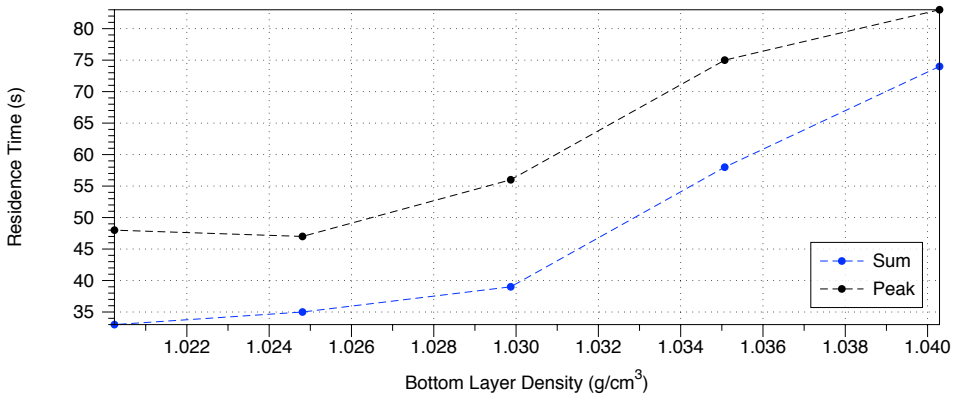


Figure 5.18: Residence time as measured by C_I (blue) and by C_P (black) as we vary the bottom layer density

CHAPTER 6

NUMERICAL IMPLEMENTATION

The resulting model provided in Equation (2.60) is a integro-differential equation in the fluid density ρ modeling a sphere settling in stratified fluid. In this chapter, we will explain the numerical tools used to implement the full theory. The simulation advects a material surface by the fluid flow $\boldsymbol{u}_s(\boldsymbol{x}, t; V) + \boldsymbol{w}(\boldsymbol{x}, t; \rho)$ in the sphere frame of reference. The initial data consist of the initial position of the sphere $Y_3(0)$, and an initial fluid density distribution $\rho_0(x_3)$, inside a cylinder of radius R_0 and outside a sphere with radius A centered along the cylinder axis (where the sphere center is constrained at all times for axially symmetric solutions). Because of the axisymmetry of the problem, we only need to consider half of the vertical cross section of the cylinder. The fluid flow and sphere velocity require ρ to define the domain of integration to calculate $\boldsymbol{w}$ and the force on the sphere due to the perturbation velocity. The majority of the computational time goes into calculating the volume integral that provides the perturbation velocity $\boldsymbol{w}$ at every time step with a changing domain of integration provided by ρ. We have made simplifications and approximations to $\boldsymbol{w}$ that speed up this expensive computation (See Theory Approximations). Many aspects of the numerics that will be discussed in this chapter are not trivial in order to obtain fast and accurate results to the problem at hand.

Under some conditions, the modeling requires implementing the complete solution of the problem. Without approximations, the volume integral becomes more expensive to compute numerically. To speed up the simulation, we integrate the kernel once in θ and reduce the 3D integral to a 2D integral. This is a cumbersome calculation resulting in long pages of formulations of elliptic integrals of the first, second, and third kind, among other expressions. In addition, the integrable singularities and removable singularities in the kernel need to be

treated properly for accurate and fast numerical integration.

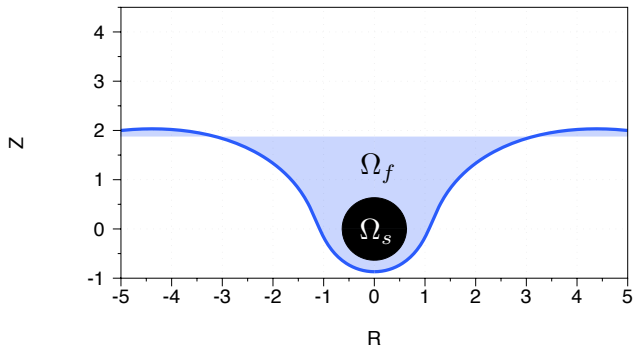


Figure 6.1: Diagram of domain of integration as determined by the entrainment and reflux regions. The fluid domain Ω_f is shaded in blue and the sphere domain is denoted Ω_s

6.1 Stokes second reflection

In the asymptotic expansion for $\boldsymbol{u} = \boldsymbol{u}^{(0)} + \boldsymbol{u}^{(1)} + \boldsymbol{u}^{(2)} + \boldsymbol{u}^{(3)} \ldots$, as formulated in [18] in cylindrical coordinate (R, θ, Z) with $r = \sqrt{R^2 + Z^2}$, the vertical component of the second reflection is given by:

$$u_Z^{(2)} = \frac{1}{2\pi} \int_0^\infty \hat{u}_Z(R, \lambda) \cos(\lambda Z) d\lambda, \tag{6.1}$$

Numerically, we compute these integrals using fast Fourier transform (FFT) *a priori* for a grid of Z values.

$$u_Z^{(2)} = \frac{1}{4\pi} \int_{-\infty}^\infty \hat{u}_Z(R, |\lambda|) \, \mathrm{e}^{-i|\lambda|Z} d\lambda, \tag{6.2}$$

$$= \frac{1}{2} \int_{-\infty}^\infty \hat{u}_Z(R, 2\pi|\hat{\lambda}|) \, \mathrm{e}^{-2\pi i|\hat{\lambda}|Z} d\hat{\lambda}, \tag{6.3}$$

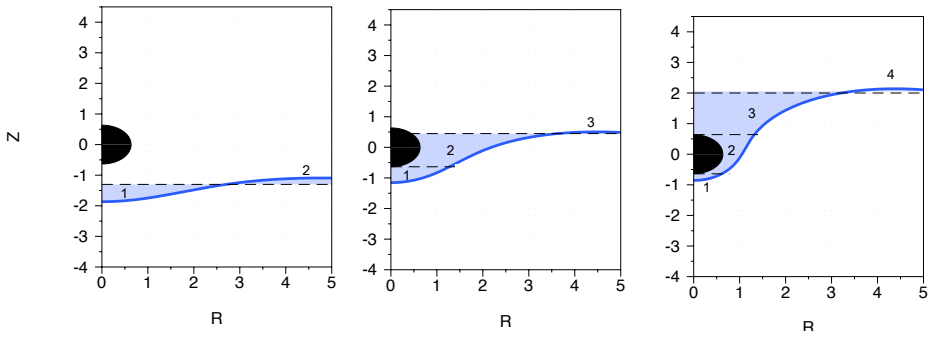
which can be expressed as a discrete Fourier transform (DFT)

$$u_{Z,k}^{(2)} = \frac{1}{2} \sum_{n=0}^{N-1} \hat{u}_{Z,n}(R, |\lambda_n|) \, \mathrm{e}^{-2\pi i|\lambda_n|Z_k}, \tag{6.4}$$

where $\lambda_n = \frac{2L}{N-1}n$

6.2 Computation of $\boldsymbol{w}$

Calculating the perturbation velocity $\boldsymbol{w}$ is the most expensive part of the code. The resulting volume integration uses the changing entrainment and reflux domain as its domain integration. We use cubic interpolation to use a uniform grid in the numerical integration. In Figure 6.2, we show the three cases the code encounters and how it subdividing the domain of integration in order to define each region as a graph for interpolation.



(a) (b) (c)

Figure 6.2: Disection of interface when (a) interface is below the sphere showing two regions of integration, (b) interface around the sphere, three regions of integration, and (c) interface above the sphere, four regions of integration.

The entrainment dominated regime requires implementing the complete solution of the problem near the sphere. Combining the more expensive full solution and the far field solution speeds up the calculation and provides accurate results for when the interface is near the sphere where the far field does not work. Numerically, our challenge is to compute the full solution of the perturbation velocity accurately and fast.

Without approximations, the volume integral becomes expensive to compute numerically. To speed up the simulation, we integrate the kernel once in θ and reduce the 3D integral to a 2D integral. This is a cumbersome calculation resulting in long expressions including elliptic integrals of the first, second, and third kind (see Appendix 4.2). In addition, the integrable

and removable singularities in the kernel need to be treated properly for accurate and fast numerical integration.

The removable singularities present in the Oseen tensor needs special care when performing an integration that requires high working precision. The removable singularities are further explained in section (4.2), where we show that they cancel out and pose no problem for the integration. These points of numerical complication occur along a line

$$\boldsymbol{y} = \alpha \boldsymbol{x}, \tag{6.5}$$

with $\alpha < 0$. When the line of removable singularities appear in the domain of integration,

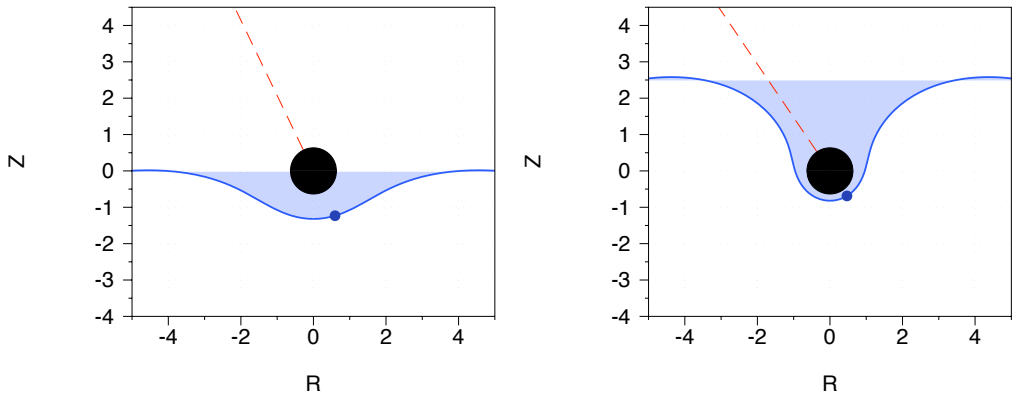


Figure 6.3: Domain of integration and removable singularities. The blue dot represents the observation interfacial point and the red dashed line indicates where its corresponding line of removable singularities occur. These points only need to be dealt with when they get inside the blue shaded region, the fluid domain of integration.

correct cancelation of large numbers must occur to obtain a continuous kernel.

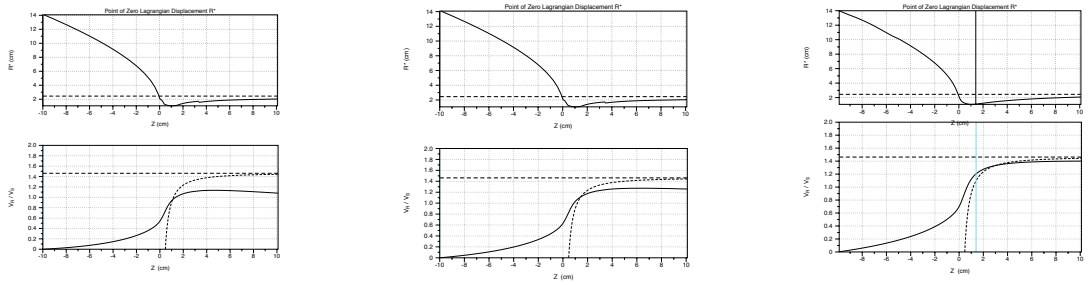
6.3 Interface Tracking Techniques and Validation

In order to determine the domain of integration for the volume integral that outputs the perturbation velocity $\boldsymbol{w}$, we must interpolate in the entrainment and reflux regions provided by the interfacial points.

6.3.1 Potential Flow

In order to validate our interface tracking, interpolation routine, and volume integrals, we perform a simple advection using potential flow and keeping everything else the same as in the full code. Using the results obtained in [10], we can compare the reflux volume V_R normalized by the volume of the sphere V_S, with its asymptotic expression.

$$V_R = 2\pi \int_{R^*}^{\infty} (f(R) - Z^*)\, RdR \tag{6.6}$$

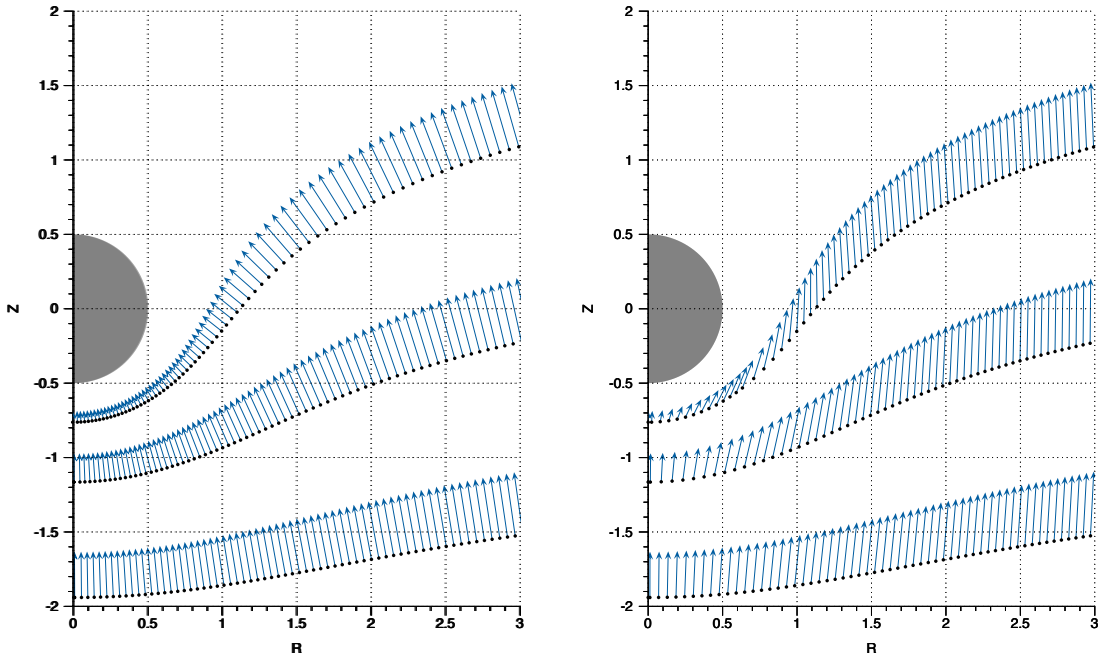


(a)

(b) (c)

Figure 6.4: Validation of interpolation, interface tracking and volume integrals by showing the point of zero Lagrangian displacement (top panels) and the normalized reflux volume (bottom panels) with potential flow for radius of sphere $A = 1\,cm$, interface initialized $y_0 = -10\,cm$ away from the sphere, and horizontal axial cut off at (a) $R_0 = 40\,cm$ (b) $R_0 = 80\,cm$ (c) $R_0 = 320\,cm$. The dashed lines represent the asymptotic value of each quantity.

6.3.2 Normal Advection



(a)

(b)

Figure 6.5: Plots of interface advected with uniform velocity free space Stokes advection using (a) normal velocity and (b) full velocity

CHAPTER 7

THEORY APPROXIMATIONS

Approximations to the model were made for different applications. In this chapter, we study the approximations and discuss their usefulness, region of validity, and comparison to the full theory.

7.1 Far Field Approximation

When the observation point $\boldsymbol{x}$ is far from the sphere, we can simplify the perturbation velocity $\boldsymbol{w}(\boldsymbol{x})$, by approximating the fundamental solution

$$\boldsymbol{W}(\boldsymbol{x},\boldsymbol{y}) \sim \boldsymbol{W}_{FF}(\boldsymbol{x},\boldsymbol{y}) \text{ as } \frac{A}{|\boldsymbol{x}|} \to 0. \tag{7.1}$$

We can use this approximation far from the sphere

$$\boldsymbol{w}_{FF}(R,Z,t) = \int_0^{2\pi} d\theta \int_0^{\infty} \int_{-\infty}^{\infty} \epsilon G(\rho,\zeta,t) \boldsymbol{W}_{FF}(R,Z,\rho,\zeta,\theta) \rho \, d\rho \, d\zeta \tag{7.2}$$

The kernel $\boldsymbol{W}_{FF}(\boldsymbol{x},\rho,\zeta,\theta)$ can be integrated once in θ analytically, simplifying the expression and reducing the dimension of the integral, thus speeding up the numerics.

$$\boldsymbol{w}_{FF}(R,Z,t) = \int_0^{\infty} \int_{-\infty}^{\infty} \epsilon G(\rho,\zeta,t) \hat{\boldsymbol{W}}(R,Z,\rho,\zeta) \rho \, d\rho \, d\zeta \tag{7.3}$$

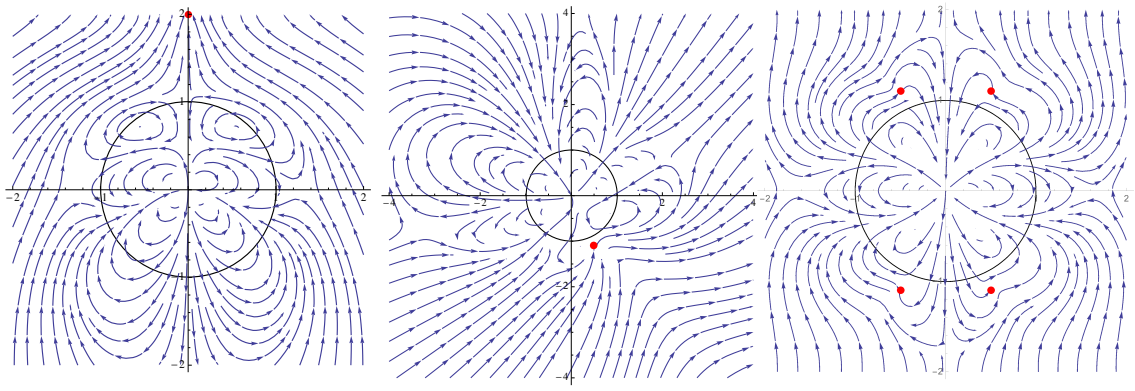
In cylindrical coordinates, integrated once in θ, the components $\left(\hat{W}_R, \hat{W}_Z\right)$ of the far field approximation of $O(A^7)$ are

$$
\begin{aligned}
\hat{W}_R = & \frac{3A^5R\left(\rho^4\left(4Z^2(8\zeta+3Z)-R^2(8\zeta+23Z)\right)\right)}{128\mu\left(\zeta^2+\rho^2\right)^{7/2}\left(R^2+Z^2\right)^{7/2}} \\
& + \frac{3A^5R\left(8\zeta\rho^2\left(-3R^4+R^2\left(\zeta^2+3Z^2+37\zeta Z\right)+Z^2\left(-4\zeta^2+6Z^2-33\zeta Z\right)\right)\right)}{128\mu\left(\zeta^2+\rho^2\right)^{7/2}\left(R^2+Z^2\right)^{7/2}} \\
& + \frac{3A^5R\left(8\zeta^3\left(2R^4+R^2\left(2\zeta^2-2Z^2-17\zeta Z\right)-2Z^2(Z-4\zeta)(2Z-\zeta)\right)\right)}{128\mu\left(\zeta^2+\rho^2\right)^{7/2}\left(R^2+Z^2\right)^{7/2}} \\
& + \frac{A^3R\left(R^2\left(2\zeta^2-\rho^2\right)(5\zeta+Z)+Z\left(3\left(2\zeta^4+3\zeta^2\rho^2+\rho^4\right)+Z^2\left(2\zeta^2-\rho^2\right)\right)\right)}{16\mu\left(\zeta^2+\rho^2\right)^{5/2}\left(R^2+Z^2\right)^{5/2}} \\
& + \frac{10A^3RZ^2\zeta\left(\rho^2-2\zeta^2\right)}{16\mu\left(\zeta^2+\rho^2\right)^{5/2}\left(R^2+Z^2\right)^{5/2}} + \frac{3ARZ\left(2\zeta^2+\rho^2\right)}{16\mu\left(\zeta^2+\rho^2\right)^{3/2}\left(R^2+Z^2\right)^{3/2}} \\
& - \frac{(Z-\zeta)\left(2\left(\zeta^2+\rho^2+R^2+2\rho R+Z^2-2\zeta Z\right)K\left(-\frac{4R\rho}{R^2-2\rho R+Z^2+\zeta^2+\rho^2-2Z\zeta}\right)\right)}{R\sqrt{\zeta^2+\rho^2+R^2-2\rho R+Z^2-2\zeta Z}\left(\zeta^2+\rho^2+R^2+2\rho R+Z^2-2\zeta Z\right)} \\
& - \frac{(Z-\zeta)\left(2\left(-\zeta^2-\rho^2+R^2-Z^2+2\zeta Z\right)E\left(-\frac{4R\rho}{R^2-2\rho R+Z^2+\zeta^2+\rho^2-2Z\zeta}\right)\right)}{R\sqrt{\zeta^2+\rho^2+R^2-2\rho R+Z^2-2\zeta Z}\left(\zeta^2+\rho^2+R^2+2\rho R+Z^2-2\zeta Z\right)}
\end{aligned}
\tag{7.4}
$$

$$
\begin{aligned}
\hat{W}_Z = & \frac{\pi}{16R}\left(\frac{3A^5R^2\left(8R^4\left(2\zeta^3-3\zeta\rho^2\right)+R^2\left(8\zeta\left(2\zeta^4+\zeta^2\rho^2-\rho^4\right)-8Z^2\left(2\zeta^3-3\zeta\rho^2\right)\right.\right.}{\left(\zeta^2+\rho^2\right)^{7/2}\left(R^2+Z^2\right)^{7/2}}\right. \\
& + \frac{Z\left(-136\zeta^4+296\zeta^2\rho^2-23\rho^4\right)\right)-4Z^2\left(8\zeta\left(2\zeta^4+\zeta^2\rho^2-\rho^4\right)+4Z^2\left(2\zeta^3-3\zeta\rho^2\right)\right.}{\left(\zeta^2+\rho^2\right)^{7/2}\left(R^2+Z^2\right)^{7/2}} \\
& \left.+\frac{\left.-3Z\left(12\zeta^4-22\zeta^2\rho^2+\rho^4\right)\right)\right)}{\left(\zeta^2+\rho^2\right)^{7/2}\left(R^2+Z^2\right)^{7/2}} + \frac{24AR^2Z\left(2\zeta^2+\rho^2\right)}{\left(\zeta^2+\rho^2\right)^{3/2}\left(R^2+Z^2\right)^{3/2}} \\
& - \frac{8A^3R^2\left(R^2\left(2\zeta^2-\rho^2\right)(5\zeta+Z)+Z\left(3\left(2\zeta^4+3\zeta^2\rho^2+\rho^4\right)+Z^2\left(2\zeta^2-\rho^2\right)\right)\right)}{\left(\zeta^2+\rho^2\right)^{5/2}\left(R^2+Z^2\right)^{5/2}} \\
& - \frac{80A^3R^2Z^2\zeta\left(\rho^2-2\zeta^2\right)}{\left(\zeta^2+\rho^2\right)^{5/2}\left(R^2+Z^2\right)^{5/2}} \\
& - \frac{16(Z-\zeta)\left(2\left(\zeta^2+\rho^2+R^2+2\rho R+Z^2-2\zeta Z\right)K\left(-\frac{4R\rho}{R^2-2\rho R+Z^2+\zeta^2+\rho^2-2Z\zeta}\right)\right.}{\pi\sqrt{\zeta^2+\rho^2+R^2-2\rho R+Z^2-2\zeta Z}\left(\zeta^2+\rho^2+R^2+2\rho R+Z^2-2\zeta Z\right)} \\
& \left.+\frac{\left.2\left(-\zeta^2-\rho^2+R^2-Z^2+2\zeta Z\right)E\left(-\frac{4R\rho}{R^2-2\rho R+Z^2+\zeta^2+\rho^2-2Z\zeta}\right)\right)}{\pi\sqrt{\zeta^2+\rho^2+R^2-2\rho R+Z^2-2\zeta Z}\left(\zeta^2+\rho^2+R^2+2\rho R+Z^2-2\zeta Z\right)}\right)
\end{aligned}
\tag{7.5}
$$

This approximation does not satisfy boundary conditions, and it is only only accurate far

from the sphere.



(a) (b) (c)

Figure 7.1: Streamlines of the Oseen Green's function far field approximation along the $x_2 = 0$ plane around a sphere of radius $A = 1$ with one Stokeslet located at (a) $\boldsymbol{y} = (0, 0, 2)$, (b) $\boldsymbol{y} = (0.5, 0, -1.1)$, (c) ring of points forces above and below

7.1.1 Region of validity

As $\rho_s - \rho_b \to \rho_b - \rho_t$, the density anomaly flow $\boldsymbol{w}$ is no longer subdominant with respect to the stokes flow $\boldsymbol{u}_s$ and thus the approximation fails due the error near the sphere affects the shape of the interface sending interface into the sphere.

It is easy to show that the far field approximation does not satisfy boundary conditions.

We can justify using the far field approximation $\boldsymbol{w}_{FF}(R, Z)$ as long as the interfacial points stay within the region of validity, and the error produced by the approximation is subdominant in comparison with the magnitude of the Stokes flow.

7.1.2 Comparison with Full Theory

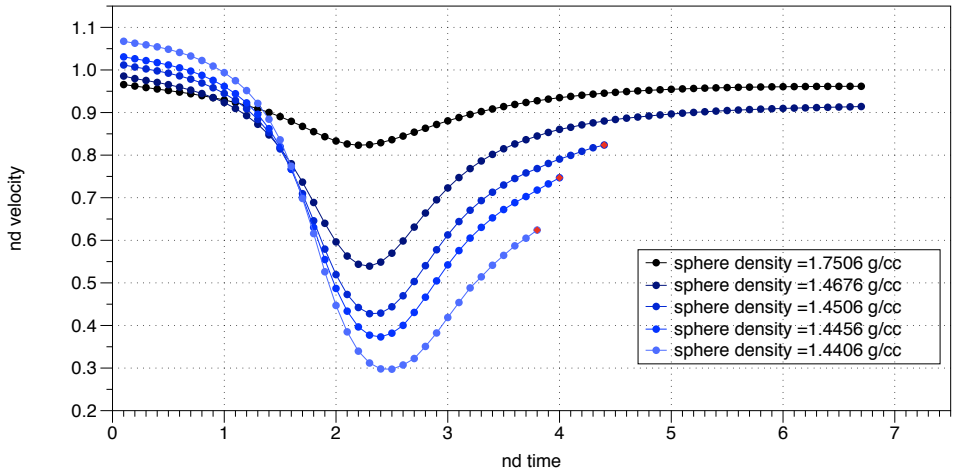


Figure 7.2: Non-dimensional numerically obtained velocity profiles using the far field approximation as the sphere density $\rho_s \to \rho_b$ with $\rho_t = 1.42760 g/cm^3$, $\rho_b = 1.43060 g/cm^3$, and $\mu = 17$ Pois. The red point represents the point that the interface goes into the sphere, deforming interface in a non-physical way and stopping the simulation

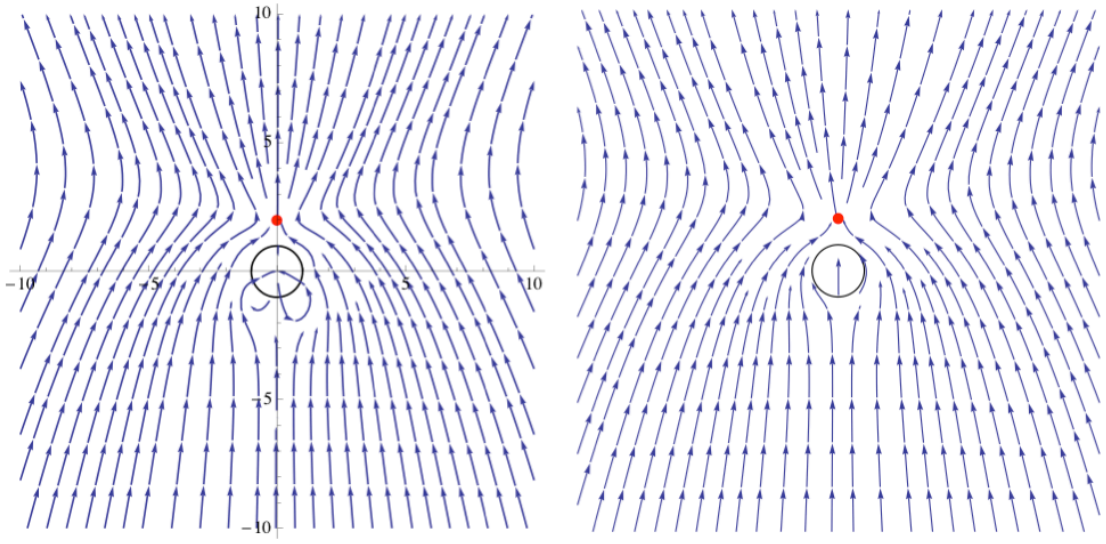
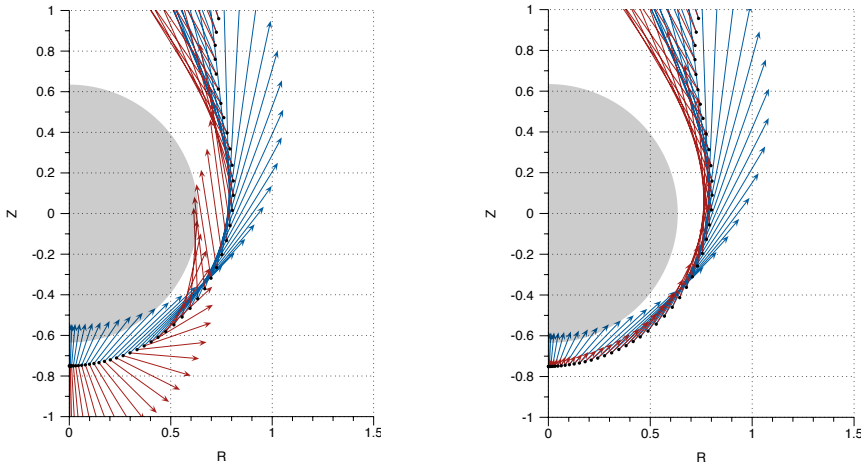


Figure 7.3: Comparison of the streamlines along the $x_2 = 0$ plane around a sphere of radius $A = 1$ with one Stokeslet located at $\boldsymbol{y} = (0, 0, 2)$ of the Oseen Green's function far field approximation $\boldsymbol{W}_{FF}$ (left) and full kernel $\boldsymbol{W}$(right)



(a) (b)

Figure 7.4: Simulation of a sphere of density $\rho_s = 1.4506\, g/cm^3$ settling in two layer fluid. Comparison of flow at the interfacial points. The blue arrows represent the stokes flow $\boldsymbol{u}_s$ while the red arrows indicate the perturbation flow using (a) the far field approximation $\boldsymbol{w}_{FF}$ and (b) the full solution $\boldsymbol{w}$.

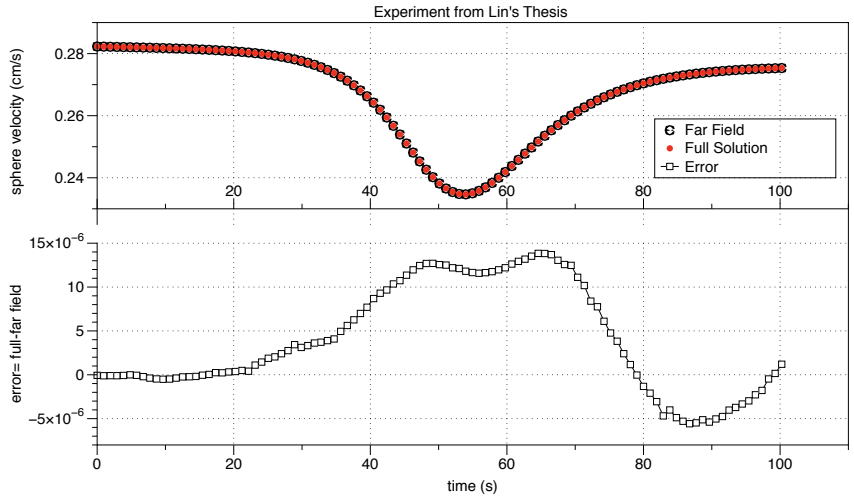


Figure 7.5: Far Field vs Full Theory in the Stokes dominated regime

7.2 Near Field Approximation

When the observation point $\boldsymbol{x}$ is very near the sphere, we can approximate the perturbation velocity, $\boldsymbol{w}(\boldsymbol{x})$ by linearizing the kernel. Expanding $\boldsymbol{w}(\boldsymbol{x})$ around $\boldsymbol{x_0}$, such that $|\boldsymbol{x_0}| = A$, we obtain,

$$\begin{aligned} \boldsymbol{w}(\boldsymbol{x}) &\sim \boldsymbol{w}(\boldsymbol{x_0}) + (\boldsymbol{x} - \boldsymbol{x_0}) \cdot \nabla_x \boldsymbol{w}\Big|_{\boldsymbol{x_0}} + \dots && (7.6) \\ \text{Since } \boldsymbol{w}(\boldsymbol{x_0}) &= 0 && (7.7) \\ \boldsymbol{w}_{NF} &= (\boldsymbol{x} - \boldsymbol{x_0}) \cdot \nabla \boldsymbol{w}\Big|_{\boldsymbol{x}=\boldsymbol{x_0}} && (7.8) \end{aligned}$$

$$\begin{aligned} \boldsymbol{w}(\boldsymbol{x}, t) &= \int_{\Omega_f(t)} \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{y}) d_3 y && (7.9) \\ \boldsymbol{w}_{NF}(\boldsymbol{x}, t) &= (\boldsymbol{x} - \boldsymbol{x_0}) \cdot \nabla_x \left(\int_{\Omega_f} \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{y}) d_3 y \right) \Big|_{\boldsymbol{x}=\boldsymbol{x_0}} && (7.10) \\ \boldsymbol{w}_{NF}(\boldsymbol{x}, t) &= \int_{\Omega_f} (\boldsymbol{x} - \boldsymbol{x_0}) \cdot \nabla_x \left(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{y}) d_3 y \right) \Big|_{\boldsymbol{x}=\boldsymbol{x_0}} && (7.11) \end{aligned}$$

$$\boldsymbol{W}_{NF}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{x_0}) \cdot \left(\nabla_x \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{y}) \right) \Big|_{\boldsymbol{x}=\boldsymbol{x_0}} \text{ as } \boldsymbol{x} \to \boldsymbol{x_0} \quad (7.12)$$

7.3 Leaky Sphere Approximation

In this section, we simplify the full model by relaxing the boundary conditions. The equations of motion are in the reference of frame of the sphere. Let $\boldsymbol{v}(\boldsymbol{z},t)$ be the solution to our problem . Let $\boldsymbol{V}(t)$ and $\boldsymbol{Y}(t)$ be the velocity and center of the sphere respectively. Then, the equations of motion in a moving frame of reference is:

$$\begin{aligned} \mu\nabla^2\boldsymbol{u} &= \nabla p - \rho\hat{g}, && (7.13)\\ \nabla\cdot\boldsymbol{u} &= 0, && (7.14)\\ \oint_S \boldsymbol{u}dS &= \boldsymbol{V}(t) \text{ as } A\to 0 && (7.15) \end{aligned}$$

$$\frac{\partial\rho}{\partial t} + \boldsymbol{u}\cdot\nabla\rho = 0. \tag{7.16}$$

Where $\boldsymbol{u} = \boldsymbol{v} - \boldsymbol{V}(t)$ and $\boldsymbol{x} = \boldsymbol{z} - \boldsymbol{Y}(t)$ and the equation of motion for the sphere can be written as

$$m_s\frac{dV(t)}{dt} = m_s\hat{g} + \oint_S \sigma\cdot\hat{n}dS, \tag{7.17}$$

where $\hat{g}$ is the gravity acceleration vector oriented along the unit vector $\hat{z}\equiv(0,0,-1)$, m_s is the mass of the sphere, σ is the stress tensor, S is the surface of the sphere, and $\hat{n}$ is the outward normal unit vector to this surface. Taking advantage of the linearity of the Stokes equations, like in equation (2.15), we split the fluid flow into two parts,

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{u}_s(\boldsymbol{x},t) + \boldsymbol{w}(\boldsymbol{x},t)\,. \tag{7.18}$$

$$\begin{aligned} \mu\nabla^2\boldsymbol{u}_s &= \nabla p_s - \rho_0(x_3 + Y_3(t))\hat{g}, && (7.19)\\ \nabla\cdot\boldsymbol{u}_s &= 0, && (7.20)\\ \oint_S \boldsymbol{u}dS &= 0 \text{ as } A\to 0 && (7.21) \end{aligned}$$

For the stratification-induced flow, we define $G(\boldsymbol{x},t) = \rho(\boldsymbol{x},t) - \rho_0(x_3 + Y_3(t))$. We can write the governing equations in a moving frame of reference,

$$\begin{aligned} \mu\nabla^2\boldsymbol{w} &= \nabla p_w - G(\boldsymbol{x},t)\rho_{ref}\hat{g}, && (7.22)\\ \nabla\cdot\boldsymbol{w} &= 0, && (7.23)\\ \oint_S \boldsymbol{u}_s dS &= 0 \text{ as } A\to 0 && (7.24)\\ & && (7.25) \end{aligned}$$

The fundamental solution of the system above satisfies the following equations.

$$\begin{aligned} \mu\nabla^2\boldsymbol{W}(\boldsymbol{x},\boldsymbol{y}) &= \nabla P(\boldsymbol{x},\boldsymbol{y}) - \hat{g}\delta(\boldsymbol{x}-\boldsymbol{y}), && (7.26)\\ \nabla\cdot\boldsymbol{W} &= 0, && (7.27)\\ \oint_S \boldsymbol{W} dS &= 0 \text{ as } A\to 0 && (7.28)\\ & && (7.29) \end{aligned}$$

$$W(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{8\pi\mu}\left(\frac{\hat{g}}{|\boldsymbol{x}-\boldsymbol{y}|} + \frac{g(x_3-y_3)(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^3} + \left(\frac{f_s}{|\boldsymbol{x}|} + \frac{(f_s\cdot\boldsymbol{x})\boldsymbol{x}}{|\boldsymbol{x}|^3}\right)\right) \quad (7.30)$$

$$f_s = \frac{-3A}{4}\left(\frac{\hat{g}}{|\boldsymbol{y}|} + \frac{gy_3\boldsymbol{y}}{|\boldsymbol{y}|^3}\right) \quad (7.31)$$

$$P = \rho_{ref}\left(\frac{\hat{g}\cdot(\boldsymbol{x}-\boldsymbol{y})}{4\pi|\boldsymbol{x}-\boldsymbol{y}|^3} + \frac{f_s.\boldsymbol{x}}{4\pi|\boldsymbol{x}|^3}\right)$$

$$\begin{aligned} w &= \int_{\Omega_f} \epsilon G(\boldsymbol{y},t) W(\boldsymbol{x},\boldsymbol{y}) d\Omega_f && (7.32)\\ p_w &= \int_{\Omega_f} \epsilon G(\boldsymbol{y},t) P_W(\boldsymbol{x},\boldsymbol{y}) d\Omega_f && (7.33) \end{aligned}$$

To find the force exerted on the sphere due to the simplified perturbation velocity, we find F_w as

$$F_w = -\oint_S \sigma_{wij} n_j dS \quad = \quad -\oint_S \left(-p_w \delta_{ij} + \mu \left(\frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i} \right) \right) n_j dS$$

Where w and p_w are the perturbation velocity and pressure respectively.

Let $\boldsymbol{n}$ be the outward unit vector normal to the surface of the sphere. When then we apply the divergence theorem on the surface that consist of the surface of the sphere of radius A and the sphere of radius R. When we take the limit of $R \to \infty$, we obtain the force exerted on the sphere of radius A in terms of a volume integral over our fluid domain. Let Ω_f the fluid domain.

$$\begin{aligned} F_w &= -\oint_{|\boldsymbol{x}|=A} \sigma_w \cdot \boldsymbol{n} dS && (7.34) \\ &= -\left(\int_{\Omega_f} \nabla \cdot \sigma_w d\Omega_f + \lim_{R\to\infty} \left(\oint_{|\boldsymbol{x}|=R} \sigma_w \cdot \boldsymbol{n} dS \right) \right) && (7.35) \\ &= -\left(\int_{\Omega_f} \epsilon G(\boldsymbol{y}, t) \rho_{ref}\ g\ \Omega_f + \lim_{R\to\infty} \left(\oint_{|\boldsymbol{x}|=R} \sigma_w \cdot \boldsymbol{n} dS \right) \right) && (7.36) \\ & && (7.37) \end{aligned}$$

By finding the contribution of $\lim_{R\to\infty} \left(\oint_{|\boldsymbol{x}|=R} \sigma_w \cdot \boldsymbol{n} dS \right)$, we can find the force on the force due to the perturbation velocity.

In order to compute the behavior of the stress at infinity, we are interested in solving the following:

$$\lim_{R\to\infty} \oint_{|\boldsymbol{x}|=R} \sigma_w \cdot \boldsymbol{n} dS \tag{7.38}$$

where

$$\sigma_{w_{ij}} = \oint_S \left(-p_w \delta_{ij} + \mu \left(\frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i} \right) \right) n_j dS \tag{7.39}$$

In order to determine the behavior of the stress at infinity, we could simplify the equations:

$$\lim_{R\to\infty}\oint_{|\boldsymbol{x}|=R}\sigma_{wij}n_j dS \tag{7.40}$$

$$\begin{aligned}
&= \lim_{R\to\infty}\oint_{|\boldsymbol{x}|=R}\int_{\Omega_f}\epsilon G(\boldsymbol{y},t)\left(-P_W(\boldsymbol{x},\boldsymbol{y})\delta_{ij}+\mu\left(\frac{\partial W_i(\boldsymbol{x},\boldsymbol{y})}{\partial x_j}+\frac{\partial W_j(\boldsymbol{x},\boldsymbol{y})}{\partial x_i}\right)\right)d\Omega_f n_j dS\\
&= \int_{\Omega_f}\epsilon G(\boldsymbol{y},t)\lim_{R\to\infty}\oint_{|\boldsymbol{x}|=R}\left(-P_W(\boldsymbol{x},\boldsymbol{y})\delta_{ij}+\mu\left(\frac{\partial W_i(\boldsymbol{x},\boldsymbol{y})}{\partial x_j}+\frac{\partial W_j(\boldsymbol{x},\boldsymbol{y})}{\partial x_i}\right)\right)n_j dS\, d\Omega_f
\end{aligned}$$

Let $F_W = \lim_{R\to\infty}\oint_{|\boldsymbol{x}|=R}\left(-P(\boldsymbol{x},\boldsymbol{y})\delta_{ij}+\mu\left(\frac{\partial W_i(\boldsymbol{x},\boldsymbol{y})}{\partial x_j}+\frac{\partial W_j(\boldsymbol{x},\boldsymbol{y})}{\partial x_i}\right)\right)n_j dS.$
The other contribution to the force exerted on the sphere is therefore.

$$\lim_{R\to\infty}\oint_{|\boldsymbol{x}|=R}\sigma_{wij}n_j dS = \frac{3A(|\boldsymbol{y}|^2+2y_3^2)}{4|\boldsymbol{y}|^3}-1 \tag{7.41}$$

$$F_w = \int_{\Omega_f}\epsilon G(\boldsymbol{y},t)\rho_{ref}\; g\left(\frac{3A(|\boldsymbol{y}|^2+2y_3^2)}{4|\boldsymbol{y}|^3}\right)\Omega_f \tag{7.42}$$

Combining the solution for Stokes flow and the results from the section above, we have the equation for the vertical component of the velocity of the sphere and the advection fluid.

$$\begin{aligned}
\frac{dY_3}{dt}(t;\rho) = V(t;\rho) &= (6\pi A\mu)^{-1}(m_s g - g\int_{\Omega_s}\rho_0(x_3+Y_3(t;\rho))d\Omega_s\\
&+ \int_{\Omega_f}\epsilon G(\boldsymbol{y},t)\rho_{ref}g\frac{3A(|\boldsymbol{y}|^2+2y_3^2)}{4|\boldsymbol{y}|^3})d\Omega_f
\end{aligned} \tag{7.43}$$

$$\frac{\partial \rho}{\partial t}(\boldsymbol{x}, t) + (\boldsymbol{u}_s(\boldsymbol{x}, t; V) + \boldsymbol{w}(\boldsymbol{x}, t; \rho)) \cdot \nabla \rho(\boldsymbol{x}, t) = 0.$$

The simplified model is compared with the far field and full theory from [9] for different sphere radii. The simplified theory satisfies the boundary conditions when the radius of the

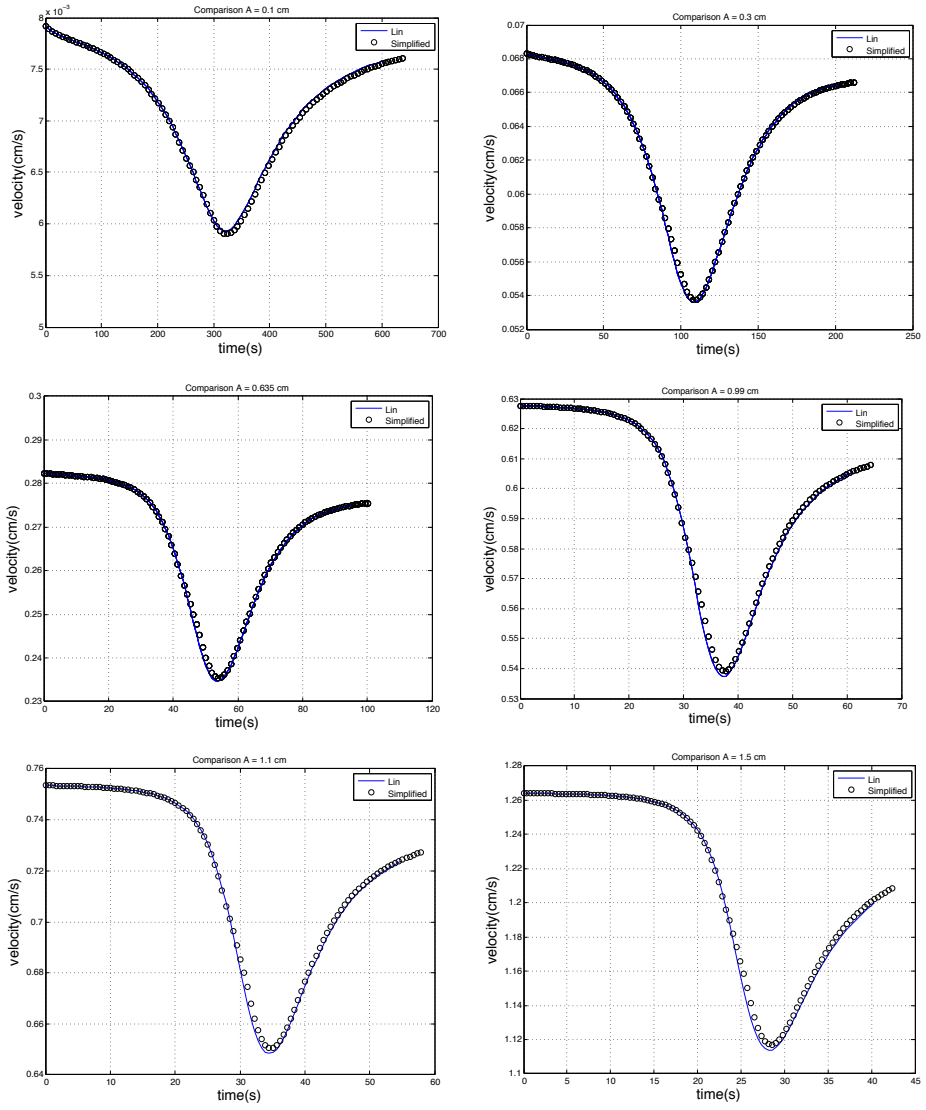


Figure 7.6: Comparison between Far Field and Simplified Theories of the sphere velocities profiles for increasing radius of the sphere: $0.1, 0.3, 0.635, 0.99, 1.1$, and $1.5cm$.

sphere $A \to 0$. The error between the velocity profiles obtained from the simplified and full theory is computed by calculating the L1 norm the L2 norm, the L infinity norm, and the normalized difference between the minimum velocity dip.

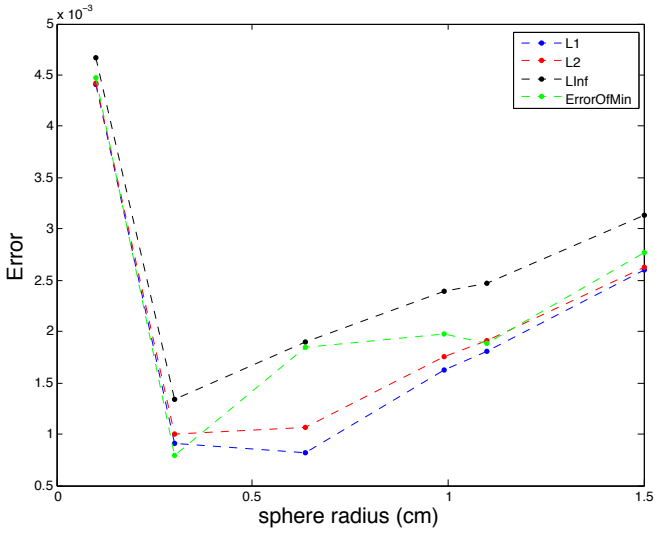


Figure 7.7: Computed Errors , showing increasing error for increasing sphere radius with the exception of the first point $A = 0.1$cm due to poor integration resolution.

7.4 Shell Model

The Shell Model assumes the entrainment shell about the sphere is spherical and omits the reflux backflow portion (see Figure 7.9). In the Entrainment regime cases, the interface wraps closely around the sphere and the reflux volume is very small, thus making this simplification close to the full theory interfacial shape. The shell thickness goes to zero with the speed distance from the sphere to the point at the south pole of the sphere. The final equations are given by,

$$\begin{aligned} V(t, y_b, y_i) &= (6\pi A\mu K)^{-1} \left(m_s g - g \int_{\Omega_s} \rho_0 (x_3 + Y_3(t)) d\Omega_s \right. \qquad (7.44) \\ &+ \left. g\,(\rho_b - \rho_t) \int_{\Omega_{fs}} \frac{u_{03}}{-V_0} d\Omega_{fs} \right) \\ \frac{dy_i}{dt} &= V(t) \qquad (7.45) \\ \frac{dy_b}{dt} &= u_s(t) + w(t) \qquad (7.46) \end{aligned}$$

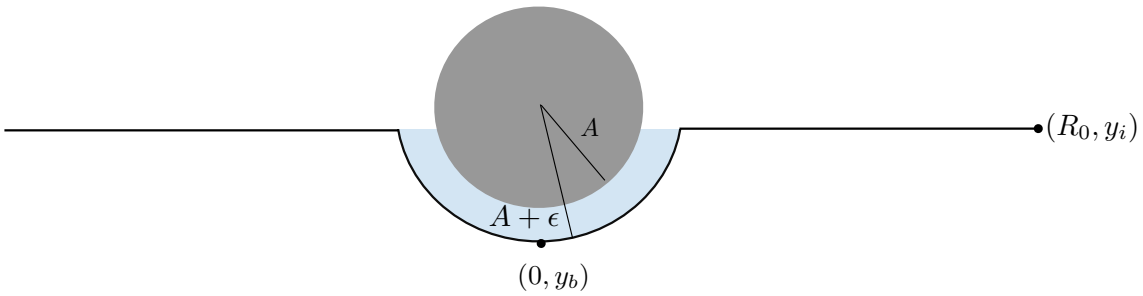


Figure 7.8: Shell model schematic showing the assumed spherical shell around the sphere.

When we run this model, the size of the shell is smaller than the full theory thus making the sphere fall faster.

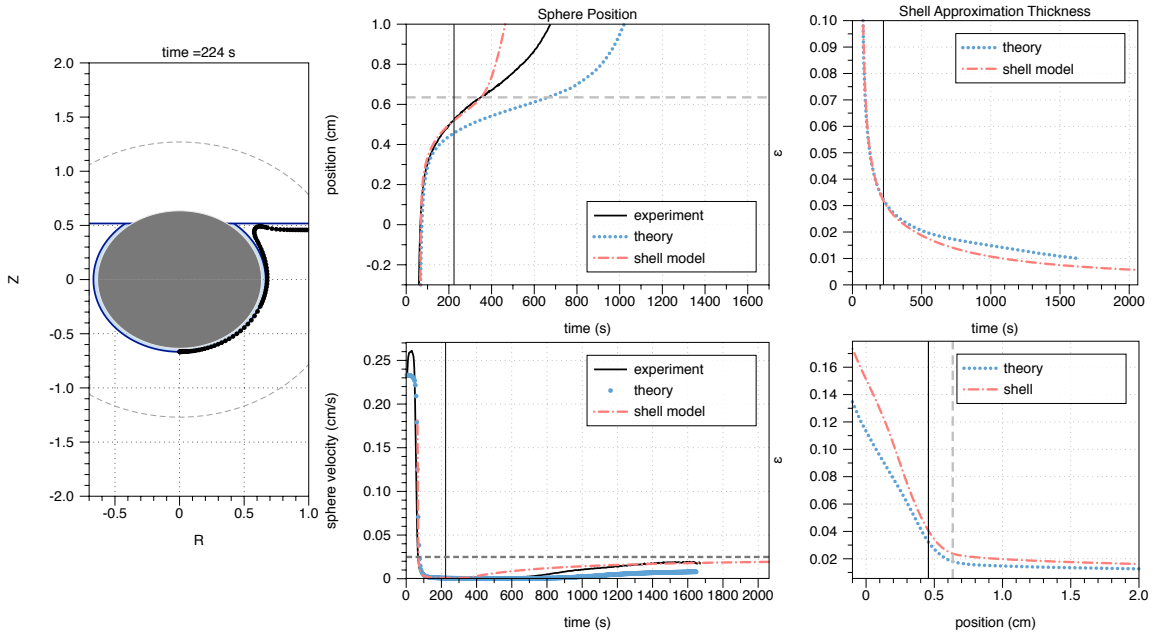


Figure 7.9: Comparison of shell model with experiment and full theory.

7.5 Perturbation Approach

Let $\boldsymbol{v}(\boldsymbol{z},t)$ be the solution to the problem of a single sphere falling in sharply stratified fluids . Let $\boldsymbol{V}(t)$ and $\boldsymbol{Y}(t)$ be the velocity and center of the sphere respectively. Then, the equations of motion in a moving frame of reference are:

$$m_s \frac{dV(t)}{dt} = m_s g\hat{z} + \oint_S \sigma \cdot \hat{n} dS \tag{7.47}$$

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2\boldsymbol{u} = \frac{1}{\rho}\nabla\cdot\sigma, \tag{7.48}$$

$$\frac{\partial\rho}{\partial t} + \boldsymbol{u}\cdot\nabla\rho = 0. \tag{7.49}$$

With boundary conditions:

$$\boldsymbol{u} = V(t)\hat{z} \text{ on } S\ , \tag{7.50}$$

$$\boldsymbol{u} \to 0 \text{ as } |\boldsymbol{x}| \to \infty \tag{7.51}$$

Where $\boldsymbol{u} = \boldsymbol{v} - \boldsymbol{V}(t)$ and $\boldsymbol{x} = \boldsymbol{z} - \boldsymbol{Y}(t)$, $\hat{g}$ is the gravity acceleration vector oriented along the unit vector $\hat{z} \equiv (0, 0, -1)$, m_s is the mass of the sphere, σ is the stress tensor, S is the surface of the sphere,μ is the viscosity of the fluid, and $\hat{n}$ is the outward normal unit vector to this surface.

In the Stokes regime, we can drop the inertial terms , and the equations become :

$$\begin{aligned} -m_s g\hat{z} - \int_S \sigma \cdot \hat{n} dS &= 0 && (7.52) \\ \mu\nabla^2 \boldsymbol{u}^2 - \nabla p = \nabla \cdot \sigma &= 0 && (7.53) \\ \frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho &= 0. && (7.54) \end{aligned}$$

with the same boundary conditions. Note that the force balance weight = stress would mean zero acceleration according to Newton but the velocity of the sphere is still allowed to depend on time, as long as law changes in time. In fact, we assume for $\epsilon = \rho_b - \rho_t << 1$:

$$\begin{aligned} \rho &= \rho_0(z,t) + \epsilon\rho_1(\boldsymbol{x},t) && (7.55) \\ \boldsymbol{u} &= \boldsymbol{u}_0 + \epsilon\boldsymbol{u}_1 + .. && (7.56) \\ p &= p_0 + \epsilon p_1 + ... && (7.57) \\ \sigma &= \sigma_0 + \epsilon\sigma_1 + ... && (7.58) \\ &= \pi p + \mu\nabla\boldsymbol{u_0} + \epsilon(-\pi p_1 + \mu\nabla\boldsymbol{u_1}) + ... && (7.59) \\ \boldsymbol{V} &= \boldsymbol{V}_0 + \epsilon\boldsymbol{V}_1 + .. && (7.60) \end{aligned}$$

so that we get :

$$\begin{aligned} \mu\nabla^2\boldsymbol{u}_0 = \nabla p_0 + g\rho_0(z),\ \boldsymbol{u}_0\Big|_S &= -V_0\hat{z} && (7.61) \\ \mu\nabla^2\boldsymbol{u}_1 = \nabla p_1 + g\rho_1(x,t),\ \boldsymbol{u}_1\Big|_S &= -V_1\hat{z} && (7.62) \end{aligned}$$

$$\int_S \sigma_0 \cdot \hat{n} = m_s g \tag{7.63}$$

$$\int_S \sigma_1 \cdot \hat{n} = 0 \tag{7.64}$$

$$\frac{\partial \rho_0}{\partial t} + \boldsymbol{u}_{0R} \cdot \nabla \rho_0 + u_{0Z} \frac{\partial \rho_0}{\partial z} \quad = \quad 0. \tag{7.65}$$

Since $\nabla \rho_0 = 0$, we need u_{0Z} independent of R or $\rho_0 = constant$. We will choose the later. And therefore, the equation

$$\frac{\partial \rho_1}{\partial t} + \boldsymbol{u}_1 \cdot \nabla_R \rho_0 + \boldsymbol{u}_0 \cdot \nabla \rho_1 \quad = \quad 0 \text{ ,becomes} \tag{7.66}$$

$$\frac{\partial \rho_1}{\partial t} + + \boldsymbol{u}_0 \cdot \nabla \rho_1 \quad = \quad 0 \tag{7.67}$$

Now, we will need to solve for $\boldsymbol{u_1}$from Stokes equation force by ρ_1 distribution with V_1 giving Dirichlet boundary conditions V_1 would then be determined by requiring,

$$\int_S \sigma_1 \cdot \hat{n} dS = 0 \tag{7.68}$$

(this is an equation for V_1 through the reciprocal Theorem) This equation would be writable compactly by using the reciprocal theorem, though u_1 would still have to be computed. For Higher order corrections (or if ρ_0 were to depend on z, which would bring in $\boldsymbol{u}_1 \cdot \rho_0$ term in advancing ρ_1.

$$(\boldsymbol{V}_1)_3 m_s g = \int_{\Omega_f} dV (\boldsymbol{f} \cdot \boldsymbol{u}_1 - \boldsymbol{h} \cdot \boldsymbol{u}_0) \tag{7.69}$$

Want $\boldsymbol{f} = 0$ (since it can be absorbed by hydrostatic pressure into p_0) and $\boldsymbol{h} = -g\rho_1(\boldsymbol{x}, t)\hat{z}$

$$(\boldsymbol{V}_1)_3 = -1/m_s \left(\int_{\Omega_f} dV \rho_1(\boldsymbol{x}, t)(\boldsymbol{u}_0)_3(\boldsymbol{x}, t) \right) \tag{7.70}$$

where Ω_f is the exterior domain and $\rho_1(\boldsymbol{x}, t)$ is obtained thru advance of rho_1 with $\boldsymbol{u}_0$

$$\frac{\partial \rho_1}{\partial t} + \boldsymbol{u}_0 \cdot \nabla \rho_1 \quad = \quad 0 \tag{7.71}$$

Taking advantage of the linearity of the Stokes equations, we use our previous split of the fluid flow into two parts to define $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$. Let the small parameter $\epsilon = \rho_b - \rho_t$, and we define the constant $\rho_0 = \rho_t$.

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}_s(\boldsymbol{x}, t) + \boldsymbol{w}(\boldsymbol{x}, t) \text{ full theory} \tag{7.72}$$

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}_0(\boldsymbol{x}, t) + \epsilon \boldsymbol{u}_1(\boldsymbol{x}, t) \text{ approximation} \tag{7.73}$$

The first term, $\boldsymbol{u}_0 = \boldsymbol{u}_s(\boldsymbol{x}, t)$, is a Stokes flow in homogenous fluid of density ρ_0. In the lab phrase:

$$(\boldsymbol{u_0})_{(3)} = (\boldsymbol{u_0})_{(Z)} \quad = \quad -V_0 \left[\frac{-3A}{4r} - \frac{3AZ^2}{4r^3} - \frac{A^3}{4r^3} + \frac{3Z^2A^3}{4r^5} \right] \tag{7.74}$$

$$(\boldsymbol{u_0})_{(R)} \quad = \quad -V_0 \left[\frac{-3ARZ}{4r^3} + \frac{3A^3RZ}{4r^5} \right] \tag{7.75}$$

While the second term $\boldsymbol{u}_1 = \boldsymbol{w}(\boldsymbol{x}, t)$ is given by the buoyancy driven flow. Combining the solution for Stokes flow and perturbation velocity, we obtain the equation for the vertical

component of the velocity of the sphere and the advection fluid.

$$\begin{aligned}\frac{dY_3}{dt}(t;\rho) &= (\boldsymbol{V_0})_3 + \epsilon(\boldsymbol{V_1})_3(t;\rho_1) \\ &= \frac{g}{6\pi A\mu}(m_s - \frac{4}{3}\pi A^3\rho_0) + \epsilon\left(-1/m_s\int_{\Omega_f} dV\rho_1(\boldsymbol{x},t)(\boldsymbol{u}_0)_3(\boldsymbol{x},t)\right)\end{aligned}$$

$$\frac{\partial\rho_1}{\partial t}(\boldsymbol{x},t) + \boldsymbol{u}_0(\boldsymbol{x},t;V)\cdot\nabla\rho_1(\boldsymbol{x},t) = 0. \tag{7.76}$$

7.5.1 Define $(V_1)_3$

Defining our desired integral in terms of the deformed interface curve $\eta = Z(R)$.

$$(\boldsymbol{V}_1)_3 = -1/m_s\left(\int_{\Omega_f} dV\rho_1(\boldsymbol{x},t)(\boldsymbol{u}_0)_3(\boldsymbol{x},t)\right) \tag{7.77}$$

$$\rho_1 = \frac{\rho-\rho_0}{\rho_b-\rho_t} = \frac{\rho-\rho_t}{\rho_b-\rho_t}; \tag{7.78}$$

$$\begin{aligned}(\boldsymbol{V}_1)_3 &= -1/m_s\left(\frac{\rho_t-\rho_t}{\rho_b-\rho_t}\int_0^{2\pi}d\theta\int_0^{\infty}R\,dR\int_{Z(R)}^{\infty}dZ\,(\boldsymbol{u}_0)_3\right. \\ &\quad + \frac{\rho_b-\rho_t}{\rho_b-\rho_t}\int_0^{2\pi}d\theta\int_0^{\infty}R\,dR\int_{-\infty}^{Z(R)}dZ(\boldsymbol{u}_0)_3 \\ &\quad \left. - \frac{\rho_t-\rho_t}{\rho_b-\rho_t}\int_S dS\rho_t(\boldsymbol{u}_0)_3\right) \\ &= -1/m_s\int_0^{2\pi}d\theta\int_0^{\infty}R\,dR\int_{-\infty}^{Z(R)}dZ(\boldsymbol{u}_0)_3 \\ &= -1/m_sI_2\end{aligned}$$

In our existing code, we already compute :

$$I_w \;=\; -\int_0^{2\pi} d\theta \int_{\sqrt{R^2+Z^2}>A}^{R_0} RdR \int_{-Y_3}^{Z(R)} dZ \frac{(\boldsymbol{u}_0)_3}{-V_0}$$

Let us note that $-Y_3 < \eta$ in the reflux region and $-Y_3 > \eta$ in the entrainment region. Therefore we can find our desired integrals:

$$I_2 \;=\; \int_0^{2\pi} d\theta \int_0^{\infty} R\, dR \int_{-\infty}^{Z(R)} dZ (\boldsymbol{u}_0)_3 \tag{7.79}$$

with the domain of integration shown below

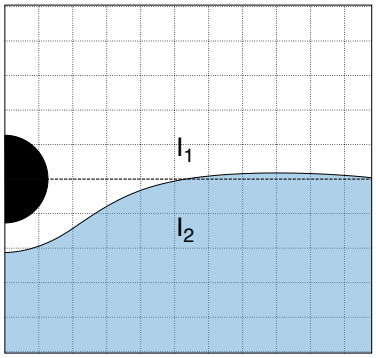


Figure 7.10: Top layer and bottom layer domains

by finding the following integrals apriori:

$$I_b \;=\; \int_0^{2\pi} d\theta \int_{-Y3}^{\infty} dZ \int_{Re\left(\sqrt{A^2-Z^2}\right)}^{\infty} R\, dR (\boldsymbol{u}_0)_3 \tag{7.80}$$

$$\tag{7.81}$$

using the domain of integrations non-dependent on the interfacial curve.

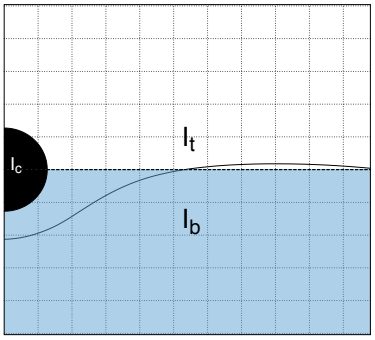


Figure 7.11: Domain of integration for integrals I_t and I_b

and using the already computed integrals in the code

$$I_w = -\int_0^{2\pi} d\theta \int_{\sqrt{R^2+Z^2}>A}^{R_0} RdR \int_{-Y_3}^{Z(R)} dZ \frac{(\boldsymbol{u}_0)_3}{-V_0} \tag{7.82}$$

$$\tag{7.83}$$

The diagram below illustrates the domain of integration of the integrals I_w

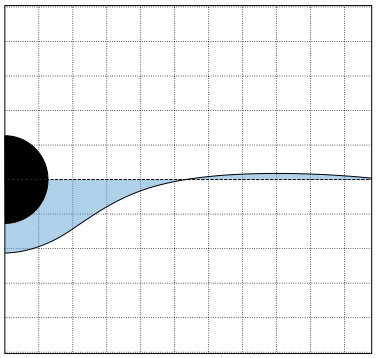


Figure 7.12: Domain of integration for I_w computed in code

Combining the integrals by using the relations,

$$I_2 = I_b - (-V_0)\, I_w \tag{7.84}$$

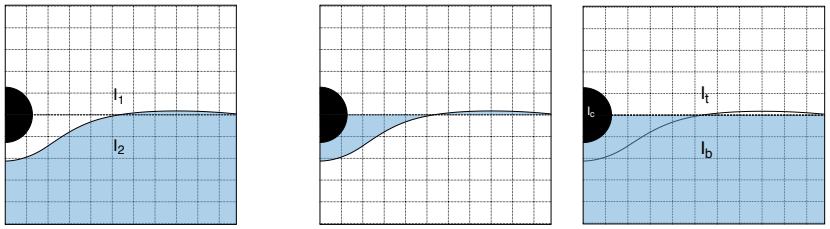


Figure 7.13: Domains of integration

Therefore, the equation of motion for the sphere becomes :

$$(\boldsymbol{V}_1)_3 \;=\; -1/m_s\,(I_b - (-V_0)\,I_w) \tag{7.85}$$

where I_w is already computed in the code.

Case 1: $-Y_3 < -A$

$$I_b \;=\; 2\pi\left(\int_{-Z_{max}}^{-Y_3} dZ \int_0^{R_0} R\,dR(\boldsymbol{u}_0)_3\right)$$

Case 2 :$|Y_3| < A$

$$I_b \;=\; 2\pi\left(\int_{-Z_{max}}^{-A} dZ \int_0^{R_0} R\,dR(\boldsymbol{u}_0)_3 + \int_{-A}^{-Y_3} dZ \int_{\sqrt{A^2-Z^2}}^{R_0} R\,dR(\boldsymbol{u}_0)_3\right)$$

Case 3: $-Y_3 > A$

$$\begin{aligned} I_b \;=\; & 2\pi\left(\int_{-Z_{max}}^{-A} dZ \int_0^{R_0} R\,dR(\boldsymbol{u}_0)_3 + \int_{-A}^{A} dZ \int_{\sqrt{A^2-Z^2}}^{R_0} R\,dR(\boldsymbol{u}_0)_3\right. \\ & +\; \left.\int_{A}^{-Y_3} dZ \int_0^{R_0} R\,dR(\boldsymbol{u}_0)_3\right) \end{aligned}$$

Even if this approach could correctly predict the position of the sphere, the shape of the interface will never be accurate.

CHAPTER 8

ENTRAINMENT DOMINATED REGIMES

In the limit as $(\rho_s - \rho_b) \to (\rho_b - \rho_t)$, the sphere velocity $\boldsymbol{V}(t)$ becomes very close to zero at the interface, the stokes velocity $\boldsymbol{u}_s$ also vanishes, and the perturbation velocity $\boldsymbol{w}$ is in charge of continuing the fluid motion and decrease the amount of entrained fluid, altering the balance of forces and allowing the sphere to change speed. To understand how the balancing of the forces behave as $\rho_s \to \rho_b$, we plot them below for the Far Field approximation. As the sphere approaches its minimum velocity at the interface, the perturbation velocity achieves its maximum. The smaller $\rho_s - \rho_b$, the closer the the balance of forces become.

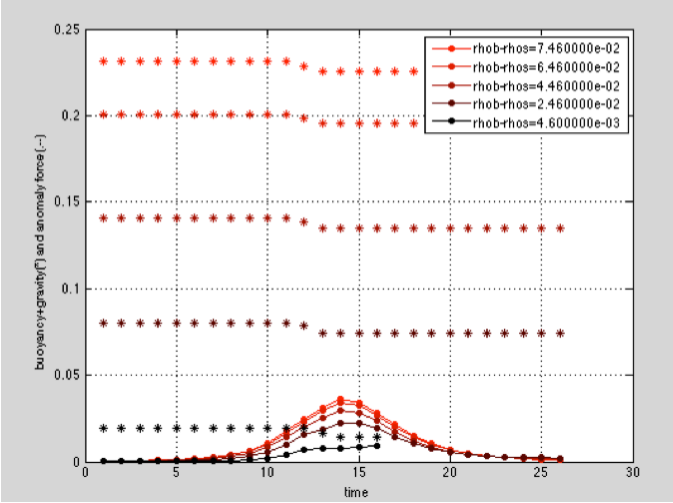


Figure 8.1: Plot of Archimedean buoyancy and density anomaly force as $\rho_s \to \rho_b$

When the density difference $(\rho_s - \rho_b) = (\rho_s - \rho_b)$ approaches the stratification jump $(\rho_b - \rho_t) = (\rho_b - \rho_t)$, the $\boldsymbol{w}$ field becomes dominant and we find ourselves in the Entrainment Regime, where the correct computation of $\boldsymbol{w}$ dictates the accuracy of the simulation output.

Using the far field approximation as a replacement for $\boldsymbol{w}$ has shown good agreement with the experiments when $(\rho_s - \rho_b) >> (\rho_b - \rho_t)$, i.e. when the Stokes velocity is dominant

with respect to the perturbation velocity, $\boldsymbol{u}(\boldsymbol{x},t) >> \boldsymbol{w}(\boldsymbol{x},t)$, since the interface does not get close to the sphere when $\boldsymbol{w}$ is at its maximum. The approximation correctly predicts the velocity profile of the particle and the changing shape of the deformed interface as the sphere settles. As we push $(\rho_s - \rho_b) \to (\rho_b - \rho_t)$, the effects of the entrainment become more prominent, since the interface gets close to the sphere when $\boldsymbol{w}$ is dominant with respect to the almost vanished stokes velocity $\boldsymbol{u}_s$. As the sphere approaches its minimum velocity at the interface, the perturbation velocity achieves its maximum. In this regime, the error given by the far field approximation is no longer negligible, and the violation of the non-slip boundary conditions (see Figure 7.4) deformes the interface in a non-physical way. In addition, because the sphere velocity and perturbation flow is computed as a volume integral over a changing domain of integration given by ρ, the shape of the interface affects the accuracy of the flow computed and the error that it propagates through each time step determines the overall success of the model.

8.1 Comparison with Experiments

A plastic sphere of radius $A = 0.635\,cm$ density $\rho_s = 1.3651\,g/cm^3$ was released in a two-layer sharply stratified tank of radius $R_0 = 5.4cm$, with top layer density $\rho_t = 1.3397\,g/cm^3$ and bottom layer density $\rho_b = 1.3649\,g/cm^3$. Viscosities are $\mu_t = 6.2349$ Poise, $\mu_b = 7.2315$ Poise for top and bottom layers respectively.

The comparison shows the best most refined version so far. This simulation includes the stokes flow up to the third correction, the full perturbation velocity in neighborhood of the sphere. The interface is shown at the last trusted time, based on the smoothness of the interface.

Note that at the last trusted time $t = 672s$, the experiment and theory have already departed significantly from each other, as seen in Figure (8.3). Because the diffusion time scales are $\sim$ 300 s, diffusion could have been making the difference seen up to the trusted

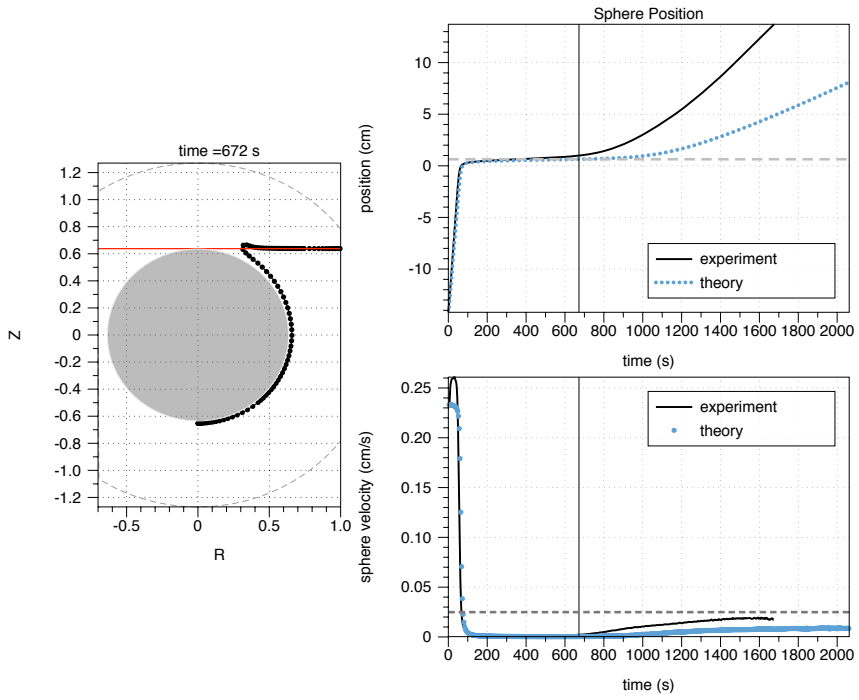


Figure 8.2: (a) Interface at time t = 672 s, the last trusted time. (b)The sphere position. (c)The sphere velocity. The black solid lines are the experiment tracking, the blue dots are the theory prediction, and the black vertical lines indicate the last trusted time t=672 s.

time.

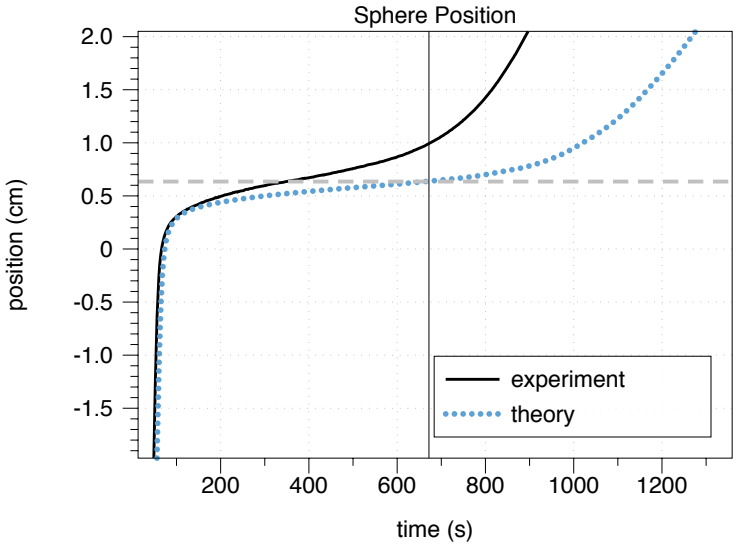


Figure 8.3: Zoomed in plots of sphere position showing the code departure from experiment.

Other examples of approaching the entrainment regimes are shown in Figures 8.4 and 8.5. In the first figure, the code agrees fairly well with the experiment and the only mismatch is due to using an average viscosity between the top and bottom layers. The next figure, however, falls into the more pronounced entrainment regimes where the residence time compares to diffusion time scales. In this case, the code does not recover in the same time as the experiment does.

8.2 Force Balance and the importance of Reflux

The forces acting on the sphere are the Archimedean force F_A and the density anomaly force F_w, such that the total force is equal to $F = F_A - F_w$. The latest can be expressed as the combination of the entrainment force F_E and the reflux force F_R , and it satisfies that $F_w = F_E - F_R$.

In the Entrainment regime, the reflux contribution is small but significant as without it the sphere would not be able to sink (see Figure 8.6). In contrast the Stokes regime would not need the reflux contribution for the motion of the sphere to continue. In the Stokes

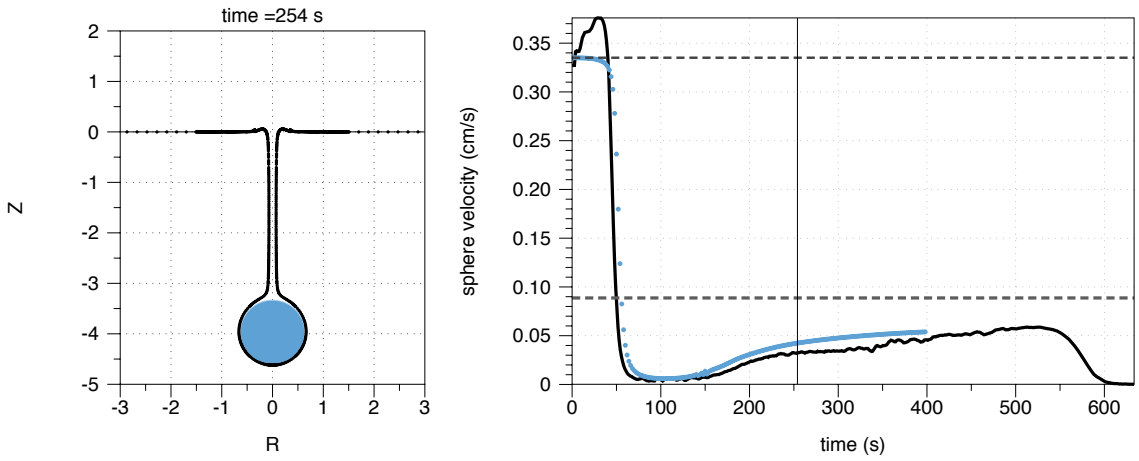


Figure 8.4: Experiment and theory comparison approaching the entrainment dominated regime. Left panel shows the model predicted interface and the right panel shows the experiment in black and the theory in blue. The experimental parameters are $A = 0.641\,cm$, $\rho_s = 1.36712\,g/cm^3$, $\rho_t = 1.34695\,g/cm^3$, $\rho_b = 1.36178\,g/cm^3$, and $\mu = 4$ Poise.

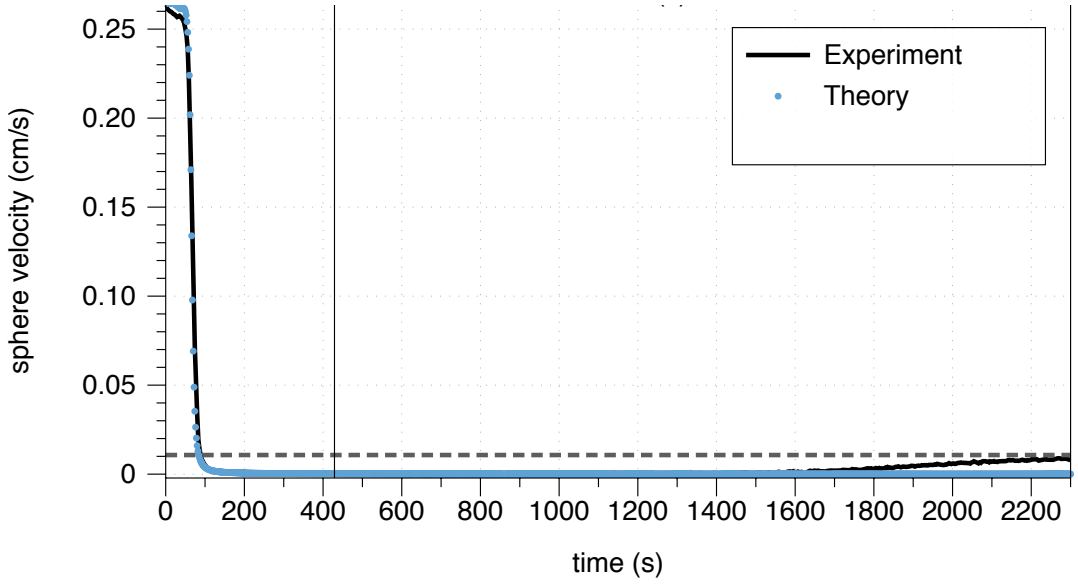


Figure 8.5: Experiment and theory comparison approaching the entrainment dominated regime. Left panel shows the model predicted interface and the right panel shows the experiment in black and the theory in blue. The experimental parameters are $A = 0.641\,cm$, $\rho_s = 1.36712\,g/cm^3$, $\rho_t = 1.34419\,g/cm^3$, $\rho_b = 1.36639\,g/cm^3$, and $\mu = 5.75$ Poise.

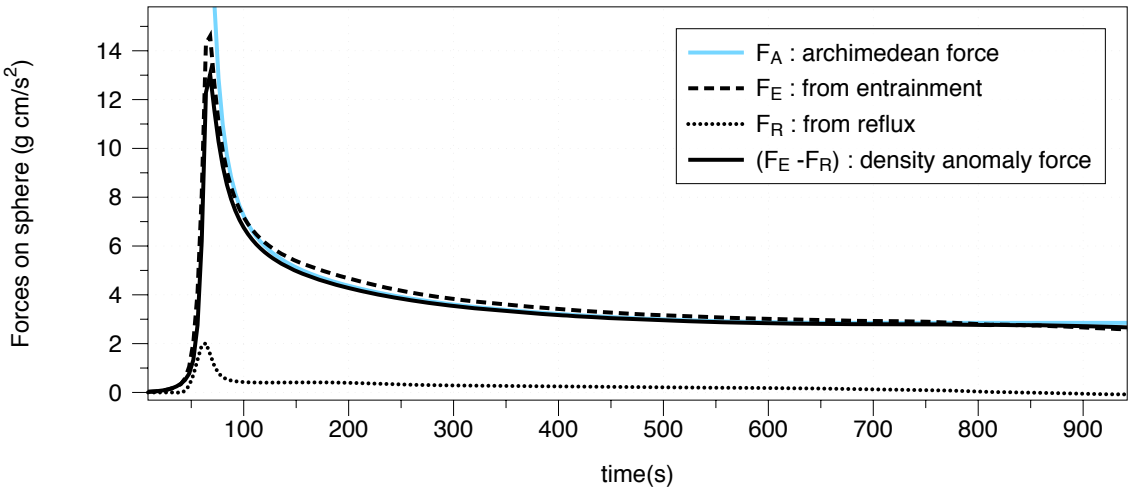


Figure 8.6: Forces on the sphere for the Entrainment (w-dominated) regime showing the importance of the reflux portion to the motion of the sphere since the entrainment force F_E is bigger than the Archimedean force F_A for a portion of time.

regime, the entrainment force F_E is smaller than the Archimedean force F_A.

8.3 Shell Depletion and Layer Thickness

A natural question to ask ourselves has to do with the shell thickness and its contribution to the the extra buoyancy force on the sphere and the advection of fluid. What parameter wins in helping the motion of the sphere? If $\rho_b - \rho_t$ is very large then the rate of depletion is fast – advecting with $\boldsymbol{w}$ that has magnitude directly proportional with $(\rho_b - \rho_t)$, therefore the shell gets very small thus helping the sphere recover. However, we have seen that a large $(\rho_b - \rho_t)$ also affects the magnitude of the extra buoyancy force on the sphere making it slow down.

Figure 8.7: Experimental picture showing the thin shell around the sphere for entrainment dominated regimes.

CHAPTER 9

CONCLUSIONS

The importance of the study of the settling of particles in stratified fluids can be extended to the many environmental applications such as better understanding the marine carbon cycle. From solid and porous particles to multi particle problem, the effects of entrainment plays a significant role in their delayed settling. Pollution clearing times and industrial applications are just of a few of the significant aspects of this study. This project focuses on miscible sharp stratification regimes that are of known importance in the ocean, where thin layers of aggregate become hotspots of bacteria remineralization. Our study focusing on a single solid sphere showed that not only a sphere can exhibit a slow down beyond its terminal velocity at the bottom layer but it can also rest at the interface for periods of time comparable to diffusion time scales of salt into the entrainment shell.

Our investigation to model the descent of a solid sphere in sharp stratifications has significantly improved in optimizing the numerics and computational time. More importantly, we have been able to implement the full theory needed the entrainment regime cases. By implementing the full solution and approximations in their respective regions of validity, the result of this investigation provides a predictive tool and describes the validity of the model for large time scales when salt diffusion plays an active role. In addition, the simulation delivers a significant time reduction to solving the proposed problem where a full Navier-Stokes simulation would be needed. The resulting solution shows the velocity of the sphere, the shaping of the interface, and the fluid flow created. With these results, we expect to give insight into the role of strong stratification in the prolonged settling of solid and porous particles in the ocean.

APPENDIX A

OSEEN INTEGRALS

In this appendix, we focus only on writing out the θ integral of the vertical component only, but the full horizontal component is also implemented in the numerical code. Let us define the vertical component of the perturbation velocity $w_Z(R,Z)$ is given by:

$$w_Z(R,Z) = \int_0^{\infty} d\rho\,\rho \int_{\zeta_1(\rho)}^{\zeta_2(\rho)} d\zeta \int_0^{2\theta} d\theta \Bigg(I_1 + I_2 + \left(-\frac{-a^2+\rho^2+\zeta^2}{\sqrt{\rho^2+\zeta^2}} \right) (I_3 + I_4 + I_5 + I_6) + \left(a^2 - R^2 - Z^2\right)(I_7 + I_8 + I_9 + I_{10} + I_{11} + I_{12}) \Bigg) \tag{A.1}$$

where $\int_0^{2\pi} d\theta I_j$ are given by:

$$\begin{aligned}
\int_0^{2\pi} d\,\theta I_1 &= F_1\left(R^2+\rho^2+(Z-\zeta)^2, -2R\rho\right) + (Z-\zeta)^2 F_2\left(R^2+\rho^2+(Z-\zeta)^2, -2R\rho\right) \\
\int_0^{2\pi} d\,\theta I_2 &= \sqrt{\frac{a^2}{\rho^2+\zeta^2}} - F_1\left(\frac{a^4 - 2a^2 Z\zeta + (R^2+Z^2)(\rho^2+\zeta^2)}{\rho^2+\zeta^2}, -\frac{2a^2R\rho}{\rho^2+\zeta^2}\right) \\
&- \sqrt{\frac{a^2}{\rho^2+\zeta^2}} \frac{a^2\left(\zeta\left(Z\zeta - a^2\right) + \rho^2 Z\right)^2 F_2\left(\frac{a^4 - 2a^2Z\zeta + \left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2}, -\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\left(\rho^2+\zeta^2\right)^3} \\
\int_0^{2\pi} d\,\theta I_3 &= \frac{a\zeta^2 F_1\left(\frac{a^4 - 2a^2Z\zeta + \left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2}, -\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\left(\rho^2+\zeta^2\right)^2} \\
\int_0^{2\pi} d\,\theta I_4 &= -\frac{a^3\zeta\left(\zeta\left(a^2 - Z\zeta\right) - \rho^2 Z\right) F_2\left(\frac{a^4 - 2a^2Z\zeta + \left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2}, -\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\left(\rho^2+\zeta^2\right)^3} \\
\int_0^{2\pi} d\,\theta I_5 &= -\frac{a^3\zeta\left(\zeta\left(a^2 - Z\zeta\right) - \rho^2 Z\right) F_2\left(\frac{a^4 - 2a^2Z\zeta + \left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2}, -\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\left(\rho^2+\zeta^2\right)^3}
\end{aligned}$$

$$
\begin{aligned}
\int_0^{2\pi} d\,\theta I_6 &= -\frac{2a^3\zeta^2\left(a^2-Z\zeta\right)F_2\left(\frac{a^4-2a^2Z\zeta+\left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2},-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\left(\rho^2+\zeta^2\right)^3} \\
&+ 2a^3\zeta^2\frac{R\rho F_4\left(\frac{a^4-2a^2Z\zeta+\left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2},-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\left(\rho^2+\zeta^2\right)^3} \\
\int_0^{2\pi} d\,\theta I_7 &= \left(\frac{3a\zeta^2}{\rho^2+\zeta^2}-\frac{3Z\zeta}{a}+a\right)F_2\left(\frac{a^4-2a^2Z\zeta+\left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2},-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right) \\
&- 3a\left(Z-\frac{a^2\zeta}{\rho^2+\zeta^2}\right)^2F_6\left(\frac{a^4-2a^2Z\zeta+\left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)}{\rho^2+\zeta^2},-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right) \\
\int_0^{2\pi} d\,\theta I_8 &= \frac{6a\zeta\left(Z-\frac{a^2\zeta}{\rho^2+\zeta^2}\right)\left(Z\zeta-a^2\right)F_6\left(\frac{a^4\rho^2}{\left(\rho^2+\zeta^2\right)^2}+\left(\frac{a^2\zeta}{\rho^2+\zeta^2}-Z\right)^2+R^2,-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\rho^2+\zeta^2} \\
&+ \frac{6a\zeta\left(Z-\frac{a^2\zeta}{\rho^2+\zeta^2}\right)R\rho F_7\left(\frac{a^4\rho^2}{\left(\rho^2+\zeta^2\right)^2}+\left(\frac{a^2\zeta}{\rho^2+\zeta^2}-Z\right)^2+R^2,-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\rho^2+\zeta^2} \\
&- \frac{2a\zeta^2F_2\left(\frac{a^4\rho^2}{\left(\rho^2+\zeta^2\right)^2}+\left(\frac{a^2\zeta}{\rho^2+\zeta^2}-Z\right)^2+R^2,-\frac{2a^2R\rho}{\rho^2+\zeta^2}\right)}{\rho^2+\zeta^2} \\
\int_0^{2\pi} d\,\theta I_{11} &= -\frac{3\left(|\boldsymbol{x}|\sqrt{\rho^2+\zeta^2}+Z\zeta\right)F_9\left(\frac{a^2|\boldsymbol{x}|\sqrt{\rho^2+\zeta^2}}{|\boldsymbol{y}|^2}+\frac{a^2Z\zeta}{|\boldsymbol{y}|^2},\frac{a^2R\rho}{|\boldsymbol{y}|^2}\right)}{a|\boldsymbol{x}|} \\
\int_0^{2\pi} d\,\theta I_{12} &= \frac{3\left(|\boldsymbol{x}|\sqrt{\rho^2+\zeta^2}+Z\zeta\right)F_9\left(\frac{a^2|\boldsymbol{x}|\sqrt{\rho^2+\zeta^2}}{|\boldsymbol{y}|^2}+\frac{a^2Z\zeta}{|\boldsymbol{y}|^2},\frac{a^2R\rho}{|\boldsymbol{y}|^2}\right)}{a|\boldsymbol{x}|}
\end{aligned}
$$

and the F_j are expressed in terms of elliptic integrals K, E, and Π of the first, second, and third kind respectively.

$$
\begin{aligned}
F_1(A,B) &= \frac{4K\left(\frac{2B}{A+B}\right)}{\sqrt{A+B}} \\
F_2(A,B) &= \frac{4E\left(\frac{2B}{A+B}\right)}{(A-B)\sqrt{A+B}} \\
F_3(A,B) &= \frac{4\left((A+B)E\left(\frac{2B}{A+B}\right) - AK\left(\frac{2B}{A+B}\right)\right)}{B\sqrt{A+B}} \\
F_4(A,B) &= -\frac{4\left(A(A+B)E\left(\frac{2B}{A+B}\right) - (A^2-B^2)\,K\left(\frac{2B}{A+B}\right)\right)}{B(A-B)(A+B)^{3/2}} \\
F_5(A,B) &= \frac{(8A^2-4B^2)\,E\left(\frac{2B}{A+B}\right) + 8A(B-A)K\left(\frac{2B}{A+B}\right)}{B^2(A-B)\sqrt{A+B}} \\
F_6(A,B) &= \frac{4\left((B-A)K\left(\frac{2B}{A+B}\right)\right.}{3(A-B)^2(A+B)^{3/2}} \\
&+ \frac{\left.4AE\left(\frac{2B}{A+B}\right)\right)}{3(A-B)^2(A+B)^{3/2}} \\
F_7(A,B) &= -\frac{4\left((A^2+3B^2)\,E\left(\frac{2B}{A+B}\right) + A(B-A)K\left(\frac{2B}{A+B}\right)\right)}{3B(A-B)^2(A+B)^{3/2}} \\
F_8(A,B) &= -\frac{4\left(2\left(A^3-3AB^2\right)E\left(\frac{2B}{A+B}\right) + \left(-2A^3+2A^2B+3AB^2-3B^3\right)K\left(\frac{2B}{A+B}\right)\right)}{3B^2(A-B)^2(A+B)^{3/2}} \\
F_9(A,B) &= \frac{2\pi}{\sqrt{A^2-B^2}} \\
F_{10}(A,B) &= \frac{2\pi A}{(A^2-B^2)^{3/2}} \\
F_{11}(A,B) &= -\frac{2\pi B}{(A^2-B^2)^{3/2}}
\end{aligned}
$$

The log terms that contains the removable singularities $I_{L3} = I_9 + I_{10} + I_{11} + I_{12}$ give rise to elliptic integrals of the third kind Π.

$$
\begin{aligned}
I_{L3} &= \frac{-|\boldsymbol{y}^*|\left(\frac{1}{|\boldsymbol{x}-\boldsymbol{y}^*|} - \frac{(x_3-y_3^*)^2}{|\boldsymbol{x}-\boldsymbol{y}^*|^3}\right) + \frac{y_3^*(x_3-y_3^*)}{|\boldsymbol{y}^*||\boldsymbol{x}-\boldsymbol{y}^*|} + 1}{-|\boldsymbol{y}^*|^2 + |\boldsymbol{y}^*||\boldsymbol{x}-\boldsymbol{y}^*| + x_1y_1^* + x_2y_2^* + x_3y_3^*} \\
&- \frac{\left(\frac{|\boldsymbol{y}^*|(x_3-y_3^*)}{|\boldsymbol{x}-\boldsymbol{y}^*|} + y_3^*\right)\left(-\frac{|\boldsymbol{y}^*|(x_3-y_3^*)}{|\boldsymbol{x}-\boldsymbol{y}^*|} + \frac{|\boldsymbol{x}-\boldsymbol{y}^*|y_3^*}{|\boldsymbol{y}^*|} + x_3 - 2y_3^*\right)}{\left(-|\boldsymbol{y}^*|^2 + |\boldsymbol{y}^*||\boldsymbol{x}-\boldsymbol{y}^*| + x_1y_1^* + x_2y_2^* + x_3y_3^*\right)^2} \\
&+ \frac{\left(\frac{|\boldsymbol{y}^*|x_3}{|\boldsymbol{x}|} + y_3^*\right)\left(\frac{|\boldsymbol{x}|y_3^*}{|\boldsymbol{y}^*|} + x_3\right)}{(|\boldsymbol{y}^*||\boldsymbol{x}| + x_1y_1^* + x_2y_2^* + x_3y_3^*)^2} - \frac{\frac{x_3y_3^*}{|\boldsymbol{y}^*||\boldsymbol{x}|} + 1}{|\boldsymbol{y}^*||\boldsymbol{x}| + x_1y_1^* + x_2y_2^* + x_3y_3^*}
\end{aligned}
$$

We can also leave this term without its closed form expression for the code. Still, I_{L3} is well understood and can be manipulated for numerical integration by analyzing its expression in terms of another integral where its regularization can be extracted.

Let,

$$
\begin{aligned}
a_1 &= (R^2+Z^2)(\rho^2+\zeta^2) + A^4 - 2A^2Z\zeta \\
b_1 &= -2A^2R\rho \\
a_2 &= Z\zeta - A^2 \\
b_2 &= R\rho \\
a_3 &= \sqrt{(R^2+Z^2)(\rho^2+\zeta^2)} + Z\zeta \\
c_0 &= \frac{6A^3\zeta - 3A(\rho^2+\zeta^2)\zeta}{(\rho^2+\zeta^2)^3} \\
s_1 &= \frac{\sqrt{a_1^2-b_1^2} - a_1}{b_1} \\
s_2 &= \frac{-\sqrt{a_1^2-b_1^2} - a_1}{b_1}
\end{aligned}
$$

and

$$\begin{aligned}
\tilde{I}_{L3} &= c_0 \frac{\partial}{\partial Z}\left(\arctan\left(2\,\frac{(b_2 r^2+2a_2 r+b_2)\sqrt{2|b_1|}\sqrt{r\,(s_1-r)\,(s_2-r)}}{2|b_1|\,r\,(s_1-r)\,(s_2-r)-\left((b_2 r^2+2a_2 r+b_2)\right)^2}\right)\right) \\
&= c_0\,\frac{\partial}{\partial Z}\left(P\,(r,\rho,\zeta,R,Z)\,\sqrt{r(s_1-r)(s_2-r)}\right) \\
&= c_0\left(-\frac{4r\left(a^2\rho\left(r^2R\zeta-2r\rho Z+R\zeta\right)+R\left(\rho^2+\zeta^2\right)\left(\left(r^2+1\right)\rho Z-2rR\zeta\right)\right)}{\left(R^2\left(\left(r^2-1\right)^2\rho^2-4r^2\zeta^2\right)+4r\left(r^2+1\right)R\rho Z\zeta-4r^2\rho^2Z^2\right)}\right. \\
&\times \left.\frac{1}{\sqrt{-r\left(a^4r-a^2\left(\left(r^2+1\right)R\rho+2rZ\zeta\right)+r\left(R^2+Z^2\right)\left(\rho^2+\zeta^2\right)\right)}}\right)
\end{aligned}$$

The kernel of the integral of $\tilde{I}_9$ only has a singularity at $r = s_1$ which behaves as :

$$I_{L3} \sim O\left(\frac{1}{\sqrt{s_1-r}}\right) \text{ as } r \to s_1 \tag{A.2}$$

To account for this integrable singularity, we regularize the function by:

$$\begin{aligned}
I_9 &= \tilde{I}_9 - \tfrac{1}{2}c_0\tfrac{1}{\sqrt{s_1-r}}\tfrac{\sqrt{s_1(s_2-s_1)}}{s_1}\tfrac{\partial s_1}{\partial Z}P\Big|_{r=s_1} \\
&= \tilde{I}_9 + c_0\tfrac{1}{\sqrt{s_1-r}}\left(\frac{\sqrt{R\rho\sqrt{\left(a^4-2a^2Z\zeta+(R^2+Z^2)(\rho^2+\zeta^2)\right)^2-4a^4R^2\rho^2}}}{\sqrt{\left(a^4-2a^2Z\zeta+(R^2+Z^2)(\rho^2+\zeta^2)\right)^2-4a^4R^2\rho^2}}\right. \\
&\quad\times\tfrac{\sqrt{2}\left(\zeta\left(Z\zeta-a^2\right)+\rho^2Z\right)\left(-2a^2r+\left(r^2+1\right)R\rho+2rZ\zeta\right)}{aR\rho} \\
&\quad\times\left.\tfrac{\left(a^4-2a^2Z\zeta-\sqrt{\left(a^4-2a^2Z\zeta+(R^2+Z^2)(\rho^2+\zeta^2)\right)^2-4a^4R^2\rho^2}+R^2\rho^2+R^2\zeta^2+\rho^2Z^2+Z^2\zeta^2\right)}{\left(8a^4r^2-8a^2r((r^2+1)R\rho+2rZ\zeta)+4r(r^2+1)R\rho Z\zeta+4r^2Z^2(\rho^2+2\zeta^2)+R^2(4r^2\zeta^2+(r^4+6r^2+1)\rho^2)\right)}\right)
\end{aligned}$$

Which regularizes our function and adds an integrand in closed form.

$$I_{L31} = \frac{\sqrt{s_1(s_2-s_1)}}{s_1}\frac{\partial s_1}{\partial Z}P\Big|_{r=s_1}\sqrt{s_1} \tag{A.3}$$

Finally, the last part of the kernel is a boundary term that came from the integration by parts of the logarithmic function. In summary, the vertical component of the perturbation velocity w_3 is:

$$
\begin{aligned}
w_3(R,Z) &= \int d\rho\,\rho \int d\zeta \left(\int_0^{2\pi} d\theta I_1 + \int_0^{2\pi} d\theta I_2 + \left(-\frac{-a^2+\rho^2+\zeta^2}{\sqrt{\rho^2+\zeta^2}} \right) \left(\int_0^{2\pi} d\theta I_3 \right.\right. \\
&+ 2\int_0^{2\pi} d\theta I_4 + \left.\int_0^{2\pi} d\theta I_6 \right) \\
&+ \left(a^2 - R^2 - Z^2\right) \left(\int_0^{2\pi} d\theta I_7 + \int_0^{2\pi} d\theta I_8 + \left.\left. \int_0^{s_1} dr\,(I_{L3}) + I_{L31} \right) \right) \right) \\
&+ \int d\rho\,\rho \int_0^{2\pi} d\theta \left(a^2 - R^2 - Z^2\right) I_{L32}
\end{aligned}
$$

A.1 Integration procedure

In this section, we go over the integration procedure on the vertical component of one of the Oseen tensor terms, I_9.

Let,

$$
\begin{aligned}
I_9 = &\Big(3\Big(\rho^2+\zeta^2\Big)\Big(\Big(\frac{r^2\zeta\Big(r^4 - 2r^2R\rho\cos(\phi-\theta) - 2r^2Z\zeta + \Big(R^2+Z^2\Big)\Big(\rho^2+\zeta^2\Big)\Big)}{\Big(\rho^2+\zeta^2\Big)^2} \\
&+ \frac{r^4\Big(Z - \frac{r^2\zeta}{\rho^2+\zeta^2}\Big)}{\rho^2+\zeta^2}\Big)\Big(Z - \frac{r^2\zeta}{\rho^2+\zeta^2}\Big) \\
&- \Big(\sqrt{\frac{r^4}{\rho^2+\zeta^2}}\Big(r^4 - 2r^2R\rho\cos(\phi-\theta) - 2r^2Z\zeta + \Big(R^2+Z^2\Big)\Big(\rho^2+\zeta^2\Big)\Big) \\
&\Big(\sqrt{\frac{r^4}{\rho^2+\zeta^2}} - \sqrt{\frac{r^4 - 2r^2R\rho\cos(\phi-\theta) - 2r^2Z\zeta + \Big(R^2+Z^2\Big)\Big(\rho^2+\zeta^2\Big)}{\rho^2+\zeta^2}}\Big)\Big)\big(\rho^2+\zeta^2\big)^{-1}\Big)\Big) \\
&\Big(r^3\Big(\frac{r^4 - 2r^2R\rho\cos(\phi-\theta) - 2r^2Z\zeta + \Big(R^2+Z^2\Big)\Big(\rho^2+\zeta^2\Big)}{\rho^2+\zeta^2}\Big)^{3/2} \\
&\Big(-\frac{r^4}{\rho^2+\zeta^2} + \frac{r^2R\rho\cos(\phi-\theta)}{\rho^2+\zeta^2} + \frac{r^2Z\zeta}{\rho^2+\zeta^2} + \sqrt{\frac{r^4}{\rho^2+\zeta^2}} \\
&\sqrt{\frac{r^4 - 2r^2R\rho\cos(\phi-\theta) - 2r^2Z\zeta + \Big(R^2+Z^2\Big)\Big(\rho^2+\zeta^2\Big)}{\rho^2+\zeta^2}}\Big)\Big)^{-1}
\end{aligned}
\tag{A.4}
$$

which can be rewritten as

$$I_9 = \frac{\alpha + \beta(A + B\cos(\phi - \theta)) + \gamma(A + B\cos(\phi - \theta))^{3/2}}{J\left((A + B\cos(\phi - \theta))^{3/2}\left(2r^2(A + B\cos(\phi - \theta))^{1/2} + 2E - B\cos(\phi - \theta)\right)\right)}$$

Let $u = A + B\cos(\phi - \theta)$

$$I_9 = \frac{\alpha + \beta u + \gamma u^{3/2}}{Ju^{3/2}\left(mu^{1/2} - u + n\right)} \tag{A.5}$$

Simplifying and applying partial fraction decomposition to the first two terms, we get:

$$\begin{aligned} I_9 = \quad & \frac{\alpha}{J}\left(\frac{\frac{n+m^2}{n^3}u - \frac{m}{n^2}u^{1/2} + \frac{1}{n}}{u^{3/2}} + \frac{\frac{n+m^2}{n^3}u^{1/2} + -m/n^2 - m/n^3(n+m^2)}{mu^{1/2} - u + n}\right) \\ + \quad & \frac{\beta}{J}\left(\frac{\frac{1}{mn}u^{1/2} + \frac{1}{n}}{u^{1/2}} + \frac{\frac{1}{mn}u - \frac{m^2+n}{nm}}{mu^{1/2} - u + n}\right) + \frac{\gamma}{J}\left(\frac{1}{mu^{1/2} - u + n}\right) \end{aligned} \tag{A.6}$$

Integrating over θ gives elliptic integrals K and E of the first and second kind respectively.

$$\begin{aligned} \int_0^{2\pi} d\theta I_9 = & \frac{\alpha}{J}\Big(4\frac{n+m^2}{n^3}\frac{K(\frac{2B}{A+B})}{\sqrt{A+B}} - \frac{2\pi\frac{m}{n^2}}{\sqrt{(A-B)/(A+B)}(A+B)} + \frac{1}{n}\frac{4E(\frac{2B}{A+B})}{\sqrt{A+B}(A-B)} \\ & + \int_0^{2\pi} d\theta \frac{\frac{n+m^2}{n^3}u^{1/2} + -m/n^2 - m/n^3(n+m^2)}{mu^{1/2} - u + n}\Big) \\ & + \frac{\beta}{J}\Big(2\pi\frac{1}{mn} + 4\frac{1}{n}\frac{K(\frac{2B}{A+B})}{\sqrt{A+B}} + \int_0^{2\pi} d\theta \frac{\frac{1}{mn}u - \frac{m^2+n}{nm}}{mu^{1/2} - u + n}\Big) \\ & + \frac{\gamma}{J}\int_0^{2\pi} d\theta \left(\frac{1}{mu^{1/2} - u + n}\right) \end{aligned} \tag{A.7}$$

The remaining integrals can be rewritten as

$$\frac{1}{J}\int_0^{2\pi} d\theta \frac{\alpha\left(\frac{n+m^2}{n^3}u^{1/2} + -m/n^2 - m/n^3(n+m^2)\right) + \beta\left(\frac{1}{mn}u - \frac{m^2+n}{nm}\right) + \gamma}{mu^{1/2} - u + n} \tag{A.8}$$

$$= \frac{1}{J}\int_0^{2\pi} d\theta\, \frac{\tau u^{1/2} + \lambda u + \omega}{m u^{1/2} - u + n} \tag{A.9}$$

Let,

$$\begin{aligned} u &= A + B\cos(\phi - \theta) \\ du &= B\sin(\phi - \theta)d\theta \\ d\theta &= \frac{du}{\pm B\sqrt{1 - \frac{(u-A)^2}{B^2}}} \end{aligned}$$

Performing this change of variables leads to the integral form:

$$\frac{1}{J}\int_0^{2\pi} d\theta\, \frac{\tau u^{1/2} + \lambda u + \omega}{m u^{1/2} - u + n} \tag{A.10}$$

$$= \frac{2}{BJ}\int_{A+B}^{A-B} du \frac{\tau u^{1/2} + \lambda u + \omega}{\Big(m u^{1/2} - u + n\Big)\sqrt{1 - \frac{(u-A)^2}{B^2}}} \tag{A.11}$$

$$= \frac{2}{J}\int_{A+B}^{A-B} du \frac{\tau\sqrt{u} + \lambda u + \omega}{\Big(m\sqrt{u} - u + n\Big)\sqrt{B^2 - (u-A)^2}} \tag{A.12}$$

In order to make use of Byrd and Friedman [7] chapter on Integrands Involving the Square Roots of Sums and Differences of Squares, we make the transformation:

$$\begin{aligned} u &= t^2 \\ du &= 2\,t\,dt \end{aligned}$$

$$= \frac{4}{J}\int_{\sqrt{A+B}}^{\sqrt{A-B}} dt \frac{\lambda t^3 + \tau t^2 + \omega t}{\Big(mt - t^2 + n\Big)\sqrt{B^2 - (t^2 - A)^2}} \tag{A.13}$$

Positive argument of radical Since $A = r^4 - 2r^2\zeta Z + (\rho^2 + \zeta^2)(R^2 + Z^2)$ and $B = -2r^2 R\rho$, we can proof that A is always bigger than B in magnitude and thus $A + B$ and $A - B$ are

always greater than zero.

Proof.

Consider the nondimensionalized expressions $A+B$ and $A-B$

$$A+B = 1-2\zeta Z+(\rho^2+\zeta^2)(R^2+Z^2)-2R\rho$$
$$A-B = 1-2\zeta Z+(\rho^2+\zeta^2)(R^2+Z^2)+2R\rho$$

We can proof that $A+B$ is sign definite, always greater than zero. As a quadratic function of Z, the discriminant is always less than or equal to zero.

$$A+B = \left(\rho^2+\zeta^2\right)Z^2-(2\zeta)Z+\left(1+R^2\left(\rho^2+\zeta^2\right)-2R\rho\right)$$
$$\text{discriminant}(A+B) = -4\left(R^2\rho^4+2R^2\rho^2\zeta^2+R^2\zeta^4-2R\rho^3-2R\rho\zeta^2+\rho\leq 0^2\right)$$

However, the only root that $A+B$ has is $R=\rho/\left(\rho^2+\zeta^2\right)$ and $Z=\zeta/\left(\rho^2+\zeta^2\right)$ does not satisfy $R^2+Z^2>1$.

Proof. Consider $f(r)=A+B=r^4-2r^2\zeta Z+(\rho^2+\zeta^2)(R^2+Z^2)-2r^2R\rho$. We want to proof that$f(r)>0\forall(R,Z,\rho,\zeta)$

$$\begin{aligned} f(r) &= r^4-2r^2\zeta Z+\rho^2R^2+\rho^2Z^2+\zeta^2R^2+\zeta^2Z^2-2r^2R\rho && \text{(A.14)}\\ &= r^4+\rho^2R^2-2r^2R\rho+\rho^2Z^2+\zeta^2R^2+\zeta^2Z^2-2r^2\zeta Z && \text{(A.15)}\\ &= (r^2-\rho R)^2+\rho^2Z^2+\zeta^2R^2+\zeta^2Z^2-2r^2\zeta Z && \text{(A.16)}\\ &> (r^2-\rho R)^2+\rho^2Z^2+\zeta^2R^2-2r^2\zeta Z && \text{(A.17)} \end{aligned}$$

Note that $f(r)\to\infty$ as $r\to\pm\infty$. We find the minimums of the function and find that they

are always greater than zero.

$$f'(r) = 4r(r^2 - \rho R) - 4a\zeta Z = 0 \tag{A.18}$$

$$4r(r^2 - \rho R - \zeta Z) = 0 \tag{A.19}$$

$$r = 0 \quad \text{and } r = \pm\sqrt{\rho R + \zeta Z} \tag{A.20}$$

$$f(0) = \rho^2 R^2 + \rho^2 Z^2 + \zeta^2 R^2 + \zeta^2 Z^2 > 0$$

$$f(\pm\sqrt{\rho R + \zeta Z}) = (\zeta Z)^2 + \rho^2 Z^2 + \zeta^2 R^2 - (\rho R + \zeta Z)\zeta Z > 0$$

Now, we have an integral of the form:

$$= \frac{4}{J}\int_{\sqrt{A+B}}^{\sqrt{A-B}} dt \frac{R(t)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \tag{A.21}$$

where r_1 and r_2 are the roots of the quadratic equation $B^2 - (t^2 - A)^2$ of t.

$$r1 = A + B$$

$$r2 = A - B$$

and $R(t) = \big(\lambda t^3 + \tau t^2 + \omega t\big)/\big(mt - t^2 + n\big)$ is a rational integral function of t. Where p_1 and p_2 satisfy the relation :

$$t^4 - (2n + m^2)t^2 + n^2 = (t^2 - p_1)(t^2 - p_2) \tag{A.22}$$

$$p_1 = 1/2\Big(2n + m^2 + m\sqrt{m^2 + 4n}\Big) \tag{A.23}$$

$$p_2 = 1/2\Big(2n + m^2 - m\sqrt{m^2 + 4n}\Big) \tag{A.24}$$

Rewriting $R(t)$ into symmetric and antisymmetric parts, we get:

$$
\begin{aligned}
R(t) &= t\,R_1(t^2) + R_2(t^2) \\
tR_1(t^2) &= \frac{t\big((n\lambda - \tau m - \omega)t^2 - \lambda t^4 + n\omega\big)}{t^4 - (2n + m^2)t^2 + n^2} \\
R_2(t^2) &= = \frac{(-\lambda m - \tau)t^4 + (n\tau - \omega m)t^2}{t^4 - (2n + m^2)t^2 + n^2}
\end{aligned}
$$

So integral becomes:

$$\frac{4}{J}\int_{\sqrt{A+B}}^{\sqrt{A-B}} dt \frac{R(t)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \tag{A.25}$$

$$= \frac{4}{J}\int_{\sqrt{A+B}}^{\sqrt{A-B}} dt \frac{tR_1(t^2)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} + \frac{4}{J}\int_{\sqrt{A+B}}^{\sqrt{A-B}} dt \frac{R_2(t^2)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \tag{A.26}$$

The left integral involving $tR_1(t^2)$ can be reduced to elementary form by substitution $t^2 = u$.

$$\frac{4}{J}\int_{\sqrt{A+B}}^{\sqrt{A-B}} dt \frac{tR_1(t^2)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \tag{A.27}$$

$$= \frac{2}{J}\int_{A+B}^{A-B} du \frac{R_1(u)}{\sqrt{-(u - r_1)(u - r_2)}} \tag{A.28}$$

$$R_1(u) = \frac{(n\lambda - \tau m - \omega)u - \lambda u^2 + n\omega}{(u - p_1)(u - p_2)} \tag{A.29}$$

Case 1: p_1 and p_2 lie outside the domain of integration $\big[A+B, A-B\big]$.

$$\frac{2}{J}\int_{A+B}^{A-B} du \frac{R_1(u)}{\sqrt{-(u-r_1)(u-r_2)}} \tag{A.30}$$

$$= \sqrt{\pi}\Big(\sqrt{\pi}\Big(\frac{2r_1^2}{\sqrt{\frac{1}{r_1-p_1}}\sqrt{\frac{p_1-r_2}{r_1-r_2}}} + \frac{2r_1}{\sqrt{\frac{1}{r_1-p_1}}\sqrt{\frac{p_1-r_2}{r_1-r_2}}} - \frac{4r_1(p_1-r_1)\Big(\sqrt{\frac{p_1-r_2}{p_1-r_1}}-1\Big)}{\sqrt{\frac{r_2-p_1}{(p_1-r_1)(r_1-r_2)}}} \tag{A.31}$$

$$- \frac{(p_1-r_1)\Big(r_1\Big(2-3\sqrt{\frac{p_1-r_2}{p_1-r_1}}\Big)+2p_1\Big(\sqrt{\frac{p_1-r_2}{p_1-r_1}}-1\Big)+r_2\sqrt{\frac{p_1-r_2}{p_1-r_1}}\Big)}{\sqrt{\frac{r_2-p_1}{(p_1-r_1)(r_1-r_2)}}} \tag{A.32}$$

$$+ \frac{2}{\sqrt{\frac{1}{r_1-p_1}}\sqrt{\frac{p_1-r_2}{r_1-r_2}}} - \frac{2(p_1-r_1)\Big(\sqrt{\frac{p_1-r_2}{p_1-r_1}}-1\Big)}{\sqrt{\frac{r_2-p_1}{(p_1-r_1)(r_1-r_2)}}}\Big)(2(p_1-p_2)(r_1-p_1))^{-1} \tag{A.33}$$

$$+ \frac{\sqrt{\pi}\Big(\frac{2r_1^2}{\sqrt{\frac{1}{r_1-p_2}}\sqrt{\frac{p_2-r_2}{r_1-r_2}}} + \frac{2r_1}{\sqrt{\frac{1}{r_1-p_2}}\sqrt{\frac{p_2-r_2}{r_1-r_2}}} - \frac{4r_1(p_2-r_1)\Big(\sqrt{\frac{p_2-r_2}{p_2-r_1}}-1\Big)}{\sqrt{\frac{r_2-p_2}{(p_2-r_1)(r_1-r_2)}}}}{2(p_1-p_2)(p_2-r_1)} \tag{A.34}$$

$$+ \frac{-\frac{(p_2-r_1)\Big(r_1\Big(2-3\sqrt{\frac{p_2-r_2}{p_2-r_1}}\Big)+2p_2\Big(\sqrt{\frac{p_2-r_2}{p_2-r_1}}-1\Big)+r_2\sqrt{\frac{p_2-r_2}{p_2-r_1}}\Big)}{\sqrt{\frac{r_2-p_2}{(p_2-r_1)(r_1-r_2)}}} + \frac{2}{\sqrt{\frac{1}{r_1-p_2}}\sqrt{\frac{p_2-r_2}{r_1-r_2}}}}{2(p_1-p_2)(p_2-r_1)} \tag{A.35}$$

$$+ \frac{-\frac{2(p_2-r_1)\Big(\sqrt{\frac{p_2-r_2}{p_2-r_1}}-1\Big)}{\sqrt{\frac{r_2-p_2}{(p_2-r_1)(r_1-r_2)}}}\Big)}{2(p_1-p_2)(p_2-r_1)}\Big) \tag{A.36}$$

$$\times \ (\sqrt{r_2-r_1})^{-1} \tag{A.37}$$

Case 2: p_1 or p_2 are one of the limits of integrations. For example, $p_1 = r_1 = A+B$, the integral is divergent. We can extract its asymptotic expression as lower limit tends to r_1.

$$\begin{aligned}\frac{2}{J}\int_{A+B}^{A-B} du \frac{R_1(u)}{\sqrt{-(u-r_1)(u-r_2)}} &= \frac{2}{J}\int_{r_1}^{r_2} du \frac{(n\lambda-\tau m-\omega)u-\lambda u^2+n\omega}{(u-r_1)(u-p_2)\sqrt{-(u-r_1)(u-r_2)}} \\ &= \frac{2}{J}\int_{r_1}^{r_2} du \frac{(n\lambda-\tau m-\omega)u-\lambda u^2+n\omega}{(u-r_1)^{3/2}(u-p_2)\sqrt{-(u-r_2)}}\end{aligned}$$

The right integral involving $R_2(t^2)$ can be split using partial fractions decomposition.

$$\begin{aligned} R_2(t^2) &= \frac{C_1 t^2}{t^2 - p_1} + \frac{C_2 t^2}{t^2 - p_2} \\ C_1 &= \frac{p_1\big(\lambda m + \tau\big) + m\omega - n\tau}{p_2 - p_1} \\ C_2 &= \frac{-p_2\big(\lambda m + \tau\big) + n\tau - m\omega}{p_2 - p_1} \end{aligned}$$

Where p_1 and p_2 satisfy the relation :

$$\begin{aligned} t^4 - (2n + m^2)t^2 + n^2 &= (t^2 - p_1)(t^2 - p_2) \\ p_1 &= 1/2\Big(2n + m^2 + m\sqrt{m^2 + 4n}\Big) \\ &= r^4 + 2r^2\sqrt{\left(R^2 + Z^2\right)\left(\rho^2 + \zeta^2\right)} + \left(R^2 + Z^2\right)\left(\rho^2 + \zeta^2\right) \\ p_2 &= 1/2\Big(2n + m^2 - m\sqrt{m^2 + 4n}\Big) \\ &= r^4 - 2r^2\sqrt{\left(R^2 + Z^2\right)\left(\rho^2 + \zeta^2\right)} + \left(R^2 + Z^2\right)\left(\rho^2 + \zeta^2\right) \end{aligned}$$

The right integral becomes:

$$\begin{aligned} \int_{\sqrt{A-B}}^{\sqrt{A+B}} dt \frac{R_2(t^2)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} &= \int_{\sqrt{A-B}}^{\sqrt{A+B}} dt \frac{(C_1 + C_2)}{\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \\ &+ C_1 p_1 \int_{\sqrt{A-B}}^{\sqrt{A+B}} \frac{dt}{(t^2 - p_1)\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \\ &+ C_2 p_2 \int_{\sqrt{A-B}}^{\sqrt{A+B}} \frac{dt}{(t^2 - p_2)\sqrt{-(t^2 - r_1)(t^2 - r_2)}} \end{aligned} \tag{A.38}$$

The first integral is a complete elliptic integral of the first kind K. The other two integrals are elliptic integrals of the third kind, that become logarithmically infinite for $t = \sqrt{p_1}$ and

$t = \sqrt{p_2}$ respectively as :

$$\pm \frac{1}{2\sqrt{(p_1^2 - r_1)(p_1^2 - r_2)}} \log(t - \sqrt{p_1})$$
$$\text{and} \quad \pm \frac{1}{2\sqrt{(p_2^2 - r_1)(p_2^2 - r_2)}} \log(t - \sqrt{p_2})$$

and for $t = -\sqrt{p_1}$ and $t = -\sqrt{p_2}$ as :

$$\mp \frac{1}{2\sqrt{(p_1^2 - r_1)(p_1^2 - r_2)}} \log(t + \sqrt{p_1})$$
$$\text{and} \quad \mp \frac{1}{2\sqrt{(p_2^2 - r_1)(p_2^2 - r_2)}} \log(t + \sqrt{p_2})$$

The poles p_1 and p_2 become one of the the limits of integration, so there are four possibilities (only two possibilities if we consider that $R > 0$ and $\rho > 0$).

$$\begin{aligned}
p_1 &= A - B \quad \text{if } R = \rho \text{ and } Z = -\zeta \\
p_1 &= A + B \quad \text{if } R = -\rho \text{ and } Z = -\zeta \\
p_2 &= A + B \quad \text{if } (R = \rho \text{ and } Z = \zeta) \Rightarrow \quad p_2 = b^2 = (r^2 - R^2 - Z^2)^2 \\
p_2 &= A - B \quad \text{if } R = -\rho \text{ and } Z = \zeta
\end{aligned}$$

In each case, $P(p) = 0$, where $p = \sqrt{p_1}$ or $p = \sqrt{p_2}$, which means the integral is elliptic of the second kind and has an infinite of (1/2) order. Therefore the last two integrals from Equation (15) can be divided into three cases . Case 1: Let us consider the case when $\sqrt{p_1}$ (or $\sqrt{p_2}$) is not inside the domain of integration $\left[\sqrt{A+B}, \sqrt{A-B}\right]$. Let $a^2 = A - B = r^4 - 2r^2\zeta Z + (R^2+Z^2)(\rho^2+\zeta^2) + 2r^2 R\rho$ and $b^2 = A+B = r^4 - 2r^2\zeta Z + (R^2+Z^2)(\rho^2+\zeta^2) - 2r^2 R\rho$, and $b < a$. In this case, the solution is an elliptic integral of the third kind and the integral is given by integrals of the form:

$$C\int_b^a \frac{t^2dt}{(t^2-p)\sqrt{(a^2-t^2)(t^2-b^2)}} = \frac{b^2 g}{b^2-p}\Pi\big(\alpha_1^2, k\big) \tag{A.39}$$

Where $g = 1/(\sqrt{A-B})$, $k^2 = (a^2-b^2)/a^2 = -2B/(A-B)$, and $\alpha_1^2 = \frac{p(a^2-b^2)}{a^2(p-b^2)}$, with $p \neq b^2$. Refer to table entry number (217.02).

Case 2: This case involves the case when $\sqrt{p_1}$ (or $\sqrt{p_2}$) is equal to the lower limit of integration. Then solution is an elliptic integral of the second kind with alegraic infinite of one-half order at point $t = \pm\sqrt{p_1}$ (or p_2).

$$\begin{aligned}\int_b^a \frac{t^2dt}{(t^2-b^2)\sqrt{(a^2-t^2)(t^2-b^2)}} &= \int_b^y \frac{t^2dt}{(t^2-b^2)\sqrt{(a^2-t^2)(t^2-b^2)}} \\ &\quad + \int_y^a \frac{t^2dt}{(t^2-b^2)\sqrt{(a^2-t^2)(t^2-b^2)}}\end{aligned}$$

The second integral can is given by Table entry number (218.07). The first integral has a divergence at $t = b$. And so we can extract its asymptotic expression as lower limit tends to b.

$$\int_b^y \frac{t^2dt}{(t^2-b^2)\sqrt{(a^2-t^2)(t^2-b^2)}} \tag{A.40}$$

$$= \int_b^y \frac{t^2dt}{(t-b)(t+b)\sqrt{(a^2-t^2)(t-b)(t+b)}} \tag{A.41}$$

$$= \int_b^y \frac{t^2dt}{(t-b)^{3/2}(t+b)\sqrt{(a^2-t^2)(t+b)}} \tag{A.42}$$

$$= \int_b^y \frac{F(t)dt}{(t-b)^{3/2}} \tag{A.43}$$

$$= \lim_{\epsilon\to b}\int_\epsilon^y \frac{F(t)dt}{(t-b)^{3/2}} \tag{A.44}$$

Since $F(t)$ is differentiable in interval, so by the M.V.T. , for every t in the interval we have a point b_t such that:

$$
\begin{aligned}
F'(b_t) &= \frac{F(t)-F(b)}{t-b} && \text{(A.45)}\\
F(t) &= F(b)+F'(b_t)(t-b) && \text{(A.46)}\\
&\int_\epsilon^y \frac{F(t)dt}{(t-b)^{3/2}} && \text{(A.47)}\\
&= \int_\epsilon^y \frac{F(b)dt}{(t-b)^{3/2}} + \int_\epsilon^y \frac{F'(b_t)(t-b)dt}{(t-b)^{3/2}} && \text{(A.48)}\\
&\sim \frac{2F(b)}{\big(\epsilon-b\big)^{1/2}} \text{ as } \epsilon \to b^+ && \text{(A.49)}
\end{aligned}
$$

Because the left integral is bounded as $\epsilon \to b^+$

$$
\Big| \int_\epsilon^y \frac{F'(b_t)dt}{(t-b)^{1/2}} \Big| \le K \text{ as } \epsilon \to b^+ \qquad \text{(A.50)}
$$

Case 3: This case involves the case when $\sqrt{p_1}$ (or $\sqrt{p_2}$) is equal to the upper limit of integration. Then solution is an elliptic integral of the second kind with alegraic infinite of one-half order at point $t = \pm\sqrt{p_1}$ (or p_2).

$$
\begin{aligned}
\int_b^a \frac{t^2dt}{(t^2-a^2)\sqrt{(a^2-t^2)(t^2-b^2)}} &= \int_b^y \frac{t^2dt}{(t^2-a^2)\sqrt{(a^2-t^2)(t^2-b^2)}}\\
&\quad + \int_y^a \frac{t^2dt}{(t^2-a^2)\sqrt{(a^2-t^2)(t^2-b^2)}}
\end{aligned}
$$

The first integral is given by table entry number (217.06). The second integral has an infinite that can be extracted similarly as in case 2.

$$\int_y^a \frac{t^2 dt}{(t^2-a^2)\sqrt{(a^2-t^2)(t^2-b^2)}} \tag{A.51}$$

$$= \int_y^a \frac{t^2 dt}{(t-a)^{3/2}(t+a)\sqrt{(t+a)(b^2-t^2)}} \tag{A.52}$$

$$= \int_y^a \frac{G(t) dt}{(t-a)^{3/2}} \tag{A.53}$$

$$\sim \frac{-2G(a)}{(\epsilon-a)^{1/2}} \text{ as } \epsilon \to a^- \tag{A.54}$$

To summarize, the integral can be rewriten as

$$\int_b^a dt \frac{R(t)}{\sqrt{-(t^2-a^2)(t^2-b^2)}} \tag{A.55}$$

$$\text{where } R(t) = \frac{\lambda t^3 + \tau t^2 + \omega t}{mt - t^2 + n} \tag{A.56}$$

$$= \int_b^a dt \frac{tR_1(t^2)}{\sqrt{-(t^2-a^2)(t^2-b^2)}} + \int_b^a dt \frac{R_2(t^2)}{\sqrt{-(t^2-a^2)(t^2-b^2)}} \tag{A.57}$$

$$= \int_b^a \frac{F(t)\,dt}{(t^2-p_1)(t^2-p_2)\sqrt{-(t^2-a^2)(t^2-b^2)}} + \int_b^a \frac{G(t)\,dt}{(t^2-p_1)(t^2-p_2)\sqrt{-(t^2-a^2)(t^2-b^2)}}$$

$$= f(a,b,p_1,p_2) + C_1 \frac{b^2 g}{b^2-p_1}\Pi(\alpha_1^2, k^2) + C_2 \frac{b^2 g}{b^2-p_2}\Pi(\alpha_2^2, k^2)$$

$$\sim \frac{\Big(F(b)+G(b)\Big)}{(b^2-p_1)b\sqrt{2b(a^2-b^2)}} \frac{1}{\big(\epsilon-b\big)^{1/2}} \text{ as } \epsilon \to b^+$$

We perform asymptotics of the solution as $p_2 \to b^2$, which means $\rho \to R$ when $\zeta = Z$, to check for integrability and regularize

Where each variable is given below as a function of the spatial variables ρ, ζ, Z, R .

$$
\begin{aligned}
b^2 &= r^4 - 2r^2R\rho - 2r^2Z\zeta + (R^2+Z^2)(\rho^2+\zeta^2) \\
p_2 &= r^4 - 2r^2\sqrt{(R^2+Z^2)(\rho^2+\zeta^2)} + (R^2+Z^2)(\rho^2+\zeta^2) \\
b^2 - p_2 &= -2r^2R\rho - 2r^2Z\zeta + 2r^2\sqrt{(R^2+Z^2)(\rho^2+\zeta^2)} \\
g &= (r^4 + 2r^2R\rho - 2r^2Z\zeta + (R^2+Z^2)(\rho^2+\zeta^2))^{-1} \\
\alpha_2^2 &= \frac{-2R\rho(r^4-2r^2\sqrt{(R^2+Z^2)(\rho^2+\zeta^2)}+(R^2+Z^2)(\rho^2+\zeta^2))}{(r^4+2r^2(R\rho-Z\zeta)(R^2+Z^2)(\rho^2+\zeta^2))(R\rho+Z\zeta-\sqrt{(R^2+Z^2)(\rho^2+\zeta^2)})} \\
k^2 &= \frac{4r^2R\rho}{r^4+2r^2R\rho-2r^2Z\zeta+(R^2+Z^2)(\rho^2+\zeta^2)}
\end{aligned}
$$

Consider the expression below that has a singularity when $p_2 \to b^2$

$$C_2 \frac{b^2 g}{b^2 - p_2} \Pi(\alpha_2^2, k^2) \tag{A.58}$$

$$p_2 \quad \to b^2 \text{ as } \rho \to R \text{ and } \zeta \to Z \tag{A.59}$$

$$\frac{b^2 g}{b^2 - p_2} \quad \to \infty \text{ as } \rho \to R \text{ and } \zeta \to Z \tag{A.60}$$

$$\alpha_2 \quad \to \infty \text{ as } \rho \to R \text{ and } \zeta \to Z \tag{A.61}$$

$$C_2 \quad \to \text{constant as } \rho \to R \text{ and } \zeta \to Z \tag{A.62}$$

In order to obtain the asymptotics of the full expression, we should consider how $\Pi(n,m) \to 0$ as $n \to \infty$ Note that for $n > 1$ the integral has a singularity at $\sin^2\theta = 1/\sqrt{n}$ and the elliptic integral of the third kind should be interpreted as a a Cauchy principal value integral. [Reference: Gil et al., Numerical Methods for Special Functions]

$$\Pi(n,m) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{(1-n\sin^2\theta)\sqrt{1-m\sin^2\theta}} \tag{A.63}$$

Definition of variables in this section

$$Y = \rho^2 + \zeta^2 \qquad X = R^2 + Z^2$$

$$B = -2r^2 R\rho \qquad A = r^4 - 2r^2\zeta Z + XY$$

$$C = r^2\zeta \qquad D = C/Y$$

$$\alpha = 3r^4(Z - D)^2 \qquad \beta = (3(Z - D)C - 3r^4)/Y$$

$$\gamma = 3r^2/Y \qquad E = -r^4 + r^2 Z\zeta$$

$$F = r^5/Y^{5/2} \qquad G = r^3 E/Y^{5/2}$$

$$H = -Br^3/(2Y^{5/2}) \qquad J = G/2E$$

$$m = 2r^2 \qquad n = 2E + A$$

$$\tau = \alpha\tfrac{n+m^2}{n^3} \qquad \omega = \alpha(-m/n^2 - m/n^3(n + m^2)) - \beta(\frac{m^2 + n}{mn}) + \gamma$$

$$\lambda = \beta\tfrac{1}{mn}$$

$$C_1 = \tfrac{p_1\left(\lambda m+\tau\right)+m\omega-n\tau}{p_2-p_1}$$

$$C_2 = \tfrac{-p_2\left(\lambda m+\tau\right)+n\tau-m\omega}{p_2-p_1}$$

APPENDIX B

HOLDING CURVE

The equations of motion for a sphere falling in sharply stratified fluid in low Re numbers are:

$$\begin{aligned}\frac{dY_3}{dt}(t;\rho) = (6\pi A\mu K)^{-1}\Bigg(m_s g - g\int_{\Omega_s}\rho_0(x_3+Y_3(t;\rho))d\Omega_s + \\ + \int_{\Omega_f} G(\boldsymbol{y},t)\frac{A\hat{g}}{4}\left\{\frac{3\left(r^2+y_3^2\right)}{r^3}+\frac{A^2\left(r^2-3y_3^2\right)}{r^5}\right\}d\Omega_f\Bigg),\end{aligned} \tag{B.1}$$

$$\frac{\partial\rho}{\partial t}(\boldsymbol{x},t) + (\boldsymbol{u}(\boldsymbol{x},t;V)+\boldsymbol{w}(\boldsymbol{x},t;\rho))\cdot\nabla\rho(\boldsymbol{x},t) = 0.$$

Let us define a *holding curve* if there exists an interface shape that makes the sphere stop and never recover. This implies the force on the sphere F_s and the sphere velocity vanishes, making the stokes velocity $\boldsymbol{u}_s = 0$. Therefore, $\partial\rho/\partial t = 0$ and the advection of the fluid is solely dictated by the perturbation velocity.

$$F_s = 0 \tag{B.2}$$

$$\boldsymbol{w}\cdot\nabla\rho = 0 \tag{B.3}$$

For equation (B.2) to be satisfied, the perturbation velocity $\boldsymbol{w} = 0$ (Case 1) or $\boldsymbol{w}$ must be tangential to the curve of ρ (Case 2).If $w \neq 0$ and it is not tangent to the curve, then ρ evolves and there is no holding curve. When the interface is away from the sphere, we can show that a holding curve does not exist.

Case 1 ($\boldsymbol{w}$=0)

If $w = 0$, the only force that holds the sphere is pressure, and the momentum equation becomes

$$\nabla p + (\rho - \rho_0)\,\hat{g} = 0. \tag{B.4}$$

Equation (B.4) has no solution unless $\rho(\boldsymbol{x}) - \rho_0(x_3)) = f(x_3)$. In other words,

$$\begin{aligned} \nabla \times (-\nabla p) &= \nabla \times (\rho - \rho_0)\hat{g} && \text{(B.5)} \\ 0 &= g\left(\hat{i}\frac{\partial(\rho - \rho_0)}{\partial x_1} - \hat{j}\frac{\partial(\rho - \rho_0)}{\partial x_2}\right) && \text{(B.6)} \end{aligned}$$

Therefore if $\boldsymbol{w} = 0$, then $\rho(x_3)$ and there is no deformation, so there is no holding curve.

Case 2 ($\boldsymbol{w}$ tangent to ρ)

If there is a holding curve, there is no evolution of ρ, then the interface becomes a stream function. Therefore $\boldsymbol{w}$ is generated by stream functions which means $\boldsymbol{w}$ would have to be tangent to this curve. However, if ρ doesnt evolve and the curve is a stream line, we show below that $\boldsymbol{w}$ must be equal to zero, reaching a contradiction and disproving the existence of a holding curve.

From the equation of the advection of the fluid, a holding curve must satisfy:

$$\nabla \cdot (\rho\,\boldsymbol{w}) = 0 \tag{B.7}$$

Assuming the above is true and that it defines a holding curve, then the stream function is constant on such curve and the holding curve is a stream line. Looking at the cross section $x_2 = 0$,

$$\begin{aligned} \nabla \cdot (\rho\,\boldsymbol{w}) &= 0 && \text{(B.8)} \\ \rho\, w_1 &= \frac{\partial \psi}{\partial x_3} && \text{(B.9)} \\ \rho\, w_3 &= -\frac{\partial \psi}{\partial x_1} && \text{(B.10)} \end{aligned}$$

Let $x_3 = \zeta(x_1)$ denote the curve where ψ is a constant. We show that if the holding curve is a graph $\zeta(x_1)$, then it cannot exists. Integrating the momentum equation over the fluid domain, we get

$$\begin{aligned}\int_{\Omega_f} |\nabla w|^2 d_3x &= 0\\ \text{third component}\\ \int_{\Omega_f} \rho\, w_3\, d_3x &= \int_{-\infty}^{\infty} dx_1 \int_0^{\zeta(x_1)} dx_3 \frac{\partial \psi}{\partial x_1}\\ &= \int_{-\infty}^{\infty} dx_1 \left(\frac{\partial}{\partial x_1} \left(\int_0^{\zeta(x_1)} dx_3 \psi(x_1, x_3) \right) - \psi\left(x_1, \zeta(x_1)\right) \frac{\partial \zeta}{\partial x_1} \right)\\ &= \int_0^{\zeta(\infty)} \psi(\infty, x_3) dx_3 - \int_0^{\zeta(-\infty)} \psi(\infty, x_3) dx_3\\ &= \psi(\infty, \zeta(\infty))\zeta(\infty) + \psi(-\infty, \zeta(-\infty))\psi(-\infty)\\ &= 0\end{aligned}$$

since

$$\frac{\partial \zeta}{\partial x_1} = \frac{\partial \psi / \partial x_1}{\partial \psi / \partial x_3} \tag{B.11}$$

$$\int_{-\infty}^{\infty} dx_1 \psi\left(x_1, \zeta(x_1)\right) \frac{\partial \zeta}{\partial x_1} = 0 \tag{B.12}$$

Translating this result to cylindrical coordinates

$$\nabla \cdot \phi = \frac{1}{r}\frac{\partial}{\partial r}(r\phi_{(r)}) + \frac{\partial}{\partial z}(\phi_{(z)}) = 0 \tag{B.13}$$

$$\phi_{(r)} = \frac{1}{r}\frac{\partial \psi}{\partial z} \tag{B.14}$$

$$\phi_{(z)} = \frac{1}{r}\frac{\partial \psi}{\partial r} \tag{B.15}$$

$$\tag{B.16}$$

$$\int_{\Omega_f} |\nabla w|^2 d_3 y \tag{B.17}$$

$$= 2\pi \int_0^\infty r dr \int_0^{\zeta(r)} dz \frac{1}{r} \frac{\partial \psi}{\partial r} \tag{B.18}$$

$$= 2\pi \int_0^\infty dr \int_0^{\zeta(r)} dz \frac{\partial \psi}{\partial r} \tag{B.19}$$

$$= 2\pi \int_0^\infty dr \frac{\partial}{\partial r} \left(\int_0^{\zeta(r)} dz \psi - \psi(r, \zeta)\zeta(r) \right) \tag{B.20}$$

$$= \psi(0, \zeta(0)) \tag{B.21}$$

$$= constant \tag{B.22}$$

By this argument, if the holding curve exits, the perturbation velocity would have to be a constant, and by boundary conditions this constant is $\boldsymbol{w} = 0$, therefore we reach a contradiction.

APPENDIX C

FLOW PAST A STOKESLET

The equations of motion for Stokes flow past a point force singularity are

$$\mu\nabla^2\boldsymbol{u} - \nabla p = -\boldsymbol{A}\,\delta(x,y,z) \tag{C.1}$$

$$\nabla \cdot u = 0 \tag{C.2}$$

$$\frac{\partial\rho}{\partial t} + \boldsymbol{u}\cdot\nabla\rho = 0 \tag{C.3}$$

$$\rho(t=0) = H(z-z_0) \tag{C.4}$$

Where z_0 is the position of the interface. The solutions to the equations is the stokeslet solution, given in terms of the strength $\boldsymbol{A} = (0,0,-1)$

$$\boldsymbol{u} = \frac{1}{8\pi\mu}\left(\frac{\boldsymbol{A}}{|\boldsymbol{x}|} + \frac{(\boldsymbol{A}\cdot\boldsymbol{x})\,\boldsymbol{x}}{|\boldsymbol{x}|^3}\right) \tag{C.5}$$

We need the stream-surface for stokeslet.

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u}(x,y,z,t) \tag{C.6}$$

$$\boldsymbol{x}\Big|_{t=0} = (x_0, y, z_0) \tag{C.7}$$

$$\tag{C.8}$$

Characteristics

$$
\begin{aligned}
\frac{dx}{dt} &= u_1 = \frac{1}{8\pi\mu}\left(\frac{-zx}{(x^2+y^2+z^2)^{3/2}}\right) \; ; x(t=0) = x_0 \\
\frac{dy}{dt} &= u_2 = \frac{1}{8\pi\mu}\left(\frac{-zy}{(x^2+y^2+z^2)^{3/2}}\right) \; ; y(t=0) = y_0 \\
\frac{dz}{dt} &= u_3 = \frac{1}{8\pi\mu}\left(\frac{-1}{(x^2+y^2+z^2)^{1/2}} + \frac{-zx}{(x^2+y^2+z^2)^{3/2}}\right) \\
&= \frac{1}{8\pi\mu}\left(\frac{-x^2-2z^2}{(x^2+y^2+z^2)^{3/2}}\right) \; ; \; z(t=0) = z_0
\end{aligned}
$$

Simplified Example :

For these notes, lets make $z_0 = 1$ and work in the y-plane, $y = 0$.

$$
\frac{dz}{dx} = \frac{x}{z} + \frac{2z}{x} \tag{C.9}
$$

Let $U = \frac{x}{z}$.

$$
\begin{aligned}
\frac{dz}{dx} &= U + \frac{2}{U} = \frac{d}{dx}\left(\frac{x}{U}\right) = \frac{1}{U} - \frac{x}{U^2}\frac{dU}{dx} && \text{(C.10)} \\
\frac{x}{U^2}\frac{dU}{dx} &= \frac{1}{U} - U - \frac{2}{U} && \text{(C.11)} \\
\frac{dU}{U^3 - U} &= \frac{dx}{x} && \text{(C.12)}
\end{aligned}
$$

Integrating both sides, we get

$$
\frac{U}{\sqrt{U^2+1}} = \frac{C}{x} \tag{C.13}
$$

Substituting $U = x/z$ gives

$$\frac{x/z}{\sqrt{(x/z)^2+1}} = \frac{C}{x} \tag{C.14}$$

$$z^2(x,C) = \frac{x^4 - C^2x^2}{C^2} \tag{C.15}$$

$$z^2(x,C) = \frac{x^4}{C^2} - x^2 \tag{C.16}$$

$$z^2 + x^2 = \frac{x^4}{C^2}. \tag{C.17}$$

Applying initial conditions $z(x(t=0)) = z(x_0) = z_0 = 1$, we get

$$C = \frac{x_0^2}{\sqrt{x_0^2 + z_0^2}} \tag{C.18}$$

$$\frac{dx}{dt} = \frac{-\left(\frac{\sqrt{x^4 - C^2x^2}}{|C|}\right)x}{\left(\frac{x^4}{C^2}\right)^{3/2}} \tag{C.19}$$

$$= -\frac{c^2\sqrt{x^2 - c^2}}{x^4} \tag{C.20}$$

$$dt = \frac{x^4}{-c^2\sqrt{x^2 - c^2}}dx. \tag{C.21}$$

APPENDIX D

DIFFUSION COEFFICIENT IN CORN SYRUP

Previous experimental measurements of diffusion coefficient of both NaCl and KI have been reported in [22] and [28]. However, [22] did so in a corn syrup solution heavily diluted by water and [28] made his recordings outside of a temperature bath.

D.1 Sodium Chloride NaCl

In [22], Lin states that for water-salt solutions, the diffusivity of salt (as well as most ionic solutes) is $1.5 \times 10^{-5}\text{cm}^2/\text{s}$ (see also, e.g., [32]), and there seems to be no reason that this diffusivity would provide faster rates in our viscous corn syrup solution. However, we have observed some fingering instabilities and the development of plumes around the interface when the concentrations of NaCl are greater than 2%. Furthermore, Moore measured the conductivity of salt obtaining a $1.3 \times 10^{-5}\text{cm}^2/\text{s}$ outside of a temperature controlled bath [28] .

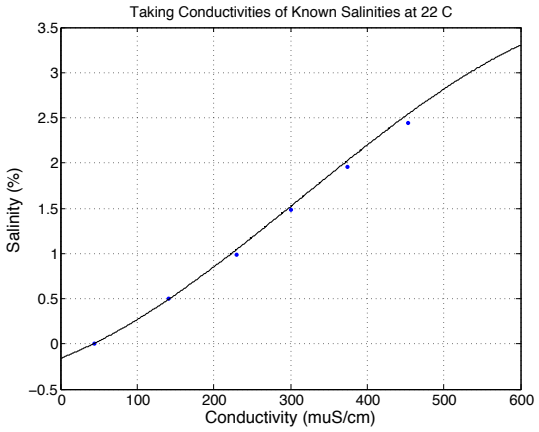


Figure D.1: Plot of percent salinity concentration vs. conductivity. The dots are measurements at 22^o C for different concentrations of NaCl using an Orion conductivity meter and probe. The black solid line is a cubic fit to the data providing a map from conductivity to salinity.

Using the function fit shown in Figure (D.3), we track the concentration of salinity given by the conductivity probe.

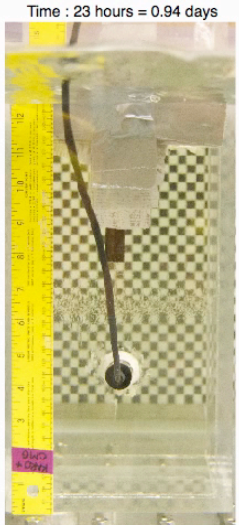


Figure D.2: Picture of the diffusion coefficient measurement set up inside the temperature bath

D.2 Potassium Iodide KI

Lin tested the diffusion of potassium iodide using its natural coloring behavior, in water diluted corn syrup obtaining an upper bound for the diffusion coefficient of the same magnitude as that of NaCl [22]. However, when we stratify using KI and non-diluted corn syrup, the sharp stratification is maintained for longer than three weeks.

A future experimental result includes an ongoing measurement comparing KI salinity in pure corn syrup and 1% NaCl in a temperature bath setting. So far, the values indicate that KI has a diffusion coefficient at least one magnitude smaller than NaCl. Therefore, comparing the entrainment regimes using NaCl and KI as the stratification agents would provide a setting in which diffusion could play a role (NaCl) and a case in which our current non-diffusive theory would agree due to the slow time scales of diffusion (KI).

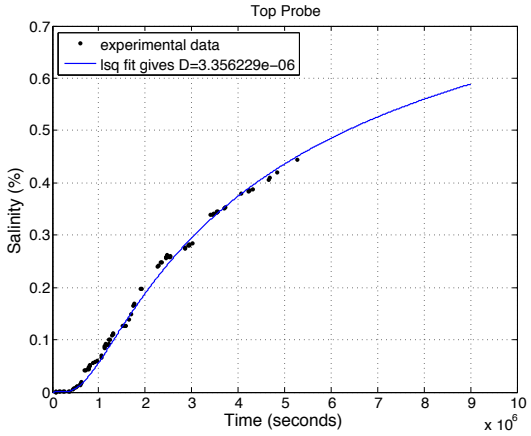


Figure D.3: Experimental measurements at a fixed location in the top layer of salinity based on the salinity-to-conductivity fit (black dots) and the solution to the diffusion equation (blue line) with D, the diffusion coefficient, chosen to best fit the measurements. Conductivity was measured using the top layer probe shown in Figure (D.2)

APPENDIX E

NOMENCLATURE

Consistent with absolute units, we have listed most of the global variables used in this thesis.

A : radius of sphere

R_0 : radius of the cylindrical tank

g : gravity acceleration constant

ρ_s : density of the sphere

ρ_t : density of the top layer

ρ_b : density of the bottom layer

μ_t : viscosity of the top layer

μ_b : viscosity of the bottom layer

$\boldsymbol{x} = (x_1, x_2, x_3) = (R, \phi, Z)$: observation point in rectangular and cylindrical coordinates respectively

$\boldsymbol{y} = (y_1, y_2, y_3) = (\rho, \theta, \zeta)$: location of stokeslet for the perturvation velocity $\boldsymbol{w}$ in rectangular and cylindrical coordinates respectively

$Y(t)$: vertical position of the sphere positon

$V(t)$: third component of the sphere velocity

$\boldsymbol{u}_s = (u_R, 0, u_Z)$: stokes flow component for static density

$\boldsymbol{u}^{(i)}$: stokes flow i$^{\text{th}}$ reflection from method of reflections

W_j : Greens function from Oseen Tensor to solve for $\boldsymbol{w}$

$\boldsymbol{w} = (w_R, 0, w_Z)$: density anomaly flow component Ω_f : fluid domain of integration Ω_s : sphere interior domain

APPENDIX F

NUMERICAL CODES

This appendix shows the fortran code developed by [22] that was modified to implement further refinements to the full flow field and tracking of interface.

```
MODULE globalinfo


!experimental parameters
real (kind=8), parameter :: rhot           =1.34647 !density of top fluid
real (kind=8), parameter :: rhob           = 1.35000 !density of bottom fluid
real (kind=8), parameter :: rhos           = 1.36712 !density of the sphere
real (kind=8), parameter :: mu      = 5.0698 !dynamic viscosity of fluid
real (kind=8) :: U                         = 0.0; !initial velocity of fluid
real (kind=8), parameter :: y0      = -13 !!initial position of interface
real (kind=8), parameter :: R       = 0.635; !radius of sphere
real (kind=8), parameter :: R0      = 5.4; !cylinder radius
real (kind=8), parameter :: maxTime  = 6000.0; !time to run to
real (kind=8), parameter :: g        = 981.0; !gravity
real (kind=8), parameter :: pi       = 3.14159265;


!numerical parameters
real (kind=8), parameter :: dt= 1; !time step
real (kind =8), parameter :: numtrapz= 0.005; !h for trapezoidal trapz1trapz1
real (kind=8), parameter :: integthres= 0.1E-5; !simpson integration
real (kind=8), parameter :: singthres= 0.1E-6; !threshold singularity x=y
real (kind=8), parameter :: logsing= 0.1; !threshold log "sing" x=y
real (kind=8), parameter :: logtrapzbig= 0.5;!h trapz for log around "sing"
real (kind=8), parameter :: logtrapz= 0.1; ! htrapz for log
real (kind = 8 ) :: FFR =2*R ! radius of full solution
```

```
real (kind =8) ,parameter:: dx= 0.05;
integer (kind=4) :: RorZ
real (kind = 8 ) :: R0cl = 2*R
integer (kind=4),parameter :: XNfar=20, XNclose=60
integer (kind=4) :: XN = XNclose+XNfar
!integer (kind=4) :: RorZ, XN=ceiling(R0/dx)

!dependent parameters
real (kind=8) :: ms= 4.0/3.0*pi*R**3*rhos;  !mass of the sphere
real (kind=8) :: oneoversixpiamuK
= (1-2.10444*(R/R0) +2.08877*((R/R0)**3))* 1.0/(6.0*pi*R*mu)
real (kind=8) :: stresspertcoeff
= -0.25*g*(rhot-rhob)*R*2.0*pi; !coefficient of the perturbatio stress
real (kind=8) :: buoyancytop= -4.0/3.0*pi*R**3*g*rhot; !buoyant force when sphere
     is above the interface
real (kind=8) :: buoyancybottom = -4.0/3.0*pi*R**3*g*rhob; !buoyant force when
    sphere is below the interface
real (kind=8) :: buoyancyCoeff1= -pi*g/3.0*(rhob-rhot); ! buoyant force coeff at
    interface
real (kind=8) :: buoyancyCoeff2= -2.0*pi*g/3.0*R**3*(rhob+rhot); ! buoyant force
    coeff at interface
real (kind=8) :: drhogover8mu= (rhob-rhot)*g/(8.0*mu); !coefficient for the
    perturbation flow
real (kind=8) :: myt= 0 !initial time

!interface
real (kind=8), dimension(:), allocatable :: x, y, sx, sy, su ,sv ,wu ,wv,
    cinterpx, cinterpy,cwinterpx, cwinterpy
real (kind=8) yend, xflagb, xflagl, px, py, myrho, myzeta
```

```
        integer (kind=4) :: flagb, flagu, flagl,flagt, cinternalcount,
           cwinternalcount

!fourier components
real (kind=8), parameter :: epsilon= 0.001
real (kind=8), parameter ::  LZ= 10.0; !u2Z lambda integral limit
real (kind=8), parameter ::  LR= 10.0; !u2R lambda integral limit
integer (kind=4), parameter :: NZ= 2**16; !discretization
integer (kind=4), parameter :: NR= 2**14;
real (kind=8), parameter :: u3coeff= -2.1044428*R/R0+2.1800173*R**3/R0**3;
real (kind=8), parameter :: upperZ= 40.0;
integer (kind=4), parameter :: upperRangeZ  = ceiling((LZ-epsilon)*upperZ/(2.0*pi
   )+1.0),&
upperRangeR    = ceiling(LR*upperZ/(2.0*pi)+1.0);
real (kind=8) :: cylindervelR(upperRangeR, XNclose+XNfar+1), cylindervelZ(
   upperRangeZ, XNclose+XNfar+1)
real (kind=8) :: zcoordinateZ(upperRangeZ), zcoordinateR(upperRangeR)
complex, parameter :: MINUS_ONE                              = -1.0
complex :: imagi                                                    = SQRT(MINUS_ONE)

real (kind=8) WZ(NZ), WR(NR), myHZ(NZ), myGZ(NZ),&
myHR(NR), myGR(NR), firstpartZ(NZ), firstpartR(NR), myk(NZ)
real (kind=8) AreaReflux,AreaSpherePortion,AreaEntrain, startx(XNclose+XNfar+1),
   starty(XNclose+XNfar+1),wforceE, wforceR, wforce, ArchBouyancy,ArchBE,
   stresspert, stresspertA,ArchBR, stresspertE,stresspertReflux

END MODULE globalinfo

program fulltime_2009_05_02_Fortran
```

```
        use globalinfo
        implicit none
        integer (kind=4) i, ierr, flag, ix, iy, iv
        real (kind=8) :: velocity(ceiling(maxTime/dt)+1), stresspertvect(ceiling(
           maxTime/dt)+1), index, abserr=0.001,&
    relerr=0.001
        real (kind=8), dimension(:), allocatable :: V, VP
Character(len=65) :: filename
        external rhoode

!initialize interface
ALLOCATE(x(XNfar+XNclose+1), STAT=ierr)
!ALLOCATE(x(XN+1), STAT=ierr)
        IF (ierr /= 0) PRINT*, "x : Allocation failed"

ALLOCATE(y(XNfar +XNclose+1), STAT=ierr)
!ALLOCATE(y(XN+1), STAT=ierr)
        IF (ierr /= 0) PRINT*, "y : Allocation failed"

DO i=1,XNclose + XNfar+1
if (i<XNclose+2) then
x (i) = (i-1)**(2) *(R0cl/XNclose**(2))
else
x( i ) = (i - XNclose-1)* (R0 -R0cl ) / XNfar + R0cl
endif
end do

y = y0
startx= x;
```

```
starty = y;



!initialize interface interpolated spherevel
ALLOCATE(cinterpx(1000000), STAT=ierr)
!ALLOCATE(x(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpx : Allocation failed"

ALLOCATE(cinterpy(1000000), STAT=ierr)
!ALLOCATE(y(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpy : Allocation failed"

!initialize interface interpolated from w
ALLOCATE(cwinterpx(1000000), STAT=ierr)
!ALLOCATE(x(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpx : Allocation failed"

ALLOCATE(cwinterpy(1000000), STAT=ierr)
!ALLOCATE(y(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpy : Allocation failed"

cinternalcount=XN+1;
cwinternalcount=XN+1;

do ix=1, max(cinternalcount,cwinternalcount)
cinterpx(ix)=x(ix);
cinterpy(ix)= y(ix);
cwinterpx(ix)=x(ix);
cwinterpy(ix)= y(ix);
```

```
end do

open (unit =9,file = 'VolumeTrack.dat')
write(9,*) "Volume of Entrainment, Volume of Reflux, Volume of Portion of Sphere"

open (unit =8,file = 'WForce.dat')
write(8,*) " ArchBER, Wforce = 6pimuAstresscoeff(wFE - wFR), SphARchBoyancy,
    ArchBR,wforceR,wforceE"

!initialize cylinder velocity
call cylindervelinit()

!********************** TIME LOOP ******************!
       do index = 0,ceiling(maxTime/dt)
              ALLOCATE(V(2*(XN+1)), STAT=ierr)
              IF (ierr /= 0) PRINT*, "V : Allocation failed"

              ALLOCATE(VP(2*(XN+1)), STAT=ierr)
              IF (ierr /= 0) PRINT*, "VP : Allocation failed"

!initialize stokes flow
ALLOCATE(su(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "x : Allocation failed"

ALLOCATE(sv(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "y : Allocation failed"

su = 0
sv = 0
```

```
!initialize w flow
ALLOCATE(wu(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "x : Allocation failed"

ALLOCATE(wv(XN+1), STAT=ierr)
IF (ierr /= 0) PRINT*, "y : Allocation failed"

wu = 0
wv = 0

V(1:XN+1) = x
V(XN+2:2*(XN+1)) = y

!====================== Write interface ====================== !
WRITE (filename, fmt='(a,f10.2,a)') 'interface',index+1,'.dat'

open (unit =2,file = filename,form='formatted')
write(2,*) "x,y at time=", myt, "rhos", rhos

do ix=1, XN+1
write(2,*), x(ix), ",", y(ix),","
end do
!====================== Write interpolated interface ================= !
WRITE (filename, fmt='(a,f10.2,a)') 'interpolation',index +1,'.dat'
!print *, "after writing filename"

open (unit =6,file = filename,form='formatted')
!open (unit =6,file = 'interpolation.dat',form='formatted')
```

```
!print *, 'before loop ', max(cinternalcount,cwinternalcount)
write(6,*) "interpolated interface x,y at time=", myt

do ix=1, max(cinternalcount,cwinternalcount)
write(6,*), cinterpx(ix), ",", cinterpy(ix),",", cwinterpx(ix), ",", cwinterpy(ix
   ),","
end do

!====================== ODE SOLVER ====================== !
flag = 1
call r8_rkf45 (rhoode, 2*(XN+1), V, VP, index*dt, (index+1.0)*dt, relerr, abserr,
    flag )

              x = V(1:XN+1)
              y = V(XN+2:2*(XN+1))

! Write data files

WRITE (filename, fmt='(a,f10.2,a)') 'stokes',index+1,'.dat'

open (unit =4,file = filename,form='formatted')
write(4,*) "us,sv at time=", myt

do ix=1, XN+1
write(4,*), su(ix), ",", sv(ix), ","
end do

WRITE (filename, fmt='(a,f10.2,a)') 'wpert',index+1,'.dat'
```

```
open (unit =5,file = filename,form='formatted')
write(5,*) "wu,wv at time=", myt

do ix=1, XN+1
write(5,*), wu(ix), ",", wv(ix), ","
end do

open (unit =9,file = 'VolumeTrack.dat')
write(9,*), AreaEntrain, ",", AreaReflux, ",", AreaSpherePortion

open (unit =8,file = 'WForce.dat')
write(8,*), ArchBE, ",", wforce, ",", ArchBouyancy, ",", ArchBR,",",wforceR,",",
    wforceE

open (unit =11,file = 'sphereVel.dat')
write(11,*), U, ",", myt, ",", yend

IF (ALLOCATED(V)) DEALLOCATE(V,STAT=ierr)
IF (ALLOCATED(VP)) DEALLOCATE(VP,STAT=ierr)

IF (ALLOCATED(wu)) DEALLOCATE(wu,STAT=ierr)
IF (ALLOCATED(wv)) DEALLOCATE(wv,STAT=ierr)

IF (ALLOCATED(su)) DEALLOCATE(su,STAT=ierr)
IF (ALLOCATED(sv)) DEALLOCATE(sv,STAT=ierr)

call fillgaps()
```

```
velocity(index+1)=U
stresspertvect(index+1)=stresspert

print *, U, ",", myt, ",", yend

end do
        print *, "velocity"
        do iv = 1, ceiling(maxTime/dt)+1
                print *, velocity(iv), ","
        end do

        print *, "stress"
        do iv = 1, ceiling(maxTime/dt)+1
                print *, stresspertvect(iv), ","
        end do

        print *, " "
        print *,
            "**********************************************************************"
        print *, "x"
        do ix=1, XN+1
                print *, x(ix), ","
        end do
        print *, "y"
        do iy=1, XN+1
                print *, y(iy), ","
        end do
        print *,
            "**********************************************************************"
```

```
        print *, " "
end program fulltime_2009_05_02_Fortran

subroutine rhoode(T, V, VP)
        use globalinfo
        implicit none

real (kind=8) :: T, sr(XN+1), V(2*(XN+1)), VP(2*(XN+1)) !, wu(XN+1), wv(XN+1), su
    (XN+1), sv(XN+1)
        integer (kind=4) ierr, i, ix, iy
        myt = T

        ALLOCATE(sx(XN+1), STAT=ierr)
        IF (ierr /= 0) PRINT*, "sx : Allocation failed"

        ALLOCATE(sy(XN+1), STAT=ierr)
        IF (ierr /= 0) PRINT*, "sy : Allocation failed"

        wu = 0.0
        wv = 0.0

        sx = V(1:XN+1)
        sy = V(XN+2:2*(XN+1))

        call sphvel()
        call specialpositions(sx, sy)
        call wTN()
        call stokes() !su, sv)
```

```
        sr = sqrt(sx**2+sy**2)
        do i = 1, XN+1
                if (sr(i) <= R) then
                        wu(i) = 0.0
                        wv(i) = 0.0
                        su(i) = 0.0
                        sv(i) = 0.0
                endif
        end do

        VP(1:XN+1) = su+wu
        VP(XN+2:2*(XN+1)) = sv+wv

        IF (ALLOCATED(sx)) DEALLOCATE(sx, STAT=ierr)
        IF (ALLOCATED(sy)) DEALLOCATE(sy, STAT=ierr)
end subroutine rhoode

subroutine sphvel()
        use globalinfo
        implicit none

real (kind=8) :: buoyancy, stressbelowsphere, stresssidesphere, stressabovesphere,
     stressbackflow, &
stressbelowsphereA, stresssidesphereA, stressabovesphereA, stressbackflowA, &
stressbelowsphereE, stresssidesphereE, stressabovesphereE, stressbackflowE,cindex

real (kind=8), external :: stresstail1D, stressIntegrandFlat1D,
    stressIntegrandsphere1D, stressIntegrand1D,&
```

```
stresstail1DA, stressIntegrandFlat1DA, stressIntegrandsphere1DA,
   stressIntegrand1DA, &
stresstail1DE, stressIntegrandFlat1DE, stressIntegrandsphere1DE,
   stressIntegrand1DE

Character(len=45):: filename
       integer i,ix, ierr

       !for a two layer fluid only
       if (sy(XN+1)>=R) then
              buoyancy = buoyancybottom;
       elseif (abs(sy(XN+1))<R) then
              buoyancy = buoyancyCoeff1*(3.0*R**2*sy(XN+1)-sy(XN+1)**3)+
                  buoyancyCoeff2;
       else
              buoyancy = buoyancytop;
       endif

!========= Initialize Interpolated ===== !
IF (ALLOCATED(cinterpx)) DEALLOCATE(cinterpx, STAT=ierr)
IF (ALLOCATED(cinterpy)) DEALLOCATE(cinterpy, STAT=ierr)

!initialize interface
ALLOCATE(cinterpx(1000000), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpx : Allocation failed"
ALLOCATE(cinterpy(1000000), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpy : Allocation failed"

cinternalcount=0.0;
```

```
cinterpx =0;
cinterpy = 0;

!calculate stress force
       stresspert = 0.0;
       if (maxval(sy) > minval(sy)) then
              call specialpositions(sx, sy)

              stressbelowsphere = 0.0;
              stresssidesphere = 0.0;
              stressabovesphere = 0.0;
              stressbackflow = 0.0;

              if  (flagu /= 0) then
              call simp(stressIntegrand1D, sy(1), max(-R, sy(1)), integthres,
                 stressbelowsphere)
              call simp(stressIntegrandsphere1D, max(-R, sy(1)), R, integthres,
                 stresssidesphere)
              call simp(stressIntegrand1D, R, yend, integthres, stressabovesphere
                 )
              elseif (flagl /= 0) then
call simp(stressIntegrand1D, sy(1), max(-R, sy(1)), integthres, stressbelowsphere
   )
call simp(stressIntegrandsphere1D, max(-R, sy(1)), yend, integthres,
   stresssidesphere)
else
call simp(stressIntegrandFlat1D, sx(1), xflagb, integthres, stressbelowsphere)
endif
```

```
if (flagb /= XN+1) then
call trapz1(stresstail1D, xflagb, sx(XN+1),real(0.01,kind=8), stressbackflow)
endif
stresspert = stresspertcoeff*(stressbelowsphere + stresssidesphere +
    stressabovesphere -stressbackflow)
endif

!calculate archimedean force of fluid
stresspertA = 0.0;
if (maxval(sy) > minval(sy)) then
call specialpositions(sx, sy)

stressbelowsphereA = 0.0;
stresssidesphereA = 0.0;
stressabovesphereA = 0.0;
stressbackflowA = 0.0;

if  (flagu /= 0) then
call simp(stressIntegrand1DA, sy(1), max(-R, sy(1)), integthres,
    stressbelowsphereA)
call simp(stressIntegrandsphere1DA, max(-R, sy(1)), R, integthres,
    stresssidesphereA)
call simp(stressIntegrand1DA, R, yend, integthres, stressabovesphereA)
elseif (flagl /= 0) then
call simp(stressIntegrand1DA, sy(1), max(-R, sy(1)), integthres,
    stressbelowsphereA)
call simp(stressIntegrandsphere1DA, max(-R, sy(1)), yend, integthres,
    stresssidesphereA)
else
```

```
call simp(stressIntegrandFlat1DA, sx(1), xflagb, integthres, stressbelowsphereA)
endif
if (flagb /= XN+1) then
call simp(stresstail1DA, xflagb, sx(XN+1), integthres, stressbackflowA)
endif
stresspertA = -g*(rhob-rhot)*(stressbelowsphereA + stresssidesphereA +
    stressabovesphereA-stressbackflowA)
endif

!calculate density anomaly force as it scales with shell size epsilon = -sy(1) -
    A

!calculate archimedean force of fluid
stresspertE = 0.0;
if (maxval(sy) > minval(sy)) then

stressbelowsphereE = 0.0;
stresssidesphereE = 0.0;
stressabovesphereE = 0.0;
stressbackflowE = 0.0;



if  (flagu /= 0) then
call simp(stressIntegrand1DE, sy(1), max(-R, sy(1)), integthres,
    stressbelowsphereE)
call simp(stressIntegrandsphere1DE, max(-R, sy(1)), R, integthres,
    stresssidesphereE)
call simp(stressIntegrand1DE, R, yend, integthres, stressabovesphereE)
elseif (flagl /= 0) then
```

```
call simp(stressIntegrand1DE, sy(1), max(-R, sy(1)), integthres,
   stressbelowsphereE)
call simp(stressIntegrandsphere1DE, max(-R, sy(1)), yend, integthres,
   stresssidesphereE)
else
call simp(stressIntegrandFlat1DE, sx(1), xflagb, integthres, stressbelowsphereE)
endif

if (flagb /= XN+1) then
call simp(stresstail1DE, xflagb, sx(XN+1), integthres, stressbackflowE)
endif
stresspertE = g*(rhob-rhot)*(stressbelowsphereE + stresssidesphereE +
   stressabovesphereE-stressbackflowE)
endif

wforceE= stressbelowsphere + stresssidesphere + stressabovesphere
wforceR=stressbackflow
wforce = oneoversixpiamuK*stresspert

ArchBouyancy =oneoversixpiamuK*(g*ms + buoyancy)
ArchBE=-stresspertA *oneoversixpiamuK
ArchBR =-g*(rhob-rhot)*(-stressbackflowA)*oneoversixpiamuK
stresspertReflux= -g* ( rhob-rhot)*stressbackflowE

U = oneoversixpiamuK*(g*ms + buoyancy + stresspert)
end subroutine sphvel

subroutine specialpositions(myx, myy)
       !determine special positions on the interface
```

```
use globalinfo
implicit none

real (kind=8) buoyancy
real (kind=8) :: myx(XN+1), myy(XN+1)
integer (kind=4) temp(1), tempmaxi

flagl = 0
flagu = 0
!find position of backflow
yend = myy(XN+1)

if(maxval(myy) > yend) then
        flagb = XN+1
        do tempmaxi = 1, XN
                if (myy(tempmaxi+1) > yend .and. myy(tempmaxi) <= yend) then
                        flagb = tempmaxi
                        exit
                endif
        end do

        tempmaxi = min(flagb-2, XN-3)
        tempmaxi = max(tempmaxi, 1)
        call interpbridge(5, myy(tempmaxi:tempmaxi+4), myx(tempmaxi:
            tempmaxi+4), yend, xflagb)
else
   flagb = XN+1;
   xflagb = myx(XN+1);
endif
```

```
        !find x position of bottom of sphere


        if (yend >= -R) then
                temp = minloc(abs(myy+R))
                flagl = temp(1)
                if (myy(1) >= -R) then
                        xflagl = 0
                else
                        !print *, 'sp 3'
                        call interpbridge(min(flagb+2, XN+1), myy(1:min(flagb+2, XN
                           +1)), myx(1:min(flagb+2, XN+1)), -R, xflagl)
                        !print *, 'sp 4'
                endif
        endif


    !find x position of top of sphere


        if (yend >= R) then


                !flagu=is 1 if interface is past sphere top
                flagu = 1;


                temp = minloc(abs(myy-R))
                flagt = temp(1)
        endif
end subroutine specialpositions
```

```
subroutine fillgaps()
       use globalinfo
       implicit none

       real (kind=8), dimension(:), allocatable :: newx, newy
       real (kind=8) dist, newpt
       integer (kind=4) internalcount, ierr, xi, posi

       !initialize new interface
       ALLOCATE(newx(2*(XN+1)), STAT=ierr)
       IF (ierr /= 0) PRINT*, "newx : Allocation failed"

       ALLOCATE(newy(2*(XN+1)), STAT=ierr)
       IF (ierr /= 0) PRINT*, "newy : Allocation failed"

       call specialpositions(x, y)

   internalcount  = 0.0
   do xi=1,XN
       internalcount     = internalcount+1.0
       newx(internalcount) = x(xi)
       newy(internalcount) = y(xi)

       dist = sqrt((x(xi+1)-x(xi))**2+(y(xi+1)-y(xi))**2)

       if  ((sqrt(x(xi)**2+y(xi)**2) < (2*R) .and. dist > dx ) .or. dist >R/2)
          then
```

```
!cubic interpolation to find point to fill gap.
internalcount         = internalcount+1
if(x(xi) > 2.0*R) then
    newx(internalcount) = 0.5*(x(xi+1)+x(xi))
    posi = min(XN+1.0, xi+3.0)
    !print *, "fillgap1"
    call interpbridge( 7, x(posi-6.0:posi), y(posi-6.0:posi), 0.5*(x(xi
       +1)+x(xi)), newpt)
    !print *, "fillgap2"
    newy(internalcount) = newpt
else
    posi = max(xi-3.0, 1.0)
    if(flagl /= 0.0 .and. xi > flagl) then
        newy(internalcount) = 0.5*(y(xi+1)+y(xi))

        !print *, "fillgap3"
        if (y(xi+1) >= y(xi)) then
                call interpbridge( 7, y(posi:posi+6.0), x(posi:posi+6.0),
                     0.5*(y(xi+1)+y(xi)), newpt)
            else
                   call interpbridge( 7, y(posi+6.0:posi:-1.0), x(posi
                      +6.0:posi:-1.0), 0.5*(y(xi+1)+y(xi)), newpt)
            endif
            !print *, "fillgap4"
                         if (newpt <= max(x(xi+1), x(xi)) .and. newpt
                            >= min(x(xi+1), x(xi))) then
                newx(internalcount) = newpt
            elseif (((0.5*(y(xi+1)+y(xi)))**2 + (0.5*(x(xi+1)+x(xi)))
                **2) > R**2) then
```

```
                        newx(internalcount) = 0.5*(x(xi+1)+x(xi))
                    else
                        newx(internalcount) = sqrt(R**2 - (0.5*(y(xi+1)+y(xi)
                            ))**2)
                    endif
                else
                    newx(internalcount) = 0.5*(x(xi+1)+x(xi))
                    call interpbridge( 7, x(posi:posi+6.0), y(posi:posi+6.0),
                        0.5*(x(xi+1)+x(xi)), newpt)

                    if (newpt <= max(y(xi+1), y(xi)) .and. newpt >= min(y(xi+1),
                         y(xi))) then
                        newy(internalcount) = newpt
                    elseif (((0.5*(y(xi+1)+y(xi)))**2 + (0.5*(x(xi+1)+x(xi)))
                        **2) > R**2) then
                        newy(internalcount) = 0.5*(y(xi+1)+y(xi))
                    else
                        newy(internalcount) = sqrt(R**2 - (0.5*(x(xi+1)+x(xi)
                            ))**2)
                    endif
                endif
            endif
        endif
    end do
    newx(internalcount+1) = x(XN+1)
    newy(internalcount+1) = y(XN+1)

    IF (ALLOCATED(x)) DEALLOCATE(x,STAT=ierr)
        IF (ALLOCATED(y)) DEALLOCATE(y,STAT=ierr)
```

```
        XN = internalcount
        ALLOCATE(x(XN+1), STAT=ierr)
        IF (ierr /= 0) PRINT*, "fillgap - x : Allocation failed"

        ALLOCATE(y(XN+1), STAT=ierr)
        IF (ierr /= 0) PRINT*, "fillgap - y : Allocation failed"

    x   = newx(1:internalcount+1)
    y   = newy(1:internalcount+1)

    IF (ALLOCATED(newx)) DEALLOCATE(newx,STAT=ierr)
    IF (ALLOCATED(newy)) DEALLOCATE(newy,STAT=ierr)
end subroutine fillgaps

subroutine interpbridge(N, interpx, interpy, xval, yval)
        use globalinfo
        implicit none

        integer (kind=4) :: N, setmin(1), mini, maxi, tempi, tempj, interpchecki=1
        real (kind=8) :: interpx(N), interpy(N), d(N), checkorder(N-1)
        real (kind=8) xval, yval, checkmin, checkmax

        if (N == 1) then
                yval = interpy(1)
        else
                checkorder = interpx(2:N) - interpx(1:N-1)
                checkmin = minval(checkorder)
                interpchecki = 1
```

```
if (checkmin <= 0) then
        mini = 1
        maxi = 1

        do interpchecki=1, N
                do tempi = maxi, N-1
                        if (checkorder(tempi) > 0) then
                                mini = tempi
                                maxi = N
                                do tempj = tempi, N-1
                                        if (checkorder(tempj) < 0) then
                                                maxi = tempj
                                                exit
                                        endif
                                end do
                                exit
                        endif
                end do
                if (xval<= interpx(maxi) .and. xval >= interpx(mini))
                    then
                       exit
                endif
        end do
else
        mini = 1
        maxi = N
endif
```

```
if (xval > interpx(maxi) .or. xval < interpx(mini) .or.
    interpchecki == N) then
        if (xval>interpx(maxi)) then
               print *, "too large"
        elseif (xval < interpx(mini)) then
               print *, "too small"
        else
               print *, "interpchecki", interpchecki, N
        endif
        print *, "out of domain error"
        print *, "time", myt, "xval", xval, "flagl", flagl, "flagb",
            flagb, "flagu", flagu
        print *, "interpx"
        do interpchecki = 1, N
               print *, interpx(interpchecki)
        end do
        print *, "interpy"
        do interpchecki = 1, N
               print *, interpy(interpchecki)
        end do
        print *, "x"
        do interpchecki = 1, XN+1
               print *, sx(interpchecki)
        end do
        print *, "y"
        do interpchecki = 1, XN+1
               print *, sy(interpchecki)
        end do
```

```
                print *, interpx(maxi), xval, interpx(mini)
                stop
        endif

        call spline_pchip_set (maxi-mini+1, interpx(mini:maxi), interpy(
           mini:maxi), d)
        call spline_pchip_val (maxi-mini+1, interpx(mini:maxi), interpy(
           mini:maxi), d, 1, xval, yval)
    endif
end subroutine


!For Non-Uniform Interface !Full2d & FF
subroutine cylindervelinit()
    use globalinfo
    implicit none

    integer index
    do index = 1, NZ
        WZ(index) = real(index-1.0, kind=8)*(LZ-epsilon)/NZ
        !myzrangeNZ(index) = real(index-1.0, kind=8)*2*pi/LZ
        myk(index) = real(index-1.0, kind =8)*2.0*pi/(LZ-epsilon)
    end do

    do index = 1, NR
        WR(index) = real(index-1.0, kind=8)*LR/NR
        !myzrangeNR (index) = real(index-1.0, kind=8)*2*pi/LR
    end do
```

```
        call Hfunc(NZ, WZ+epsilon, myHZ)
        call Gfunc(NZ, WZ+epsilon, myGZ)
        call Hfunc(NR, WR, myHR)
        call Gfunc(NR, WR, myGR)

        firstpartZ = (WZ+epsilon)/2.0*(myHZ+myGZ);
        firstpartR = WR/2.0*(myHR+myGR);

        do index = 1, upperRangeZ
               zcoordinateZ(index) = (index-1.0)*2.0*pi/(LZ-epsilon)
        end do

        do index = 1, upperRangeR
               zcoordinateR(index) = (index-1.0)*2.0*pi/LR
        end do

        call cylindervelgrid()

end subroutine cylindervelinit

subroutine cylindervelgrid()
        use globalinfo
        implicit none

        real (kind=8) :: FZ(NZ), FR(NR), BESSI
        !real (kind=8), external :: sign
        integer myi, WRi, WZi
        real   ( kind = 4 ) wsavez(4*NZ+15), wsaver(4*NR+15)
```

```
complex ( kind = 4 ) tempFR(NR), tempFZ(NZ)
real (kind = 8) :: besseli0R(NR), besseli1R(NR), besseli0Z(NZ), besseli1Z(
    NZ)



do myi = 1, XN+1
        !r-component of velocity
        do WRi = 1, NR
                besseli0R(WRi) = BESSI(0,WR(WRi)*x(myi))
                besseli1R(WRi) = BESSI(1,WR(WRi)*x(myi))
        end do

        tempFR = real((x(myi)*firstpartR*besseli0R-myGR*besseli1R)*0.5,
            kind=4)
        tempFR(1) = 0.0

        call cffti ( NR, wsaver )
        call cfftb ( NR, tempFR, wsaver )
        FR = real(aimag(tempFR)/NR*LR/pi, kind=8)
        cylindervelR(1:upperRangeR, myi) = FR(1:upperRangeR)

        !z-component of velocity
        do WZi = 1, NZ
                besseli0Z(WZi) = BESSI(0,(WZ(WZi)+epsilon)*x(myi))
                besseli1Z(WZi) = BESSI(1,(WZ(WZi)+epsilon)*x(myi))
        end do

        tempFZ = real((x(myi)*firstpartZ*besseli1Z+myHZ*besseli0Z)*0.5,
            kind=4)
```

```
            call cffti ( NZ, wsavez )
            call cfftb ( NZ, tempFZ, wsavez )
            FZ = real(exp(imagi*epsilon*abs(myk))*real(tempFZ/NZ, kind=8)*(LZ-
                epsilon)/pi+3.0*R/pi*epsilon*((-2.0*R**2/(3.0*R0**2)+1.0)*x(myi)
                **2/R0**2+log(epsilon*0.5*R0)-1.0), kind=8)
            cylindervelZ(1:upperRangeZ, myi) = FZ(1:upperRangeZ)

    end do

end subroutine cylindervelgrid

subroutine stokes () !(su, sv)
       use globalinfo
       implicit none

       real (kind=8) :: k1(XN+1), k2(XN+1),u3r(XN+1), u3z(XN+1), myr(XN+1), tempx
           (4), tempy(4),&
                            myinterp1, myinterp2, myinterp3, myinterp4 !, su(XN
                               +1), sv(XN+1)
       real (kind=8), external :: sign
       integer (kind=4) :: starti, sizecyl, i, j

       k1 = 0
       k2 = 0
       sizecyl = size(cylindervelR, 2)
       myr  = sqrt(sx**2+sy**2)
```

```
       do i=1, XN+1

!for non-uniform interface
if (sx(i)<=R0cl) then
starti = floor((sx(i)*(XNclose**2.0)/R0cl)**(1.0/2.0)+1.0)
else
starti = floor((sx(i)-R0cl)*XNfar/(R0-R0cl)+XNclose +1.0)
endif

               starti = max(starti, 1);
               starti = min(starti, sizecyl-3);

               !horizontal velocity component
               call interpbridge(upperRangeR, zcoordinateR, cylindervelR(1:
                  upperRangeR, starti), abs(sy(i)), myinterp1)
               call interpbridge(upperRangeR, zcoordinateR, cylindervelR(1:
                  upperRangeR, starti+1), abs(sy(i)), myinterp2)
               call interpbridge(upperRangeR, zcoordinateR, cylindervelR(1:
                  upperRangeR, starti+2), abs(sy(i)), myinterp3)
               call interpbridge(upperRangeR, zcoordinateR, cylindervelR(1:
                  upperRangeR, starti+3), abs(sy(i)), myinterp4)

!              tempx = (/ ((starti-1+j)*dx, j=0,3) /)
       tempx = (/(startx(starti+j), j=0,3)/) !try for non-uniform interface
               tempy = (/ myinterp1, myinterp2, myinterp3, myinterp4 /)
               tempy=tempy*sign(sy(i))

               call interpbridge(4, tempx, tempy, max(min(sx(i), R0), real(0.0,
                  kind=8)), k1(i))
```

```
            !vertical velocity component

            call interpbridge(upperRangeZ, zcoordinateZ, cylindervelZ(1:
               upperRangeZ, starti), abs(sy(i)), myinterp1)
            call interpbridge(upperRangeZ, zcoordinateZ, cylindervelZ(1:
               upperRangeZ, starti+1), abs(sy(i)), myinterp2)
            call interpbridge(upperRangeZ, zcoordinateZ, cylindervelZ(1:
               upperRangeZ, starti+2), abs(sy(i)), myinterp3)
            call interpbridge(upperRangeZ, zcoordinateZ, cylindervelZ(1:
               upperRangeZ, starti+3), abs(sy(i)), myinterp4)

            tempy = (/ myinterp1, myinterp2, myinterp3, myinterp4 /)

            call interpbridge(4, tempx, tempy, max(min(sx(i), R0), real(0.0,
               kind=8)), k2(i))

        end do

        !print *, sx(XN+1), sy(XN+1), k1(XN+1), k2(XN+1)
        !print *, "stokes 2"


!third reflection minus stokes part



u3z = -(R**2*(12.62665286929866*R0**2*(sx**2 + sy**2)**3*(sx**2 + 2*sy**2) +&
   1.276699789481672*R**6*(3*sx**4 - 24*sx**2*sy**2 + 8*sy**4) +&
```

```
   2*R**2*(sx**2 + sy**2)**2*(2.10444214488311*R0**2*(sx**2 - 2*sy**2) -&
 6.260028903991107*(sx**4 + 3*sx**2*sy**2 + 2*sy**4)) -&
   R**4*(sx**2 + sy**2)*(-0.14001148948167197*(sx**4 + 20*sx**2*sy**2 - 16*sy**4)
       -&
      2.18001729431815*(-2*sx**4 + 2*sx**2*sy**2 + 4*sy**4) +&
      1.1366883*(3*sx**4 - 24*sx**2*sy**2 + 8*sy**4)))*U)/&
(8.*R0**3*(sx**2 + sy**2)**4.5)



u3r =   -(R**2*sx*sy*(-6.38349894740836*R**6*&
      (3*sx**2 - 4*sy**2) +&
     12.62665286929866*R0**2*(sx**2 + sy**2)**3 +&
     2*R**2*(sx**2 + sy**2)**2*&
      (-6.31332643464933*R0**2 -&
        6.260028903991107*(sx**2 + sy**2)) +&
     R**4*(sx**2 + sy**2)*&
      (1.5401263842983917*sx**2 -&
        3.3602757475601273*sy**2 +&
        1.1366883*(23*sx**2 - 12*sy**2) +&
        13.080103765908902*(sx**2 + sy**2)))*U)/&
(8.*R0**3*(sx**2 + sy**2)**4.5)

!Stokes Flow with reflections

su = U*((-0.75*R*sx*sy/myr**3+0.75*R**3*sx*sy/myr**5)-k1) + u3r
sv = U*(1+(-0.75*R/myr-0.75*R*sy**2/myr**3-0.25*R**3/myr**3+0.75*sy**2*R**3/myr
    **5)-k2)+u3z;
```

```
!Stokes Free Space

!su = U*(1)*((-0.75*R*sx*sy/myr**3+0.75*R**3*sx*sy/myr**5))
!sv = U*(1+(1)*(-0.75*R/myr-0.75*R*sy**2/myr**3-0.25*R**3/myr**3+0.75*sy**2*R**3/
    myr**5));


end subroutine stokes


subroutine Hfunc(Num, lambda, ReturnH)
        use globalinfo
        implicit none

        integer index, Num
        real (kind=8) :: lambda(Num), besselk0(Num), besselk1(Num),&
                besseli1(Num), besseli2(Num), besseli0(Num), ReturnH(Num),&
                BESSK, BESSI

        do index = 1, Num
                besselk0(index) = BESSK(0,R0*lambda(index))
                besselk1(index) = BESSK(1,R0*lambda(index))
                besseli0(index) = BESSI(0,R0*lambda(index))
                besseli1(index) = BESSI(1,R0*lambda(index))
                besseli2(index) = BESSI(2,R0*lambda(index))
        end do

        ReturnH = R*(3.0-(6.0+R**2*lambda**2)*(real(besselk0)*besseli2+besseli1*
            real(besselk1)))/(besseli0*besseli2-besseli1**2);
```

```
end subroutine


subroutine Gfunc(Num, lambda, ReturnG)
       use globalinfo
       implicit none

       integer index, Num
       real (kind=8) :: lambda(Num), besselk1(Num), besselk2(Num), besseli1(Num),
            besseli2(Num), besseli0(Num), ReturnG(Num),&
              BESSK, BESSI

       do index = 1, Num
              besselk1(index) = BESSK(1,R0*lambda(index))
              besselk2(index) = BESSK(2,R0*lambda(index))
              besseli0(index) = BESSI(0,R0*lambda(index))
              besseli1(index) = BESSI(1,R0*lambda(index))
              besseli2(index) = BESSI(2,R0*lambda(index))
       end do

       ReturnG = R*(-3.0+R**2*lambda**2*(real(besselk1)*besseli1+besseli0*real(
           besselk2)))/(besseli0*besseli2-besseli1**2);
end


function sign(val)
       implicit none
```

```
        real(kind=8) val, sign

        if (val< 0) then
               sign = -1.0
        else
               sign = 1.0
        endif
end function


!outputs interpolated interfaces

function stresstail1D ( xval )
        use globalinfo
        implicit none

        real (kind=8) eta, xval, stresstail1D
        integer (kind=4) startingi

        startingi = max(flagb-2, 1)
        call interpbridge(XN+2-startingi, sx(startingi:XN+1), sy(startingi:XN+1),
            xval, eta)

        stresstail1D =-xval*(eta*(R**2-3.0*(eta**2+xval**2))/sqrt(eta**2+xval**2)
            **3)&
                +xval*(yend*(R**2-3.0*(xval**2+yend**2)))/sqrt(xval**2+yend**2)**3&
                -xval*6.0*log(eta+sqrt(xval**2+eta**2))+xval*6.0*log(yend+sqrt(xval
                    **2+yend**2))
```

```
cinternalcount = cinternalcount +1;
cinterpy(cinternalcount) = eta;
cinterpx(cinternalcount) = xval;

end

function stressIntegrandFlat1D ( xval )
       use globalinfo
       implicit none

       real (kind=8) xval, eta, stressIntegrandFlat1D
       integer (kind=4) endingi

       endingi = min(flagb+2, XN+1)
       call interpbridge( endingi, sx(1:endingi), sy(1:endingi), xval, eta)
       stressIntegrandFlat1D = -xval*(yend*(R**2-3.0*(yend**2+xval**2))/sqrt(yend
           **2+xval**2)**3)&
               +xval*(eta*(R**2-3.0*(xval**2+eta**2)))/sqrt(xval**2+eta**2)**3&
               -6.0*log((yend+sqrt(xval**2+yend**2))**xval)+6.0*log((eta+sqrt(xval
                  **2+eta**2))**xval)

cinternalcount = cinternalcount +1;
cinterpy(cinternalcount) = eta;
cinterpx(cinternalcount) = xval;
end

function stressIntegrandsphere1D(yval)
       use globalinfo
       implicit none
```

```
        real (kind=8) yval, eta, stressIntegrandsphere1D
        integer (kind=4) tempflagb

        tempflagb = min(flagb+2, XN+1)

        call interpbridge( tempflagb, sy(1:tempflagb), sx(1:tempflagb), yval, eta
            )
        stressIntegrandsphere1D = 2.0/R*(R**2-yval**2)-eta**2*(-R**2+3.0*(yval**2+
            eta**2))/(yval**2+eta**2)**(1.5)

cinternalcount = cinternalcount +1;
cinterpy(cinternalcount) = yval;
cinterpx(cinternalcount) = eta;
end


function stressIntegrand1D(yval)
        use globalinfo
        implicit none

        integer (kind=4) tempflagb
        real (kind=8) yval, eta, stressIntegrand1D

        tempflagb = min(flagb+2, XN+1)
        call interpbridge ( tempflagb, sy(1:tempflagb), sx(1:tempflagb), yval, eta
             )
!print *, "tempflagb",tempflagb, "yval", yval,"eta",eta
```

```
        stressIntegrand1D =- eta**2*(-R**2+3.0*(yval**2+eta**2))/((yval**2+eta**2)
            **(1.5));
cinternalcount = cinternalcount +1;
cinterpy(cinternalcount) = yval;
cinterpx(cinternalcount) = eta;
end

subroutine wTN () !(wu, wv)
use globalinfo
implicit none



real (kind = 8) wbackflow(XN+1), wsidesphere(XN+1), wbelowsphere(XN+1),
    wabovesphere(XN+1),&
tempbackflow, tempsidesphere, tempbelowsphere, tempabovesphere !, wu(XN+1), wv(XN
    +1),
real (kind=8), external :: w2IntegrandBackflow, w2IntegrandBelowSphere,
    w2IntegrandPartialSphere,&
w2IntegrandZetaSphere, w2IntegrandZetaVert, w2IntegrandR, w2IntegrandZ
integer (kind=4) wi
real (kind=8) AEbelow,AEside,AEabove!,AreaEntrain, AreaReflux, AreaSpherePortion
real (kind=8), external :: RefluxAreaFunction,EntrainPartialSphere,
    EntrainZetaSphere,EntrainZetaVert,EntrainBelowSphere, AreaElementZeta,
    AreaElementRho

integer ierr
```

```
wbackflow             = real(0.0, kind=8)
wsidesphere    = real(0.0, kind=8)
wbelowsphere   = real(0.0, kind=8)
wabovesphere   = real(0.0, kind=8)
AreaReflux     = real(0.0, kind=8)
AreaEntrain    = real(0.0, kind=8)

call specialpositions(sx, sy)




!========== FIND ENTRAINMENT AND REFLUX VOLUMES =========!

!reflux

if(flagb <XN+1.0 ) then
!print *, "w0"
call simp(RefluxAreaFunction, xflagb, sx(XN+1), integthres, AreaReflux)

endif
if (flagu /= 0.0) then
AreaSpherePortion= (4/3*pi*R**3);


call simp(EntrainPartialSphere,real(0.0,kind=8),xflagl,integthres,AEbelow)
```

```
call simp(EntrainZetaSphere, max(-R, sy(1)), R, integthres, AEside)


call simp(EntrainZetaVert, R, yend, integthres, AEabove)

AreaEntrain=AEbelow+AEside+AEabove;

elseif (flagl /= 0.0) then


!find area of the sphere

AreaSpherePortion= (pi*(yend+R)**2/3)*(3*R - (yend+R));


call simp(EntrainPartialSphere, real(0.0, kind=8), xflagl, integthres,AEbelow)

call simp(EntrainZetaSphere, max(-R, sy(1)), yend, integthres, AEside)


AreaEntrain=AEbelow+AEside;


else
!print *, "w6"
AreaSpherePortion= 0.0;
```

```
call simp(EntrainBelowSphere, max(sx(1), real(0.0, kind=8)), xflagb, integthres,
    AEbelow)

AreaEntrain=AEbelow;

endif



!========== FIND perturbation velocity w =========!

do RorZ = 0, 1
if (minval(sy) < maxval(sy)) then
do wi = 1, XN+1
px = sx(wi)
py = sy(wi)



!========= Initialize Interpolated ===== !



IF (ALLOCATED(cwinterpx)) DEALLOCATE(cwinterpx, STAT=ierr)
IF (ALLOCATED(cwinterpy)) DEALLOCATE(cwinterpy, STAT=ierr)

!initialize interface interpolated
ALLOCATE(cwinterpx(1000000), STAT=ierr)
```

```
IF (ierr /= 0) PRINT*, "cintertpx : Allocation failed"

ALLOCATE(cwinterpy(1000000), STAT=ierr)
IF (ierr /= 0) PRINT*, "cintertpy : Allocation failed"

!print *, 'after pertvelT.f90'
cwinternalcount=0.0;
cwinterpx =0;
cwinterpy = 0;



tempbackflow  = real(0.0, kind=8)
tempsidesphere = real(0.0, kind=8)
tempbelowsphere = real(0.0, kind=8)
tempabovesphere = real(0.0, kind=8)

if(flagb <XN+1.0 ) then
!print *, "w0"
call trapz1(w2IntegrandBackflow, xflagb, sx(XN+1), numtrapz, tempbackflow)

endif
if (flagu /= 0.0) then
!print *, "w1"
call trapz1(w2IntegrandPartialSphere, real(0.0, kind=8), xflagl, numtrapz,
   tempbelowsphere)
!print *, "w2"
call trapz1(w2IntegrandZetaSphere, max(-R, sy(1)), R, numtrapz, tempsidesphere)
!print *, "w3"
```

```
call trapz1(w2IntegrandZetaVert, R, yend, numtrapz, tempabovesphere)
!print *, "w3.5"
elseif (flagl /= 0.0) then
!print *, "w4"
call trapz1(w2IntegrandPartialSphere, real(0.0, kind=8), xflagl, numtrapz,
   tempbelowsphere)
!print *, "w5"
call trapz1(w2IntegrandZetaSphere, max(-R, sy(1)), yend, numtrapz, tempsidesphere
   )
!print *, "w5.5"
else
!print *, "w6"

!print *, "below sphere limits", real(0.0, kind=8), xflagb
call trapz1(w2IntegrandBelowSphere, max(sx(1), real(0.0, kind=8)), xflagb,
   numtrapz, tempbelowsphere)

!print *, "belowsphere", tempbelowsphere

endif

wbackflow(wi)         = tempbackflow
wbelowsphere(wi)      = tempbelowsphere
wsidesphere(wi)       = tempsidesphere
wabovesphere(wi)      = tempabovesphere
end do
endif
if (RorZ > 0) then
wu = (-wbackflow+wbelowsphere+wsidesphere+wabovesphere)*drhogover8mu
```

```
else
wv = (-wbackflow+wbelowsphere+wsidesphere+wabovesphere)*drhogover8mu
!print *, sx(2), sy(2)
!                     print *, "wbackflow", wbackflow
!                     print *, "wbelowsphere",wbelowsphere(2)
!                     print *, "wsidesphere",wsidesphere(2)
!                     print *, "wabovesphere",wabovesphere(2)
endif
!print *, "wv", wv



end do

!print *, "flow components", wbelowsphere(3), wsidesphere(3)
if(abs(wbelowsphere(1)) > 1000.0) then
stop
endif
!print *, "w7"
end subroutine wTN




!========== FOR VOLUME TRACK =========!

function AreaElementRho (rho)
use globalinfo
implicit none

real (kind =8) rho, AreaElementRho
```

```
AreaElementRho = 2*pi*rho;
return
end


function AreaElementZeta (zeta)
use globalinfo
implicit none

real (kind =8) zeta, AreaElementZeta

AreaElementZeta = 2*pi*myrho;
return
end


function RefluxAreaFunction(rho)
use globalinfo
implicit none

real (kind=8) rho, zcoord, RefluxAreaFunction
real (kind=8), external :: AreaElementZeta

call interpbridge( XN+2-max(flagb-2, 1), sx(max(flagb-2,1):XN+1), sy(max(flagb-2,
    1):XN+1), rho, zcoord)

myrho  = rho
call simp2(AreaElementZeta, yend, zcoord, integthres, RefluxAreaFunction)
```

```
end


function EntrainZetaSphere(zeta)
use globalinfo
implicit none

real (kind=8) zeta, xupper, xlower, EntrainZetaSphere
real (kind=8), external :: AreaElementRho

myzeta = zeta
call interpbridge(min(flagb+2, XN+1), sy(1:min(flagb+2, XN+1)), sx(1:min(flagb+2,
    XN+1)), zeta, xupper)

xlower = sqrt(R**2 - zeta**2)

call simp2(AreaElementRho, xlower, xupper, integthres, EntrainZetaSphere)
end

function EntrainPartialSphere(rho)
use globalinfo
implicit none

real (kind=8) rho, zcoord, EntrainPartialSphere
real (kind=8), external :: AreaElementZeta

myrho = rho
```

```
call interpbridge(min(flagl+3, XN+1), sx(1:min(flagl+3, XN+1)), sy(1:min(flagl+3,
    XN+1)), rho, zcoord)


call simp2(AreaElementZeta, zcoord, -R, integthres, EntrainPartialSphere)
end

function EntrainZetaVert(zeta)
use globalinfo
implicit none

real (kind=8) zeta, xupper, EntrainZetaVert
real (kind=8), external :: AreaElementRho
integer (kind=4) temp(1), tempmini

myzeta = zeta

call interpbridge(min(flagb+2, XN+1), sy(1:min(flagb+2, XN+1)), sx(1:min(flagb+2,
    XN+1)), zeta, xupper)

call simp2(AreaElementRho, real(0.0, kind=8), xupper, integthres,EntrainZetaVert)
end


function EntrainBelowSphere(rho)
use globalinfo
implicit none

real (kind=8) rho, zcoord, EntrainBelowSphere
```

```
real (kind=8), external :: AreaElementZeta



call interpbridge(max(flagl+2, XN+1), sx(1:max(flagl+2, XN+1)), sy(1:max(flagl+2,
     XN+1)), rho, zcoord)

myrho  = rho
call simp2(AreaElementZeta, zcoord, yend, integthres, EntrainBelowSphere)
end


!========== FOR VOLUME TRACK =========!

function w2IntegrandBackflow(rho)
use globalinfo
implicit none

real (kind=8) rho, zcoord, w2IntegrandBackflow
real (kind=8), external :: w2IntegrandZeta

!print *, "w1"
call interpbridge( XN+2-max(flagb-2, 1), sx(max(flagb-2,1):XN+1), sy(max(flagb-2,
     1):XN+1), rho, zcoord)
!print *, "w1"
myrho  = rho
call trapz1(w2IntegrandZeta, yend, zcoord, real(min(0.001,numtrapz),kind=8),
    w2IntegrandBackflow)
```

```
cwinternalcount = cwinternalcount +1;
cwinterpx(cwinternalcount) = rho;
cwinterpy(cwinternalcount) = zcoord;

end

function w2IntegrandZetaSphere(zeta)
use globalinfo
implicit none

real (kind=8) zeta, xupper, xlower, w2IntegrandZetaSphere
real (kind=8), external :: w2IntegrandRho

myzeta = zeta
!print *, "w2"
call interpbridge(min(flagb+2, XN+1), sy(1:min(flagb+2, XN+1)), sx(1:min(flagb+2,
    XN+1)), zeta, xupper)
!print *, "w2"
xlower = sqrt(R**2 - zeta**2)
call trapz1(w2IntegrandRho, xlower, xupper, numtrapz, w2IntegrandZetaSphere)

cwinternalcount = cwinternalcount +1;
cwinterpx(cwinternalcount) = xupper;
cwinterpy(cwinternalcount) = zeta;
end

function w2IntegrandPartialSphere(rho)
use globalinfo
```

```
implicit none

real (kind=8) rho, zcoord, w2IntegrandPartialSphere
real (kind=8), external :: w2IntegrandZeta

myrho = rho
!print *, "w3"
call interpbridge(min(flagl+3, XN+1), sx(1:min(flagl+3, XN+1)), sy(1:min(flagl+3,
    XN+1)), rho, zcoord)
!print *, "w3"
call trapz1(w2IntegrandZeta, zcoord, -R, numtrapz, w2IntegrandPartialSphere)

cwinternalcount = cwinternalcount +1;
cwinterpx(cwinternalcount) = rho;
cwinterpy(cwinternalcount) = zcoord;
end

function w2IntegrandZetaVert(zeta)
use globalinfo
implicit none

real (kind=8) zeta, xupper, w2IntegrandZetaVert
real (kind=8), external :: w2IntegrandRho
integer (kind=4) temp(1), tempmini

myzeta = zeta
!temp = minloc( sy(1:min(flagb, XN+1)))
!tempmini = temp(1)
!print *, "w4"
```

```
!call interpbridge(min(flagb, XN+1)-tempmini+1, sy(tempmini:min(flagb, XN+1)), sx
   (tempmini:min(flagb, XN+1)),zeta, xupper)
call interpbridge(min(flagb+2, XN+1), sy(1:min(flagb+2, XN+1)), sx(1:min(flagb+2,
    XN+1)), zeta, xupper)
!print *, "w4"
!print *, yend
call trapz1(w2IntegrandRho, real(0.0, kind=8), xupper, numtrapz,
   w2IntegrandZetaVert)

cwinternalcount = cwinternalcount +1;
cwinterpx(cwinternalcount) = xupper;
cwinterpy(cwinternalcount) = zeta;
end



function w2IntegrandBelowSphere(rho)
use globalinfo
implicit none

real (kind=8) rho, zcoord, w2IntegrandBelowSphere
real (kind=8), external :: w2IntegrandZeta



!print *, "w5p2"
call interpbridge(max(flagl+2, XN+1), sx(1:max(flagl+2, XN+1)), sy(1:max(flagl+2,
    XN+1)), rho, zcoord)
!print *, "w5p2"
myrho  = rho
!print *, zcoord, yend, myrho
```

```
call trapz1(w2IntegrandZeta, zcoord, yend, numtrapz, w2IntegrandBelowSphere)



cwinternalcount = cwinternalcount +1;

cwinterpx(cwinternalcount) = rho;

cwinterpy(cwinternalcount) = zcoord;

end




!=====================================================

!Integrands

!=====================================================


function w2IntegrandZeta(zeta)

use globalinfo

implicit none


real (kind=8) zeta, w2IntegrandZeta

real (kind=8), external :: w2IntegrandR, w2IntegrandZ, w2IntegrandRFF,

    w2IntegrandZFF




Real*8 ellipticE, ellipticK,ellipticE1, ellipticK1

DOUBLE PRECISION k,k1, kbar, tempK, tempE, DRF, DRD, ex, ey, ez

integer ier
```

```
myzeta = zeta


if ( ((py-myzeta)**2+(px-myrho)**2)>singthres) then

!============================================================
!inputs for ellipticK, ellipticE,ellipticK1, and ellipticE1
!============================================================
k      = 4.0*px*myrho/((py-myzeta)**2+(px-myrho)**2)

kbar   = k/(k+1.0)

ex = 0.0
ey = 1.0-kbar
ez = 1.0

tempK  = DRF(ex, ey, ez, ier)
tempE  = tempK-1.0/3.0*kbar*DRD(ex, ey, ez, ier)



ellipticE     = sqrt(1.0+k)*tempE
ellipticK     = (1.0/sqrt(1.0+k))*tempK

endif


k1 = -((-4.0)*R**2.0*px*myrho*(R**4.0+(-2.0)* &
```

```
R**2.0*(px*myrho+py*myzeta)+(px**2.0+py**2.0)*(myrho**2.0+myzeta**2.0))**(-1.0))

kbar   = k1/(k1+1.0)

ex = 0.0
ey = 1.0-kbar
ez = 1.0

tempK  = DRF(ex, ey, ez, ier)
tempE  = tempK-1.0/3.0*kbar*DRD(ex, ey, ez, ier)



ellipticE1    = sqrt(1.0+k1)*tempE
ellipticK1    = (1.0/sqrt(1.0+k1))*tempK




if (RorZ >0.0) then

if (px**2+py**2 >= (FFR)**2) then
w2IntegrandZeta= w2IntegrandRFF()
else
w2IntegrandZeta =w2IntegrandR(ellipticK,ellipticE,ellipticK1,ellipticE1)
endif

else
if (px**2+py**2 >= (FFR)**2) then
w2IntegrandZeta = w2IntegrandZFF()
!print *,'FarField', 'px',px,'py',py, 'zeta',zeta,' rho',myrho
```

```
!print *, "integrand value", w2IntegrandZeta
else

w2IntegrandZeta =w2IntegrandZ(ellipticK,ellipticE,ellipticK1,ellipticE1)
!print*,'3D Oseen', 'px',px,'py',py, 'zeta',zeta,' rho',myrho
!print *, "integrand value", w2IntegrandZeta
endif

endif




!print *, "integrand value", w2IntegrandZeta
!print *, "eval pts", zeta, px, py
!print *, "eval pt and result", myzeta, w2IntegrandZeta, myt
!endif
end



function w2IntegrandRho(rho)
use globalinfo
implicit none

real (kind=8) rho, w2IntegrandRho
real (kind=8), external :: w2IntegrandR, w2IntegrandZ, w2IntegrandRFF,
    w2IntegrandZFF
```

```
Real*8 ellipticE, ellipticK,ellipticE1, ellipticK1
DOUBLE PRECISION k, k1, kbar, tempK, tempE, DRF, DRD, ex, ey, ez
integer ier




myrho = rho
if ( ((py-myzeta)**2+(px-myrho)**2)>singthres) then

!=====================================================
! inputs for ellipticK, ellipticE,ellipticK1, and ellipticE1
!=====================================================
k      = 4.0*px*myrho/((py-myzeta)**2+(px-myrho)**2)

kbar   = k/(k+1.0)

ex = 0.0
ey = 1.0-kbar
ez = 1.0

tempK  = DRF(ex, ey, ez, ier)
tempE  = tempK-1.0/3.0*kbar*DRD(ex, ey, ez, ier)



ellipticE     = sqrt(1.0+k)*tempE
ellipticK     = (1.0/sqrt(1.0+k))*tempK
```

```
endif

k1 = -((-4.0)*R**2*px*myrho*(R**4.0+(-2.0)* &
R**2.0*(px*myrho+py*myzeta)+(px**2.0+py**2.0)*(myrho**2.0+myzeta**2.0))**(-1.0));

kbar   = k1/(k1+1.0)

ex = 0.0
ey = 1.0-kbar
ez = 1.0

tempK  = DRF(ex, ey, ez, ier)
tempE  = tempK-1.0/3.0*kbar*DRD(ex, ey, ez, ier)



ellipticE1     = sqrt(1.0+k1)*tempE
ellipticK1     = (1.0/sqrt(1.0+k1))*tempK



if (RorZ >0.0) then

if (px**2+py**2 >= (FFR)**2) then
w2IntegrandRho = w2IntegrandRFF()
else
w2IntegrandRho = w2IntegrandR(ellipticK,ellipticE,ellipticK1,ellipticE1)
endif

else
if (px**2+py**2 >= (FFR)**2) then
```

```
w2IntegrandRho = w2IntegrandZFF()
else
w2IntegrandRho = w2IntegrandZ(ellipticK,ellipticE,ellipticK1,ellipticE1)
endif

endif
end

!====================== Far Field kernel for px^2+py^2 > 4R^2
   ====================!
function w2IntegrandRFF()
use globalinfo
implicit none

real (kind=8) w2IntegrandRFF, ellipticE, ellipticK
DOUBLE PRECISION k, kbar, tempK, tempE, DRF, DRD, ex, ey, ez
integer ier

w2IntegrandRFF = 0.0
if (px > 0.0 .and. ((py-myzeta)**2+(px-myrho)**2)>singthres) then
k      = 4.0*px*myrho/((py-myzeta)**2+(px-myrho)**2)
kbar   = k/(k+1.0)

ex = 0.0
ey = 1.0-kbar
ez = 1.0

tempK  = DRF(ex, ey, ez, ier)
tempE  = tempK-1.0/3.0*kbar*DRD(ex, ey, ez, ier)
```

```
!tempK=0.0
!tempE=0.0

ellipticE      = sqrt(1.0+k)*tempE
ellipticK      = (1.0/sqrt(1.0+k))*tempK

w2IntegrandRFF = 2.0*(py-myzeta)/(pi*px*sqrt((px-myrho)**2+(py-myzeta)**2)&
*((px+myrho)**2+(py-myzeta)**2))*((px**2-myrho**2-(py-myzeta)**2)*ellipticE&
+((myrho+px)**2+(py-myzeta)**2)*ellipticK)&
-3.0*R*px*py*(2.0*myzeta**2+myrho**2)/(2.0*sqrt(px**2+py**2)**3*sqrt(myzeta**2+
    myrho**2)**3)&
+(R**3*px*(px**2*(py + 5.0*myzeta)* (2.0*myzeta**2 - myrho**2)&
+ py*(py**2*(2.0*myzeta**2 - myrho**2) + 10.0*py*myzeta*(-2.0*myzeta**2 + myrho
    **2)&
+3.0*(2.0*myzeta**4 + 3.0*myzeta**2*myrho**2 + myrho**4))))/(2.0*(px**2 + py**2)
    **(2.5)*(myzeta**2 + myrho**2)**(2.5))&
- (3.0*R**5*px*(8.0*px**4*(2.0*myzeta**3 - 3.0*myzeta*myrho**2) +px**2* (-8.0*py
    **2*(2.0*myzeta**3&
- 3.0*myzeta*myrho**2) + py*(-136.0*myzeta**4 + 296.0*myzeta**2*myrho**2 - 23.0*
    myrho**4)&
+ 8.0*myzeta*(2.0*myzeta**4 + myzeta**2*myrho**2 - myrho**4)) -4.0*py**2*(4.0*py
    **2*(2.0*myzeta**3&
- 3.0*myzeta*myrho**2) + 8.0*myzeta*(2.0*myzeta**4 + myzeta**2*myrho**2 - myrho
    **4)&
- 3.0*py*(12.0*myzeta**4 - 22.0*myzeta**2* myrho**2 + myrho**4))))/(16.0*(px**2 +
     py**2)**(3.5)*&
(myzeta**2 + myrho**2)**(3.5))

w2IntegrandRFF = w2IntegrandRFF*myrho
```

```
endif
end

function w2IntegrandZFF()
use globalinfo
implicit none

real (kind=8) w2IntegrandZFF, ellipticE, ellipticK
DOUBLE PRECISION k, kbar, tempK, tempE, DRF, DRD, ex, ey, ez
integer ier

w2IntegrandZFF = 0.0
!if (px > 0.0 .and. ((py-myzeta)**2+(px-myrho)**2)>1.0E-006) then
if (((py-myzeta)**2+(px-myrho)**2)>singthres) then
k      = 4.0*px*myrho/((py-myzeta)**2+(px-myrho)**2)
kbar   = k/(k+1.0)

ex = 0.0
ey = 1.0-kbar
ez = 1.0

tempK  = DRF(ex, ey, ez, ier)
tempE  = DRF(ex, ey, ez, ier)-1.0/3.0*kbar*DRD(ex, ey, ez, ier)
!tempK=0.0
!tempE=0.0

ellipticE     = sqrt(1.0+k)*tempE
ellipticK     = (1.0/sqrt(1.0+k))*tempK
```

```
w2IntegrandZFF = 4.0*((py-myzeta)**2*ellipticE+((px+myrho)**2+(py-myzeta)**2)*
    ellipticK)/&
(pi*sqrt((px-myrho)**2+(py-myzeta)**2)*((px+myrho)**2+(py-myzeta)**2))&
-3.0*R*(px**2+2.0*py**2)*(2.0*myzeta**2+myrho**2)/(2.0*sqrt(px**2+py**2)**3*sqrt(
    myzeta**2+myrho**2)**3)&
-R**3/(2.0*sqrt(px**2+py**2)**5*sqrt(myrho**2+myzeta**2)**5)*&
(px**4*(-2.0*myzeta**2 + myrho**2) - 2.0*py**2*(2.0*myzeta**4 + 3.0*myzeta**2*
    myrho**2&
+ myrho**4 + py**2*(2.0*myzeta**2 - myrho**2) + 5.0*py*myzeta*(-2.0*myzeta**2 +
    myrho**2))&
+ px**2*(2.0*myzeta**4 + 3.0*myzeta**2*myrho**2 + myrho**4 + 5.0*py*myzeta*(-2.0*
    myzeta**2 + myrho**2)&
+py**2* (-6.0*myzeta**2 + 3.0*myrho**2)))&
-(3.0*R**5*(px**4*(8.0*myzeta**4 - 24.0*myzeta**2*myrho**2 + 3.0*myrho**4 + 8.0*
    py*(2.0*myzeta**3&
- 3.0*myzeta*myrho**2)) - 8.0*py**3*(4.0*myzeta**5 + 2.0*myzeta**3*myrho**2 -
    2.0*myzeta*myrho**4&
+ py**2*(4.0*myzeta**3 - 6.0*myzeta*myrho**2) - py*(12.0*myzeta**4 - 22.0*myzeta
    **2* myrho**2&
+ myrho**4)) - 8.0*px**2*py*(-6.0*myzeta**5 - 3.0*myzeta**3* myrho**2 + 3.0*
    myzeta* myrho**4&
+ py**2* (2.0* myzeta**3 - 3.0* myzeta* myrho**2) + py* (22.0*myzeta**4 - 45.0*
    myzeta**2* myrho**2&
+ 3.0*myrho**4))))/(16.0*(px**2 + py**2)**(3.5)* (myzeta**2 + myrho**2)**(3.5))

w2IntegrandZFF = w2IntegrandZFF*myrho
endif
end
```

```
function w2IntegrandR(ellipticK,ellipticE,ellipticK1, ellipticE1)
! this is the horizontal component of the velocity
use globalinfo
implicit none

real (kind=8) w2IntegrandR,ellipticE, ellipticK,ellipticE1, ellipticK1,ILogR
real (kind=8), external :: I1R,I2R,I3R,I4R,I5R,I6R,I7R,I8R,LogTermThetaR

w2IntegrandR=0.0;
ILogR = 0.0;

if (px > 0.0) then

if (abs (myzeta*px + py * myrho ) < logsing) then
!call simp2(LogTermThetaR, real(0.0,kind=8), real(2.0*pi,kind=8),integthres,
    ILogR)

call trapz1(LogTermThetaR, real(0.0,kind=8),real(2.0*pi,kind=8), logtrapzbig,
    ILogR)

else
call trapz1(LogTermThetaR, real(0.0,kind=8),real(2.0*pi,kind=8), logtrapz, ILogR)
endif

w2IntegrandR = I1R(ellipticK,ellipticE)+I2R(ellipticK1,ellipticE1)&
+I3R(ellipticK1,ellipticE1)+I4R(ellipticK1,ellipticE1)&
+I5R(ellipticK1,ellipticE1)+I6R(ellipticK1,ellipticE1)&
+I7R(ellipticK1,ellipticE1)+I8R(ellipticK1,ellipticE1)+ILogR
```

```
w2IntegrandR = w2IntegrandR*myrho/pi
endif
!print *, "w2IntegrandR", w2IntegrandR

end

function w2IntegrandZ(ellipticK, ellipticE,ellipticK1, ellipticE1)
use globalinfo
implicit none

real (kind=8) w2IntegrandZ,ellipticE, ellipticK,ellipticE1, ellipticK1,ILogZ
real (kind=8), external :: I1Z,I2Z,I3Z,I4Z,I5Z,I6Z,I7Z,I8Z,LogTermThetaZ

ILogZ = 0.0;

if (abs (myzeta*px + py * myrho ) < logsing .and. abs (myzeta*px + py * myrho ) >
     0.0) then
call trapz1(LogTermThetaZ,real(0.0,kind=8),real(2.0*pi,kind=8),logtrapzbig, ILogZ
    )

elseif (abs (myzeta*px + py * myrho ) > logsing )then

call trapz1(LogTermThetaZ, real(0.0,kind=8),real(2.0*pi,kind=8),logtrapz, ILogZ)


endif

w2IntegrandZ = I1Z(ellipticK,ellipticE)+I2Z(ellipticK1,ellipticE1)&
+I3Z(ellipticK1,ellipticE1)+I4Z(ellipticK1,ellipticE1)&
```

```
+I5Z(ellipticK1,ellipticE1)+I6Z(ellipticK1,ellipticE1) &
+I7Z(ellipticK1,ellipticE1)+I8Z(ellipticK1,ellipticE1)+ILogZ

w2IntegrandZ = w2IntegrandZ*myrho/pi

end



function I1R(ellipticK,ellipticE)
use globalinfo
implicit none

Real*8 I1R, ellipticE, ellipticK

I1R=0
if ( ((py-myzeta)**2+(px-myrho)**2)>singthres) then

I1R= 2.0*px**(-1.0)*(px**2.0+(-2.0)*px*myrho+myrho**2.0+(py+(-1.0)*myzeta)**2.0)
   **(-1.0/2.0)* &
(px**2.0+2.0*px*myrho+myrho**2.0+(py+(-1.0)*myzeta)**2.0)**(-1.0)*(py+(-1.0)*
   myzeta)*( &
(px**2.0+(-1.0)*myrho**2.0+(-1.0)*(py+(-1.0)*myzeta)**2.0)*ellipticE+(px
   **2.0+2.0* &
px*myrho+myrho**2.0+(py+(-1.0)*myzeta)**2.0)*ellipticK)
endif

end

function I1Z(ellipticK,ellipticE)
```

```
use globalinfo
implicit none

Real*8 I1Z, ellipticE, ellipticK

I1Z=0

if ( ((py-myzeta)**2+(px-myrho)**2)>singthres) then


I1Z=4.0*(py+(-1.0)*myzeta)**2.0*(px**2.0+(-2.0)*px*myrho+myrho**2.0+((-1.0)*py+
   myzeta)**2.0) &
**(-1.0/2.0)*(px**2.0+2.0*px*myrho+myrho**2.0+((-1.0)*py+myzeta)**2.0)**(-1.0)* &
ellipticE+4.0*(px**2.0+(-2.0)*px*myrho+myrho**2.0+((-1.0)*py+myzeta)**2.0)**( &
-1.0/2.0)*ellipticK;
endif

end


function I2R (ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I2R, ellipticE1, ellipticK1
Real*8 :: a

a=R;
```

```
I2R=    (-2*(a**2/(myrho**2 + myzeta**2))**2.5*&
(a**2*myzeta - (myrho**2 + myzeta**2)*py)*&
((a**4 - 2*a**2*myzeta*py - (myrho**2 + myzeta**2)*(px**2 - py**2))*&
ellipticE1 - &
(a**4 + 2*a**2*(myrho*px - myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticK1))/&
(a**2*px*(a**4 + 2*a**2*(myrho*px - myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
Sqrt((a**4 - 2*a**2*(myrho*px + myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))/(myrho**2 + myzeta**2)))
end



function I2Z(ellipticK1,ellipticE1)
use globalinfo
implicit none



Real*8 I2Z, ellipticE1, ellipticK1
Real*8 :: a
a=R;

I2Z= 4.0*(a**2.0*(myrho**2.0+myzeta**2.0)**(-1.0))**(1.0/2.0)*((myrho**2.0+myzeta
    **2.0)**(-1.0) &
*(a**4.0+(-2.0)*a**2.0*(px*myrho+py*myzeta)+(px**2.0+py**2.0)*(myrho**2.0+myzeta
    **2.0) &
))**(-1.0/2.0)*((-1.0)*a**2.0*(myrho**2.0+myzeta**2.0)**(-2.0)*(myrho**2.0*py+
    myzeta* &
```

```
((-1.0)*a**2.0+py*myzeta))**2.0*(a**4.0+2.0*a**2.0*(px*myrho+(-1.0)*py*myzeta)+(
    &
px**2.0+py**2.0)*(myrho**2.0+myzeta**2.0))**(-1.0)*ellipticE1+(-1.0)*ellipticK1)

end

function I3R (ellipticK1, ellipticE1)
use globalinfo
implicit none



Real*8 I3R, ellipticE1, ellipticK1,coeff
Real*8 :: a

a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I3R=     coeff*( (-2*myzeta*((a**4 - 2*a**2*(myrho*px + myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))* &
ellipticE1 - &
(a**4 - 2*a**2*myzeta*py + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticK1  ))/&
(a*(myrho**2 + myzeta**2)**2*px*&
Sqrt((a**4 - 2*a**2*(myrho*px + myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))/&
(myrho**2 + myzeta**2))))
return
```

```
end

function I3Z (ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I3Z, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I3Z=coeff*(4.0*a*myzeta**2.0*(myrho**2.0+myzeta**2.0)**(-2.0)*((myrho**2.0+myzeta
    **2.0)**(-1.0)*( &
a**4.0+(-2.0)*a**2.0*(px*myrho+py*myzeta)+(px**2.0+py**2.0)*(myrho**2.0+myzeta
    **2.0))) &
**(-1.0/2.0)*ellipticK1)
return

end

function I4R(ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I4R, ellipticE1, ellipticK1,coeff,a
a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);
```

```
I4R=-coeff* ((2*a*(-((a**2*myzeta)/(myrho**2 + myzeta**2)) + py)*&
((a**4 - 2*a**2*myzeta*py + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
(a**4 - 2*a**2*(myrho*px + myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticE1 - &
(a**8 - 4*a**6*myzeta*py - &
4*a**2*myzeta*(myrho**2 + myzeta**2)*py*&
(px**2 + py**2) + &
(myrho**2 + myzeta**2)**2*(px**2 + py**2)**2 + &
a**4*(-2*myrho**2*(px**2 - py**2) + &
2*myzeta**2*(px**2 + 3*py**2)))*&
ellipticK1))/&
((myrho**2 + myzeta**2)**2*px*&
(a**4 + 2*a**2*(myrho*px - myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
((a**4 - 2*a**2*(myrho*px + myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))/&
(myrho**2 + myzeta**2))**1.5))
return

end

function I4Z(ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I4Z, ellipticE1, ellipticK1,coeff
```

```
real*8 :: a
a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I4Z=-coeff* ((4*a**3*myzeta*(-(a**2*myzeta) + (myrho**2 + myzeta**2)*py)*&
ellipticE1)/((myrho**2 + myzeta**2)**2*(a**4 + 2*a**2*(myrho*px &
- myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
Sqrt((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 +&
myzeta**2)*(px**2 + py**2))/(myrho**2 + myzeta**2))))
return
end

function I5R(ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 ellipticE1, ellipticK1,coeff,I5R
real*8 :: a
a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I5R= -coeff*((-2*a**3*myzeta*((a**4 - 2*a**2*myzeta*py - (myrho**2 + myzeta**2)*(
    px**2 &
- py**2))*ellipticE1 - &
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*
    ellipticK1))/&
```

```
((myrho**2 + myzeta**2)**2*px*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 +
     &
myzeta**2)*(px**2 + py**2))*&
sqrt((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 +&
py**2))/(myrho**2 + myzeta**2))))
return
end

function I5Z (ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I5Z, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I5Z= -coeff*( (4*a**3*myzeta*(-(a**2*myzeta) + (myrho**2 + myzeta**2)*py)*
    ellipticE1 )/&
((myrho**2 + myzeta**2)**2*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 +
    myzeta**2)*(px**2 + py**2))*&
Sqrt((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py
    **2))/(myrho**2 + myzeta**2))) )
end

function I6R (ellipticK1, ellipticE1)
use globalinfo
implicit none
```

```
Real*8 I6R, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;

coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I6R= coeff* ((4*myzeta*((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 +
   myzeta**2)*(px**2 + py**2))* &
((a**8 - 4*a**6*myzeta*py - 4*a**2*myzeta*(myrho**2 + myzeta**2)*py*(px**2 + py
   **2) + &
(myrho**2 + myzeta**2)**2*(px**2 + py**2)**2 + 2*a**4*(myrho**2*py**2 + myzeta
   **2*(px**2 + 3*py**2)))*ellipticE1- &
(a**4 - 2*a**2*myzeta*py + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*
   ellipticK1 ) + &
a**2*(-a**2 + myzeta*py)*((a**4 - 2*a**2*myzeta*py + (myrho**2 + myzeta**2)*(px
   **2 + py**2))*&
(a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*
   ellipticE1- &
(a**8 - 4*a**6*myzeta*py - 4*a**2*myzeta*(myrho**2 + myzeta**2)*py*(px**2 + py
   **2) + &
(myrho**2 + myzeta**2)**2*(px**2 + py**2)**2 + a**4*(-2*myrho**2*(px**2 - py**2)
   + 2*myzeta**2*(px**2 + 3*py**2)))*&
ellipticK1 )))/(a*(myrho**2 + myzeta**2)**3*px*(a**4 + 2*a**2*(myrho*px - myzeta*
   py) + (myrho**2&
+ myzeta**2)*(px**2 + py**2))*((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2
   + myzeta**2)*&
```

```
(px**2 +py**2))/(myrho**2 + myzeta**2))**1.5))
end


function I6Z (ellipticK1, ellipticE1)
use globalinfo
implicit none


Real*8 I6Z, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;


coeff=(a**2+(-1.0)*myrho**2+(-1.0)*myzeta**2)*(myrho**2+myzeta**2)**(-1.0/2.0);

I6Z= coeff* (  (-4*a*myzeta**2*((a**4 - (myrho**2 + myzeta**2)*(px**2 + py**2))*
    ellipticE1 &
+ (a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))
    *ellipticK1&
))/((myrho**2 + myzeta**2)**2*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 +
     myzeta**2)*(px**2 + py**2))*&
Sqrt((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py
    **2))/(myrho**2 + myzeta**2))))

end

function I7R (ellipticK1, ellipticE1)
use globalinfo
```

```
implicit none

Real*8 I7R, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;

coeff=(-1.0/2.0)*(a**2.0+(-1.0)*px**2.0+(-1.0)*py**2.0)*(myrho**2.0+myzeta**2.0)
   **(-3.0/2.0)*(( &
-1.0)*a**2.0+myrho**2.0+myzeta**2.0)

I7R=- coeff* ( (2*((-3*myzeta*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 +
     myzeta**2)*(px**2 + py**2))*&
(a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
(-((a**4 - 2*a**2*myzeta*py - (myrho**2 + myzeta**2)*(px**2 - py**2))*ellipticE1
   ) + &
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*
   ellipticK1 ))/&
(myrho**2 + myzeta**2) - a**2*(-((a**2*myzeta)/(myrho**2 + myzeta**2)) + py)*&
(-((a**8 - 4*a**6*myzeta*py - 4*a**2*myzeta*(myrho**2 + myzeta**2)*py*(px**2 + py
   **2) + &
(myrho**2 + myzeta**2)**2*(px**2 + py**2)**2 + 2*a**4*(myrho**2*(7*px**2 + py**2)
     +&
myzeta**2*(px**2 + 3*py**2)))*ellipticE1 ) + (a**4 - 2*a**2*myzeta*py + (myrho**2
     + &
myzeta**2)*(px**2 + py**2))*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 +&
myzeta**2)*(px**2 + py**2))*ellipticK1 + 2*(myrho**2 + myzeta**2)*px**2*(4*(a**4
   -&
2*a**2*myzeta*py + (myrho**2 + myzeta**2)*(px**2 + py**2))*ellipticE1 - &
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
```

```
ellipticK1 )))))/(a*px*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta
    **2)&
*(px**2 + py**2))**2*((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta
    **2)&
*(px**2 + py**2))/(myrho**2 + myzeta**2))**1.5))

end

function I7Z (ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I7Z, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;

coeff=(-1.0/2.0)*(a**2.0+(-1.0)*px**2.0+(-1.0)*py**2.0)*(myrho**2.0+myzeta**2.0)
    **(-3.0/2.0)*(( &
-1.0)*a**2.0+myrho**2.0+myzeta**2.0)

I7Z=- coeff* ((4*(myrho**2 + myzeta**2)**2*(((a**2*(myrho**2 + 4*myzeta**2) - 3*
    myzeta*(myrho**2 + myzeta**2)*py)*&
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
(a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*
    ellipticE1 )/&
(a*(myrho**2 + myzeta**2)**3) - (a*(-((a**2*myzeta)/(myrho**2 + myzeta**2)) + py)
    **2*&
(4*(a**4 - 2*a**2*myzeta*py + (myrho**2 + myzeta**2)*(px**2 + py**2))*ellipticE1-
     &
```

```
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticK1 ))/(myrho**2 + myzeta**2)))/((a**4 + 2*a**2*(myrho*px-myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))**2*((a**4 - 2*a**2*(myrho*px + myzeta*py)
    &
+ (myrho**2 + myzeta**2)*(px**2 + py**2))/(myrho**2 + myzeta**2))**1.5))

end

function I8R (ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I8R, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;

coeff=(-1.0/2.0)*(a**2.0+(-1.0)*px**2.0+(-1.0)*py**2.0)*(myrho**2.0+myzeta**2.0)
    **(-3.0/2.0)*(( &
-1.0)*a**2.0+myrho**2.0+myzeta**2.0)

I8R=- coeff* ( (4*myzeta*Sqrt(myrho**2 + myzeta**2)*((a**12 - 5*a**10*myzeta*py -
     a**8*(myrho**2 + &
5*myzeta**2)*(px**2 - 2*py**2) + (myrho**2 + myzeta**2)**3*px**2*(px**2 + py**2)
    **2 + &
a**2*myzeta*(myrho**2 + myzeta**2)**2*py*(3*px**4 + 2*px**2*py**2 - py**4) + &
2*a**6*myzeta*py*(myzeta**2*(7*px**2 - 5*py**2) + myrho**2*(9*px**2 - 3*py**2)) -
     &
a**4*(myzeta**4*(5*px**4 + 12*px**2*py**2 - 5*py**4) + 6*myrho**2*myzeta**2*(px
    **4&
```

```
+ 4*px**2*py**2 - py**4) + myrho**4*(px**4 + 12*px**2*py**2 - py**4)))*ellipticE1-
     &
(a**12 + a**10*(2*myrho*px - 5*myzeta*py) + (myrho**2 + myzeta**2)**3*px**2*(px
    **2 + py**2)**2 + &
a**2*(myrho**2 + myzeta**2)**2*(px**2 + py**2)*(2*myrho*px**3 - myzeta*py*(3*px
    **2 + py**2)) - &
a**4*(myrho**2 + myzeta**2)*(px**2 + py**2)*(2*myrho*myzeta*px*py + myrho**2*(px
    **2 - py**2)&
- myzeta**2*(px**2 + 5*py**2)) + a**8*(-6*myrho*myzeta*px*py - myrho**2*(px**2 -
    2*py**2) +&
myzeta**2*(px**2 + 10*py**2)) - 2*a**6*(-3*myrho*myzeta**2*px*py**2 + 3*myrho**2*
    myzeta*py**3 &
+ myzeta**3*py*(2*px**2 + 5*py**2) + myrho**3*(2*px**3 - px*py**2)))*ellipticK1 )
    )/&
(a*px*(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py
    **2))**2*&
(a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))
    **1.5))
end

function I8Z (ellipticK1, ellipticE1)
use globalinfo
implicit none

Real*8 I8Z, ellipticE1, ellipticK1,coeff
real*8 :: a
a=R;
```

```
coeff=(-1.0/2.0)*(a**2.0+(-1.0)*px**2.0+(-1.0)*py**2.0)*(myrho**2.0+myzeta**2.0)
    **(-3.0/2.0)*(( &
-1.0)*a**2.0+myrho**2.0+myzeta**2.0)

I8Z=- coeff* ( (4*a*myzeta*((-2*myzeta*(a**4 + 2*a**2*(myrho*px - myzeta*py) + &
(myrho**2 + myzeta**2)*(px**2 + py**2))*&
(a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticE1 )/&
(myrho**2 + myzeta**2) + (-((a**2*myzeta)/(myrho**2 + myzeta**2)) + py)*&
(2*(-a**2 + myzeta*py)*(4*(a**4 - 2*a**2*myzeta*py + (myrho**2 + myzeta**2)*(px
    **2 + py**2))*&
ellipticE1 - &
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticK1 ) + &
((a**8 - 4*a**6*myzeta*py - 4*a**2*myzeta*(myrho**2 + myzeta**2)*py*(px**2 + py
    **2) + &
(myrho**2 + myzeta**2)**2*(px**2 + py**2)**2 + &
2*a**4*(myrho**2*(7*px**2 + py**2) + myzeta**2*(px**2 + 3*py**2)))*&
ellipticE1 - &
(a**4 - 2*a**2*myzeta*py + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
(a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))*&
ellipticK1 )/a**2&
)))/((a**4 + 2*a**2*(myrho*px - myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py
    **2))**2*&
((a**4 - 2*a**2*(myrho*px + myzeta*py) + (myrho**2 + myzeta**2)*(px**2 + py**2))
    /&
(myrho**2 + myzeta**2))**1.5))

end
```

```
function LogTermThetaR(theta)
!---------------------------------------
!  Integrand coming from log term to integrate 3D
!---------------------------------------
use globalinfo
implicit none

real*8  phi,x2,coeffn,LogTermThetaR
real*8 theta,pxx,y1,y2,y3,x1,x3,pxy,ys1,ys2,ys3,pxs,pxxys
real*8 :: a
a=R;

y1 = myrho *cos(theta)
y2= myrho* sin(theta)
y3 = myzeta
x1 = px
x2= 0
x3 = py

pxy=(myrho**2.d0+myzeta**2.d0)**(1.d0/2.d0)
pxx=(px**2.d0+py**2.d0)**(1.d0/2.d0)

ys1 = a**2/pxy**2.d0*y1
ys2= a**2/pxy**2.d0*y2
ys3 = a**2/pxy**2.d0*y3

coeffn=   (-3.0*(a**2 - x1**2 - x2**2 - x3**2)*(a**2 - y1**2 - y2**2 - y3**2))/&
(2.0*a*(y1**2 + y2**2 + y3**2))
```

```
pxs=(ys1**2.d0+ys2**2.d0+ys3**2.d0)**(1.d0/2.d0)
pxxys=((x1-ys1)**2 + (x2-ys2)**2 +(x3-ys3)**2)**(1.d0/2.d0)

LogTermThetaR= coeffn *((((pxs*x1 + pxx*ys1)*(pxs*x3 + pxx*ys3))/ &
(pxx*(pxs*pxx + x1*ys1 + x2*ys2 + x3*ys3)**2) - &
(x1*ys3)/(pxx*(pxs*pxx + x1*ys1 + x2*ys2 + x3*ys3)) + &
((pxs - pxxys)*(pxs*(x1 - ys1) + pxxys*ys1)*(pxs*(x3 - ys3) + pxxys*ys3))/&
(pxxys**2*(-pxs**2 + pxs*pxxys + x1*ys1 + x2*ys2 + x3*ys3)**2) + &
((x1 - ys1)*(pxs**2*(x3 - ys3) + pxxys**2*ys3))/&
(pxxys**3*(-pxs**2 + pxs*pxxys + x1*ys1 + x2*ys2 + x3*ys3)))/pxs)
return
end

function LogTermThetaZ(theta)
!---------------------------------------
!  Integrand coming from log term to integrate 3D
!---------------------------------------
use globalinfo
implicit none

real*8  phi,x2,coeffn,LogTermThetaZ
real*8 theta,pxx,y1,y2,y3,x1,x3,pxy,ys1,ys2,ys3,pxs,pxxys
real*8 :: a
a=R;

phi=0
y1 = myrho *cos(theta)
y2= myrho* sin(theta)
```

```
y3 = myzeta

x1 = px

x2= 0

x3 = py


pxy=(myrho**2.d0+myzeta**2.d0)**(1.d0/2.d0)

pxx=(px**2.d0+py**2.d0)**(1.d0/2.d0)


ys1 = a**2/pxy**2.d0*y1

ys2= a**2/pxy**2.d0*y2

ys3 = a**2/pxy**2.d0*y3


coeffn=   (-3.0*(a**2 - x1**2 - x2**2 - x3**2)*(a**2 - y1**2 - y2**2 - y3**2))/&

(2.0*a*(y1**2 + y2**2 + y3**2))


pxs=(ys1**2+ys2**2+ys3**2)**(1.d0/2.d0)

pxxys=((x1-ys1)**2 + (x2-ys2)**2 +(x3-ys3)**2)**(1.d0/2.d0)


LogTermThetaZ=coeffn*(((pxs*x3 + pxx*ys3)**2/(pxx*(pxs*pxx + x1*ys1 + x2*ys2 + x3

    *ys3)**2) - &

(pxs*pxx + x3*ys3)/(pxx*(pxs*pxx + x1*ys1 + x2*ys2 + x3*ys3)) + &

((pxs - pxxys)*(pxs*(x3 - ys3) + pxxys*ys3)**2)/ &

(pxxys**2*(-pxs**2 + pxs*pxxys + x1*ys1 + x2*ys2 + x3*ys3)**2) + &

(pxs*pxxys**2*(-pxs + pxxys) + pxs**2*(x3 - ys3)**2 + pxxys**2*(x3 - ys3)*ys3)/&

(pxxys**3*(-pxs**2 + pxs*pxxys + x1*ys1 + x2*ys2 + x3*ys3)))/pxs )


return

end
```

REFERENCES

[1] Abaid, N., Adalsteinsson, D., Agyapong, A. & McLaughlin, R. M. 2004 An internal splash: Levitation of falling spheres in stratified fluids. *Phys. Fluids* **16** (5), 1567–1580.

[2] Abramowitz, M. & Stegun, I. 1965 *Handbook of Mathematical Functions*. Dover Publications

[3] Akers, B. & Belmonte, A. 2006 Impact dynamics of a solid sphere falling into a viscoelastic micellar fluid. *J. Non-Newton. Fluid* **135**, 97–108.

[4] Batchelor, G. K. 1967 *An Introduction to Fluid Dynamics.* Cambridge: Cambridge University Press.

[5] Bercovici, D. & Mahoney, J. 1994 Double flood basalts and plume head separation at the 660-kilometer discontinuity. *Science* **266**, 1367–1369.

[6] Brenner, H. 1961 The slow motion of a sphere through a viscous fluid towards a plane surface. *Chem. Eng. Sci.* **16**, 242–251.

[7] Byrd, P. F. & Friedman, M., D. 1954 *Handbook of elliptic integrals for engineers and physicists*. Berlin: Springer.

[8] Camassa, R., Falcon, C., Lin, J., McLaughlin, R. M. & Parker, R. 2009 Prolonged residence times for particles settling through stratified miscible fluids in the Stokes regime. *Phys. Fluids* **21**, 031702-1–4.

[9] Camassa, R., Falcon, C., Lin, J., McLaughlin, R. M. & Mykins, N. 2010 A first-principle predictive theory for a sphere falling through sharply stratified fluid at low Reynolds number. *J. Fluid Mech.* **664**, 436–465.

[10] Camassa, R., McLaughlin, R. M., Moore, M. & Vaidya, A. 2008 Brachistochrone paths in potential and stokes flow past a sphere. *Phys. Lett. A* .

[11] Camassa, R., Khatri, S., McLaughlin, R. M., Prairie, J. C., & White, B. L. 2013 Retention and entrainment effects: Experiments and theory for porous spheres settling in sharply stratified fluids. *Phys. Fluids* **25**, 081701-1–7

[12] Oseen, C. W. 1927 *Hydrodynamik*. Leipzig: Academische Verlagsgesellschaft M. B. H.

[13] Condie, S. A. & Bormans, M. 1997 The influence of density stratification on particle settling, dispersion and population growth. *J. of Theor. Biol.* **187**, 65–75.

[14] Darwin, C. 1953 Note on hydrodynamics. *Proc. Camb. Phil. Soc.* **49**, 342–254.

[15] Denman, K. L. & Gargett, A. E. 1995 Biological-physical interactions in the upper ocean: The role of vertical and small scale transport processes. *Annu. Rev. Fluid Mech.* **27**, 225–255.

[16] Economidou, M & Hunt, G. G. 2009 Density stratified environments: the double-tank method.*Exp.Fluids* **46**, 453-466.

[17] Happel, J. & Brenner, H. 1965 *Low Reynolds Number Hydrodynamics with Special Applications to Particulate Media.* Leyden: Noordhoff International Publishing.

[18] Happel, J. & Byrne, B. J. 1954 Motion of a sphere and fluid in a cylindrical tube. *Ind. Eng. Chem.* **46** (6), 1181–1186.

[19] Higdon, J. J. L. 1979 A hydrodynamic analysis of flagellar propulsion. *J. Fluid Mech.* **90** (4), 685–711.

[20] Jurine, D., Jaupart, C., Brandeis, G. & Tackley, P. J. 2005 Penetration of mantle plumes through depleted lithosphere. *J. Geopyhs. Res.* .

[21] Kellogg, W. W. 1980 Aerosols and climate. In *Interaction of Energy and Climate* (ed. W. Bach, J. Prankrath & J. Williams). D. Reidel.

[22] Lin, J. 2009 An Experimental and Mathematical Study on the Prolonged Residence Time of a Sphere Falling through Stratified Fluids at Low Reynolds Number. PhD thesis(Univ. North Carolina, North Carolina).

[23] MacIntyre, S., Alldredge, A. L. & Gottschalk, C. C. 1995 Accumulation of marine snow at density discontinuities in the water column. *Limnol. Oceanogr.* **40** (3), 449–468.

[24] Linton, C. M. 1995 Multipole methods for boundary-value problems involving a sphere in a tube. *IMA J. Appl. Math.* **55**, 187–204.

[25] Ma, T. & Wang, S. 2001 A generalized Poincaré-Hopf index formula and its applications to 2-D incompressible flows. *Nonlinear Anal–Real* **2**, 467–482.

[26] Manga, M. & Stone, H. A. 1995 Low reynolds number motion of bubbles, drops and rigid spheres through fluid-fluid interfaces. *J. Fluid Mech.* **287**, 279–298.

[27] Maxey, M. R. & Riley, J. J. 1983 Equation of motion for a small rigid sphere in a nonuniform flow. *Phys. Fluids* **26** (4), 883–889.

[28] Moore, M. N. J 2010 Stratified flows with vertical layering of density: theoretical and experimental study of the time evolution of flow configurations and their stability. PhD thesis(Univ. North Carolina, North Carolina).

[29] Oseen, C. W. 1927 *Hydrodynamik*. Leipzig: Academische Verlagsgesellschaft M. B. H.

[30] Parsons, J. D., Bush, J. W. M. & Syvitski, J. P. M. 2001 Hyperpycnal plume formation from riverine outflows with small sediment concentrations. *Sedimentology* **48**, 465–478.

[31] Prairie, J. C., Ziervogel, K., Arnosti, C., Camassa, R., Falcon, C., Khatri, S., McLaughlin, R. M., White, B. L., &Yu, S. 2013 Delayed settling of marine snow at sharp density transitions driven by fluid entrainment and diffusion-limited retention. *Mar.Ecol Prog Ser* **487**, 185–200

[32] Rard, J. A. & Miller, D. G. 1979 The mutual diffusion coefficients of NaCl–H_2O and $CaCl_2$–H_2O at 25^{o}C from Rayleigh interferometry. *J. Sol. Chem.* **8**, 701–716.

[33] Srdić-Mitrović, A. N., Mohamed, N. A. & Fernando, H. J. S. 1999 Gravitational settling of particles through density interfaces. *J. Fluid Mech.* **381**, 175–198.

[34] Sutor, M. M. & Dagg, M. J. 2008 The effects of vertical sampling resolution on estimates of plankton biomass and rate calculations in stratified water columns. *Estuar. Coast. Shelf Sci.* **78**, 107–121.

[35] Tanner, R. I.1963 End effects in falling-ball viscometry. *J. Fluid Mech.* **17**, 161–170.

[36] Taylor, G. 1953 Dispersion of soluble matter in solvent flowing slowly through a tube. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* pp. 186–203.

[37] Turco, R. P., Toon, O. B., Ackerman, T. P., Pollack, J. B. & Sagan, C. 1990 Climate and smoke: An appraisal of nuclear winter. *Science* **247** (4939), 662–665.

[38] Widder, E. A., Johnsen, S., Bernstein, S. A., Case, J. F. & Neilson, D. J. 1999 Thin layers of bioluminescent copepods found at density discontinuities in the water column. *Mar. Biol.* **134**, 429–437.

[39] Yick, K. Y., Torres, C. R., Peacock, T. & Stocker, R. 2009 Enhanced Drag of a Sphere Settling in a Stratified Fluid at Small Reynolds Number. *J. Fluid. Mech.* **632**, 49–68.