

---

# Study of the Flow Around a Wedge-Shaped Object in a Stratified Flow

Matthew Hurst

September 10, 2018

Dissertation submitted in partial fulfillment of the MSc Mathematical Modelling,

UCL

Supervised by:

Prof. Erik Burman, Department of Mathematics.

Prof. Edward Johnson, Department of Mathematics.

## Abstract

Sloping surfaces in a stratified fluid generate diffusion-driven flow, which produces a bulk fluid motion along the surfaces in an otherwise quiescent fluid. This phenomenon was first discovered and studied in the context of geophysical flows, such as those in the ocean and in fissures inside rocks. More recently, it was shown experimentally that this same phenomenon applies to a wedge-shaped object submerged in a stratified fluid, creating a net force on the wedge and causing it to move at constant velocity through the fluid at a speed on the order of centimeters per hour. The aim of this project is to study this flow numerically and to characterize the speed at which the wedge moves as a function of the governing parameters of any given physical scenario.

Three models of the flow around a two-dimensional wedge in a stratified fluid are developed: 1) a nonlinear, steady model based on the full conservation of momentum and conservation of mass for a diffusive fluid, 2) a time-dependent extension of this model, and 3) a linear model in the limiting case of a non-diffusive fluid. It is shown that both the steady and the non-diffusive models do not accurately replicate the physics of the problem because the fundamental assumptions of these models impose constraints on the physics under study. Assuming a steady solution places significant limitations on the convergence of the solution and assuming a non-diffusive fluid eliminates the propulsive effect of interest.

Using the time-dependent, nonlinear model, key experimental results are verified numerically. The structure of the flow qualitatively resembles the flow observed experimentally and the source of the forward propulsion is also explained based on the computational results. The speed at which the wedge moves is computed in several different cases with varying parameters; it is shown that this speed increases with an increase in the stratification of the fluid, decreases with an increase in wedge angle (but only for angles greater than  $5^\circ$ ), and increases with an increase in the diffusivity of the fluid.

Ultimately, these results serve to validate the numerical model, which could be used in the future for the analysis of this effect in more extreme cases, such as those with thermal stratification or more complicated geometries.

## Acknowledgments

I would like to extend my gratitude to my supervisors Prof Erik Burman and Prof Edward Johnson from the University College London (UCL) Department of Mathematics, who developed the initial idea for the research and provided invaluable guidance and expertise throughout the project. In particular, I thank Professor Burman for providing a starting point for the computational code and sharing his wealth of knowledge about the implementation of the finite element method. I thank Professor Johnson for assisting with the development of the theoretical framework and guiding me to a much better understanding of the problem as a whole.

In addition, I acknowledge Aaron Solomon for providing his expertise in setting up Amazon Web Services to run the more time-consuming simulations. I also thank Lauren Hurst for her thorough reading of drafts of this report and for providing helpful feedback.

I am grateful to my parents, Dale and Michelle Hurst, for supporting my education and consistently inspiring me to pursue my curiosity. I also thank Cassandra Giammo for her constant encouragement throughout my time as a graduate student and while working on this project.

Finally, I thank the Marshall Aid Commemoration Commission and all of those involved with the Marshall Scholarship for providing the funding for me to study at UCL and pursue this research project.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Theoretical Background of Diffusion-Driven Flow . . . . .	7
1.2	Flow Around Wedge-Shaped Objects . . . . .	9
<b>2</b>	<b>Problem Definition</b>	<b>13</b>
2.1	Geometric Setup . . . . .	13
2.2	Stationary Case . . . . .	15
2.2.1	Conservation of Momentum . . . . .	16
2.2.2	Conservation of Mass . . . . .	17
2.2.3	Boundary Conditions . . . . .	17
2.3	Time-Dependent Case . . . . .	18
2.3.1	Conservation of Momentum . . . . .	19
2.3.2	Conservation of Mass . . . . .	19
2.3.3	Initial and Boundary Conditions . . . . .	20
2.4	Non-dimensionalization . . . . .	20
2.4.1	Non-dimensional Variables . . . . .	20
2.4.2	Stationary Case . . . . .	22
2.4.3	Time-Dependent Case . . . . .	23
2.4.4	Initial and Boundary Conditions . . . . .	24
2.4.5	Discussion . . . . .	24
2.5	Non-Diffusive Limit . . . . .	25
<b>3</b>	<b>Numerical Implementation</b>	<b>26</b>
3.1	Weak Formulation . . . . .	26
3.1.1	Stationary Case . . . . .	26
3.1.2	Time-Dependent Case . . . . .	28
3.1.3	Non-Diffusive Limit . . . . .	28
3.2	Finite Element Method . . . . .	29
3.2.1	Mesh Discretization . . . . .	30
3.2.2	Interpolation, Galerkin Projection, Integration, and Solving .	31
3.3	Non-Diffusive Case Pressure Recovery . . . . .	31
3.4	Computing Forces . . . . .	32
3.5	Fixed Point Iteration . . . . .	33

3.6	Time Stepping . . . . .	34
3.7	Zero Finding . . . . .	36
<b>4</b>	<b>Numerical Results: Steady Case</b>	<b>37</b>
4.1	Negative Net Force . . . . .	37
4.2	Zero Net Force . . . . .	38
4.3	Positive Net Force . . . . .	40
4.4	High Stratification . . . . .	40
<b>5</b>	<b>Numerical Results: Time-Dependent Case</b>	<b>42</b>
5.1	Mesh Refinement Study . . . . .	42
5.2	Domain Size Study . . . . .	43
5.3	Experimental Case . . . . .	44
5.4	Stratification Study . . . . .	50
5.4.1	Schmidt Number = 1 . . . . .	50
5.4.2	Schmidt Number = 100 . . . . .	51
5.4.3	Schmidt Number = 1000 . . . . .	51
5.4.4	Discussion . . . . .	52
5.5	Wedge Slope Study . . . . .	53
5.6	Schmidt Number Study . . . . .	54
5.6.1	Low Stratification . . . . .	54
5.6.2	High Stratification . . . . .	55
5.7	Discussion . . . . .	57
<b>6</b>	<b>Numerical Results: Non-Diffusive Limit</b>	<b>58</b>
<b>7</b>	<b>Conclusion</b>	<b>60</b>
7.1	Summary of Research Conducted . . . . .	60
7.2	Strengths and Weaknesses . . . . .	60
7.3	Ideas for Future Work . . . . .	61

# 1 Introduction

Understanding the dynamics of fluid flow plays a crucial role in many fields of science and engineering. Aerodynamic models allow the flow over aircraft to be predicted, enabling the design of efficient aircraft. When geophysical flows in the oceans and atmosphere are understood, scientists can develop accurate predictions of weather and climate. A deeper understanding of the flow of fluid in pipes, pumps, compressors, and turbines facilitates innovative industrial design and power generation. The field of fluid dynamics is also closely coupled with thermodynamics and heat transfer and proves essential in the study of heat exchange and refrigeration.

Although fluid dynamics has been studied extensively for hundreds, if not thousands, of years, many open questions still exist and research in the field is ongoing. A relatively recent addition to the research in this field is the study of diffusion-driven flow, which is the subject of this project. Diffusion-driven flow has been shown to play a key role in geophysical fluid dynamics, for example in the melting of icebergs [1] and the motion of tectonic plates [2]. Recently, it was shown experimentally that diffusion-driven flow can result in the propulsion of asymmetric, submerged objects in a stratified fluid, i.e. one in which density varies in the vertical direction [3]. This effect causes such an object to “self-propel” through a fluid with no obvious cause. Since the publication of these experimental results in 2010, some additional work has been focused on studying this effect computationally. This research has largely aimed to understand the structure of such flows.

The aim of this project is to develop a model of the diffusion-driven flow around a wedge-shaped object in a stratified flow and to utilize this model to numerically study the impact of various physical parameters on the nature of the flow. Inspired by the 2010 paper, the main objective will be to predict the speed at which a wedge-shaped object will travel through a given fluid. While the 2010 paper explored the effect of physical parameters, such as the magnitude of the fluid stratification or the wedge angle, on this wedge speed, none of the recent numerical work has studied this. Computational work in fluid mechanics, as in many fields, allows scientists and engineers to predict outcomes without running physical experiments, which can be expensive and time consuming. A computational model of a problem allows for the efficient prediction of flows, thus enabling a better understanding of the relevant phenomena and intelligent design.

The remainder of this report will be organized as follows. First, a more detailed overview of the previous work will be presented. This will include an overview of the original papers on diffusion-driven flow from nearly 50 years ago and a review of both the experimental and computational work that has taken place since then. Throughout this overview of the relevant literature, the fundamentals of diffusion-driven flow will be detailed. Then, the governing equations and boundary conditions for the problem will be derived. The basic assumptions and non-dimensionalization procedure will also be covered in this section. Three models will be developed: a stationary nonlinear model, a time-dependent nonlinear model, and a linear model in the limiting case of a non-diffusive fluid. Next, the numerical finite element implementation will be discussed. Results from the numerical simulations will be presented, including an exploration of the effect of various physical parameters on the wedge speed using the time-dependent nonlinear model. Finally, an overview of the work conducted, a discussion of the strengths and weaknesses of the approach, and ideas for future work will be discussed.

## 1.1 Theoretical Background of Diffusion-Driven Flow

In 1970, Wunsch [4] and Phillips [5] published papers describing the effects of sloping boundaries in stably stratified diffusive fluids. Stability in this case refers to density decreasing with height. If the sidewalls of a salt-stratified fluid are vertical, the isopycnals (lines of constant density) remain horizontal, as is natural for a stably stratified fluid. The impermeability of the walls necessitates that there is no salt flux across them. This means that the gradient of density must vanish at the walls, which results in the isopycnals intersecting the walls at right angles. However, when the sidewalls are sloping, their impermeability requires the isopycnals curve to meet the surface at a right angle. This effect is shown in Fig. 1. In this figure and others, gravity is pointing down.

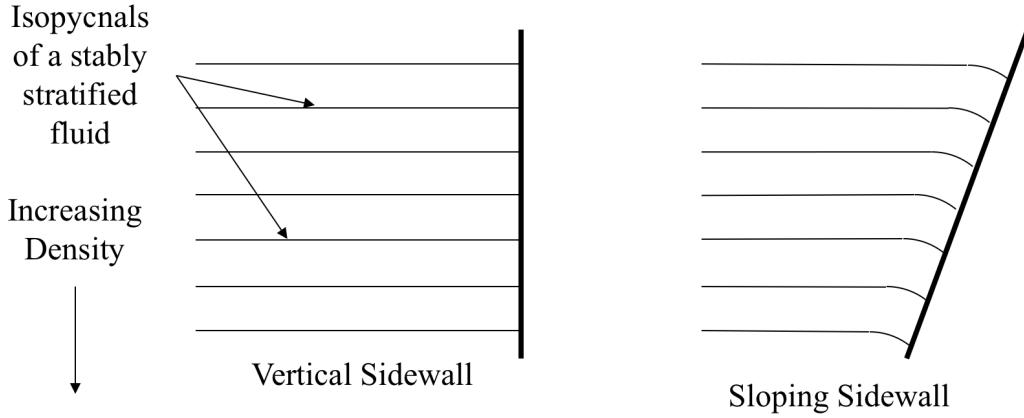


Figure 1: The effect of sloping, impermeable sidewalls in a stably stratified fluid [4] [5]

In the case of a sloping sidewall, the fluid is no longer in hydrostatic equilibrium. Due to the curved isopycnals, the fluid near the wall is less dense than the adjacent fluid. This causes a bulk motion of fluid up the sidewall, until the viscous forces counteract this pressure imbalance [5]. This is known as diffusion-driven flow, or buoyancy-driven flow, because it is caused by the combined effects of diffusion and buoyancy.

Phillips presents another way of thinking about the same phenomenon using a conservation argument about the density flux [5]. In the stably stratified fluid, the diffusive flux is constant and upward due to the linear density gradient. Away from the sloping sidewall, the diffusive flux at a point is supplied by the diffusive flux below that point. Just above the sloping sidewall, there must also be a diffusive flux upward, but just below there is not a diffusive flux upward to balance this. Thus, a convective flux upward along the sloping wall must exist in order to feed this diffusive flux. This is visualized in Fig. 2, a re-created image from the Phillips paper.

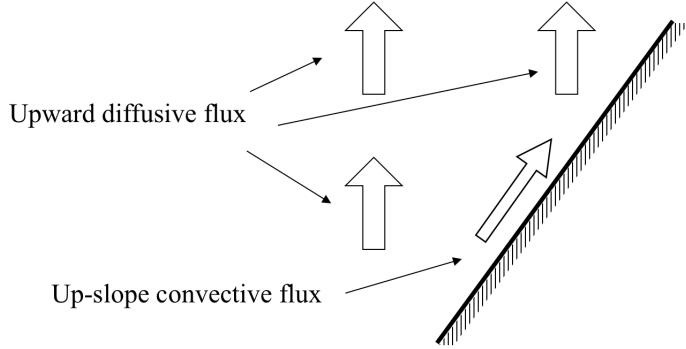


Figure 2: Visualization of convective flux along a sloping wall [5]

In both frameworks, a bulk fluid motion up the sloping sidewall is generated. Wunsch and Phillips derived theoretical results describing the flow near the wall, including the velocity profile in this boundary layer. These results will be used later in this project. The original application of their work was understanding oceanic mixing and mineral transport in rocks.

Following the initial work, Huppert and Turner studied this effect in melting icebergs [1]. They showed that when melting icebergs as a source of fresh water, the effect of the density gradient in the top 250 m of the ocean facilitates the production of diffusion-driven flow, which in turn means that isolating fresh water from salt water would prove more difficult than theorized. Gurnis then showed the application of this theory in modeling convection in the Earth's mantle [2]. While diffusion-driven flow is a relatively small effect in salt-stratified fluids, it can be much larger in thermally stratified fluids involving liquid metals.

## 1.2 Flow Around Wedge-Shaped Objects

Prior to 2010, much of the work in diffusion-driven flow was focused on developing a better understanding of geophysical fluids. However, in 2010, Allshouse et. al. published a paper [3] describing the results of an experiment involving a wedge-shaped object submerged in a tank of stably stratified fluid. They showed that the diffusion-driven flow along the sloping surfaces of the wedge creates a bulk fluid motion that results in regions of relative low pressure near the sloping surfaces. These low pressure regions create a forward propulsive force, which causes the wedge to accelerate. This propulsive force is counter-balanced by a viscous drag, resulting in the wedge moving through the fluid at constant velocity.

In the experiment, a wedge was lowered into the fluid and settled at the point of neutral buoyancy. A side view of this experimental setup is re-created in Fig. 3. In three dimensions, this wedge cross section is extended horizontally in the fluid.

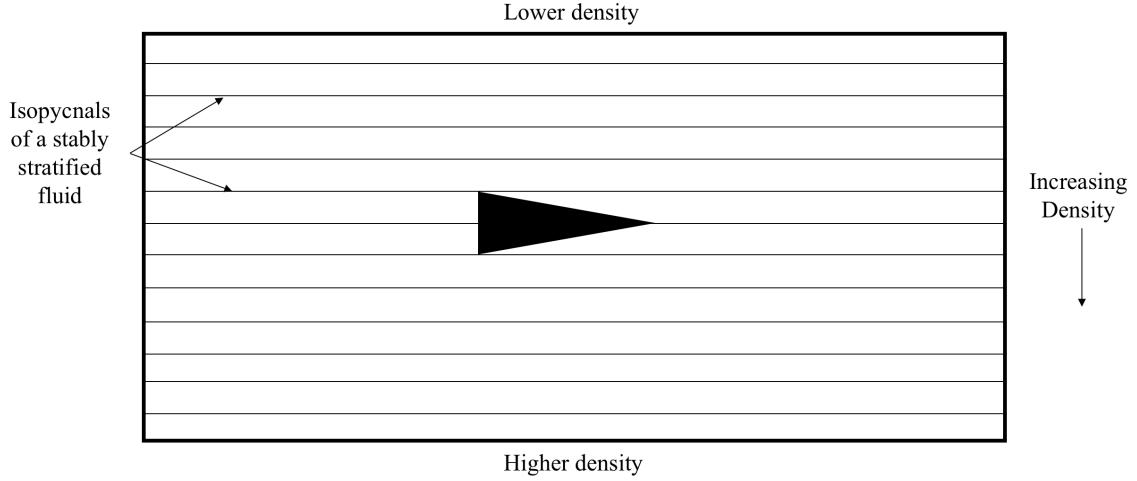


Figure 3: Orientation of the wedge in the stratified fluid, side view

Since the wedge is impermeable, the same effect described by Wunsch and Phillips is present on the sloping walls of the wedge: the isopycnals bend so that there is no density flux across the surface of the wedge. Then, fluid particles on the top surface of the wedge are less dense than those adjacent in the surrounding fluid. Thus, they move up the surface of the wedge, creating a bulk convection along the sloping surfaces of the wedge. The converse is true on the bottom surface. This is illustrated in Fig. 4.

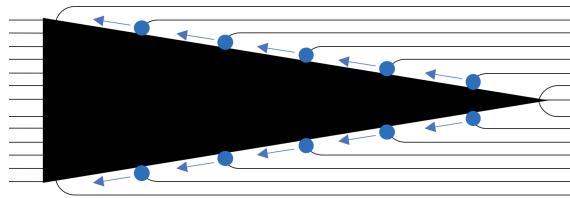


Figure 4: Visualizing the source of the convective flow along the sloping surfaces of the wedge

This fluid motion creates regions of relative low pressure above and below the top and bottom sloping surfaces of the wedge, respectively, which result in a net force to the right, as shown in Fig. 5.

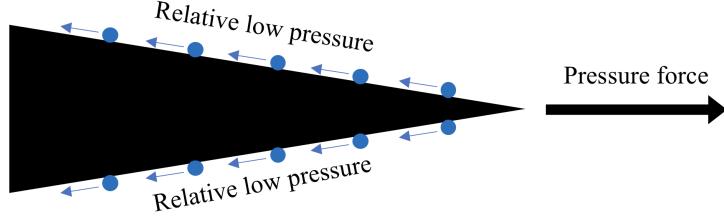


Figure 5: Pressure force generated by diffusion-driven flow

The pressure force causes the wedge to accelerate to the right, but as the wedge velocity increases, the viscous force pulling to the left also increases. When these two forces balance, the wedge moves at constant velocity. This is illustrated in Fig. 6.

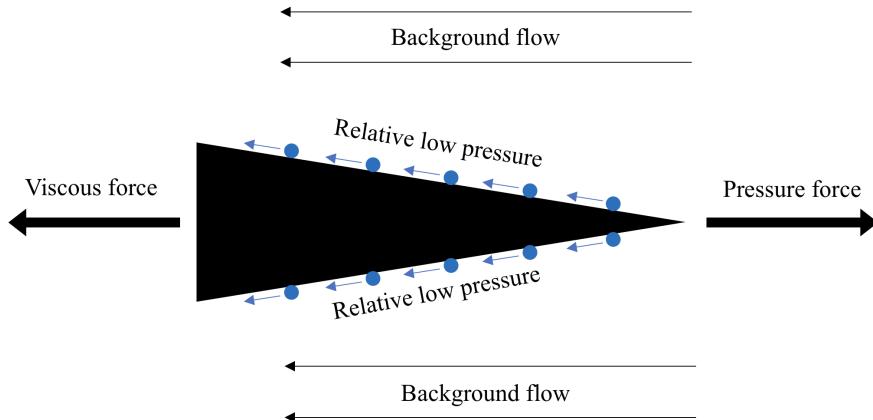


Figure 6: Wedge with zero net force moving at constant velocity to the right

Note that only horizontal force is considered. It is assumed that the wedge is settled at the vertical location of neutral buoyancy. For the remainder of this report, only the horizontal force will be studied.

Allshouse et. al. provided several key experimental results [3] that will be discussed later in this paper. In addition to these experimental results, they conducted a numerical simulation with limited scope. In this simulation, the net force on a stationary wedge in stationary fluid was computed, then a separate simulation was conducted in a non-stratified fluid to determine the background velocity that would result in a drag of the magnitude of the propulsion seen in the stratified case. This background velocity was provided as an estimate for the speed at which the wedge would travel. This approach assumes that the nonlinear effects can be ignored and that the two flows can be added. In this project, both effects (diffusion-driven

propulsion and viscous drag) will be studied in the same simulation by considering a moving wedge. Also, Allshouse et. al. conducted the numerical simulation at a different set of parameter values than the experiment. In particular, the Schmidt number, which describes viscosity and diffusivity, was order 1 in the simulations and order 1000 in the experiments. It will be shown later that this has a significant effect on the solution. The key contribution of Allshouse et. al. was to show experimentally that diffusion-driven flow, which had been studied for decades, could create a propulsive force on an asymmetric object submerged in a stably stratified fluid.

It is worth noting that this propulsion seems to come from nowhere and thus be “free energy,” but in fact, the energy is derived from the density gradient itself. In order to maintain the density gradient, solute must be pumped in to the bottom of the tank. This is the source of energy that allows the wedge to move [3] [6].

In 2014, Mercier et. al. studied this effect in more detail in two fluid systems, showing that the natural convection due to a heated surface creates the same effect [7]. Both density-stratified and thermally-stratified fluids can exhibit this property. In the latter case, the relevant mechanism is thermal diffusion, not molecular diffusion, and the sloping surfaces must be thermally insulating, not impermeable to solute. Due to the relative strength of the thermal effects, the propulsion in this case can be much more significant.

Recent computational work by Chashechkin [8] and Dimitrieva [9] has further explored the structure of the flow generated around the wedge and Doyle [10] carried out numerical studies to model the effect in a thermally stratified fluid.

However, no known work to date has attempted to numerically study the phenomena described in the 2010 experimental paper, in particular the dependence of the resultant wedge speed on the physical parameters of the fluid and the wedge. This is the aim of the present work: to characterize the dependence of the steady wedge speed on magnitude of the background stratification, the diffusivity of the fluid, and the wedge angle. In the next section, the governing equations of diffusion-driven flow will be derived.

## 2 Problem Definition

The goal of this project is to study the flow generated by and surrounding a wedge-shaped object in a stratified fluid. In particular, the focus is on characterizing the flow both qualitatively and quantitatively and predicting the steady propulsive speed of a wedge through a stably stratified fluid.

The working begins with a problem definition. This includes a description of the problem geometry, a discussion and derivation of the governing equations and boundary conditions, and a description of the non-dimensionalization procedure.

Initially, the steady (or stationary) equations are studied because the behavior under study is that of the wedge moving at constant velocity through the flow. Although the fluid is moving, the problem is stationary because the solution variables do not change in time.

The time-dependent case is also studied for two primary reasons: adding time dependence helps to facilitate convergence in some cases and at high levels of stratification, the solution is inherently unsteady.

Finally, the limiting case of a slow-moving, non-diffusive fluid is presented. This so-called non-diffusive limit greatly simplifies the governing equations and results in a linear system.

In all of the following working, the basic equations of fluid mechanics are assumed and not derived for the sake of brevity. The simplifications and transformations of these equations, however, will be described.

### 2.1 Geometric Setup

As previously discussed, the problem geometry under consideration is a wedge-shaped object submerged in a tank of stably stratified fluid. The previous experimental work considered a three-dimensional wedge with a finite ratio of length to width. In this project, the case of an infinite length to width ratio is considered, i.e. a wedge with a triangular cross section that is extended to infinity. When studying this problem, the physics can be reduced to two dimensions as it can be assumed that the flow is identical in each cross section of the three-dimensional fluid. More simply, any three-dimensional effects that arise due to the sides of the wedge and the corresponding edges are ignored. This decision was made for the sake of simplicity and computational tractability. For the remainder of this report, integrated surface

quantities will be computed per unit width. For example, a pressure integrated over the surface of the wedge will be in units of N/m and will represent pressure force per unit width. Non-dimensional force will represent non-dimensional force per unit non-dimensional width.

The wedge being studied is shown in Fig. 7.

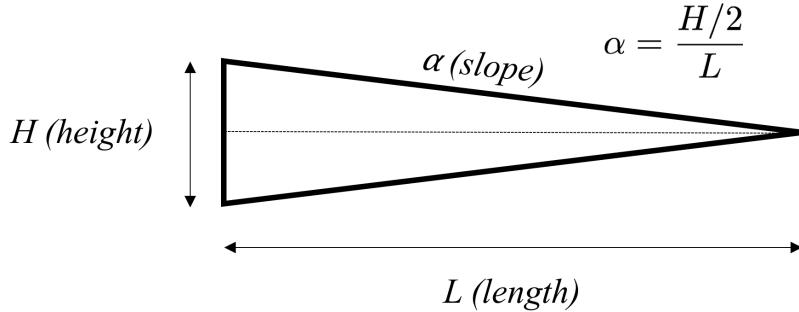


Figure 7: Wedge geometry and key geometric variables

The wedge height is given by  $H$  and the length is given by  $L$ . The aspect ratio of the wedge, or the slope of both the top and bottom surfaces, is determined by the ratio of the half-height to the length. A slope of zero ( $\alpha = 0$ ) corresponds to the limiting case of a flat plate. For the remainder of this project, the wedge geometry will be parameterized by the length  $L$  and slope  $\alpha$ ; the height  $H$  is simply a function of these two.

The edges of the tank will be denoted  $\Gamma_1$  and the boundaries of the wedge will be denoted  $\Gamma_2$ . The tank shape under consideration will always be rectangular and is parameterized by variables `xlength` and `zlength` which represent one half of the total width and height of the tank respectively (as viewed from the side). This setup is shown in Fig. 8. Note that the coordinate system is  $(x, z)$ , with  $x$  as the horizontal coordinate and  $z$  as the vertical coordinate. Gravity is in the  $-z$  direction and the wedge is expected to move in the  $+x$  direction. The coordinate frame is centered vertically at the mid-line of the wedge and horizontally halfway between the sharp corner and blunt back edge of the wedge.

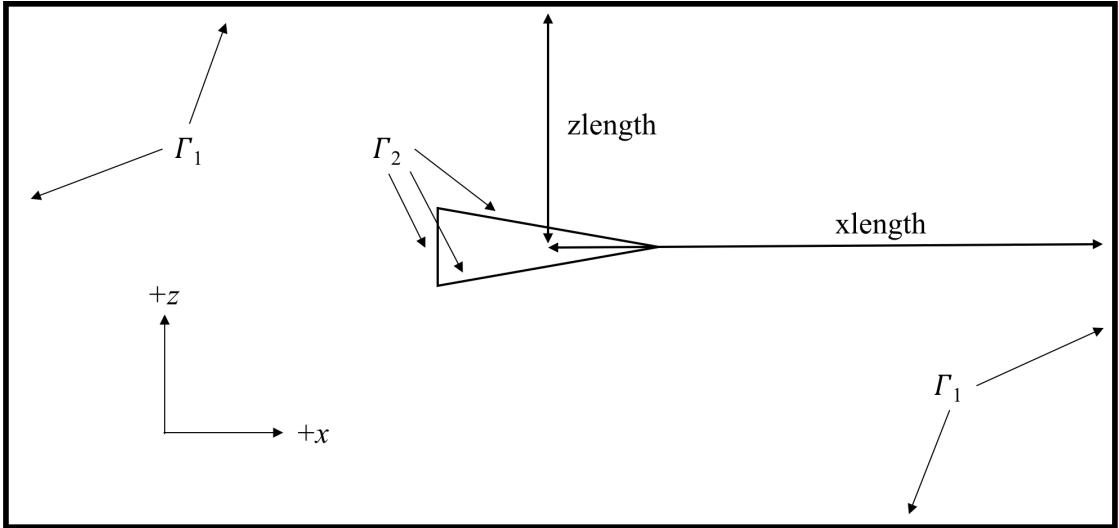


Figure 8: Illustration of the wedge submerged in the tank

In the physical case, the wedge would move slowly through a stationary fluid. Instead of simulating this directly, the coordinate frame will be fixed and the surrounding fluid will be moving over the wedge. Prescribing the fluid velocity on the edge of the domain is equivalent to the wedge moving at that velocity through the fluid. This is an important consideration that will be especially relevant later in this report when computing the net force on the wedge. Throughout the rest of this report, the term background velocity will sometimes be used to denote the steady wedge speed.

In the experimental setup, the tank size was selected so that the boundaries would have no effect on the flow around the wedge. This will be taken under consideration later in this report when determining values for `xlength` and `zlength`. With the geometric setup defined, the governing equations will now be derived.

## 2.2 Stationary Case

First, the stationary (steady) case will be presented. The governing equations used here are the fundamental equations of fluid mechanics. Importantly, conservation of momentum and conservation of mass are considered, but not conservation of energy as it is assumed that temperature is uniform and there is no internal heat generation or external energy forcing. As described earlier, the problem is limited to two spatial dimensions.

### 2.2.1 Conservation of Momentum

Consider the momentum equation (2.1) for a steady, viscous fluid under a gravitational force, where  $\rho$  is density,  $\mathbf{u}$  is velocity with  $x$  and  $z$  components  $u$  and  $w$ , respectively,  $p$  is pressure,  $\mathbf{g}$  is the gravity vector in the direction of gravity with magnitude of  $9.81 \text{ m/s}^2$ , and  $\mu$  is dynamic viscosity.

$$\rho(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u} \quad (2.1)$$

This partial differential equation (PDE) is nonlinear, vector-valued, and contains the three unknown solution variables velocity, pressure, and density. Consider perturbations of density and pressure from their background, hydrostatic values  $\rho_0$  and  $p_0$  respectively:  $\rho = \rho_0 + \tilde{\rho}$  and  $p = p_0 + \tilde{p}$ . These background values are linked by the relation  $\nabla p_0 = \rho_0 \mathbf{g}$  [11]. Note that  $\rho_0$  is constant and  $p_0$  increases linearly in the direction of gravity.

$$(\rho_0 + \tilde{\rho})(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla(p_0 + \tilde{p}) + (\rho_0 + \tilde{\rho})\mathbf{g} + \mu \nabla^2 \mathbf{u} \quad (2.2)$$

Using the relation  $\nabla p_0 = \rho_0 \mathbf{g}$  and dropping the tildes for simplicity,

$$(\rho_0 + \rho)(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u}, \quad (2.3)$$

where the variables  $\rho$  and  $p$  now represent perturbations from their background values. Note that no assumption has yet been made about the relative magnitude of these perturbations. These perturbations can sometimes be assumed to be small relative to their background values. However for highly stratified fluids this is not necessarily the case. Instead, the perturbation  $\rho$  can be further decomposed into a component due to the background density which is linearly varying (with slope  $\partial \rho / \partial z$ ) and a perturbation from that linear variation. Note that  $\partial \rho / \partial z$  is a constant parameter of the problem that defines the background stratification, not a point-wise derivative of density. The perturbation from that linear variation is assumed to be small relative to both the constant reference value and the background stratification  $\partial \rho / \partial z$ . This is a modified form of the Boussinesq approximation for a stratified fluid [11]. The small density term can now be ignored in the convective term, but must be retained in the buoyancy term. Dropping the tilde again for simplicity, (2.3) can

be written as

$$\left(\rho_0 + z\frac{\partial\rho}{\partial z}\right)(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u}. \quad (2.4)$$

In (2.4),  $p$  and  $\rho$  are still perturbations from their hydrostatic reference values.

Assuming that the fluid is incompressible, the divergence of velocity must also be zero.

$$\nabla \cdot \mathbf{u} = 0 \quad (2.5)$$

The system (2.4) and (2.5) represents conservation of momentum for an incompressible, diffusive, linearly stratified fluid.

### 2.2.2 Conservation of Mass

Next consider the continuity equation (conservation of mass) for a steady, diffusive fluid,

$$\nabla \cdot (\rho \mathbf{u}) - \nabla \cdot (\kappa_0 \nabla \rho) = 0, \quad (2.6)$$

where the first and second terms represent convection and diffusion respectively. The parameter  $\kappa_0$  is the molecular diffusivity and is assumed constant throughout the fluid, so upon expanding the divergence operators,

$$\nabla \rho \cdot \mathbf{u} + \rho \nabla \cdot \mathbf{u} - \kappa_0 \nabla^2 \rho = 0. \quad (2.7)$$

Upon substituting (2.5) into (2.7), the equation governing the convection and diffusion of density through the fluid is found.

$$\mathbf{u} \cdot \nabla \rho = \kappa_0 \nabla^2 \rho \quad (2.8)$$

The density variable used here still includes the hydrostatic value. Now assume the same perturbation from a constant background value:  $\rho = \rho_0 + \tilde{\rho}$ . Since density only appears under a differential in (2.8) and  $\rho_0$  is constant, this equation also holds for the variations  $\tilde{\rho}$ . Dropping the tilde, (2.8) is the convection-diffusion equation for density, where  $\rho$  is the difference in density from a constant background value.

### 2.2.3 Boundary Conditions

The equations presented in the previous subsections must be supplemented with boundary conditions before they can be solved.

In the far-field, the flow is uniform in the negative  $x$  direction with speed  $U$ . The density (difference from background) in the far-field varies linearly as in a stably stratified fluid. This linear variation is assumed to take the form  $\rho = -\beta_0 \rho_0 z/L$ , where  $L$  is the length of the wedge and the unit-less parameter  $\beta_0$  describes the magnitude of the stratification; in particular,  $\frac{\partial \rho}{\partial z} = -\beta_0 \rho_0/L$ , so the magnitude of  $\beta_0$  describes the magnitude of stratification. This form is assumed in order to simplify the non-dimensionalization later.

On the wedge, no slip and no diffusion are enforced. To apply the no flux condition, there must be no change in density across the wedge, i.e.  $\partial \rho / \partial n = 0$ , where  $n$  is the outward normal. These boundary conditions are shown in (2.9) and (2.10) below.

$$(u, w, \rho) \rightarrow (-U, 0, -\beta_0 \rho_0 z/L) \quad \text{as } x^2 + z^2 \rightarrow \infty \quad (2.9)$$

$$(u, w, \rho_n) = (0, 0, 0) \quad \text{on the wedge} \quad (2.10)$$

In practice, the domain is non-infinite and the far-field conditions are imposed on the boundaries of the tank. Using the labels in Fig. 8, (2.9) is enforced on  $\Gamma_1$  and (2.10) is enforced on  $\Gamma_2$ .

Together, (2.4), (2.5), and (2.8) describe the physics of a steady, incompressible, viscous, diffusive fluid under a gravitational force. Along with (2.9) and (2.10), they form a complete system.

### 2.3 Time-Dependent Case

As discussed earlier, the desired solution is the stationary case with zero net force on the wedge. However, for high levels of stratification, a stationary solution may not exist due to vortex shedding or other unsteady processes. This unsteadiness will result in a periodic solution and the macroscopic behavior of the wedge will be the time average of this solution. In addition to cases where unsteady behavior occurs, adding time dependence assists with the convergence of the numerical method. If a stationary solution exists, the time dependent solution should converge to it.

The derivation of the governing equations presented here will be brief because it only differs slightly from that of the stationary case.

### 2.3.1 Conservation of Momentum

The equation for conservation of momentum in the time-dependent case is the same as that of the stationary case (2.1) with the addition of the time derivative of momentum.

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u} \quad (2.11)$$

The above equation is in its dimensional form with  $t$  representing time in seconds. Expanding the time differential yields

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \frac{\partial \rho}{\partial t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u}. \quad (2.12)$$

Recall that  $\rho$  is made up of three components: a constant background value, a linearly varying part due to the stratification, and a small deviation. Because the first two components of this do not vary in time and are assumed to be much larger than the small deviation, the time derivative of density can be ignored. In addition, the density terms can be rewritten as before to simplify further. Now, the variable  $\rho$  represents the perturbation from a constant background value and the variable  $p$  represents the perturbation from a linearly varying hydrostatic pressure field.

$$\left( \rho_0 + z \frac{\partial \rho}{\partial z} \right) \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u}. \quad (2.13)$$

### 2.3.2 Conservation of Mass

Next the equation of conservation of mass for the time-dependent case is derived. Again, the conservation law here is very similar to that of the stationary case with the addition of a time-dependent term. In particular, adding unsteadiness to the convection-diffusion equation in (2.6) yields

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) - \nabla \cdot (\kappa_0 \nabla \rho) = 0. \quad (2.14)$$

Imposing incompressibility and writing density as the sum of its uniform background value and a perturbation yields (2.15) below.

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = \kappa_0 \nabla^2 \rho \quad (2.15)$$

### 2.3.3 Initial and Boundary Conditions

The boundary conditions for the time-dependent case are identical to those of the stationary case, but must now be supplemented with an initial condition. The choice of initial condition is somewhat arbitrary because the transient solution is not being studied. The goal of the time-dependent solutions is to allow for periodic steady states and to facilitate convergence; as long as the initial condition does not cause the numerical solution to become unstable or converge to some non-physical solution, the initial condition should not matter. With that said, the initial condition chosen will always be a fluid with a uniform velocity field and density matching the background stratification. Recall that this pressure and density are deviations from hydrostatic values, so non-positive values are permitted.

For completeness, the initial and boundary conditions are given below.

$$(u, w, \rho) = (-U, 0, -\beta_0 \rho_0 z / L) \quad \text{at } t = 0 \quad (2.16)$$

$$(u, w, \rho) \rightarrow (-U, 0, -\beta_0 \rho_0 z / L) \quad \text{as } x^2 + z^2 \rightarrow \infty \quad (2.17)$$

$$(u, w, \rho_n) = (0, 0, 0) \quad \text{on the wedge} \quad (2.18)$$

Along with (2.13), (2.15), and (2.5), these equations describe the unsteady problem.

## 2.4 Non-dimensionalization

Before solving, the system will be non-dimensionalized in order to determine the unit-less parameters that define the solution and to simplify the discussion of the simulations run with different sets of physical parameters.

### 2.4.1 Non-dimensional Variables

First, several reference variables are identified. The characteristic length scale  $L$  is chosen to be the length of the wedge. Note that the height  $H$  could be introduced as a second characteristic length scale for the  $z$  direction. This would result in a more complicated transformation of the Laplacian operator requiring the introduction of the aspect ratio  $\alpha$  into the equations. For the sake of simplifying the equations and leaving this additional complexity in the boundary condition, only one length scale  $L$  will be used.

Two options are available for a reference velocity. In many problems in fluid

mechanics, the reference velocity is chosen to be the background velocity [12] because this velocity determines the nature of the flow. In this case, choosing the background velocity as a reference is limiting because it does not allow for the solution with zero background velocity to be computed. In addition, the background velocity is not the most relevant scaling in the problem. Instead, a characteristic tangential velocity for diffusion-driven flow was selected [5]. In particular, the reference velocity  $u_0$  is given by

$$u_0 = 2\kappa_0 \gamma \cot(\theta), \quad (2.19)$$

where  $\theta$  is the angle-equivalent of the aspect ratio  $\alpha$  given by  $\theta = \tan^{-1}(\alpha)$ . The parameter  $\gamma$  is inversely related to the length scale of the boundary layer formed next to the sloping surface. This parameter [5] is given by

$$\gamma = \left( \frac{N^2 \sin^2(\alpha)}{4\nu\kappa_0} \right)^{1/4}, \quad (2.20)$$

where  $N$  is the buoyancy frequency [11] given by

$$N^2 = -\frac{g}{\rho_0} \frac{\partial \rho}{\partial z}. \quad (2.21)$$

Note that several new parameters have been introduced, but they merely serve as intermediaries in the computation of the reference velocity. Selecting diffusivity  $\kappa_0$ , aspect ratio  $\alpha$ , kinematic diffusivity  $\nu$ , reference density  $\rho_0$ , and far-field stratification  $\partial\rho/\partial z$  fixes the reference velocity. The buoyancy frequency  $N$  will also be used to define a characteristic time for the unsteady case.

Non-dimensional spatial coordinates  $\tilde{x} = x/L$  and  $\tilde{z} = z/L$  are now introduced. Using the chain rule, the differential operators transform as  $\frac{\partial}{\partial x} = \frac{1}{L} \frac{\partial}{\partial \tilde{x}}$  and  $\frac{\partial}{\partial z} = \frac{1}{L} \frac{\partial}{\partial \tilde{z}}$ . For now,  $\tilde{\nabla}$  will represent the vector of partial derivatives with respect to these non-dimensional coordinates. Like the partial differential operators, the gradient and divergence operators with respect to the non-dimensional coordinates are scaled by the factor  $1/L$ .

The non-dimensional time  $\tilde{t} = tN$  is also introduced. By the chain rule,  $\frac{\partial}{\partial t} = N \frac{\partial}{\partial \tilde{t}}$ .

The velocity  $\mathbf{u}$  is non-dimensionalized on the characteristic velocity  $u_0$ , so that

$$\tilde{\mathbf{u}} = (\tilde{u}, \tilde{w})^T = \left( \frac{u}{u_0}, \frac{w}{u_0} \right)^T = \frac{\mathbf{u}}{u_0}. \quad (2.22)$$

Pressure is non-dimensionalized on  $\mu u_0/L$  and density on  $\beta_0 \rho_0$  so that

$$\tilde{p} = \frac{p}{\left(\frac{\mu u_0}{L}\right)} \quad \text{and} \quad \tilde{\rho} = \frac{\rho}{\beta_0 \rho_0}. \quad (2.23)$$

#### 2.4.2 Stationary Case

These non-dimensional operators and variables are now substituted into the equation of steady conservation of momentum (2.4).

$$\frac{(\rho_0 - \rho_0 \beta_0 \tilde{z}) u_0^2}{L} (\tilde{\mathbf{u}} \cdot \tilde{\nabla} \tilde{\mathbf{u}}) = -\frac{\mu u_0}{L^2} \tilde{\nabla} \tilde{p} + \beta_0 \rho_0 \tilde{\rho} \mathbf{g} + \frac{\mu u_0}{L^2} \tilde{\nabla}^2 \tilde{\mathbf{u}}. \quad (2.24)$$

Rearranging and canceling, using  $\mathbf{g} = -g \hat{\mathbf{k}}$ , where  $\hat{\mathbf{k}}$  is the unit vector in the  $z$ -direction,

$$\frac{u_0 L \rho_0}{\mu} (1 - \beta_0 \tilde{z}) (\tilde{\mathbf{u}} \cdot \tilde{\nabla} \tilde{\mathbf{u}}) = -\tilde{\nabla} \tilde{p} - \frac{\beta_0 \rho_0 g L^2}{(u_0 \mu)} \tilde{\rho} \hat{\mathbf{k}} + \tilde{\nabla}^2 \tilde{\mathbf{u}}. \quad (2.25)$$

Here it becomes clear that Reynolds number,  $\text{Re} = u_0 L \rho_0 / \mu$  (the ratio of momentum to viscosity) controls the magnitude of the convective term. The effect of assuming small density perturbations relative to the stably stratified fluid has also become apparent. The term  $(1 - \beta_0 \tilde{z})$  modifies the local Reynolds number at different vertical locations throughout the fluid. In particular, the local Reynolds number decreases with an increase in height. Assuming that density perturbations were small relative to a constant value would result in a constant Reynolds number throughout the fluid.

The non-dimensional parameter  $\beta = \beta_0 \rho_0 g L^2 / (u_0 \mu)$  is also introduced in order to simplify the coefficient of the buoyancy term. Dropping the tildes and using these definitions, the non-dimensional momentum equation is

$$\text{Re}(1 - \beta_0 z)(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p - \beta \rho \hat{\mathbf{k}} + \nabla^2 \mathbf{u}. \quad (2.26)$$

For completeness, this equation is written in scalar form below.

$$\text{Re}(1 - \beta_0 z)(uu_x + wu_z) = -p_x + \nabla^2 u \quad (2.27)$$

$$\text{Re}(1 - \beta_0 z)(uw_x + ww_z) = -p_z - \beta \rho + \nabla^2 w \quad (2.28)$$

It is important to note that the incompressibility equation does not change form upon non-dimensionalization, i.e. the non-dimensional velocity must also be diver-

gence free.

Two parameters determine which terms dominate in the problem:  $\text{Re}$  and  $\beta$ . The Reynolds number describes the ratio of momentum to viscosity and the stratification parameter  $\beta$  describes the ratio of buoyancy to viscosity. In a problem dominated by convection, the Reynolds number will be large. In a problem with strong stratification,  $\beta$  will be large.

Returning to the non-dimensional variables and operators represented by the tilde variables and substituting these into (2.8), the following PDE can be found.

$$\frac{u_0 \beta_0 \rho_0}{L} \tilde{\mathbf{u}} \cdot \tilde{\nabla} \tilde{\rho} = \frac{\kappa_0 \beta_0 \rho_0}{L^2} \tilde{\nabla}^2 \tilde{\rho} \quad (2.29)$$

Cancelling and rearranging, it is found that

$$\tilde{\mathbf{u}} \cdot \tilde{\nabla} \tilde{\rho} = \frac{\kappa_0}{u_0 L} \tilde{\nabla}^2 \tilde{\rho}. \quad (2.30)$$

Another non-dimensional parameter  $\kappa = \kappa_0/(u_0 L)$  is introduced, representing the ratio of molecular diffusivity to momentum. Upon inserting into the above equation and dropping the tildes,

$$\mathbf{u} \cdot \nabla \rho = \kappa \nabla^2 \rho. \quad (2.31)$$

For completeness, the same equation is written below in component form.

$$u \rho_x + w \rho_z = \kappa \nabla^2 \rho \quad (2.32)$$

### 2.4.3 Time-Dependent Case

The non-dimensionalization procedure for the unsteady equations is very similar to that of the stationary equations, so it will be discussed briefly. Substituting the non-dimensional operators and variables into (2.13), the following equation is found.

$$(\rho_0 - \rho_0 \beta_0 \tilde{z}) \left( N u_0 \frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{t}} + \frac{u_0^2}{L} (\tilde{\mathbf{u}} \cdot \tilde{\nabla} \tilde{\mathbf{u}}) \right) = -\frac{\mu u_0}{L^2} \tilde{\nabla} \tilde{p} + \beta_0 \rho_0 \tilde{\rho} \mathbf{g} + \frac{\mu u_0}{L^2} \tilde{\nabla}^2 \tilde{\mathbf{u}} \quad (2.33)$$

This equation is further simplified as shown below.

$$\frac{u_0 L \rho_0}{\mu} (1 - \beta_0 \tilde{z}) \left( \frac{N L}{u_0} \frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{t}} + (\tilde{\mathbf{u}} \cdot \tilde{\nabla} \tilde{\mathbf{u}}) \right) = -\tilde{\nabla} \tilde{p} - \frac{\beta_0 \rho_0 g L^2}{(u_0 \mu)} \tilde{\rho} \hat{\mathbf{k}} + \tilde{\nabla}^2 \tilde{\mathbf{u}} \quad (2.34)$$

The parameters  $\beta$  and  $\text{Re}$  can be seen in the equation above along with a new parameter, the Strouhal number  $\text{St} = NL/u_0$ . Dropping the tildes and rewriting, the final non-dimensional, unsteady equation for conservation of mass is found.

$$\text{Re}(1 - \beta_0 \tilde{z}) \left( \text{St} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla \mathbf{u}) \right) = -\nabla p - \beta \rho \hat{\mathbf{k}} + \nabla^2 \mathbf{u} \quad (2.35)$$

The same procedure can be followed for conservation of mass and the resulting non-dimensional equation will be

$$\text{St} \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = \kappa \nabla^2 \rho. \quad (2.36)$$

#### 2.4.4 Initial and Boundary Conditions

In addition to non-dimensionalizing the governing equations, the boundary and initial conditions must also be non-dimensionalized. Here, the fourth non-dimensional parameter  $\alpha$  arises, as discussed previously. Dropping the tildes, the new initial and boundary conditions in the non-dimensional variables are immediately found.

$$(u, w, \rho) = (-U/u_0, 0, -z) \quad \text{at } t = 0 \quad (2.37)$$

$$(u, w, \rho) \rightarrow (-U/u_0, 0, -z) \quad \text{as } x^2 + z^2 \rightarrow \infty \quad (2.38)$$

$$(u, w, \rho_n) = (0, 0, 0) \quad \text{on the wedge} \quad (2.39)$$

In non-dimensional units, the wedge has unit length and height  $2\alpha$ .

#### 2.4.5 Discussion

The four non-dimensional parameters  $\text{Re}$ ,  $\beta$ ,  $\kappa$ , and  $\alpha$  play a vital role in determining the behavior of a solution. By examining their relative sizes in any particular case, it can be determined which physical phenomena dominate the solution.

It is important to also introduce the Schmidt number  $\mu/(\rho_0 \kappa_0)$ , which describes the ratio of viscous to molecular diffusion. The Schmidt number is related to the above parameters:  $\text{Sc} = (\kappa \text{Re})^{-1}$ . This parameter does not introduce another degree of freedom, but rather is determined by the four original parameters. The Schmidt number will be used throughout this paper because of its common usage in describing stratified fluids.

## 2.5 Non-Diffusive Limit

With the non-dimensional equations in hand, a special limiting case of the steady problem is now considered. In particular, consider a very viscous and slow-moving but non-diffusive fluid. In this case  $\text{Re}, \kappa \rightarrow 0$  and thus  $\text{Sc} \rightarrow \infty$ .

A stream function is introduced such that

$$u = -\psi_z \quad \text{and} \quad w = \psi_x. \quad (2.40)$$

This results in the automatic satisfaction of the incompressibility condition. The convection-diffusion equation (2.32) can be rewritten as

$$-\psi_z \rho_x + \psi_x \rho_z = 0. \quad (2.41)$$

This implies that all derivatives of  $\psi$  and  $\rho$  with respect to  $x$  and  $z$  must be zero, i.e.  $\rho = \rho(\psi)$  [13].

The boundary condition in the far-field implies that  $\psi \rightarrow z$  far from the wedge. This implies that the far-field flow has speed  $-1$  in the  $-x$  direction, so in this case the reference velocity is chosen as the background velocity. Also in the far-field,  $\rho \rightarrow -z$ , so in the far-field  $\rho \rightarrow -\psi$ . In fact, because  $\rho = \rho(\psi)$  throughout the fluid, this condition holds at all points in the flow field. Thus, if the stream function can be found, the density field is also known.

The stream function can be substituted in to the scalar momentum equations (2.27) and (2.28). Taking the limit as  $\text{Re}$  and  $\kappa \rightarrow 0$ ,

$$0 = p_x + \psi_{xxz} + \psi_{zzz} \quad \text{and} \quad 0 = -p_z + \beta\psi + \psi_{xxx} + \psi_{zzx}. \quad (2.42)$$

Differentiating the first with respect to  $z$  and the second with respect to  $x$  and adding yields a variation of the biharmonic equation.

$$\nabla^4 \psi + \beta \psi_x = 0 \quad (2.43)$$

This equation is subject to  $\psi$  and  $\psi_n$  vanishing on the wedge, which fixes the zero streamline on the wedge and enforces the no slip condition. Note that when  $\beta = 0$ , this reduces to Stokes flow past a wedge and when  $\beta \rightarrow \infty$ , the flow is uniform in the  $x$  direction.

### 3 Numerical Implementation

In the previous section, the strong formulation of the problem was derived, including the governing PDEs and boundary conditions. Now these equations will be solved approximately using the finite element method. Note that the governing equations require the computation of second derivatives (or fourth derivatives in the case of the non-diffusive limit); the aim is to reduce this requirement and allow a lower order of continuity. This is the basis of the finite element method and it will allow the solution to be approximated weakly.

In this section, the weak formulation of all three models will be derived. Then, the interpolation, integration, and linear system solver will be discussed briefly. The aim of this report is not to summarize the finite element method, but rather to implement it for an interesting problem, so where the approaches given are standard they will not be described in detail [14] [15]. In addition, the implementation of the stabilization methods will not be presented. Finally, the solution method including fixed point iteration, time stepping, and zero-finding will be described.

#### 3.1 Weak Formulation

First, the weak formulation of each system will be derived. This weak formulation will allow for solutions which don't satisfy the same smoothness properties required in the strong formulation. This is a key step in the finite element method because it reduces the requirement of high order continuity across element boundaries.

##### 3.1.1 Stationary Case

The non-dimensional, steady conservation of momentum equation (2.4) and incompressibility condition (2.5) are rewritten here for convenience.

$$\text{Re}(1 - \beta_0 z)(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p - \beta \rho \hat{\mathbf{k}} + \nabla^2 \mathbf{u} \quad (3.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3.2)$$

Now, in (3.1) take a suitable inner product with a velocity test function  $\mathbf{v}$ . In this case, the  $L^2$  inner product over the domain  $\Omega$  is chosen and it is required that  $\mathbf{v} \in H^1(\Omega)^2$ , i.e.  $\mathbf{v}$  has first derivatives that are square integrable. In the following

working,  $d\Omega$  is a differential element.

$$\int_{\Omega} (\operatorname{Re}(1 - \beta_0 z)(\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}) d\Omega = - \int_{\Omega} (\nabla p \cdot \mathbf{v}) d\Omega - \int_{\Omega} (\beta \rho \hat{\mathbf{k}} \cdot \mathbf{v}) d\Omega + \int_{\Omega} ((\nabla^2 \mathbf{u}) \cdot \mathbf{v}) d\Omega \quad (3.3)$$

Second derivatives of velocity appear in the diffusion term on the right hand side, so the Gauss-Green identity is applied to that term in order to reduce the order of differentiation. This identity is also applied to the pressure term in order to reduce the order of differentiation on pressure. This allows  $p \in L^2$  and  $\mathbf{u} \in H^1$ . It is important to note that it is assumed that the test function  $\mathbf{v}$  satisfies the homogeneous form of the Dirichlet boundary conditions; thus, the boundary terms vanish.

$$\int_{\Omega} (\operatorname{Re}(1 - \beta_0 z)(\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}) d\Omega - \int_{\Omega} (p(\nabla \cdot \mathbf{v})) d\Omega + \int_{\Omega} (\beta \rho \hat{\mathbf{k}} \cdot \mathbf{v}) d\Omega + \int_{\Omega} (\nabla \mathbf{u}) \cdot (\nabla \mathbf{v}) d\Omega = 0 \quad (3.4)$$

This expression must hold for all  $\mathbf{v} \in H^1(\Omega)^2$ . Note that the buoyancy term only exists in the second component, i.e. only  $v_2$  will appear in that integral.

To obtain the weak formulation of the incompressibility equation, multiply by the pressure test function  $q \in L^2(\Omega)$ .

$$\int_{\Omega} (\nabla \cdot \mathbf{u}) q d\Omega = 0 \quad (3.5)$$

Since  $q \in L^2$ , it is not helpful to integrate by parts. Instead, consider the Galerkin projection and replace the test function with the trial function. In this case, replace  $q$  with  $p$  in (3.5). In (3.4), replace  $\mathbf{v}$  with  $\mathbf{u}$ . Adding these two now eliminates the pressure term. The resultant equation, retaining the test functions in the other terms, is shown below.

$$\int_{\Omega} (\operatorname{Re}(1 - \beta_0 z)(\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}) d\Omega + \int_{\Omega} (\beta \rho \hat{\mathbf{k}} \cdot \mathbf{v}) d\Omega + \int_{\Omega} (\nabla \mathbf{u}) \cdot (\nabla \mathbf{v}) d\Omega = 0 \quad (3.6)$$

This is the weak form of conservation of momentum with the pressure term eliminated using the incompressibility condition.

A similar process is followed for the convection-diffusion equation. The PDE (2.8) is given again here for convenience.

$$\mathbf{u} \cdot \nabla \rho = \kappa \nabla^2 \rho \quad (3.7)$$

Take the  $L^2$  inner product with a test function  $v$  which again is required to have square integrable derivatives. Note that this  $v$  is different from  $\mathbf{v}$ , which is the test function for velocity.

$$\int_{\Omega} (\mathbf{u} \cdot \nabla \rho) v d\Omega = \int_{\Omega} (\kappa \nabla^2 \rho) v d\Omega \quad (3.8)$$

By applying Gauss-Green, the order of differentiation on  $\rho$  is reduced.

$$\int_{\Omega} (\mathbf{u} \cdot \nabla \rho) v d\Omega + \int_{\Omega} \kappa (\nabla \rho) \cdot (\nabla v) d\Omega = 0 \quad (3.9)$$

Again, the boundary terms do not appear. This weak form must be supplemented with the strongly enforced boundary condition  $\rho = -z$  in the far-field.

In both of these equations, an interior point penalty method was implemented. This results in an additional term in the weak formulation which allows the gradient fluctuation over element boundaries to be controlled. This also ensures stability of the velocity-pressure coupling and improves numerical stability in the high Reynolds number regime [16]. The derivation and implementation of this method are beyond the scope of this project, but [17], [18], and [19] provide further information.

### 3.1.2 Time-Dependent Case

The additional complexity in the time-dependent case is the presence of a time derivative. This will be handled using a backward Euler approximation and thus does not require any special treatment when formulating the weak problem. In particular, the time derivatives of velocity and pressure are approximated as

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\Delta t} \quad \text{and} \quad \frac{\partial \rho}{\partial t} = \frac{\rho_i - \rho_{i-1}}{\Delta t}, \quad (3.10)$$

respectively, where  $i$  indexes through time. The algorithmic implementation of this time stepping procedure will be discussed after the description of the finite element method.

### 3.1.3 Non-Diffusive Limit

The governing equation for the non-diffusive case (2.43) is reproduced here with the differential operators expanded for convenience.

$$\psi_{xxxx} + 2\psi_{xxzz} + \psi_{zzzz} + \beta\psi_x = 0 \quad (3.11)$$

As before, this equation can be converted into a weak form by taking the inner product with a test function  $q \in H^2(\Omega)$ .

$$\int_{\Omega} \psi_{xxxx} q d\Omega + 2 \int_{\Omega} \psi_{xxzz} q d\Omega + \int_{\Omega} \psi_{zzzz} q d\Omega + \int_{\Omega} \beta \psi q d\Omega = 0 \quad (3.12)$$

From here, Gauss-Green is applied twice in order to reduce all of the fourth derivatives to second derivatives. Then, a  $C^0$  interior penalty method was implemented. The derivation and implementation of this approach are beyond the scope of this project, but further information can be found in [20] and [21].

## 3.2 Finite Element Method

With the weak forms now in hand, the finite element method can be used to find approximate solutions. The open source package **FreeFem++** [22] was used to run all of the simulations in this report. This software package is based on C++ but has specialized commands for solving PDEs using the finite element method. It has built in mesh generation, interpolation, numerical integration, and a variety of linear system solvers. The package allows the user to define the weak form of a PDE along with boundary conditions and then in one line, solve the problem. This eliminates the need to write a mesh generator, interpolator, integrator, and sparse solver and allows the user to relatively quickly obtain solutions to a given problem.

Because this package relies simply on the weak formulations, which were found in the previous section, the steps of the finite element method will only be discussed briefly. The aim of this project was not to write a finite element solver but rather to use the finite element method to explore an interesting physical phenomenon, namely the effect of diffusion-driven flow over a wedge. Thus, much of the focus of the rest of this report will be on the results themselves and not on the underlying mechanics of the **FreeFem++** software.

A sample of the **FreeFem++** code used for this project can be found in the Appendix along with a link to a repository containing all of the code and the simulation results. Many of the simulations for this project were run remotely through Amazon Web Services (AWS). This allowed several machines, each with 8 cores, to run the simulations simultaneously.

### 3.2.1 Mesh Discretization

`FreeFem++` provides automatic mesh generation based on a Delaunay triangulation [23]. Upon specifying a number of points on each section of the domain boundary, the algorithm computes the node and edge locations that result in a Delaunay triangulation, i.e. one in which the minimum angle in the mesh is maximized. This results in well-conditioned elements and a regular mesh.

An example of the generated mesh is shown in Fig. 9.

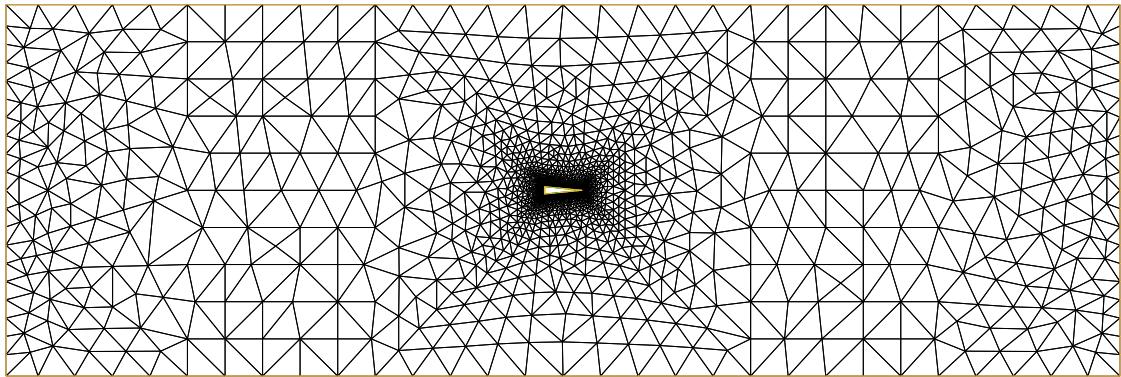


Figure 9: An example of the mesh generated automatically by `FreeFem++`

This particular mesh is more refined near the wedge, as seen in Fig. 10.

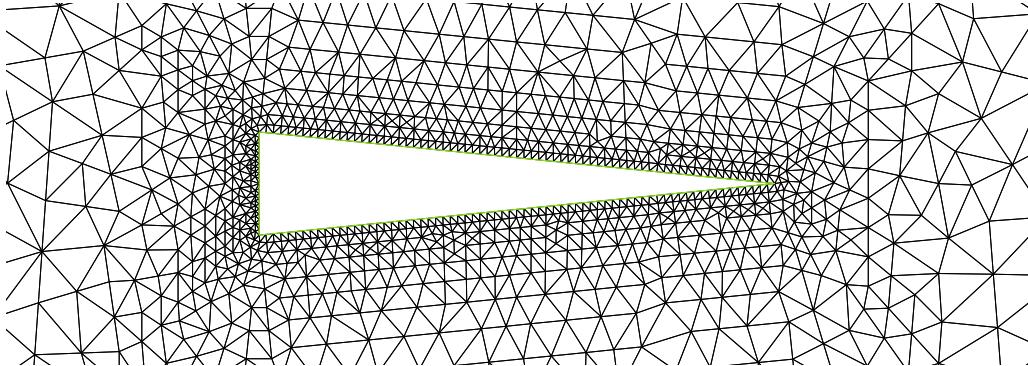


Figure 10: Mesh refinement near the wedge

For all of the simulations in this project, the refinement of the mesh is based on a refinement parameter  $r$ . The number of points on each edge is based on this refinement parameter. In particular, the top and bottom of the domain have  $2r$  points each, the left and right sides of the domain have  $r$  points each, the sloping surfaces of the wedge have  $8r$  points each, and the back of the wedge has  $2r$  points. The relative number of points on each edge of the domain was determined empirically based on the local variation of the flow in each region of the domain. This method

of refining the mesh was not optimized, and it should be noted that simulation efficiency could be improved with a detailed study on the number of points required on each edge.

### 3.2.2 Interpolation, Galerkin Projection, Integration, and Solving

**FreeFem++** also provides an interpolation tool. Upon specifying the element type, all interpolation is automatic. For this project, all elements were selected as P2 Lagrangian elements. These elements are six-node triangles with quadratic shape functions.

Upon specifying the weak formulation and the element types, **FreeFem++** performs the interpolation and Galerkin projection, replacing the test function and trial function with the element shape functions multiplied by the nodal values of the unknowns. It is important to note that the non-linear problem is solved iteratively, where each iteration involves the solution of a linear problem. This procedure will be discussed in more detail later in this section.

In order to compute the element stiffness and boundary vector (where applicable), **FreeFem++** uses a 3-point Gaussian quadrature integration rule [23].

Upon computing the element data for each triangular element, these are assembled into the global linear system. The sparse system is solved using the Unsymmetric MultiFrontal method (UMFPACK), which is designed to solve unsymmetric sparse systems [24].

Given the weak forms and the **FreeFem++** implementation of the finite element method, the nodal values of the solution variables can be computed. The rest of this section will detail the auxiliary calculations required in order to obtain a full solution and perform the additional analysis.

## 3.3 Non-Diffusive Case Pressure Recovery

In the case of the non-diffusive limit, the finite element solver will determine the nodal values of the stream function  $\psi$ . This allows for the direct computation of  $\rho = -\psi$ ,  $u = -\partial\psi/\partial z$ , and  $w = \partial\psi/\partial x$ . In order to compute the force on the wedge, however, the pressure must be determined. To accomplish this, the known density and velocity are input into the stationary momentum equation solver and the pressure is computed.

### 3.4 Computing Forces

A key component of the numerical simulation is the computation of the net force on the wedge due to both pressure and viscous stresses. As stated earlier, because the problem is two-dimensional, all forces are per unit width. A positive net force indicates a forward (to the right in all diagrams) propulsion on the wedge. A negative net force corresponds to a drag (to the left) on the wedge. The computation of net force will be used to determine the steady speed of the wedge. A solution that results in zero net force on the wedge indicates that the background velocity is the steady speed of the wedge. If the net force is positive, the wedge will move faster than the background velocity and if the net force is negative, the wedge will move more slowly than the background velocity. Thus, finding the steady speed of the wedge is equivalent to finding the background velocity that results in zero net force on the wedge. Recall that only horizontal forces are considered because the wedge is assumed to settle at the vertical location of neutral buoyancy.

The non-dimensional pressure force on the wedge in the  $x$  direction is given by the equation below,

$$F_{pressure,x} = - \int_{\Gamma_2} p n_x ds, \quad (3.13)$$

where  $n_x$  is the component of the wedge unit outward normal in the  $x$  direction.

The viscous force is determined by the strain rate tensor  $\epsilon(\mathbf{u})$ , given below.

$$\epsilon(u, w) = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \frac{1}{2} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) & \frac{\partial w}{\partial z} \end{pmatrix}. \quad (3.14)$$

In non-dimensional form, the viscous force is given by

$$F_{viscous} = \int_{\Gamma_2} \epsilon(u, w) \mathbf{n} ds, \quad (3.15)$$

where  $n$  is the unit outward normal vector of the wedge written in  $x$  and  $z$  components. Thus, the  $x$  component of the viscous force is given by

$$F_{viscous,x} = \int_{\Gamma_2} \left( \frac{\partial u}{\partial x} n_x + \frac{1}{2} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) n_z \right) ds \quad (3.16)$$

Again, these forces are non-dimensional. In order to scale these to their dimensional counterparts, they can be multiplied by  $\mu u_0$ . For the remainder of this report,

only non-dimensional forces will be considered.

### 3.5 Fixed Point Iteration

Due to the non-linearity in the momentum equation, a fixed point iteration algorithm is used to solve for both velocity and density. For convenience, the weak form (3.6) is reproduced below.

$$\begin{aligned} \int_{\Omega} (\text{Re}(1 - \beta_0 z)(\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}) d\Omega + \int_{\Omega} (\beta \rho \hat{\mathbf{k}} \cdot \mathbf{v}) d\Omega + \\ \int_{\Omega} (\nabla \mathbf{u}) \cdot (\nabla \mathbf{v}) d\Omega + \int_{\Omega} \mathbf{u} \cdot (\nabla q) d\Omega = 0 \end{aligned} \quad (3.17)$$

The nonlinearity is due to the product of velocities in the first term. To create a linear problem that can be iteratively solved, the equation is rewritten as

$$\begin{aligned} \int_{\Omega} (\text{Re}(1 - \beta_0 z)(\mathbf{u}_{iter} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}) d\Omega + \int_{\Omega} (\beta \rho \hat{\mathbf{k}} \cdot \mathbf{v}) d\Omega + \\ \int_{\Omega} (\nabla \mathbf{u}) \cdot (\nabla \mathbf{v}) d\Omega + \int_{\Omega} \mathbf{u} \cdot (\nabla q) d\Omega = 0, \end{aligned} \quad (3.18)$$

where  $\mathbf{u}_{iter}$  is the solution from the previous iteration. Since  $\mathbf{u}_{iter}$  is not a variable during a given iteration, the problem is now linear at each iteration.

This iterative procedure also lends itself to the coupling between the conservation of momentum and conservation of mass equations. With an initial guess for density, pressure and velocity can be found using the conservation of momentum equation, then input into the conservation of mass equation to find an updated density. The procedure can be conducted iteratively, where each iteration updates the values of density, velocity, and pressure that are used in the next iteration. This algorithm will be presented in more detail below.

Before doing so, a notion of convergence must be defined. For the steady case, convergence is tracked using a relative difference in the solution variables between successive iterations. This difference is computed using an  $L^2$  norm, as shown below for a generic solution variable  $\mathbf{x}$  and its previous value  $\mathbf{x}_{iter}$ .

$$\text{increment} = \sqrt{\frac{\int_{\Omega} \|\mathbf{x} - \mathbf{x}_{iter}\|_2^2 d\Omega}{\int_{\Omega} \|\mathbf{x}_{iter}\|_2^2 d\Omega}} \quad (3.19)$$

While the equation above is written for a vector quantity  $\mathbf{x}$  (such as  $\mathbf{u}$ ), it holds for scalar quantities as well.

In order to facilitate convergence, relaxation parameters were used to update the values of velocity and density between successive iterations. These parameters were set to 0.1 for all simulations. They were determined empirically and should not be taken as the optimal values. Note that the increment is computed prior to the relaxation, which means that the convergence criteria are stronger than if this increment was computed post-relaxation. The algorithm is presented below.

---

**Algorithm 1** Fixed point iteration

---

- 1: Set initial values for  $\mathbf{u}$  and  $\rho$  (uniform flow, linear stratification)
  - 2:  $\mathbf{u}_{iter} = \mathbf{u}$  and  $\rho_{iter} = \rho$
  - 3: **while** (fixed point increment > fixed point tolerance) & (iteration < iteration max) **do**
  - 4:     iteration = iteration + 1
  - 5:     Solve momentum equation for velocity and pressure
  - 6:     Compute velocity increment
  - 7:      $\mathbf{u} = \mathbf{u} * (\text{urelax}) + \mathbf{u}_{iter} * (1 - \text{urelax})$
  - 8:      $\mathbf{u}_{iter} = \mathbf{u}$
  - 9:     Solve convection-diffusion equation for density
  - 10:    Compute density increment
  - 11:     $\rho = \rho * \text{densityrelax} + \rho_{iter} * (1 - \text{densityrelax})$
  - 12:     $\rho_{iter} = \rho$
  - 13:    fixed point increment = velocity increment + density increment
  - 14: **end while**
- 

In all cases, the fixed point tolerance was set to 0.01, representing a 1% relative change between solutions in successive iterations. This value was selected empirically because it provides a balance between solution accuracy and required computation time.

### 3.6 Time Stepping

In the time-dependent case, the nonlinearity is handled in the same manner as in the stationary case. However, the additional complexity of the time derivative is introduced. As stated before, a backward Euler approximation was used in place of the time derivative. Using the index  $i$  through time and the same fixed point iteration variables, the equations for conservation of momentum and conservation mass in the unsteady case are given by (3.20) and (3.21) below, where  $\Delta t$  is the

Euler time step in non-dimensional units.

$$\begin{aligned} & \int_{\Omega} (\text{Re}(1 - \beta_0 z) \left( \text{St} \frac{\mathbf{u}_i - \mathbf{u}_{i-1}}{\Delta t} + (\mathbf{u}_{iter} \cdot \nabla \mathbf{u}_i) \right) \cdot \mathbf{v}) d\Omega + \\ & \int_{\Omega} (\beta \rho_i \hat{\mathbf{k}} \cdot \mathbf{v}) d\Omega + \int_{\Omega} (\nabla \mathbf{u}_i) \cdot (\nabla \mathbf{v}) d\Omega + \int_{\Omega} \mathbf{u}_i \cdot (\nabla q) d\Omega = 0 \end{aligned} \quad (3.20)$$

$$\int_{\Omega} \left( \frac{\rho_i - \rho_{i-1}}{\Delta t} \right) v d\Omega + \int_{\Omega} (\mathbf{u}_i \cdot \nabla \rho_i) v d\Omega + \int_{\Omega} \kappa (\nabla \rho_i) \cdot (\nabla v) d\Omega = 0 \quad (3.21)$$

It is important to again emphasize that (3.20) is used to solve for the velocity and pressure while (3.21) is used to solve for the density. At each iteration of the time stepping procedure, fixed point iteration is used to solve for the approximate solution at the given time. The algorithm is summarized below.

This time, the stopping criteria are based on the “time increment” and the “force increment” divided by the time step “dt” representing  $\Delta t$ . The “time increment” is given by the same  $L^2$  norm formulation as (3.19), but using successive time steps instead of successive fixed point iterations. The “force increment” is simply the relative change in force between successive time steps and it was used to monitor the convergence of the force to a steady value. Both of these increments are now metrics for the rate of change of the solution.

---

### Algorithm 2 Time stepping

---

- 1: Set initial values for  $\mathbf{u}_0$  and  $\rho_0$  (uniform flow, linear stratification)
  - 2:  $i=0$
  - 3: **while** (time increment / dt > time tolerance) OR (force increment / dt > force tolerance) **do**
  - 4:      $i = i+1$
  - 5:     Using fixed point iteration, solve for  $\mathbf{u}_i$  and  $\rho_i$
  - 6:     Compute time increment
  - 7:     Compute force increment
  - 8: **end while**
- 

In all simulations, the “time tolerance” and the “force tolerance” were set to 1e-5. This value was determined empirically to provide a balance between required computation time and convergence to the steady solution. Note that these tolerances are much smaller than those used in the fixed point iteration; in practice, the fixed point iteration at each time step was not required to fully converge, but rather to facilitate the convergence of the time stepping algorithm. If the time dependence of the solution was being studied, it would be important to resolve the solution at

each time step more precisely.

### 3.7 Zero Finding

The main aim of this project is to determine the steady wedge speed as a function of the physical parameters. Recall that this is equivalent to finding the background velocity that results in zero net force on the wedge. In order to do so, a method was developed to change the background velocity iteratively until the net force on the wedge was found to be sufficiently close to zero. This algorithm is based on a secant method for root finding and is functional because the force as a function of background velocity is well-behaved. The stopping conditions here are the relative change in force between successive iterations and the relative change in background velocity between successive iterations, where iterations now refer to secant method iterations, not time steps or fixed point iterations.

In this method, two initial simulations are run using different background velocities. This results in the computation of two net force values. Using these two data points, a line is drawn and the point of intersection of this line with the line representing zero net force is computed. The background velocity associated with this point is selected as the next background velocity to simulate. This procedure is repeated, using the previous two simulations to compute the next background velocity, until either the change in force or the change in background velocity between successive iterations is sufficiently close to zero.

---

#### **Algorithm 3** Zero finding

---

```

1:  $U_{background,1} = 0$ 
2: Compute the force  $f_1$ 
3:  $U_{background,2} > 0$  (choose value)
4: Compute the force  $f_2$ 
5:  $i = 2$ 
6: while (force increment > secant tolerance) AND (velocity increment > velocity
   tolerance) do
7:    $i = i+1$ 
8:    $f'_{i-1} = ((f_{i-1} - f_{i-2}) / (U_{background,i-1} - U_{background,i-2}))$ 
9:    $U_{background,i} = U_{background,i-1} - f_{i-1} / f'_{i-1}$ 
10:  Compute the force  $f_i$ 
11:  Compute force increment
12:  Compute velocity increment
13: end while
```

---

## 4 Numerical Results: Steady Case

In the following sections, numerical results from each of the three models will be presented. First, the results from the steady nonlinear system are presented. It will be shown in this section that the stationary model is lacking, especially at high levels of stratification. Thus, few results will be presented here in order to allow the results and discussion to be focused on the time-dependent case. These results will be qualitative and a discussion of quantitative results will be deferred to the next section.

### 4.1 Negative Net Force

Running the stationary simulation at a background velocity of  $U = 1\text{e}-6 \text{ m/s}$ , stratification of  $\partial\rho/\partial z = 1\text{e}-6$  and  $\text{Sc} = 1$  results in the flow field in Fig. 11, viewed in the entire domain.

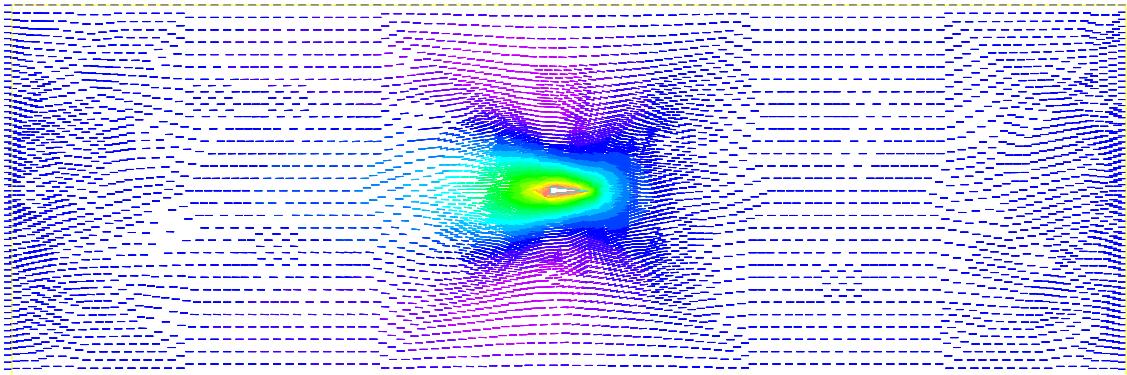


Figure 11: Stationary model with  $U = 1\text{e}-6 \text{ m/s}$

In this case, the velocity field is near uniform, except near the wedge, as expected given the experimental results. The flow field is altered by the presence of the wedge, which can be seen in more detail in Fig. 12.

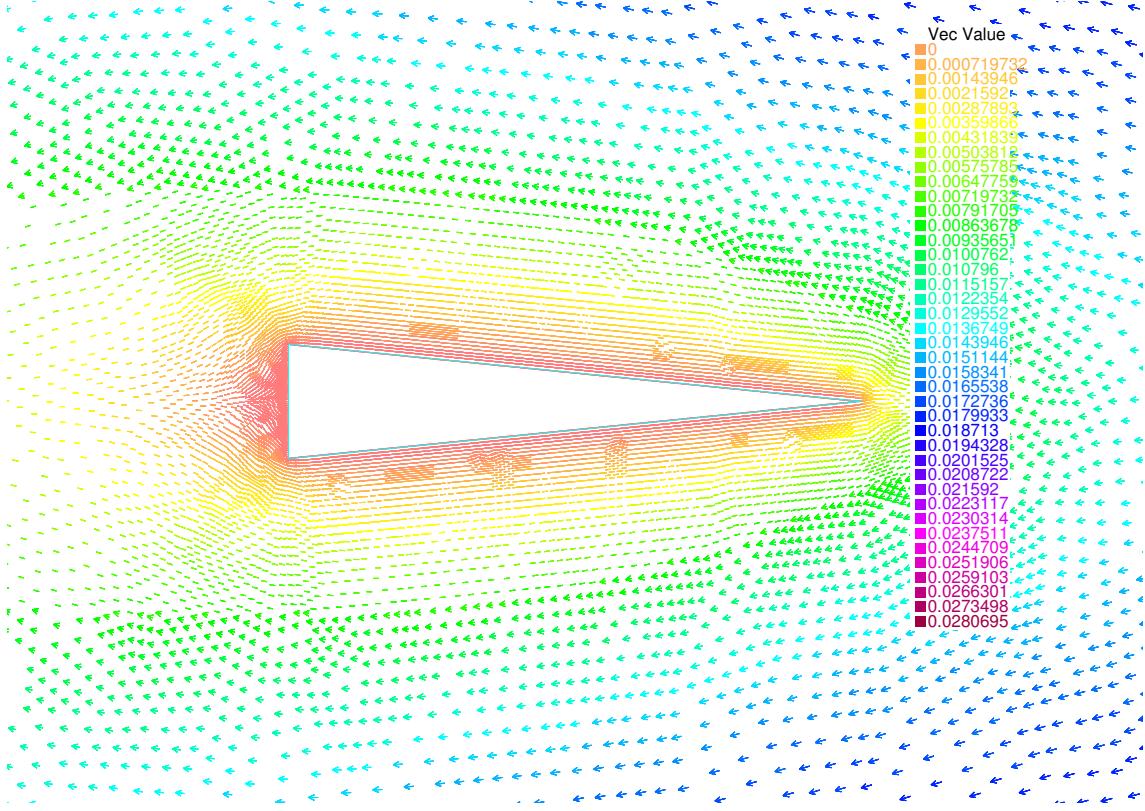


Figure 12: Stationary model with  $U = 1\text{e}-6 \text{ m/s}$ , near the wedge

Now, the development of a boundary layer can be seen in the stagnated flow near the wedge. A large wake is also present behind the wedge. This flow qualitatively matches what is expected for a wedge in a uniform flow. The diffusive propulsion effects are small because the background velocity is large. In this case, there is a negative net force on the wedge, i.e. a drag pulling it to the left in the image above.

## 4.2 Zero Net Force

Next consider the same case with a slower background velocity of  $U = 1\text{e}-7 \text{ m/s}$ . The background velocity has decreased by a factor of 10, which should decrease the backward force and allow the diffusive effects to become visible. This is shown in Fig. 13.

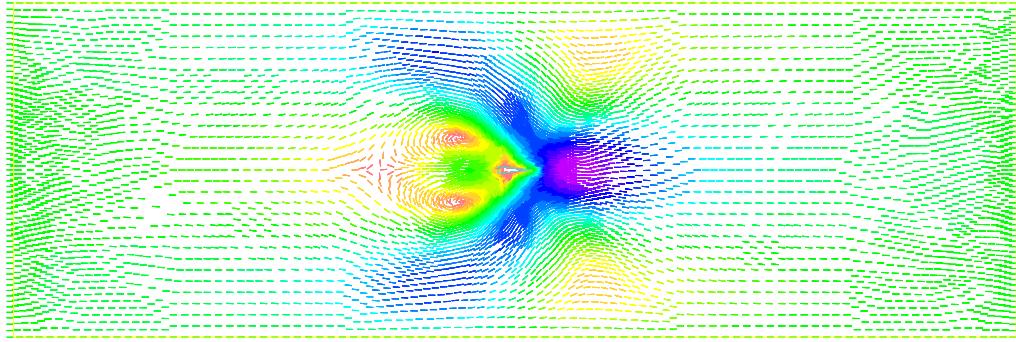


Figure 13: Stationary model with  $U = 1e-7$  m/s

The background flow is less uniform and flow farther from the wedge has been affected by the presence of the wedge. Two regions of re-circulation have appeared just behind the wedge. The solution near the wedge is shown in Fig. 14.

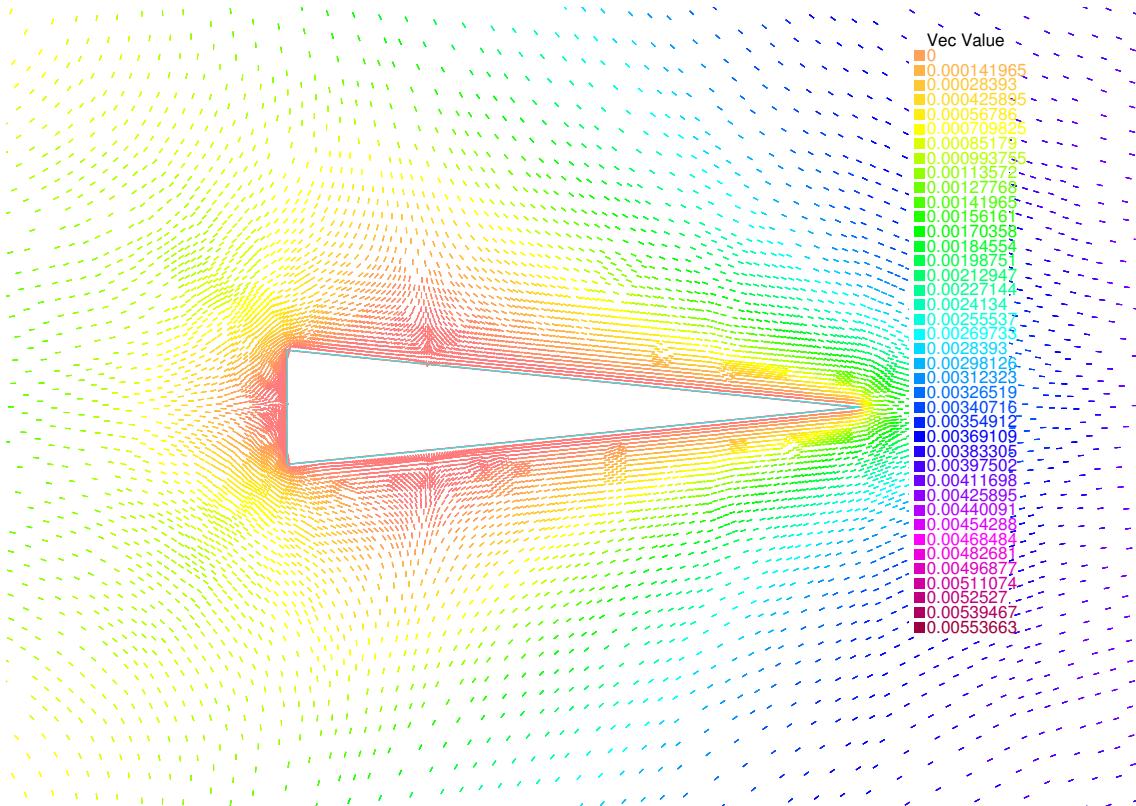


Figure 14: Stationary model with  $U = 1e-7$  m/s, near the wedge

While the solution appears qualitatively similar to the solution with faster background velocity, the flow on the wedge in this case has reversed near the back face. This flow reversal has created a region with positive viscous force which counterbalances the negative viscous force in the leading part of the boundary layer. The net force in this case is very close to zero, indicating that this background velocity is close to the steady speed of the wedge.

### 4.3 Positive Net Force

Finally, consider the same case with an even slower background velocity:  $U = 1\text{e}-8$  m/s. The full flow field can be seen in Fig. 15.

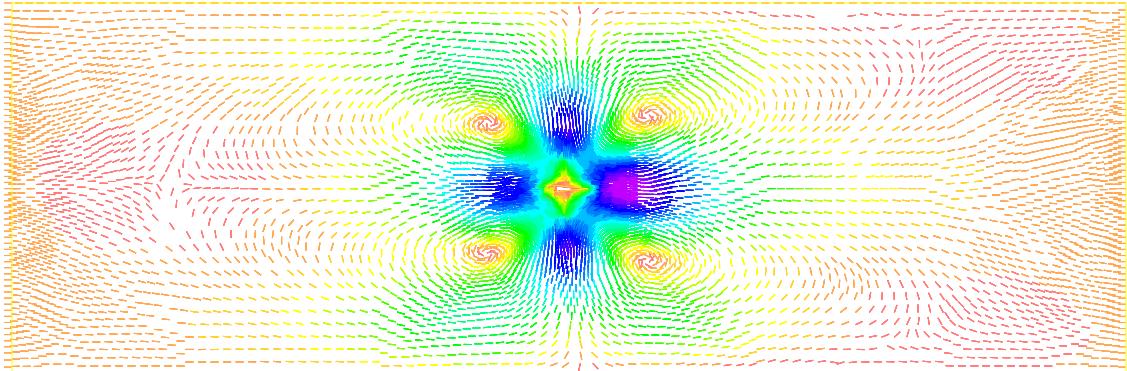


Figure 15: Stationary model with  $U = 1\text{e}-8$  m/s

Four regions of re-circulation now exist and the solution differs drastically from a uniform flow. The re-circulation is pushing fluid forward along the wedge, contributing to the positive net force on the wedge.

These results do not match what was observed in the experiment. In particular, no regions of re-circulation were present and the background velocity remained relatively uniform in all experimental cases. Here, both the re-circulation and the non-uniformity of the background velocity suggest that these results don't make sense. However, it is important to note that the net force in this case is positive. Thus, in a time-dependent scenario, the wedge would not remain in this state. It would accelerate through the flow, resulting in an increase in the background velocity until the pressure and viscous forces balanced, as in the previous case. These regions of re-circulation may never exist in the real scenario because the wedge would accelerate before they could develop. This scenario is effectively simulating the wedge being held in place in the stratified fluid, not allowed to freely move.

### 4.4 High Stratification

The primary reason the stationary model is not the best approach to further explore the problem is that it breaks down in a highly stratified fluid. Four studies were run, each with different background density gradient and each with the goal of finding the background velocity that resulted in zero net force on the wedge. The results are shown in Fig. 16.

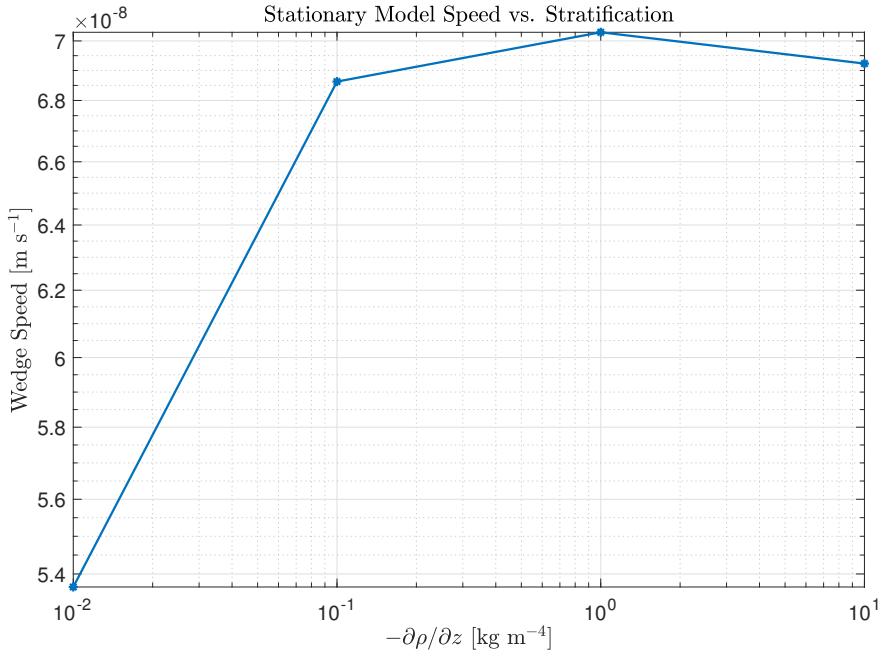


Figure 16: Steady wedge speed vs. stratification for the stationary model

It is expected that the steady wedge speed will increase with increased stratification, however that is not observed in these results. In fact, the steady speed of the wedge begins to decrease as  $-\partial\rho/\partial z$  increases past 1. As the stratification is increased, the flow fields also begin to resemble the expected flow field less and less. It is worth noting that a simulation was run at  $\partial\rho/\partial z = -100$  but it did not converge, even with a further relaxation of the solution. The experimental studies were conducted with  $\partial\rho/\partial z$  of order 100, so it is very limiting that this model cannot simulate those results.

The breakdown of this model at high stratification is the primary reason it will not be used for the remainder of this report. It is important to point out that studies of the effect of mesh refinement, domain size, and fixed point tolerance were all conducted to ensure that those factors were not affecting the solution. For the sake of brevity, the results will not be presented here but will be presented for the unsteady model.

## 5 Numerical Results: Time-Dependent Case

In this section, the main results from this project will be presented. Due to the shortcomings of stationary model discussed earlier and the non-diffusive model which will be discussed later, neither of these models were used for the main study.

### 5.1 Mesh Refinement Study

First, a mesh refinement study was conducted. The aim of this study was to determine a sufficient resolution for the mesh that effectively balanced the accuracy of the solution and the computational complexity. For this study, a nominal setup was considered with  $\text{Sc} = 1000$ ,  $\frac{\partial \rho}{\partial z} = -1 \text{ kg/m}^4$ , zero background velocity, and a domain with total width 10 and height 5, in non-dimensional units.

The refinement parameter  $r$  (which determines the refinement on each boundary of the domain) was set to 4, 8, and 12 and the simulation was run until the solution converged. The net force values computed during this process were saved and are shown in Fig. 17. The net force was chosen as a metric for determining the mesh refinement for two reasons. First, it is easy to compute and compare between meshes with different numbers of elements and nodes. Comparing the velocity, pressure, or density fields directly would be a more robust indicator, but comparing these fields is non-trivial. Second, net force is the metric which will be analyzed for the remainder of this report. It is most important that a simulation can accurately predict the net force for a given setup.

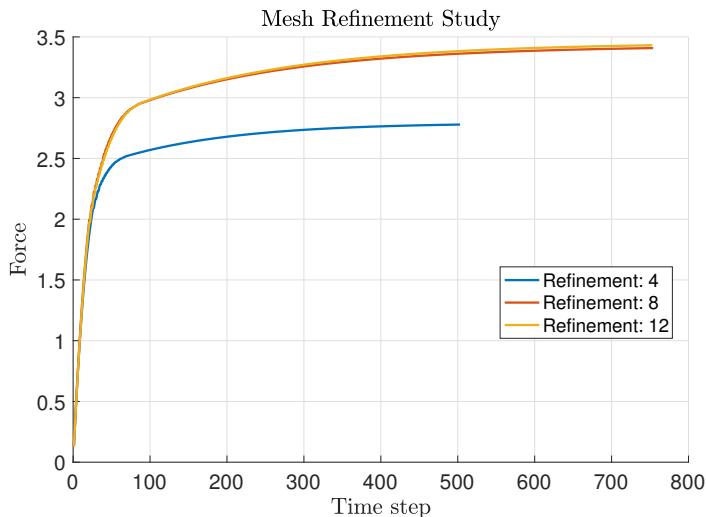


Figure 17: Mesh refinement study results using three different refinement levels

As seen in Fig. 17, the simulation with  $r = 4$  converged to a lower force than the simulations with  $r = 8$  or  $r = 12$ . It can be assumed that a more refined mesh will provide a more accurate solution. The simulation with  $r = 8$  closely approximates the  $r = 12$  simulation. The  $r = 12$  simulation took roughly twice as long as the  $r = 8$  simulation and the additional accuracy is not necessary for the purposes of this project, so the  $r = 8$  refinement will be used for the remainder of this project.

In later studies, this refinement process was repeated in order to ensure that changing parameters such as the Schmidt number or the density gradient did not change the required mesh refinement. It was always the case that  $r = 8$  was an appropriate refinement of the mesh.

## 5.2 Domain Size Study

Following the mesh refinement study, a study to determine an appropriate domain size was carried out. The objective of this study was originally to determine at what point increasing the domain size no longer has an effect on the solution. In the experiments, the tank was chosen to be sufficiently large that the local flow around the wedge did not impinge on the boundaries. In order to determine the appropriate domain size, several simulations were run with different values of the parameters `xlength` and `zlength`. For each case, the aspect ratio of the domain was kept the same so that the length was always twice the height. This was done in order to decrease the dimension of the search space and simplify the search for the appropriate domain size. Again, the metric used is the net force on the wedge. The results from this study are shown in Fig. 18.

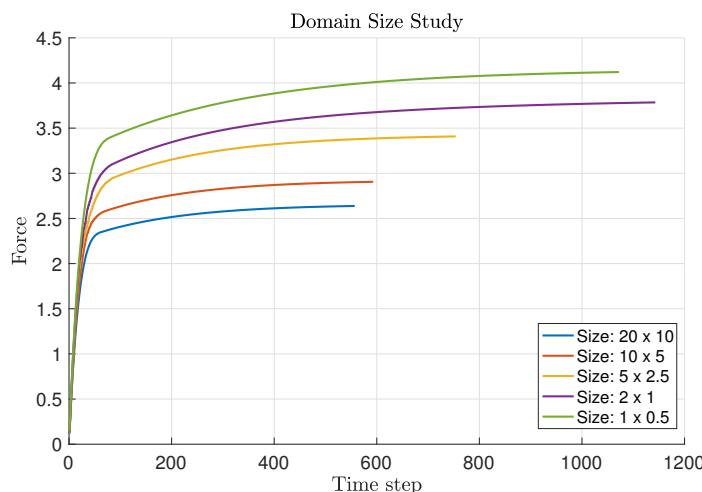


Figure 18: Domain size study results

As the domain size increases, the net force on the object continues to decrease. However for larger domains, the solution becomes unstable more easily. No simulations past  $20 \times 10$  were run because of this instability. Since the domain size continued to have an effect even with the largest domain considered, the  $5 \times 2.5$  mesh was selected because it was less computationally complex than the large domains but it was large enough to allow the flow around the wedge to develop. It is important to emphasize that this domain size should not be considered optimal. However, in the 2010 paper [3], the size of the localized flow field observed in the experiment does fit within this boundary.

This selection of domain size can also be viewed as a slight modification of the problem. Instead of a wedge submerged in a very large tank, the problem is now a wedge submerged in a channel. This channel is not so narrow that it confines the flow, but is sufficiently narrow that it results in a higher net force than a simulation with a larger domain.

For the remainder of the studies, a domain with  $x\text{length}=5$  and  $z\text{length}=2.5$  was used, representing a full domain that is 10 non-dimensional units wide and 5 non-dimensional units high. The wedge is centered in this domain. This results in a mesh with 2490 elements and 1341 vertices.

### 5.3 Experimental Case

With a mesh refinement and domain size selected, the aim of the next study was to examine a baseline case that closely resembles that of the experimental setup from the 2010 paper [3].

For this case, the physical parameters used were as follows: gravitational acceleration  $g = 9.81 \text{ m/s}^2$ , background constant density  $\rho_0 = 1115 \text{ kg/m}^3$ , kinematic viscosity  $\nu = 1e-6 \text{ m}^2/\text{s}$ , molecular diffusivity  $\kappa_0 = 1e-9 \text{ m}^2/\text{s}$ , wedge slope  $\alpha = 0.1$ , and stratification  $\partial\rho/\partial z = -500 \text{ kg/m}^4$ . These physical parameter values result in the following derived parameters: Schmidt number  $\text{Sc} = 1000$ , buoyancy frequency  $N = 2.097 \text{ s}^{-1}$ , inverse boundary layer thickness  $\gamma = 1817 \text{ m}^{-1}$ , non-dimensional buoyancy parameter  $\beta = 121100$ , non-dimensional diffusivity parameter  $\kappa = 0.002752$ , Reynolds number  $\text{Re} = 0.3633$ , and Strouhal number  $\text{St} = 577.3$ .

The simulation was run using non-dimensional time steps of 10 (which was determined empirically to facilitate fast convergence without promoting numerical in-

stability), fixed point tolerance of 0.01, force convergence tolerance of 1e-5, time convergence tolerance of 1e-5, and secant tolerance of 1e-3.

The force convergence plot is shown in Fig. 19.

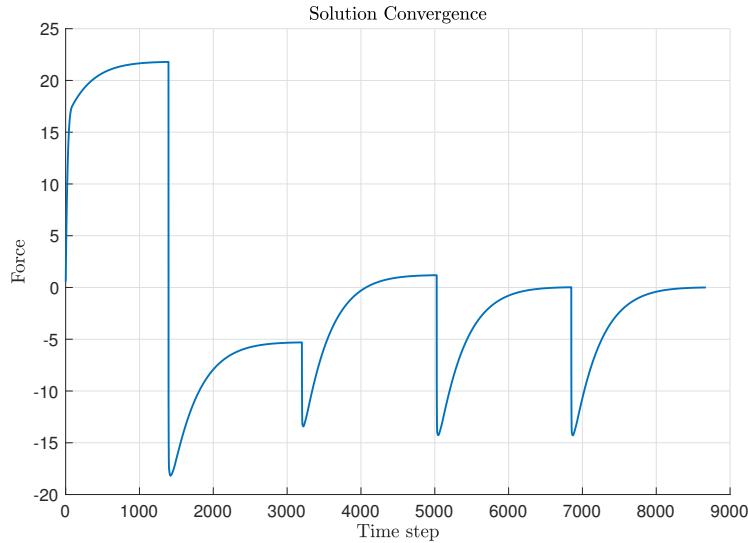


Figure 19: Visualizing solution convergence for the re-created experimental case

This chart shows the convergence of net force for each successive background velocity simulated. The jumps correspond to new runs with updated background velocity. Five background velocities were tested in search of the background velocity that yields zero net force, and for each of these background velocities, it is clear where the force has sufficiently converged.

For the same simulation, the force at each background velocity is plotted in Fig. 20.

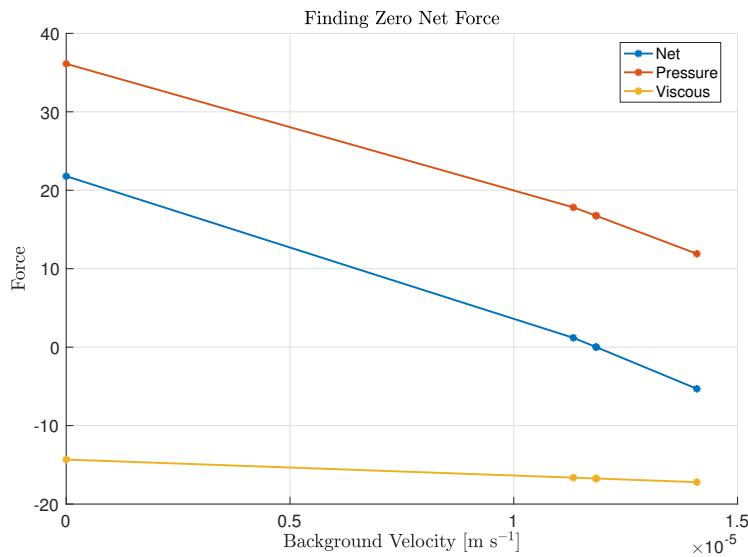


Figure 20: Force as a function of background velocity for the re-created experimental case

The first background velocity of zero results in a net force of just over 20 in non-dimensional units. The pressure force is positive and the viscous force is negative. As the background velocity is increased, the pressure force drops. The viscous force is always negative, but increases in magnitude as the background velocity increases. The net force decreases as the velocity is increased, primarily due to the drop in pressure force and not the increase in viscous force. The different iterations of the zero-finding algorithm are visible and the solution with zero net force occurs at a background velocity of about  $1.2\text{e-}5$  m/s. In the experiments, the speed of the wedge was found to be  $2.83\text{e-}6$  m/s. The difference can be attributed to several factors. First, these simulations were two-dimensional and the experiment involved a three dimensional wedge. The sides of the wedge would increase drag, resulting in a lower speed. Second, the exact parameters of the experiment were not quoted in the paper. While the Schmidt number is known, the exact diffusivity and viscosity are not. Thus, there is still freedom in the choice of parameters in these simulations.

The rest of this subsection will explore the zero net force simulation in more detail. The full velocity field is shown in Fig. 21.

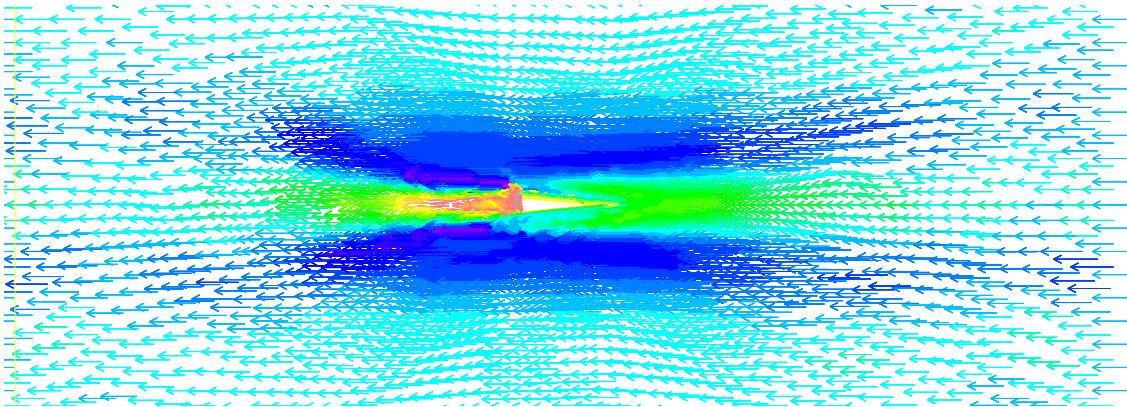


Figure 21: Velocity field for the re-created experimental case

The uniform background velocity is visible in the far field and the velocity field is clearly altered by the presence of the wedge. This diagram does not have values for the velocity labeled, but serves to qualitatively illustrate the flow. In Fig. 22 is a snapshot of the same velocity field with velocity magnitudes labeled.

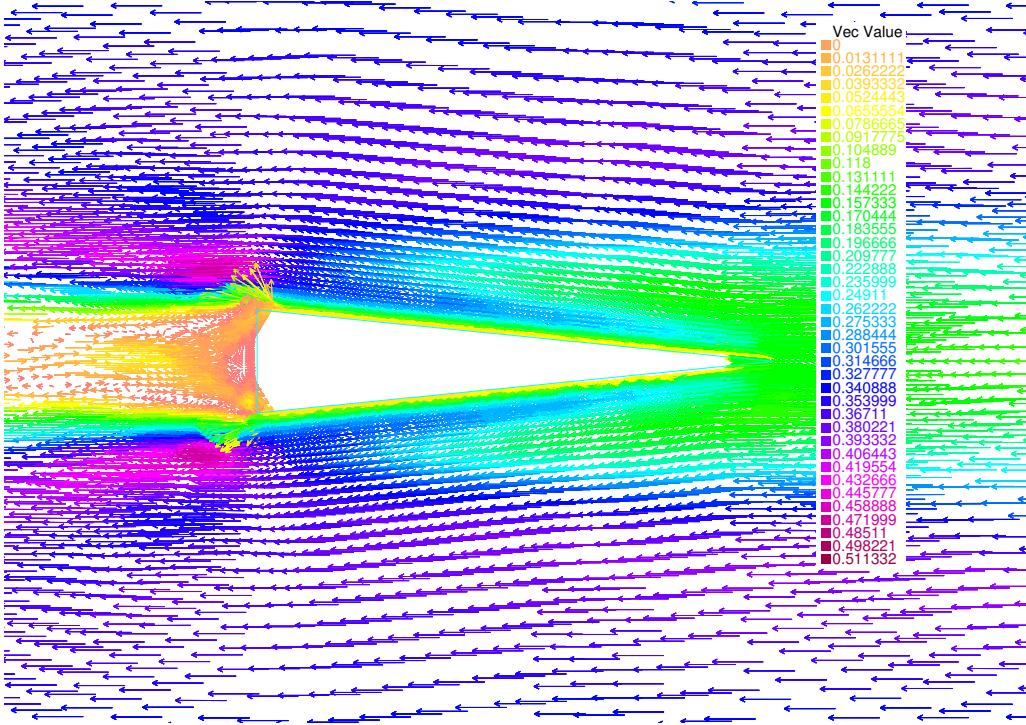


Figure 22: Velocity field near the wedge for the re-created experimental case

Now, the effect of the wedge is more clear. These speed values are non-dimensional. For reference, the non-dimensional background velocity is about 0.326, which corresponds to the dark blue vectors on the diagram. The boundary layer on the wedge is visible and a region of slightly accelerated flow exists outside of this boundary layer. This region of accelerated flow serves to create a relative low pressure region that is the source of some of the propulsion. Behind the wedge, a wake has developed and some of the flow is reversed and moving in the direction of the wedge. Just outside of this wake is another region of accelerated flow, which also serves to provide the forward propulsion. Because this is the zero net force case, these pressure forces are counter-balanced by the viscous force.

To study these effects in more detail, the solution variables on the surface of the wedge will now be studied. First, the density on the sloping surfaces of the wedge is shown in Fig. 23. Note that in all of the plots of the solution variables on the wedge, the first and last points are omitted. These surface values were post-processed by projecting the finite element solution on to a P1 space and then writing the nodal values to a file. This process caused the values on the corners of the wedge to display with discontinuity. Thus, only for the purposes of post-processing in MATLAB, these points were omitted. This did not change anything about the finite element solver or the computation of force and only affected the following plots. It is also worth

noting that all variables are non-dimensional and that the  $x$  axis is a shifted version of the original coordinate frame, this time centered at the tip of the wedge.

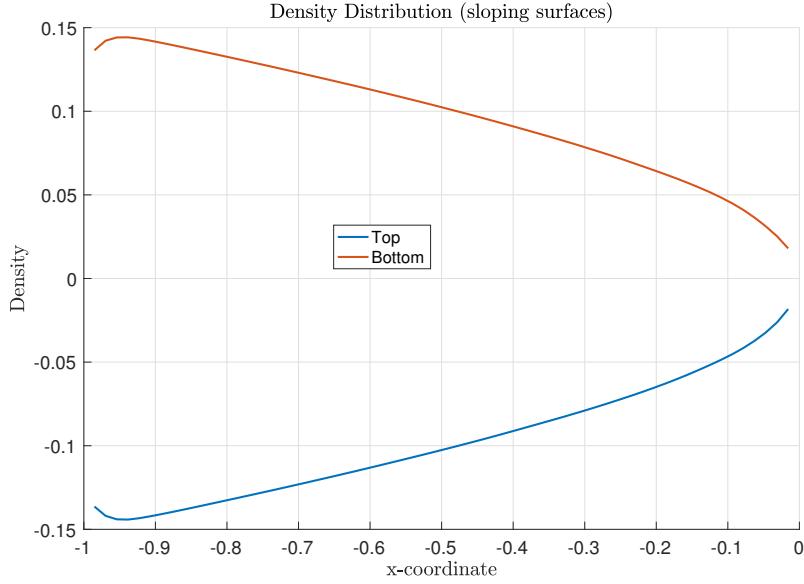


Figure 23: Density distribution along the wedge for the re-created experimental case

The effect of the stratification is visible in this diagram. Generally, the density varies linearly along the wedge, corresponding to the linear variation of height along the sloping surfaces of the wedge. As expected, the density is greater on the bottom surface. This density distribution is the main source of the propulsion, as described earlier in this report. Near the back of the wedge, the density gradient changes, corresponding to the flow matching the background flow away from the wedge.

Next, the pressure distribution on the sloping surfaces is shown in Fig. 24.

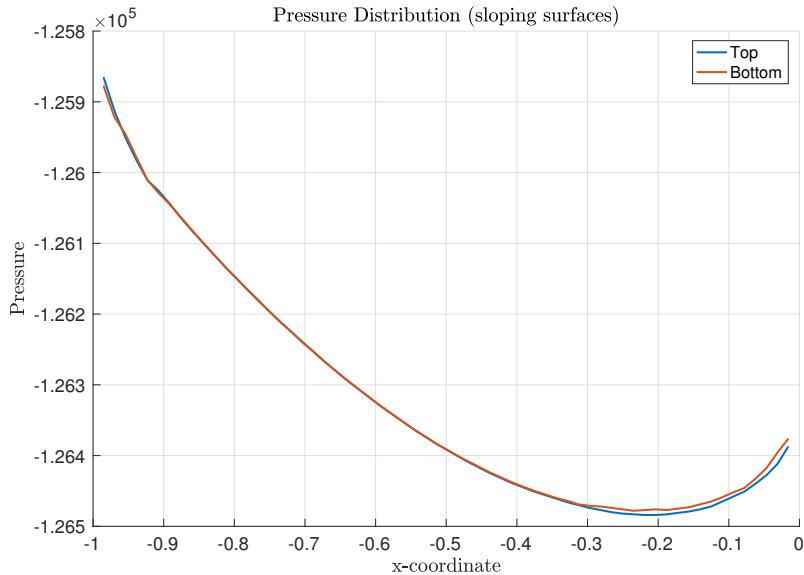


Figure 24: Pressure distribution along the wedge for the re-created experimental case

The pressure on the top and bottom surfaces are roughly the same and are minimized near the leading edge of the wedge. This negative pressure is the source of the forward propulsion, but these values alone are somewhat misleading. In order to fully understand the effect of pressure on the wedge, the pressure distribution on the back of the wedge must also be studied. This is shown in Fig. 25.

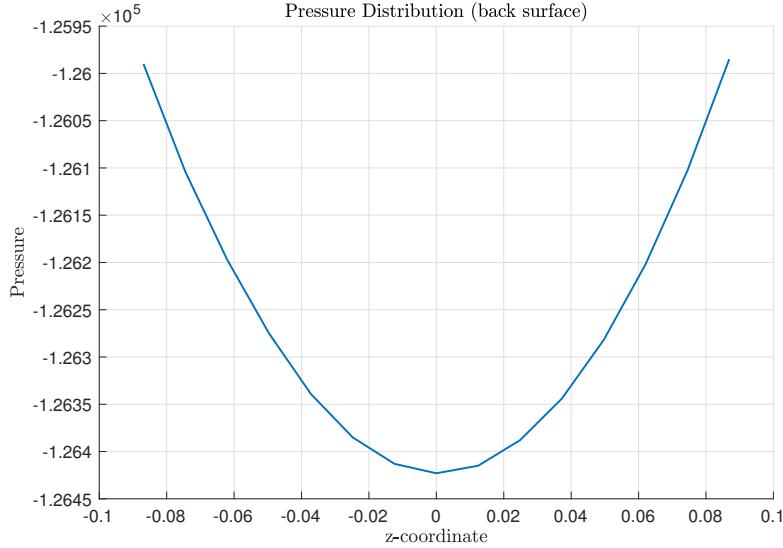


Figure 25: Pressure distribution along the back of the wedge for the re-created experimental case

As expected, this graph is symmetric. The values here are also negative, but comparing with Fig. 24, the back of the wedge has a higher relative pressure. This difference in relative pressure is the source of the positive pressure force.

The horizontal viscous stress on the sloping surfaces is plotted in Fig. 26.

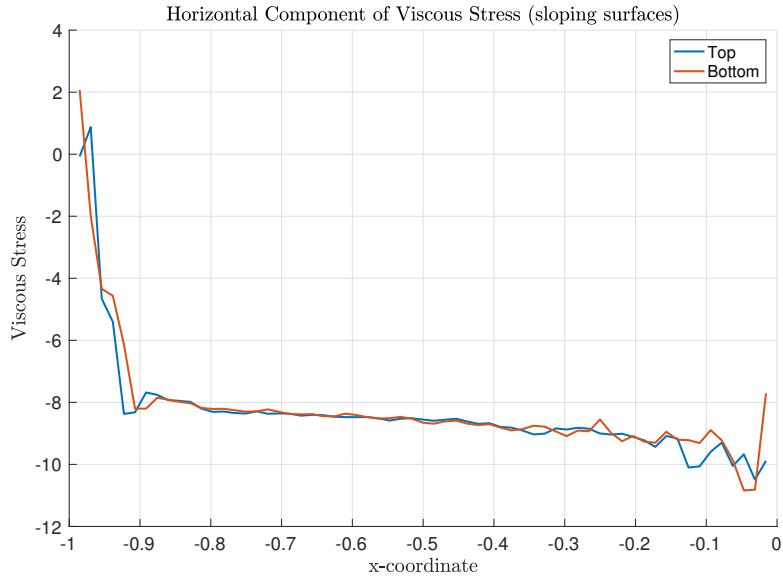


Figure 26: Viscous stress distribution along the wedge for the re-created experimental case

As expected, the viscous stress is negative along the top and bottom surfaces of the wedge. Near the back of the wedge, where the density gradient reverses, there is a small region of forward flow and a corresponding positive viscous stress.

In the following studies, the results will not be analyzed with such granularity. Instead, the focus will now be turned to understanding how the steady wedge velocity varies as a function of the parameters of the fluid.

## 5.4 Stratification Study

Now, the effect of stratification on the steady wedge speed is explored. Increasing the stratification should increase the wedge speed because the stratification is the source of the positive propulsion. Three separate studies were conducted at different Schmidt numbers; the results will be presented in the following subsections.

### 5.4.1 Schmidt Number = 1

First, the speed as a function of stratification was explored for  $Sc = 1$ . For these simulations, all of the nominal parameter values from earlier were used. To change the Schmidt number, the dimensional molecular diffusivity was changed while holding all other parameters constant. The steady speed is plotted in Fig. 27 against the background density gradient. For each of the data points on the graph, the time-dependent model was run several times to find the background velocity resulting in zero net force on the wedge.

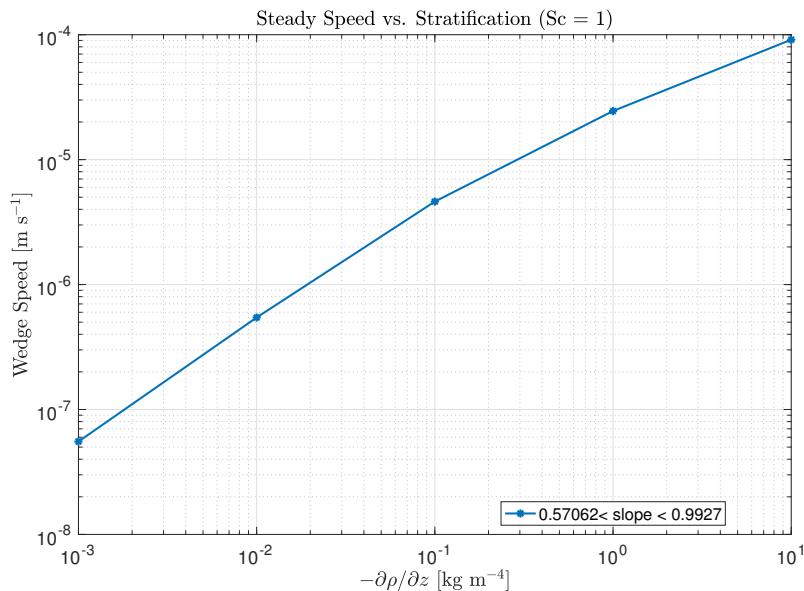


Figure 27: Wedge speed as a function of  $\partial\rho/\partial z$  for  $Sc = 1$

As the figure illustrates, the steady speed increases as the background density gradient increases, which matches the expectation. In fact, the background velocity  $U \propto (-\partial\rho/\partial z)^\zeta$ , where  $\zeta$  is the slope on this log-log plot. For  $Sc = 1$  and the range of stratification shown, this slope  $\zeta$  ranges from about 0.6 to 1.0.

#### 5.4.2 Schmidt Number = 100

Next, the same study was carried out with  $Sc = 100$ . The results are shown in Fig. 28.

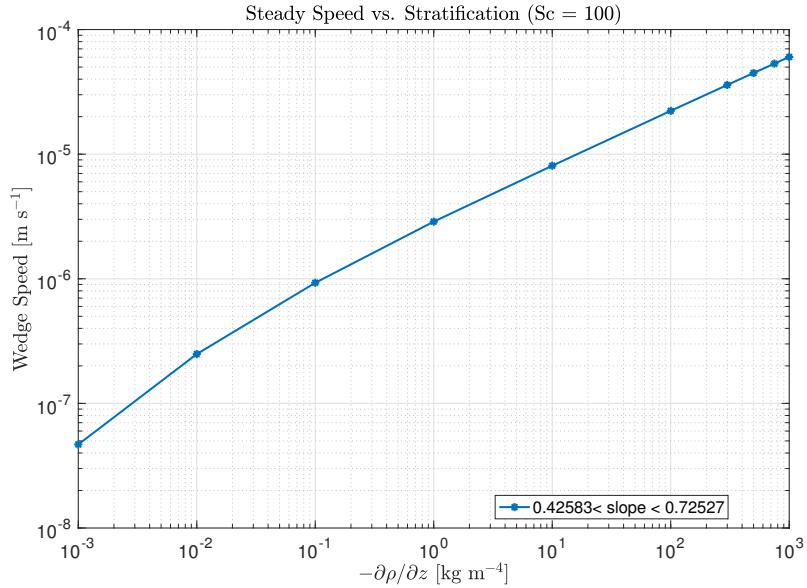


Figure 28: Wedge speed as a function of  $\partial\rho/\partial z$  for  $Sc = 100$

This time, the slope varies from about 0.4 to 0.7, with the larger slope only occurring at low stratification. Thus, when the Schmidt number is set to 100 as opposed to 1, the steady velocity increases at a slower rate with increased stratification. It is also worth noting that at the same stratification of  $-1 \text{ kg/m}^4$ , the steady wedge speed is lower for a higher Schmidt number. This matches the trend observed in the experiments [3].

#### 5.4.3 Schmidt Number = 1000

Finally, the Schmidt number was increased to 1000 and the same simulations were run. The results are shown in Fig. 29.

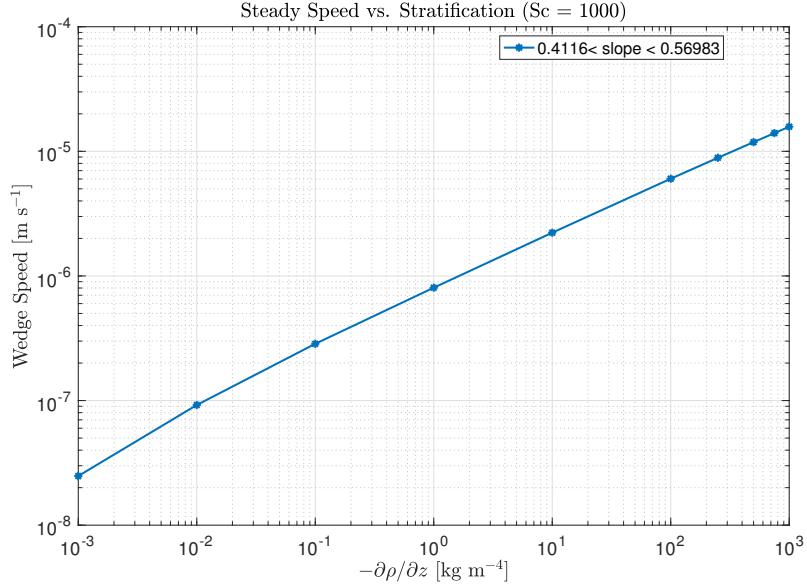


Figure 29: Wedge speed as a function of  $\partial\rho/\partial z$  for  $Sc = 1000$

In this case, the slope varies from about 0.41 to 0.57. Again, at the same stratification the steady speed is lower with the increased Schmidt number, as expected.

#### 5.4.4 Discussion

In all three studies, the steady speed increases roughly as  $(-\partial\rho/\partial z)^\zeta$ , with  $\zeta$  ranging from about 0.4 to 1.0. As the Schmidt number was increased, the range of the slope  $\zeta$  narrowed and centered around 0.5. In all cases, the slope at low stratification was higher than the slope at high stratification, meaning increases in stratification have a larger effect on the steady wedge speed when the stratification is smaller. In the experimental results [3], steady wedge speed was shown to increase as  $(-\partial\rho/\partial z)^{0.25}$  for Schmidt number greater than 100 and background density gradient stronger than  $10\ kg/m^4$ . The difference in these results can be attributed to several factors.

First, and most importantly, only some of the physical parameters of the fluid were quoted in the experimental paper. For example, either a change in viscosity or diffusivity could result in a change in Schmidt number. When the authors of the study increased the Schmidt number of the fluid, it is likely that both viscosity and diffusivity were changed but it is unclear exactly how. In this project, the Schmidt number was changed by modifying the diffusivity alone. Additionally, changing the stratification could change the reference density, but again the exact manner in which these parameters were changed in the experimental paper remains unknown.

Thus, it is difficult to fully validate the results of the study in this project because

no known experiment matches the parameters exactly. It would be a worthwhile exercise to re-examine this computational study and change the parameters in different ways. For example, change the viscosity to change the Schmidt number instead of changing the diffusivity.

Even with the uncertainty around the agreement of experimental results with the computational results in this paper, the key takeaway between both studies is that the steady speed increases sub-linearly with an increase in the stratification. The exact relationship depends on what other parameters are changed, but when considering how to take advantage of this effect in an engineering sense, it is important to understand the effect of increasing the stratification.

## 5.5 Wedge Slope Study

Next, a study of the effect of wedge slope  $\alpha$  was conducted. Again, the quantity of interest is the steady wedge speed as a function of the wedge slope (or wedge angle). This study was conducted in order to examine at what point diffusion-driven flow breaks down. This is greatly important in understanding when this effect is significant and when it can be ignored. The results of the study are shown in Fig. 30, where the wedge angle is in degrees. Note that non-dimensional background velocity is plotted against wedge angle. Non-dimensional velocity was selected as a metric because it more clearly illustrates the trends. Wedge angle was plotted as opposed to wedge slope because it is easier to interpret physically.

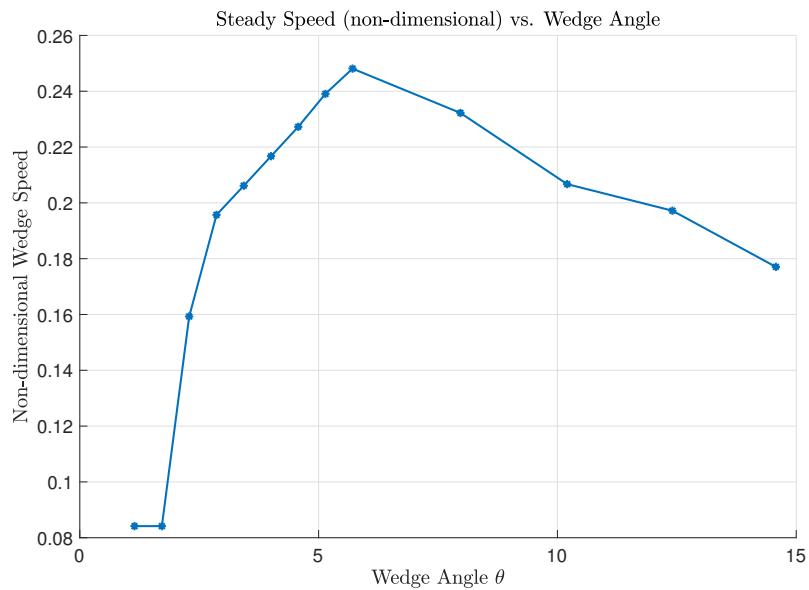


Figure 30: Non-dimensional wedge speed as a function of wedge angle

At wedge angles greater than  $6^\circ$ , the background velocity steadily drops. This is the result of greater drag on a wedge with a larger cross-section blocking the incoming flow. At wedge angles less than  $6^\circ$ , the background velocity drops off more quickly. It has been shown previously that the theory of diffusion driven flow breaks down at small angles [4] and this has been verified experimentally [25]. A breakdown at  $5^\circ$  was observed in the 2010 paper [3]. Thus, for large wedge angles, diffusion-driven flow generates propulsion that is counterbalanced by increasing viscous drag, but for small angles, the diffusion-driven flow no longer provides significant propulsion. In this case,  $6^\circ$  is an optimal wedge angle for maximizing the wedge speed.

## 5.6 Schmidt Number Study

The final study examines the effect of changing the Schmidt number while holding all other parameters constant. This study was conducted for a range of Schmidt numbers at two different levels of stratification.

### 5.6.1 Low Stratification

First, the study was conducted for  $\partial\rho/\partial z = -1 \text{ kg m}^{-4}$ . The non-dimensional wedge velocity resulting in zero net force on the wedge is plotted in Fig. 31 as a function of the Schmidt number.

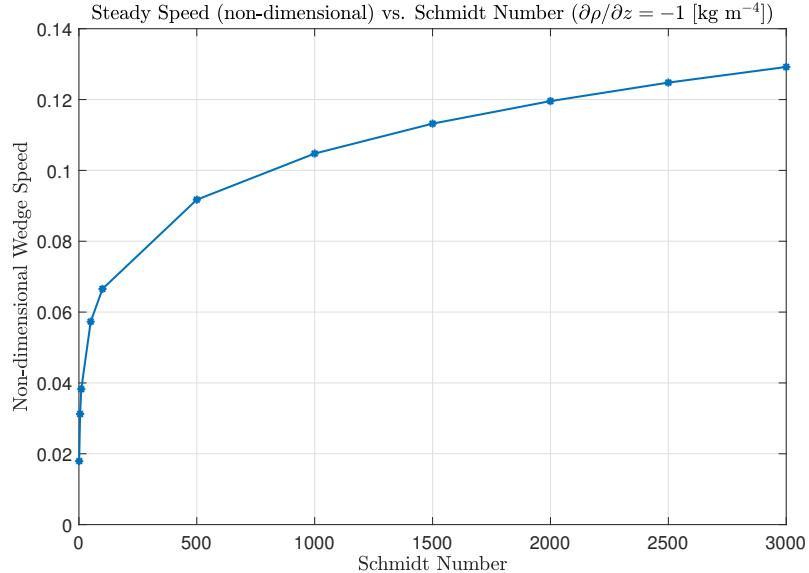


Figure 31: Non-dimensional wedge speed as a function of Schmidt number for  $\partial\rho/\partial z = -1 \text{ [kg m}^{-4}\text{]}$

In the 2010 experimental study, it was shown that non-dimensional velocity was

not a function of Schmidt number for  $527 < \text{Sc} < 3143$  [3]. In particular, the non-dimensional velocity remained near 0.06 in all trials. In the computational study conducted for this project, the non-dimensional velocity is not independent of Schmidt number. As Schmidt number is increased, the non-dimensional velocity also increases.

There are two key differences between the experimental and computational results. First, the computational study presented here explores a wider range of Schmidt numbers. At low Schmidt number, the dependence of wedge speed on Schmidt number is more significant. Second, as discussed earlier, the Schmidt number was changed by changing only the diffusivity, whereas in the experiment the viscosity of the fluid was likely also changed. This would not only affect other parameters of the problem, like the Reynolds number, but it would also change the reference velocity and thus would affect the non-dimensional velocity.

For reference and comparison with the next study, the dimensional results for the same study are shown in Fig. 32.

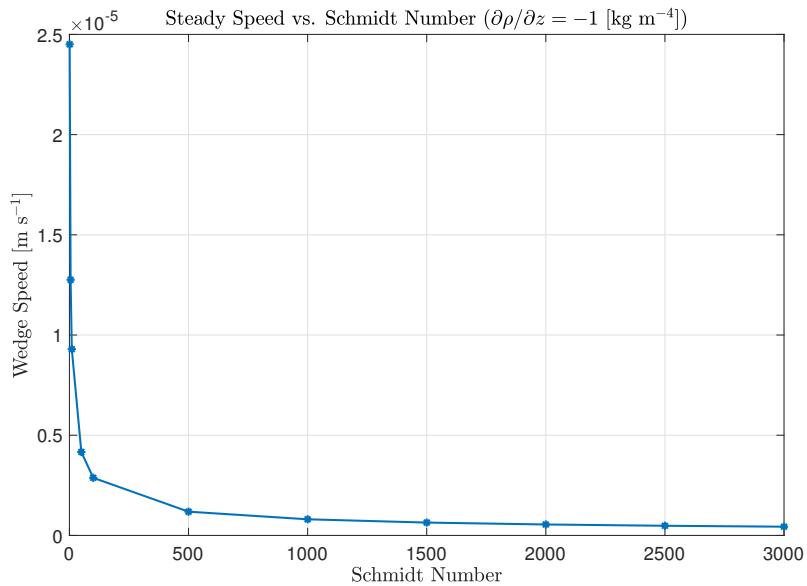


Figure 32: Wedge speed as a function of Schmidt number for  $\partial\rho/\partial z = -1 \text{ [kg m}^{-4}\text{]}$

### 5.6.2 High Stratification

In the experimental study, the dependence on Schmidt number was characterized for fluids with a greater stratification. In order to test this numerically, the computational study was also run for  $\partial\rho/\partial z = -100 \text{ kg/m}^4$ . The same procedure was followed to determine the background speed at which the net force on the wedge was zero for a range of Schmidt numbers. The results are shown in Fig. 33.

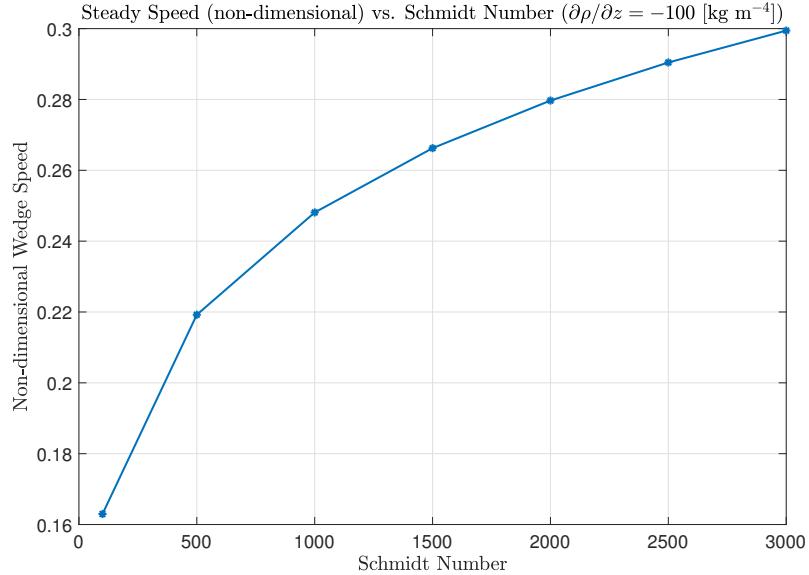


Figure 33: Non-dimensional wedge speed as a function of Schmidt number for  $\partial\rho/\partial z = -100 \text{ [kg m}^{-4}\text{]}$

The profile shown in the figure above has the same general shape as that of the study with lower stratification. It is important to note that the scale is different and that very small values of Schmidt number were not studied. The primary difference between these results and the previous results is the magnitude of the non-dimensional velocity. Because the stratification directly affects the reference velocity, it is more appropriate to examine the dimensional background velocity in order to compare the two studies.

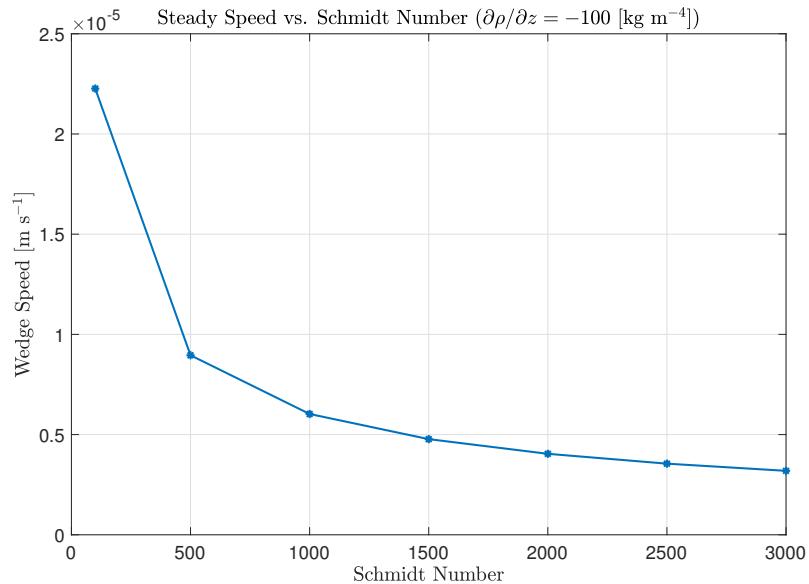


Figure 34: Wedge speed as a function of Schmidt number for  $\partial\rho/\partial z = -100 \text{ [kg m}^{-4}\text{]}$

Comparing Fig. 34 with Fig. 32 from earlier, it is clear that the background

velocity is greater in the case with greater background stratification. This makes sense given the previous discussion of the effect of stratification on wedge speed.

## 5.7 Discussion

In all three studies, the numerical results make sense given the experimental results from the 2010 paper [3]. The dependence of wedge speed on stratification is both sub-linear, with a stronger dependence in the computational results. The effect of changing the wedge angle in the numerical simulations matched the expectation closely, illustrating a breakdown of the effect of diffusion-driven flow around  $6^\circ$ . Finally, increasing the diffusivity was found to increase the wedge speed, a result that is difficult to verify using the experimental results but which makes sense given the relevant mechanisms in the problem.

The discrepancy between experimental and numerical results in all cases can be attributed to three primary factors. First, the numerical simulation was two dimensional while the experiments involved a three dimensional wedge. Adding a third dimension results in additional drag on the side of the wedge, which certainly decreases the wedge speed and possibly changes the relationship between wedge speed and other parameters of the fluid. Second, the domain in the numerical simulation was smaller than the experimental apparatus because it facilitated better convergence and shorter computation time. This likely results in a slightly higher resultant wedge speed in all cases, as shown in the domain size study. Finally and most significantly, the exact parameters of the fluid did not necessarily match between the experimental and numerical results. With a better understanding of the experimental setup, it would be possible to directly validate the numerical model.

Even with the discrepancies stated, the numerical simulation re-creates the same qualitative trends found in the experiments. This serves to partially validate the model and provides a better understanding of the flow as a whole.

## 6 Numerical Results: Non-Diffusive Limit

The implementation of the non-diffusive case is much simpler. The only parameters prescribed are the non-dimensional stratification  $\beta$  and the slope of the wedge  $\alpha$ . In a basic case with  $\beta = 0$  and  $\alpha = 0.1$ , the problem reduces to Stokes flow around a wedge. The velocity field can be seen in Fig. 35.

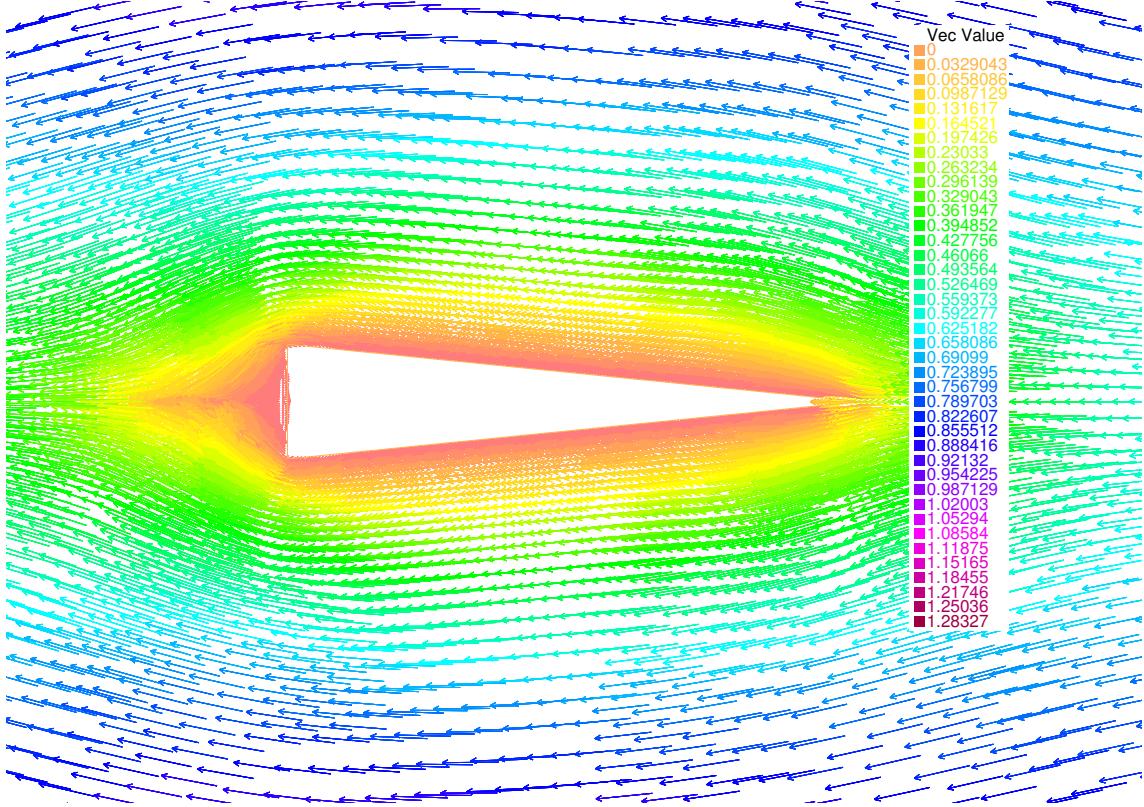


Figure 35: Non-diffusive limit with  $\beta = 0$ : Stokes flow

In this case, the force on the wedge is negative, as expected because there is no density gradient to create the diffusion-driven flow. Increasing  $\beta$  to 1000 results in the flow field in Fig. 36.

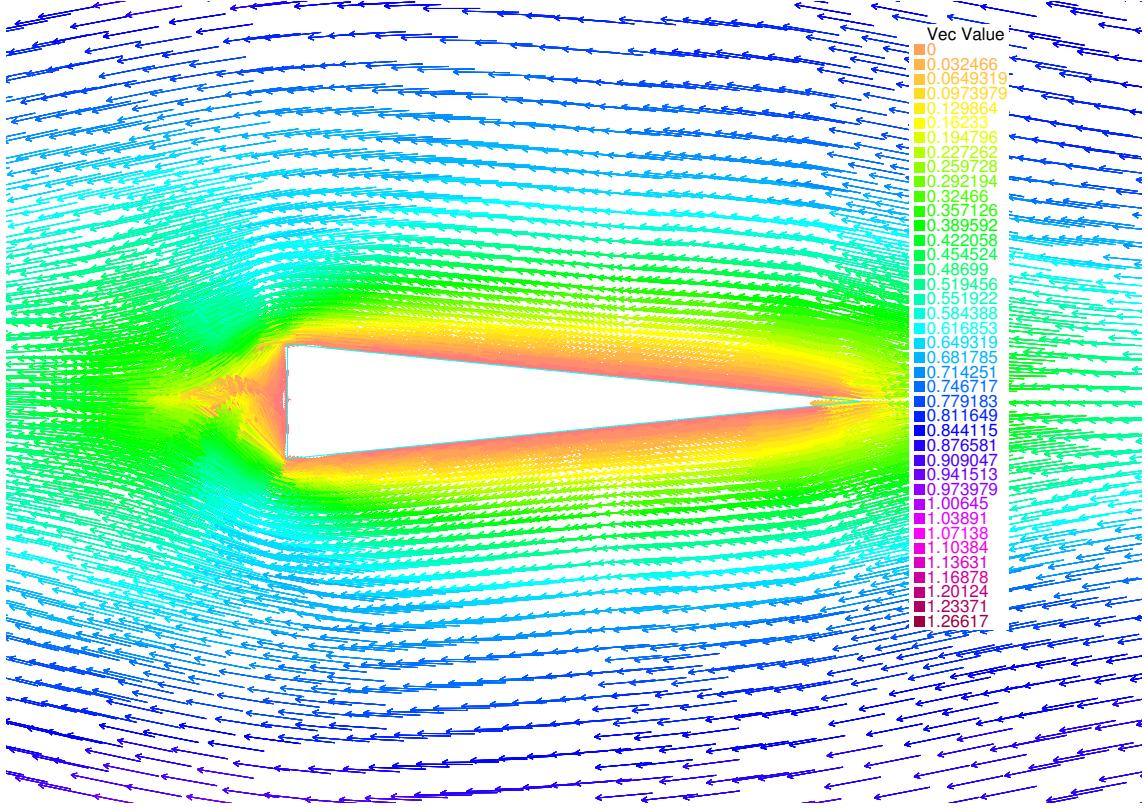


Figure 36: Non-diffusive limit with  $\beta = 1000$

Again in this case, the force on the wedge is negative. In fact, for every value of  $\beta$  tested, the wedge force found was negative, suggesting that this model does not support a forward propulsive force being generated on the wedge. This makes sense, given that the source of the propulsion is the effect of diffusion-driven flow above and below the sloping surfaces of the wedge. In the non-diffusive limit, the diffusion of density and the convection of momentum are both neglected, and these are significant mechanisms in the flow under study. As shown in the time-dependent case, increasing the Schmidt number (decreasing the diffusivity) results in a significant reduction of propulsion. In the limiting case of zero diffusivity, it is reasonable to expect that no propulsion would be generated. Thus, the non-diffusive case was not studied further. In the future, it would be worth exploring this case in more detail in order to re-examine the initial assumptions and further study the effects of increasing  $\beta$ . The value of a linear model would be significant in assisting with the efficient simulation of these flows.

## 7 Conclusion

In this project, the flow of a stratified fluid around a wedge-shaped object was studied. Based on the original work on diffusion-driven flow in 1970 and the experimental work studying the propulsion of a wedge, the aim of this project was to develop a model of this flow and explore the effect of various parameters on the model solution.

### 7.1 Summary of Research Conducted

Three models were developed: a stationary nonlinear model, a time-dependent nonlinear model, and a linear, non-diffusive model. It was shown that both the stationary and the non-diffusive models had shortcomings. In particular, the stationary model struggled to converge at high stratification. The non-diffusive model appeared to neglect the terms which are vital in generating the propulsion. Thus, the numerical study of these two models was very limited.

The time-dependent nonlinear model, however, was successfully implemented to re-create the trends of the 2010 experimental work. It was shown that increasing the background stratification of the fluid increases the wedge velocity, as expected. It was also shown that decreasing the wedge angle below about  $6^\circ$  results in the breakdown of the theory of diffusion driven flow, as has been shown both theoretically and experimentally. Finally, decreasing the diffusivity or increasing the Schmidt number results in a decrease of the wedge speed. This further verifies the source of the propulsion: diffusion-driven flow up the surface of the wedge creating regions of relative low pressure along the sloping surfaces.

The research conducted in this project serves to expand the body of understanding of diffusion-driven flow around a wedge. This is the first computational work that has studied in detail the same effects studied in the 2010 experimental paper, in particular the dependence of the wedge speed on the parameters of the flow.

### 7.2 Strengths and Weaknesses

The models and their implementations each have strengths and weaknesses, many of which have already been discussed. The primary weaknesses of the stationary and non-diffusive models are the assumptions under which they were derived, in particular that the solution is steady and that diffusive effects can be ignored. Their

strength is in their simplicity and computational tractability. The remainder of the discussion of strengths and weaknesses will be focused on the nonlinear time-dependent model.

The principal strength of the time-dependent model is the accuracy with which it models the physical effects taking place in the fluid. By making few assumptions, retaining time dependence, and solving the full nonlinear problem, the model closely replicates the underlying physics. This provided results that made sense and followed the same trends found in the experimental study.

This accuracy comes at significant cost. Because of the fixed point iteration and the time-stepping, computing the force at a given background velocity proved computationally intensive. In order to obtain a well-converged solution, the computation took close to one hour. This was simply to study one background velocity; in practice, finding the background velocity which resulted in zero drag on the wedge required several of these simulations. This computational complexity was distributed over many instances of Amazon Web Services compute nodes, making it feasible for this project, however this could prove limiting in the future. Another weakness of the model is the domain size. By not extending the domain, the fluid was forced to flow in a more channel-like configuration than in the experiments. Finally, the two-dimensional assumption was limiting. While it serves as an effective analogue for an infinite wedge, in order to truly estimate the force on the wedge the sides of the wedge must also be considered.

The weaknesses of the model in this project provide the basis for ideas for future work.

### 7.3 Ideas for Future Work

There is still much left to explore in the area of diffusion-driven flow over a wedge in a stratified fluid. In the future, it would be interesting to explore the same model in three-dimensions. This would allow for a more accurate solution (and the study of edge effects), but would prove more computationally expensive. In order to scale up the computation, it would be worthwhile to parallelize the code so that it could be run more efficiently on multi-core systems. Parallel solvers exist and would be useful to improve the speed of the finite element solution when the number of elements and nodes increases significantly.

It would be interesting to further explore the parameter space, for instance by altering the viscosity of the underlying fluid. Extending this analysis to a thermally stratified fluid would also be a worthwhile exercise in order to determine the magnitude of propulsion that is feasible before turbulent effects arise.

The non-diffusive limit deserves further study both analytically and numerically, in order to determine if it is possible to develop a linear model of the flow for more tractable computation. If so, the applicability of such a linear model should be explored in detail in order to determine when the nonlinear effects may become too significant to be ignored.

Finally, this problem should be studied with other interesting geometries. This same effect can take place on any submerged, asymmetric object, so it may be possible to reduce the drag or increase the propulsion on a differently shaped object. Perhaps a wedge-shaped object with an entrant corner on the back edge or the limiting case of the shell of the sloping surfaces would prove interesting for further study.

## References

1. Huppert, H. E. & Turner, J. S. On melting icebergs. *Nature* **271**, 46–48 (Jan. 1978).
2. Gurnis, M. Large-scale mantle convection and the aggregation and dispersal of supercontinents. *Nature* **332**, 695–699 (Apr. 1988).
3. Allshouse, M. R., Barad, M. F. & Peacock, T. Propulsion generated by diffusion-driven flow. *Nature Physics* **6**, 516–519 (2010).
4. Wunsch, C. On oceanic boundary mixing. *Deep Sea Research and Oceanographic Abstracts* **17**, 293–301 (1970).
5. Phillips, O. M. On flows induced by diffusion in a stably stratified fluid. **17**, 435–443 (1970).
6. Page, M. A. Propelled by diffusion. *Nature Physics* **6**, 486–487 (June 2010).
7. Mercier, M. J., Ardekani, A. M., Allshouse, M. R., Doyle, B. & Peacock, T. Self-propulsion of immersed objects via natural convection. *Physical review letters* **112**, 204501 (2014).
8. Chashechkin, Y. & Dimitrieva, N. *Dynamics and Structure of Stratified Flows Around Wedge: from Diffusion Induced up to Transient Vortex Regime* Feb. 2016.
9. Dimitrieva, N. F. & Chashechkin, Y. D. Fine Structure of a Stratified Flow around a Fixed and Slow-Moving Wedge. *Oceanology* **58**, 340–349 (May 2018).
10. Doyle, B. P. *Propulsion via buoyancy driven boundary layer* PhD thesis (Massachusetts Institute of Technology, 2011).
11. Turner, J. S. *Buoyancy effects in fluids* (Cambridge University Press, 1979).
12. Anderson, J. *Fundamentals of Aerodynamics* (McGraw-Hill Education, 2010).
13. Johnson, E. & Burman, E. *Fluid Flow around a Wedge in Stratified Fluid* Mar. 2018.
14. Hughes, T. J. *The finite element method: linear static and dynamic finite element analysis* (Courier Corporation, 2012).
15. Zienkiewicz, O. C., Taylor, R. L., Zienkiewicz, O. C. & Taylor, R. L. *The finite element method* (McGraw-Hill London, 1977).

16. Burman, E. *Email: Re: Updated Draft* Sept. 2018.
17. Burman, E. & Hansbo, P. Edge stabilization for the generalized Stokes problem: A continuous interior penalty method. *Computer Methods in Applied Mechanics and Engineering* **195**, 2393–2410 (2006).
18. Burman, E., Fernández, M. & Hansbo, P. Continuous Interior Penalty Finite Element Method for Oseen’s Equations. *SIAM Journal on Numerical Analysis* **44**, 1248–1274 (July 2006).
19. Burman, E. & Fernández, M. A. Continuous interior penalty finite element method for the time-dependent Navier–Stokes equations: space discretization and convergence. *Numerische Mathematik* **107**, 39–77 (July 2007).
20. Kim, T.-Y., Park, E.-J. & Shin, D.-w. A C0-discontinuous Galerkin method for the stationary quasi-geostrophic equations of the ocean. *Computer Methods in Applied Mechanics and Engineering* **300**, 225–244. ISSN: 0045-7825 (2016).
21. Engel, G. *et al.* Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Computer Methods in Applied Mechanics and Engineering* **191**, 3669–3750 (2002).
22. Hecht, F. New development in FreeFem++. *J. Numer. Math.* **20**, 251–265 (2012).
23. Hecht, F. & Pironneau, O. *FreeFem++ Manual* 2018.
24. Davis, T. A. Algorithm 832: UMFPACK V4.3—an Unsymmetric-pattern Multifrontal Method. *ACM Trans. Math. Softw.* **30**, 196–199 (June 2004).
25. Peacock, T., Stocker, R. & Aristoff, J. M. An experimental investigation of the angular dependence of diffusion-driven flow. *Physics of Fluids* **16**, 3503–3505 (2004).

# Appendix

The code used for this project and the simulation results can be found in the following GitHub repository:

<https://github.com/mhurst4186/Hurst-UCL-MSc-Code-Repository>

## Nonlinear Time-Dependent Code with Zero Finding

An example of the nonlinear time-dependent model code is presented here in its entirety.

```
// Time-Dependent Nonlinear Model with Zero Finding
// Author: Matthew Hurst
// Based on Codes from Erik Burman

// ##### Preliminaries to set up input/output #####
ifstream filein("timeanddate");
string datestr;
filein >> datestr;
string dir = "Results/" + datestr + "/";
ofstream setup(dir + "problem_parameters.sol");

// ##### Define some useful functions #####
macro dn(u) (N.x*dx(u)+N.y*dy(u)) // normal derivative
macro div(u,v) (dx(u)+dy(v)) // divergence
macro eps1n(u,v) (dx(u)*N.x+0.5e0*(dy(u)+dx(v))*N.y) // drag stress
macro eps2n(u,v) (dy(v)*N.y+0.5e0*(dy(u)+dx(v))*N.x) // lift stress (not used)

// ##### Define and initialize variables #####
// Stabilisation (all are turned off except for gammap – pressure stabilization)
real gammap, gammau, gammabc, gdiv;
gammap=0.1e-1;
gammau=0.1e-1;
gammabc=10.0e0;
gdiv = 0.;

// Relaxation
real relax, urelax;
relax = 0.1; // Density relaxation
```

```

setup << "relax: " << relax << endl;
urelax = 0.1; // Velocity relaxation
setup << "urelax: " << urelax << endl;

// Time stepping
real dt = 10.0; // Non-dimensional time step (determined empirically)
setup << "dt: " << dt << endl;

// Convergence tolerance and increments
real fixedpointtol = 0.01; // used at each time step
setup << "fixedpointtol: " << fixedpointtol << endl;
real timetol = 1.0e-5; // used to determine when solution no longer changes in time
real forcetol = 1.0e-5; // used to determine when force no longer changes
real newtontol = 1e-3; // also called "secant tol," used in zero finding
real veltol = 1e-3; // used in zero finding
real incrhocurrent, incvelcurrent, inctimecurrent, incforce, veldiffcurrent;

// Counters
int i; // used for plotting
int fixedpointiter=0;
int fixedpointitermax=250;
int timeiter=0;
int newtoniter=2;

// Geometry
real Ldim=0.01; // dimensional wedge length (m)
setup << "Ldim: " << Ldim << endl;
real L=1.0; // non-dimensional wedge length
setup << "L: " << L << endl;
real alpha=0.1; // non-dimensional wedge slope
setup << "alpha: " << alpha << endl;
real alpharad = atan(alpha); // as an angle (radians)
setup << "alpharad: " << alpharad << endl;
real area = Ldim^2*alpha; // not used, but could be used to model the transient
cout << "area: " << area << endl;

// Physical fluid parameters (fixed in all simulations)
real g=9.81; // gravitational acceleration (m s^-2)
setup << "g: " << g << endl;
real nu = 1.e-6; // kinematic viscosity (m^2 s^-1)

```

```

setup << "nu: " << nu << endl;
real rho00=1115.0; // dimensional reference density (kg m^-3)
setup << "rho00: " << rho00 << endl;
real mu = nu*rho00; // dynamic viscosity (kg m^-1 s^-1)
setup << "mu: " << mu << endl;

// Controlling the schmidt number
real kappa0=1.e-9; // dimensional diffusivity constant (m^2 s^-1)
setup << "kappa0: " << kappa0 << endl;
real Sc=nu/kappa0; // Schmidt Number
setup << "Sc: " << Sc << endl;

// Controlling the density gradient
real drhodz=-5.0e3; // density gradient (kg m^-4)
setup << "drhodz: " << drhodz << endl;

// Determining the reference velocity
real Nvar=(-g*drhodz/rho00)^(0.5); // buoyancy frequency (s^-1)
setup << "Nvar: " << Nvar << endl;
real gamma=(Nvar^2*(sin(alpharad))^2/(4.0*nu*kappa0))^(0.25);
setup << "gamma: " << gamma << endl;
real u0=2.0*kappa0*gamma/tan(alpharad); // reference velocity (m s^-1)
setup << "u0: " << u0 << endl;

// Buoyancy parameter
real beta0=-Ldim*drhodz/rho00; // non-dimensional stratification parameter
setup << "beta0: " << beta0 << endl;
real beta=beta0*rho00*g*Ldim^2/(u0*mu); // scaled stratification parameter
setup << "beta: " << beta << endl;

// Reynolds number
real Re=u0*Ldim/nu; // Reynolds Number
setup << "Re: " << Re << endl;

// Diffusivity parameter
real kappa=1./(Re*Sc); // Non-dimensional diffusivity
setup << "kappa: " << kappa << endl;

// Strouhal number
real St=Nvar*Ldim/u0; // Strouhal number

```

```

setup << "St: " << St << endl;

// Variables to keep track of force and velocity for zero-finding
real force1, force2, force3;
real forceold;
real Ubackground1, Ubackground2, Ubackground3;
real U;

// Domain setup
real xlen=5.0, ylen=2.5; // size of domain
setup << "xlen: " << xlen << endl;
setup << "ylen: " << ylen << endl;
real S=99; // label for the wedge border

// Plotting option
bool toPlot=true;

// ##### Design (and plot) the geometry of the domain #####
// Outside border
border C1(t=0,1) { x=-xlen*(1-t)+xlen*t-L/2; y=-ylen; label= 1; }
border C2(t=0,1) { x=xlen-L/2; y=-ylen*(1-t)+ylen*t; label= 1; }
border C3(t=0,1) { x=xlen*(1-t)-xlen*t-L/2; y=ylen; label= 1; }
border C4(t=0,1) { x=-xlen-L/2; y=ylen*(1-t)-ylen*t; label= 1; }

// Wedge border
border Splus(t=L,0){ x = -t; y = alpha*t ; label=S; }
border Sminus(t=0,L){ x = -t; y = -alpha*t ; label=S; }
border Left(t=0,1){ x = -L; y = -alpha*L + t*2*alpha*L ; label = S; }

// Refinement parameter(s) (can be changed independently)
int outrefinement = 8;
int wedgerefinement = 8;

// Number of points along domain boundaries
int topandbottom = 2*outrefinement;
setup << "topandbottom: " << topandbottom << endl;
int leftandright = 1*outrefinement;
setup << "leftandright: " << leftandright << endl;
int wedgeslope = 8*wedgerefinement;

```

```

setup << "wedgeslope: " << wedgeslope << endl;
int wedgeback = 2*wedgerefinement;
setup << "wedgeback: " << wedgeback << endl;
setup.flush;

// Create and save mesh (and plot optionally)
mesh th= buildmesh(C1(topandbottom)+C2(leftandright)+C3(topandbottom)+C4(leftandright)
    ↪ +Splus(wedgeslope)+Left(wedgeback)+Sminus(wedgeslope));
savemesh(th,dir+"Th.msh");
if (toPlot) plot(th,bb=[[-xlen-L/2,-ylen],[xlen-L/2,ylen]],ps=dir+"mesh.eps");
if (toPlot) plot(th,bb=[1.2*[-L,-alpha*L],[0.2*L,1.2*alpha*L]],ps=dir+"meshzoom.eps");

// ##### Define the approximation space and finite element functions
↪ #####
fespace Vh(th,P2); // p2 space (used for all solution variables)
Vh u1,u2,p; // velocity and pressure (solution variables)
Vh v1,v2,q; // test functions in NS
Vh rho; // density (solution variable)
Vh v; // test function in convection-diffusion
Vh u1iter,u2iter,rhoiter; // place-holders for fixed point iteration
Vh u1old,u2old,rhoold; // place-holders for time-stepping

fespace Xh(th,P1); // p1 polynomial space (only used for plotting)
Xh p1,u11,u21,rho1;
Xh visc1,visc2;

// ##### Define some useful functionals #####
func pressureforce = int1d(th,S)(p*N.x);
func viscousforce = int1d(th,S)(-eps1n(u1,u2));
func incrho = sqrt(int2d(th)((rhoiter-rho)^2)/int2d(th)(rho^2));
func incvel = sqrt(int2d(th)((u1-u1iter)^2+(u2-u2iter)^2)/int2d(th)(u1^2+u2^2));
func incetime = sqrt(int2d(th)((rhoold-rho)^2)/int2d(th)(rho^2))+sqrt(int2d(th)((u1-u1old)
    ↪ ^2+(u2-u2old)^2)/int2d(th)(u1^2+u2^2));

// ##### Variational form of the convection-diffusion equation #####
problem convdiff(rho,v,solver=UMFPACK) =
    int2d(th)(St*rho/dt*v + kappa*(dx(rho)*dx(v) + dy(rho)*dy(v)) ) // kappa fusion

```

```

+ int2d(th)( u1iter*dx(rho)*v + u2iter*dy(rho)*v ) // advection
// stabilization necessary if kappa << |b| h$
// + intalledges(th)( ((gammau*abs(b1*N.x + b2*N.y))*lenEdge^(2))*(jump(dn(rho))*jump(dn
    ↪ (v)))*(nTonEdge-1) ) // IP method
-int2d(th)(St*rhoold/dt*v)
+ on(1,rho=-y)
;

// ##### Variational form of the navier-stokes equation #####
problem NS([u1,u2,p],[v1,v2,q],solver=UMFPACK) =
int2d(th)(Re*St*(u1/dt*v1+u2/dt*v2)
+ (dx(u1)*dx(v1) + dy(u1)*dy(v1) + dx(u2)*
    ↪ dx(v2) + dy(u2)*dy(v2)) // comes
    ↪ from IBP of Laplacian operator
+ Re*(1-y*beta0)*(u1iter*dx(u1) + u2iter*dy(u1))*v1 // comes
    ↪ directly from conservation of x momentum
+ Re*(1-y*beta0)*(u1iter*dx(u2) + u2iter*dy(u2))*v2 // comes
    ↪ directly from conservation of y momentum
+ 0.0000000001*p*q // this small penalty term
    ↪ improves solvability
+ gdiv*div(u1iter,u2iter)*(u1*v1+u2*v2) // this term is turned off
- p*div(v1,v2) // comes from IBP of pressure
    ↪ term in conservation of momentum
+ q*(div(u1,u2)) // comes directly from the
    ↪ incompressibility equation
)
+ int2d(th)(rho*beta*v2) // comes from buoyancy term
-int2d(th)(Re*St*(u1old/dt*v1+u2old/dt*v2))

// Interior penalty stabilization of the streamline derivative + 5% crosswind diffusion
// + intalledges(th)(
// (gammau*(abs(b1*N.x + b2*N.y) +
// 0.05*(b1^2 + b2^2)^0.5))*lenEdge^(2)*
// (jump(dn(u1))*jump(dn(v1))+jump(dn(u2))*jump(dn(v2))*(nTonEdge-1)))
+ intalledges(th)( (gammap*min(lenEdge^(3)/nu,lenEdge^(2)/(1.e-16+(u1iter^2+
    ↪ u2iter^2)^0.5)))*
*(jump(dn(p))*jump(dn(q)))*(nTonEdge-1)) // this penalty on gradient
    ↪ fluctuations ensures pressure stability
// Strongly imposed boundary conditions
+ on(S,u1=0,u2=0)

```

```

+ on(1,u1=-U,u2=0)

;

// ##### MAIN PROGRAM
← #####
// Open output files
ofstream resultsout(dir+"results.sol");
ofstream forceout(dir+"force.sol");

resultsout << "Program Started: " << endl;

// Keep track of the time
real cpu = clock();

// ##### BEGIN FIRST SOLUTION LOOP
← #####
resultsout << "##### Iteration: 1 #####" << endl;

Ubackground1=0.; // background velocity (m s^-1) starting guess
resultsout << "Ubackground: " << Ubackground1 << endl;

U=Ubackground1/u0; // non-dimensional background velocity
resultsout << "U: " << U << endl;

// Set initial conditions (uniform flow at non-dimensional background velocity)
u1=-U;
u2=0.;
u1iter=u1;
u2iter=u2;
u1old=u1;
u2old=u2;
p=0;
rho=-y;
rhoiter=rho;
rhoold=rho;

// Start time iteration
timeiter=0;

```

```

inctimecurrent = 2*timetol*dt;
incforce = 2*forcetol*dt;
forceold = 0;

while ((inctimecurrent/dt > timetol) || (incforce/dt > forcetol)){
    timeiter++;
    resultsout << "time iteration: " << timeiter << endl;

    // Start fixed point iteration
    fixedpointiter = 0;
    incrhocurrent=2*fixedpointtol;
    incvelcurrent=2*fixedpointtol;

    while ((incrhcurren+incvelcurrent>fixedpointtol) && (fixedpointiter<
        ↪ fixedpointitermax)) {

        fixedpointiter++;
        resultsout << "fixed point iteration: " << fixedpointiter << endl;

        NS;
        incvelcurrent=incvel;
        u1=u1*urelax+(1.-urelax)*uliter;
        u2=u2*urelax+(1.-urelax)*u2iter;
        uliter=u1;
        u2iter=u2;
        convdiff;
        incrhcurren=incrhc;
        rho = relax*rho+(1.-relax)*rhoiter;
        rhoiter=rho;

        // Check the convergence progress
        resultsout << "fixed point increment: " << incrhcurren+incvelcurrent <<
            ↪ endl;
        resultsout.flush();
        //if (toPlot) plot(rho);
        //plot([u1,u2],bb=[[-xlen-L/2,-ylen],[xlen-L/2,ylen]],coef=1);
    }

    inctimecurrent=inctime;
}

```

```

        force1 = pressureforce+viscousforce;
        incforce = abs(force1 - forceold);
        resultsout << "time increment/dt: " << inctimecurrent/dt << endl;
        resultsout << "force increment/dt: " << incforce/dt << endl;
        resultsout.flush;

        forceout << force1 << endl;
        forceout.flush;

        // Reset old solution
        u1old=u1;
        u2old=u2;
        rhoold=rho;
        forceold = force1;
    }

resultsout << "time: " << clock() - cpu << endl;
resultsout << "timeiter: " << timeiter << endl;

force1 = pressureforce+viscousforce;
resultsout << "pressure force: " << pressureforce << endl;
resultsout << "viscous force: " << viscousforce << endl;
resultsout << "force: " << force1 << endl;

// ##### BEGIN SECOND SOLUTION LOOP
→ #####
resultsout << "##### Iteration: 2 #####" << endl;

Ubackground2=1.409e-5; // background velocity (m s^-1) starting guess
resultsout << "Ubackground: " << Ubackground2 << endl;

U=Ubackground2/u0; // non-dimensional background velocity
resultsout << "U: " << U << endl;

// Set initial conditions (uniform flow)
u1=-U;
u2=0.;
u1iter=u1;
u2iter=u2;

```

```

u1old=u1;
u2old=u2;
p=0;
rho=-y;
rhoiter=rho;
rhoold=rho;

// Start time iteration
timeiter=0;
inctimecurrent = 2*timetol*dt;
incforce = 2*forcetol*dt;
forceold = 0;

while ((inctimecurrent/dt > timetol) || (incforce/dt > forcetol)){
    timeiter++;
    resultsout << "time iteration: " << timeiter << endl;

    // Start fixed point iteration
    fixedpointiter = 0;
    incrhocurrent=2*fixedpointtol;
    incvelcurrent=2*fixedpointtol;

    while ((incrhcurren+incvelcurrent>fixedpointtol) && (fixedpointiter<
        ↪ fixedpointitermax)) {
        fixedpointiter++;
        resultsout << "fixed point iteration: " << fixedpointiter << endl;

        NS;
        incvelcurrent=incvel;
        u1=u1*urelax+(1.-urelax)*u1iter;
        u2=u2*urelax+(1.-urelax)*u2iter;
        u1iter=u1;
        u2iter=u2;
        convdiff;
        incrhcurren=incrhc;
        rho = relax*rho+(1.-relax)*rhoiter;
        rhoiter=rho;
    }
}

```

```

// Check the convergence progress
resultsout << "fixed point increment: " << incrhocurrent+incvelcurrent <<
    ↪ endl;
resultsout.flush();
//if (toPlot) plot(rho);
//plot([u1,u2],bb=[[-xlen-L/2,-ylen],[xlen-L/2,ylen]],coef=1);
}

inctimecurrent=inctime;
force2 = pressureforce+viscousforce;
incforce = abs(force2 - forceold);
resultsout << "time increment/dt: " << inctimecurrent/dt << endl;
resultsout << "force increment/dt: " << incforce/dt << endl;
resultsout.flush();

forceout << force2 << endl;
forceout.flush();

// Reset old solution
u1old=u1;
u2old=u2;
rhoold=rho;
forceold = force2;
}

resultsout << "time: " << clock() - cpu << endl;
resultsout << "timeiter: " << timeiter << endl;

force2 = pressureforce+viscousforce;
resultsout << "pressure force: " << pressureforce << endl;
resultsout << "viscous force: " << viscousforce << endl;
resultsout << "force: " << force2 << endl;

// ##### BEGIN ADDITIONAL SOLUTION LOOPS
    ↪ #####
veldiffcurrent = 2*veltol;
while ((veldiffcurrent>veltol) && (abs(force2)>newtontol)){
    newtoniter++;
}

```

```

resultsout << "##### Iteration: "+newtoniter+" #####"
    ↪ << endl;

Ubackground3=Ubackground2 - force2/((force2-force1)/(Ubackground2-
    ↪ Ubackground1)); // background velocity (m s^-1)
resultsout << "Ubackground: " << Ubackground3 << endl;

U=Ubackground3/u0; // non-dimensional background velocity
resultsout << "U: " << U << endl;

// Set initial conditions (uniform flow)
u1=-U;
u2=0.;
u1iter=u1;
u2iter=u2;
u1old=u1;
u2old=u2;
p=0;
rho=-y;
rhoiter=rho;
rhoold=rho;

// Start time iteration
timeiter=0;
inctimecurrent = 2*timetol*dt;
incforce = 2*forcetol*dt;
forceold = 0;

while ((inctimecurrent/dt > timetol) || (incforce/dt > forcetol)){
    timeiter++;
    resultsout << "time iteration: " << timeiter << endl;

    // Start fixed point iteration
    fixedpointiter = 0;
    incrhcurren=2*fixedpointtol;
    incvelcurrent=2*fixedpointtol;

    while ((incrhcurren+incvelcurrent>fixedpointtol) && (fixedpointiter<
        ↪ fixedpointitermax)) {

```

```

fixedpointiter++;
resultsout << "fixed point iteration: " << fixedpointiter << endl;

NS;
incvelcurrent=incvel;
u1=u1*urelax+(1.-urelax)*u1iter;
u2=u2*urelax+(1.-urelax)*u2iter;
u1iter=u1;
u2iter=u2;
convdiff;
incrhocurrent=incrho;
rho = relax*rho+(1.-relax)*rhoiter;
rhoiter=rho;

// Check the convergence progress
resultsout << "fixed point increment: " << incrhocurrent+
    ↪ incvelcurrent << endl;
resultsout.flush;
//if (toPlot) plot(rho);
//plot([u1,u2],bb=[[-xlen-L/2,-ylen],[xlen-L/2,ylen]],coef=1);
}

inctimecurrent=inctime;
force3 = pressureforce+viscousforce;
incforce = abs(force3 - forceold);
resultsout << "time increment/dt: " << inctimecurrent/dt << endl;
resultsout << "force increment/dt: " << incforce/dt << endl;
resultsout.flush;

forceout << force3 << endl;
forceout.flush;

// Reset old solution
u1old=u1;
u2old=u2;
rhoold=rho;
forceold = force3;
}

```

```

resultsout << "time: " << clock() - cpu << endl;
resultsout << "timeiter: " << timeiter << endl;

force3 = pressureforce+viscousforce;
resultsout << "pressure force: " << pressureforce << endl;
resultsout << "viscous force: " << viscousforce << endl;
resultsout << "force: " << force3 << endl;

Ubackground1=Ubackground2;
Ubackground2=Ubackground3;
veldiffcurrent = abs(Ubackground2-Ubackground1)/Ubackground1;

force1=force2;
force2=force3;

}

// ##### Plotting #####
plot([u1,u2],bb=[[-xlen-L/2,-ylen],[xlen-L/2,ylen]],coef=1,ps=dir+"vel.eps");
plot([u1,u2],bb=[1.2*[-L,-alpha*L],[0.2*L,1.2*alpha*L]],coef=1,value=true,nbarrow=40,ps=dir
    ↪ +"velzoom.eps");

plot(p,nbiso=100,fill=true,ps=dir+"p.eps");
plot(p,bb=[1.2*[-L,-alpha*L],[0.2*L,1.2*alpha*L]],value=true,nbiso=100,fill=true,ps=dir+
    ↪ pzoom.eps);

plot(rho,nbiso=100,fill=true,ps=dir+"rho.eps");
plot(rho,bb=[1.2*[-L,-alpha*L],[0.2*L,1.2*alpha*L]],value=true,nbiso=100,fill=true,ps=dir+
    ↪ rhozoom.eps");

// ##### Writing outputs #####
// Interpolate into P1 space for graphic representation
visc1=dx(u1);
visc2=0.5e0*(dy(u1)+dx(u2));
p1=p;
rho1=rho;
u11=u1;
u21=u2;

```

```

ofstream pout(dir+"p.sol");
ofstream rhout(dir+"rho.sol");
ofstream uxout(dir+"ux.sol");
ofstream uyout(dir+"uy.sol");
ofstream visc1out(dir+"visc1.sol");
ofstream visc2out(dir+"visc2.sol");

pout << p1.n << endl;
rhout << rho1.n << endl;
uxout << u11.n << endl;
uyout << u21.n << endl;
visc1out << visc1.n << endl;
visc2out << visc2.n << endl;

// Write solution variables to be post-processed in MATLAB
for (i=0;i<rho1.n;i++)
{
    pout << p1[i] << endl;
    rhout << rho1[i] << endl;
    uxout << u11[i] << endl;
    uyout << u21[i] << endl;
    visc1out << visc1[i] << endl;
    visc2out << visc2[i] << endl;
}

```