University of Alberta

Transient Deposition of Particles with Applications to Inhaled Pharmaceutical Aerosols

by

Philip S. Ophus

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

Spring 2011
Edmonton, Alberta

Examining Committee

Carlos F. Lange, Department of Mechanical Engineering

Warren Finlay, Department of Mechanical Engineering

Jason Olfert, Department of Mechanical Engineering

Faye Hicks, Department of Civil Engineering

This thesis is dedicated to Sisyphus. Don't give up hope!

Abstract

Most current models for the deposition of aerosol particles in the human lung are based on a Lagrangian reference frame, which is ill-suited for modeling transient effects. Deposition models based on an Eulerian reference frame are much better at capturing instantaneous time-dependent effects, though they are difficult to create. In the interest of developing such models, mathematical techniques were used to describe the velocity fields of simple particle flows. Analytic expressions describing the time-dependent flow of particles through a curved pipe were created and implemented numerically. The numerical simulations were used to determine which flow regimes required the use of Eulerian modelling for deposition prediction.

Preface

This thesis comprises six chapters. Chapter 1 outlines the current state of lung-deposition simulations and describes the motivation for creating an Eulerian framework for deposition. Chapter 2 describes attempted solution methods that were unsuccessful. Chapter 3 describes the method that eventually succeeded and the derivation of the primary analytic solution. Chapter 4 demonstrates the numerical implementation of the successful method. Chapter 5 describes and discusses the results and validation of the numerical simulations. Finally, chapter 6 contains conclusions and potential next steps of this work.

Acknowledgements

Thanks be to NSERC and the University of Alberta for their financial support;
to my supervisor, Dr. Carlos Lange, whose insight and guidance helped me along the path;
to my family for their support and encouragement;
and most of all to my brother, Dr. Colin Ophus, for his technical assistance, math help, brainstorming, motivation, and inflatable mattress.

Contents

List of Figures

List of Symbols

Symbol	Meaning [units]	Symbol	Meaning [units]
A	Fluid function intercept	t	Time [s]
a	Pipe radius [m]	t_f	Simulation end time [s]
A_{dep}	Deposited area [m^2]	t_i	Simulation start time [s]
A_{ol}	Overlap area [m^2]	t_0	Particle injection time [s]
B	Fluid function slope	u	Particle x-velocity [m/s]
b	Ellipse major axis [m]	$\vec{u}$	Particle velocity [m/s]
C	Relative concentration	v	Particle y-velocity [m/s]
d	Particle diameter [m]	W	Fluid speed [m/s]
f	Fluid velocity function	$\vec{w}$	Fluid velocity vector [m/s]
h	Ellipse offset distance [m]	w_x	Fluid x-velocity [m/s]
K_D	Coefficient of drag	w_y	Fluid y-velocity [m/s]
N_{CV}	Control volume count	y_0	Particle initial position [m]
R	Pipe bend radius [m]		
R_p	Pipe ratio		
Re_p	Particle Reynolds number	λ_n	Lagrange multiplier
S	Stokes number	μ	Dynamic viscosity [kg/m·s]
s	Complex particle position	ν	Kinematic viscosity [m^2/s]
Str	Strouhal number	ρ	Particle density [kg/m^3]
T	Fluid flow period [s]	ρ_c	Concentration [kg/m^3]

1

Introduction

1.1 Background and Motivation

1.1.1 The Utility of Lung Deposition Simulations

The study of inhaled aerosols is very important because of the significant impact, both positive and negative, that they can have on human health. Many dangerous aerosols are present in industrial environments, and some can even be found in residential environments (e.g. asbestos). But aerosols can also be helpful; the field of pharmaceutical aerosols is continually expanding because of the benefits this method of drug delivery has compared to others

such as injection and ingestion [3]. These benefits include safety, convenience, and rapid efficacy.

For both harmful and helpful aerosols, an accurate method of determining the fraction of an inhaled aerosol that actually deposits inside the respiratory system is very important. For harmful aerosols, this fraction combined with the known toxicity of a given material can be used to determine safe exposure levels. For medical aerosols, the efficacy of the treatment depends on the amount of drug delivered, so a fairly accurate prediction of this amount is required to prevent under- or over-dosing, either of which could be quite harmful to a patient.

Performing experiments to calculate aerosol deposition, either in real humans or using human respiratory system analogs, while feasible, can be difficult, costly, and time-consuming. An alternative solution is to predict deposition using mathematical models which are solved numerically. Such models can be based either on empirical observations from the aforementioned experiments, or on the physical laws governing the motion of particles in respiratory flow. The latter is preferable because the resulting models will be more generalized, though likely less accurate for a given specific case [4].

If a model can be created which accurately predicts deposition as a function of case-dependent variables such as particle size and density, lung geometry, and inhalation flow rate, such a model could be used with the new generation of aerosol inhalation devices to create a simple, safe, and effective way of treating a wide variety of diseases.

1.1.2 Eulerian Versus Lagrangian

Most existing aerosol deposition models use a Lagrangian reference frame in which the coordinate system follows a cloud of particles as it moves through

the respiratory tract. The primary advantage of this type of reference frame is its ease of implementation in numerical code. Usually, models of this sort are steady state; this is because it is very difficult to capture time-dependent effects using a Lagrangian reference frame [5]. Since respiration is transient by nature, some degree of error will result from these simplified models. For slow breathing cases this error may be negligible, but for high-acceleration cases such as inhalation through a jet nebulizer the error may be significant.

The alternative to the Lagrangian reference frame is the Eulerian reference frame in which the coordinate system is fixed to the respiratory tract geometry and the inhaled particles move through it. This type of reference frame is harder to implement, but it is inherently well-suited to modelling transient effects. There are some deposition models available which are based on an Eulerian framework, such as [6], [7], and [8]. These models predict deposition by solving a time-dependent transport equation which determines particle concentration throughout the lung, allowing them to accommodate transient boundary conditions and to predict instantaneous deposition rates within each lung generation. However, the underlying terms representing the physical deposition phenomena which make up the transport equations are not themselves transient, meaning that the time-history of the particles is neglected and inertial effects are lost. A model with a fully transient basis is the eventual goal of this project.

1.1.3 A Pseudo-Eulerian Method

The present work builds on that of [9], in which a more detailed discussion of current lung deposition models can be found. The goal of that work was to create a numerical conversion scheme which could be used to adapt standard Lagrangian deposition models to an Eulerian framework. Two such schemes were tested. The results were partially successful for the deposition mechanism

of sedimentation. In the end, this approach is more Lagrangian than Eulerian; it allows for the calculation of instantaneous deposition rate, but still does not account for the time-history of the particles and thus cannot properly model flow acceleration.

1.1.4 One-Dimensional vs. Three-Dimensional Models

The model that the present work contributes to will be of the simplified one-dimensional (1D) type that has existed for several decades. The examples of Eulerian models given above belong to this type. A more modern alternative is to model deposition using three-dimensional computation fluid dynamics (3D CFD) simulations. This approach is gaining popularity with the increasing power and decreasing cost of computer processors. The relative merits of 1D models such as [10], [11] compared to 3D CFD approaches such as [12] are discussed in Byron [13]. Byron concludes that CFD models are quite useful for revealing localized deposition behaviour and can be especially helpful in individualized cases such as targeted medication for a specific person's lung geometry. However, due to their high computational cost and lack of generality, CFD models cannot yet replace 1D models in the "development of nonindividualized aerosol drug delivery".

1.1.5 Goals

The present work sought to create a generalized Eulerian deposition model which utilized transient, fully analytic mathematical expressions based on physical laws. A full deposition simulation model is quite complex and large in scale, so simplified geometry was used and focus was given to deposition by inertial impaction. Impaction is the dominant effect for larger particles and is of particular importance in the mouth-throat region; thus, most of the

modelling considered heavy particles depositing in a curved pipe, which can be used as a rudimentary representation of the throat. Deposition also occurs by sedimentation due to gravity and diffusion due to Brownian motion, but these phenomena are less sensitive to transient effects and thus can be fairly well represented by Lagrangian models.

Since the solution of transient analytic equations in one or more dimensions is often difficult or even impossible, some simplifications were required to proceed. For this reason, rather than attempt to produce exact, accurate expressions for particle deposition, this work instead tried to merely identify functional forms for deposition. These forms could be modified using some number of fitting parameters to fit numerical and experimental results to create an analytic predictive model. The main advantage of such a model is that it is based on physical laws rather than being purely empirical, and thus it should apply to a wider variety of flow cases.

1.2 Methods

1.2.1 The Continuum Approach

The original approach taken to solve particle flow behaviour involved treating the aerosol particles as a separate continuum overlapping the fluid continuum. Simple geometries were used for the fluid flow, such as a perpendicular jet impinging on a flat wall and flow past a triangular wedge. These cases are roughly representative of flow passing from the mouth to the throat and the flow past a bifurcation in the lung, respectively. In this approach, the drag of the fluid on the particles creates a force vector field which, along with boundary conditions for the particle flow, was used to solve simplified versions of the Navier-Stokes functions with the goal of producing velocity and concentration

fields for the particles. Though this approach can be successful for simple cases, it ultimately proved unfeasible for flows with two spatial dimensions and a temporal dimension on any sort of useful geometry. Thus, the decision was made to switch to a single-particle approach.

1.2.2 The Single-Particle Approach

The alternative to an entire field of particles is to consider one particle at a time. The same known fluid flow fields can be used, and the same expression for the drag force, but instead of the Navier-Stokes equations, one or more PDEs are created with the direct application of Newton's second law. Solution of the PDEs yields a particle trajectory function which depends on time, initial conditions, and the fluid flow function. The trajectory function can be applied to a group of particles numerically and deposition determined, or the function can be combined with an appropriate geometric function to create an analytic deposition function.

2

Solution Attempts: The Continuum Approach

Initial attempts to solve particle motion in a fluid used the so-called "continuum" approach, in which the aerosol particles were treated as a second separate continuum overlaying the fluid continuum. With this method, particle-fluid interactions are represented by a force vector field, and particle behaviour is fully described by a vector field of particle velocity and a scalar field of particle concentration. Such an approach is inherently Eulerian.

Simple geometries were studied using this approach: first, a particle-laden jet impinging on a flat wall, which is representative of the flow through the mouth hitting the back of the throat; and later, flow past a wedge, which is repre-

sentative of any of the bifurcations in the lung. It was hoped that relatively simple particle flow fields could be obtained by solving the the differential equations describing these cases, which would then be extended to deposition models utilizing more realistic lung geometry.

Unfortunately, the continuum approach requires the use of the Navier-Stokes equations to describe particle motion, which are inherently difficult to solve even in simplified cases. The requirement of the additional time-dimension for the purposes of this work made the solution on any sort of useful geometry impossible beyond crude approximations. It was for this reason that the continuum approach was eventually abandoned. Two techniques used in this approach are described below.

2.1 The Perturbation Method

The perturbation method of solution of a differential equation involves identifying the dominant terms of the equation, eliminating the non-dominant terms, and solving the resulting simplified equation [14]. This creates a baseline solution to which a number of correction terms are added depending on the level of accuracy desired. The first correction term is found by substituting the baseline solution back into the original equation, and then solving a new equation created by the next-most dominant terms, and subsequent correction terms are found by repeating this procedure. The following section gives an example of this method being applied to the particle flow problem.

2.1.1 Application of the Perturbation Method to Simplified Navier-Stokes Equations

For continuum particle flow, the Navier-Stokes equations are simplified as follows: viscous and pressure forces are represented by a drag force term, and all body forces are neglected. For small particles, Stokes' Law can be used to model drag force as a constant coefficient of drag multiplied by the difference between the fluid and particle velocities. The PDE is thus

$$\rho_c \left(\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = K_D(\vec{w} - \vec{v}) \tag{2.1.1}$$

where ρ_c is particle mass concentration, $\vec{v}$ is particle velocity, K_D is the coefficient of drag, and $\vec{w}$ is the fluid velocity. The particle mass concentration is calculated as the multiple of the number of particles per unit volume multiplied by the mass of a single particle.

For this example, the case of steady stagnation in-plane flow is used, which is a simplified form of the impinging jet in two dimensions. Near the stagnation point, the fluid velocity in the x- and y-directions can be approximated as $w_x = cx$ and $w_y = -cy$ respectively, where c is some constant. The PDE becomes

$$\begin{aligned} \rho_c \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \right) &= K_D(w_x - u) \\ \rho_c \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} \right) &= K_D(w_y - v) \end{aligned}$$

Here the PDE has been separated into its x- and y-components, with u being the particle x-velocity and v the y-velocity. Now, to begin using the perturbation method, the dominant terms of the equations must be identified. The LHS of the equation represents the inertia of the particles; this term is dominant for relatively heavy particles. The RHS represents the drag force; this term is dominant for relatively light particles. Heavy particles are assumed for this example; thus, the LHS is dominant and the coefficient of drag is assumed to

be relatively small. The following forms are assumed for the particle velocity solutions:

$$\begin{aligned} u &= \Phi_0 + K_D\Phi_1 + K_D^2\Phi_2 + ... \\ v &= \Psi_0 + K_D\Psi_1 + K_D^2\Psi_2 + ... \end{aligned}$$

The zeroth order equation is created by substituting these forms into the PDEs and gathering all of the terms of order $O(K_D)^0$. The zeroth order system is

$$\begin{aligned} \Phi_0\frac{\partial\Phi_0}{\partial x} + \Psi_0\frac{\partial\Phi_0}{\partial y} = 0 \\ \Phi_0\frac{\partial\Psi_0}{\partial x} + \Psi_0\frac{\partial\Psi_0}{\partial y} = 0 \end{aligned}$$

The solution to this system is

$$\begin{aligned} \Phi_0 &= c_1 \\ \Psi_0 &= c_2 \end{aligned}$$

where c_1 and c_2 are constants. The x-velocity boundary condition at the particle jet outlet is $u = 0$, and the y-velocity condition is $v = -V = -cY$, where $y = Y$ is the location of the jet outlet (thus $c = V/Y$). These conditions must be satisfied by the zeroth order components of the velocity solutions, thus

$$\begin{aligned} \Phi_0 &= 0 \\ \Psi_0 &= -V \end{aligned}$$

To solve for particle concentration, the conservation of momentum equations must be supplemented by the continuity equation:

$$\rho_c\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + u\frac{\partial\rho_c}{\partial x} + v\frac{\partial\rho_c}{\partial y} = 0 \qquad (2.1.2)$$

Assuming a similar form for the concentration function as for the velocity functions and applying the same procedure, the zeroth order concentration solution is

$$\eta_0 = P$$

where P is a constant representing the particle concentration at the jet outlet. A constant was chosen for simplicity, but a function of x would also have satisfied the PDE. Next, the PDE terms of order $O(K_D)^1$ are gathered; after substituting in the zeroth order solutions, they are

$$\begin{aligned} -VP\frac{\partial \Phi_1}{\partial y} &= \frac{V}{Y}x \\ -YP\frac{\partial \Psi_1}{\partial y} &= -y + Y \\ P\left(\frac{\partial \Phi_1}{\partial x} + \frac{\partial \Psi_1}{\partial y}\right) - V\frac{\partial \eta_1}{\partial y} &= 0 \end{aligned}$$

The solution of these equations is

$$\begin{aligned} \Phi_1 &= \frac{x}{2P}\left(1 - \frac{y}{Y}\right) \\ \Psi_1 &= \frac{y}{P} - \left(\frac{y^2 + Y^2}{2YP}\right) \\ \eta_1 &= \frac{3Y}{4V}\left[\frac{2y}{Y}\left(1 - \frac{y}{2Y}\right) - 1\right] \end{aligned}$$

Further correction terms could be calculated, but they are unnecessary since all terms in the original PDE have now been accounted for. Figure 2.1 shows a sample plot of the particle paths given by $\Phi_0 + K_D\Phi_1$ and $\Psi_0 + K_D\Psi_1$ overlaid on streamlines of the fluid flow. The solution is physically reasonable; the paths begin with the inlet velocity and deviate somewhat due to fluid drag. The amount of deviation depends on K_D and P. The concentration solution generated is somewhat less useful since it depends on y alone; a solution that also depends on x could have been generated by defining $\eta_0 = P(x)$ instead of the constant P.

This solution shows the potential of the perturbation method; however, as mentioned above, the addition of the third dimension makes the transient versions of these PDEs too complex to solve using this method.

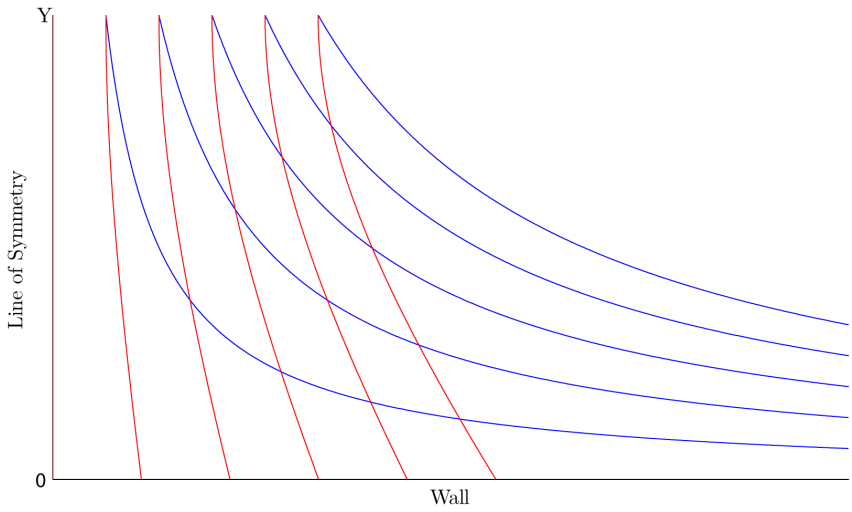

Figure 2.1: *Plot of the solution generated by the perturbation method for a steady two-dimensional case of impinging jet particle flow; fluid streamlines are shown in blue and particle paths are in red.*

2.2 The Variational Iteration Method

The variational iteration method (VIM) was originally proposed in [15]. To quote He, the method was created "to solve nonlinear partial differential equations without linearization or small perturbations." The method is similar to perturbation theory in that successive correction terms are applied to a baseline solution. Rather than being based on sub-dominant terms, the corrections are instead calculated using an integral formula that is constructed using a problem-specific Lagrange multiplier. The steps of this procedure are illustrated in detail using an example in the next section.

2.2.1 Application of the Variational Iteration Method

The following example from [15] illustrates the procedure of the VIM. The ODE to be solved is

$$\frac{\mathrm{d}y}{\mathrm{d}t} + y^2 = 0, \quad y \geq 0, \quad y(0) = 1$$

The VIM correction equation for this ODE is written as

$$y_{n+1}(t) = y_n(t) + \int_0^t \lambda_n \left[\frac{\mathrm{d}y_n(\tau)}{\mathrm{d}\tau} + {y_n}^2(\tau) \right] \mathrm{d}\tau \tag{2.2.1}$$

where y_n is the old solution approximation, y_{n+1} is the new solution approximation, λ_n is a general Lagrange multiplier, and the independent variable t has been replaced with the integration variable τ in the integral on the RHS. The selection of an appropriate Lagrange multiplier, which may change at each iteration, is the key to the successful application of the VIM. To determine λ_n, first make the following substitution in the correction equation:

$$y_n(t) = y_e(t) + \delta y(t)$$

Here, y_e is the exact solution of the ODE and δy is the correction function that must be applied to the current approximation to obtain the exact solution. The correction function is chosen such that it is stationary at $t = 0$, meaning that $\delta y(0) = 0$. With this substitution Equation 2.2.1 becomes

$$y_{n+1}(t) = y_e(t) + \delta y(t) + \int_0^t \lambda_n \left[\frac{\mathrm{d}\delta y(\tau)}{\mathrm{d}\tau} + 2y_e(\tau)\delta y \right] \mathrm{d}\tau \tag{2.2.2}$$

where the definition $y_e' + {y_e}^2 = 0$ has been used and the δy^2 term neglected under the assumption that the correction function is small. Now, the goal of the correction is to have $y_{n+1} = y_e$; consequently, λ_n should be chosen to minimized all of the terms of the RHS of Equation 2.2.2 after the first. It is not immediately obvious what form λ_n needs to take, since the character of δy and its derivative is unknown. To remove one of these unknowns, integration

by parts is applied to the first term in the integral of Equation 2.2.2, yielding

$$y_{n+1}(t) = y_e(t) + \delta y(t)\left[1 + \lambda_n(\tau)\Big|_{\tau=t}\right] + \int_0^t \delta y(\tau)\left[2y_e(\tau)\lambda_n(\tau) - \frac{\mathrm{d}\lambda_n(\tau)}{\mathrm{d}\tau}\right]\mathrm{d}\tau$$

From this equation, the optimum form of λ_n is apparent. It must satisfy the ODE

$$\begin{aligned} \frac{\mathrm{d}\lambda_n}{\mathrm{d}\tau} - 2y_e(\tau)\lambda_n(\tau) &= 0 \\ \lambda_n(t) &= -1 \end{aligned} \tag{2.2.3}$$

Since y_e is unknown, it is replaced with y_0. The solution of Equation 2.2.3 is

$$\lambda_n(t,\tau) = -e^{2\int_t^\tau y_0(\eta)\mathrm{d}\eta} \tag{2.2.4}$$

With the Lagrange multiplier known, only the initial solution approximation y_0 is required to solve the correction equation. Obviously the initial guess should be as close to the exact solution as possible under the circumstances, but the only restriction placed on it by the VIM is that it must satisfy the boundary condition(s) of the original ODE. This is a result of the correction function δy being defined as stationary at $t = 0$. In this case the BC is $y(0) = 1$. A very simple first guess of $y_0 = 1$ is chosen first; if this is unsuccessful, a new function can be tried. This makes the first Lagrange multiplier

$$\lambda_0 = -e^{2\int_t^\tau y_0(\eta)\mathrm{d}\eta} = -e^{2(\tau - t)}$$

Substituting λ_0 and y_0 into Equation 2.2.1 and solving gives

$$y_1 = \frac{1}{2}\left(1 + e^{-2t}\right)$$

Repeating the correction integral with y_1 in place of y_0 gives

$$y_2 = \frac{1}{2}\left(1 + e^{-2t}\right) + \frac{1}{2}te^{-2t} - \frac{1}{8}\left(1 - e^{-4t}\right)$$

The correction can theoretically be applied as many times as desired, and the accuracy of the approximation should continue to improve if the Lagrange

multiplier was chosen correctly; however, the integrals will quickly become quite complicated. Since the exact solution to the example ODE is known, it can be compared against the approximations produced by the VIM. This comparison is shown in Figure 2.2. After just two iterations, the approximation is very accurate in the neighbourhood of $t = 0$, which shows the potential of this method.

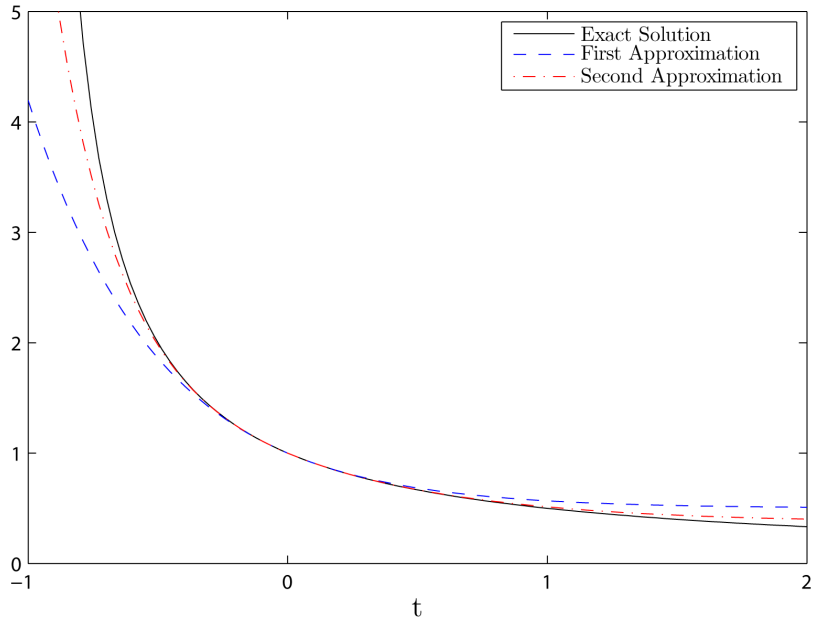

Figure 2.2: *The exact solution of the VIM test case ODE is compared to two levels of the approximate solution.*

After trying several applications to some simplified Navier-Stokes equations, it was determined that the VIM is best suited to equations of only one dependent variable. With two spatial dimensions plus time, three equations must be corrected. This raises issues of which approximation is corrected first, whether new corrections are included in the correction of other equations, and so on. The end result is a huge number of possible combinations of Lagrange multipliers, 0^{th} order function approximations, and solution orders. This leads to a method that is more about subjectivity and trial-and-error instead of the

strict rigorousness that is desired. As a result, this method was abandoned for use in the solution of the Navier-Stokes equations. The VIM is recommended as a potential technique for transforming a steady-state solution of a particle velocity or density field into a transient solution, which only requires the correction of one variable.

3

A Working Solution: The Single-Particle Approach

It is simpler to solve the equations of motion of a single particle and extend the solution to a collection of particles than to solve the equations for all of the particles at once. Although this approach is Lagrangian in nature, the resulting particle path functions can be adapted to an Eulerian framework and used to create velocity and concentration field functions.

In order to solve single particle motion, a soluble equation must be formulated which relates the motion of the particle to the forces acting on it. For this solution attempt, an equation based on Newton's second law of motion was used. The only other pieces of information required are expressions for the

mass of and forces acting on the particle.

3.1 Formulating The Equation

Calculating the actual exact mass of a given aerosol particle is difficult considering the irregularity and variability of the shape of such particles. However, most aerosol particles can be sufficiently accurately approximated as spherical, which makes the mass calculation trivial. This approximation also serves to greatly simplify the calculation of the drag force. If, however, the spherical approximation is not sufficiently accurate, a dynamic shape factor can be used to scale the drag force. The dynamic shape factor describes the effect the true particle shape has on the drag force as compared to a perfect sphere. This correction can be applied at any time, including at the final solution; simply divide the Stokes number (seen later) by the appropriate shape factor. The Cunningham slip correction [16] can also be applied in this way.

The standard drag force formula, in which drag is proportional to the square of the velocity difference, could be used, but the squared velocity term leads to a highly nonlinear equation which is quite difficult to solve. Instead, the small size of the particles of interest in this case allowed the use of Stokes' law for calculating drag [17],

$$\vec{F}_d = 6\pi\mu\frac{d}{2}(\vec{w} - \vec{u})$$

where $\vec{F}_d$ is the drag force, μ is dynamic viscosity of the fluid, d is the particle diameter, $\vec{w}$ is the fluid velocity, and $\vec{u}$ is the particle velocity. This expression for drag is typically valid for $Re_p \ll 1$, where $Re_p = d(\vec{w} - \vec{u})/\nu$ is the particle Reynolds number and ν the kinematic viscosity of the fluid. Typical values of Re_p for the cases considered here range from ~ 0.01 to ~ 0.1. Combining Stokes' law and the formula for the mass of a sphere with the equation of

motion gives

$$\begin{aligned}\frac{\mathrm{d}^2\vec{x}}{\mathrm{d}t^2} &= \frac{\vec{F}}{m} \\ &= \frac{6\pi\mu\frac{d}{2}}{\frac{4}{3}\pi\left(\frac{d}{2}\right)^3\rho}(\vec{w}-\vec{u}) \\ &= \frac{18\mu}{d^2\rho}\left(\vec{w}-\frac{\mathrm{d}\vec{x}}{\mathrm{d}t}\right) \end{aligned} \tag{3.1.1}$$

where $\vec{x}$ is the particle position vector and ρ is the particle density. Now, because the equation is linear, it can be split into Cartesian components:

$$\begin{aligned}\ddot{x} &= \frac{18\mu}{d^2\rho}(w_x-\dot{x}) \\ \ddot{y} &= \frac{18\mu}{d^2\rho}(w_y-\dot{y})\end{aligned}$$

Here a dot above a variable represents differentiation with respect to time.

3.1.1 Non-dimensionalization

To aid in manipulation and interpretation, the particle equations were non-dimensionalized. The following quantities were defined:

$$\begin{aligned}x^* &= \frac{x}{R} \\ y^* &= \frac{y}{R} \\ t^* &= \frac{W}{R}(t-t_0) \\ S &= \frac{\rho d^2 W}{18\mu R}\end{aligned} \tag{3.1.2}$$

where x^*, y^*, and t^* are the dimensionless independent variables and R and W are a (constant) representative length and speed of the particular fluid flow field being considered, and S is the Stokes number of the flow. The Stokes number represents the ratio of the particle stopping distance to a characteristic

length of the flow. The non-dimensionalized equations are

$$\begin{aligned} \ddot{x}^* &= \frac{1}{S}\left({w_x}^* - \dot{x}^*\right) \\ \ddot{y}^* &= \frac{1}{S}\left({w_y}^* - \dot{y}^*\right) \end{aligned}$$

For the remainder of the thesis, all variables appearing in equations will be dimensionless unless otherwise specified, so the asterisk next to the symbols will be dropped for the sake of clarity.

3.2 Solving the Equation

With the particle trajectory equation defined and written in a useful form, the only remaining step before solution was to choose a fluid velocity function. The geometry under consideration, in this case a curved pipe, dictates what shape the fluid flow must take. Thus, two circular flow functions were chosen: one with constant θ-velocity at all points, and one with θ-velocity increasing linearly in the radial direction. The former was selected as a simple first case and the latter because it is a better representation of the actual flow in a curved pipe. Of course, neither function includes details such as boundary layer effects or secondary flow, which were assumed to have little effect on the bulk particle flow. If necessary, correction terms can potentially be added to the solution *a posteriori* to account for the neglected phenomena. The pipe geometry and flow field for each of the two cases is shown in Figure 3.1. The characteristic length R chosen for this geometry is the pipe bend radius; also defined are the pipe cross-section radius a and the ratio $R_p = R/a$, hereafter referred to as the "pipe ratio".

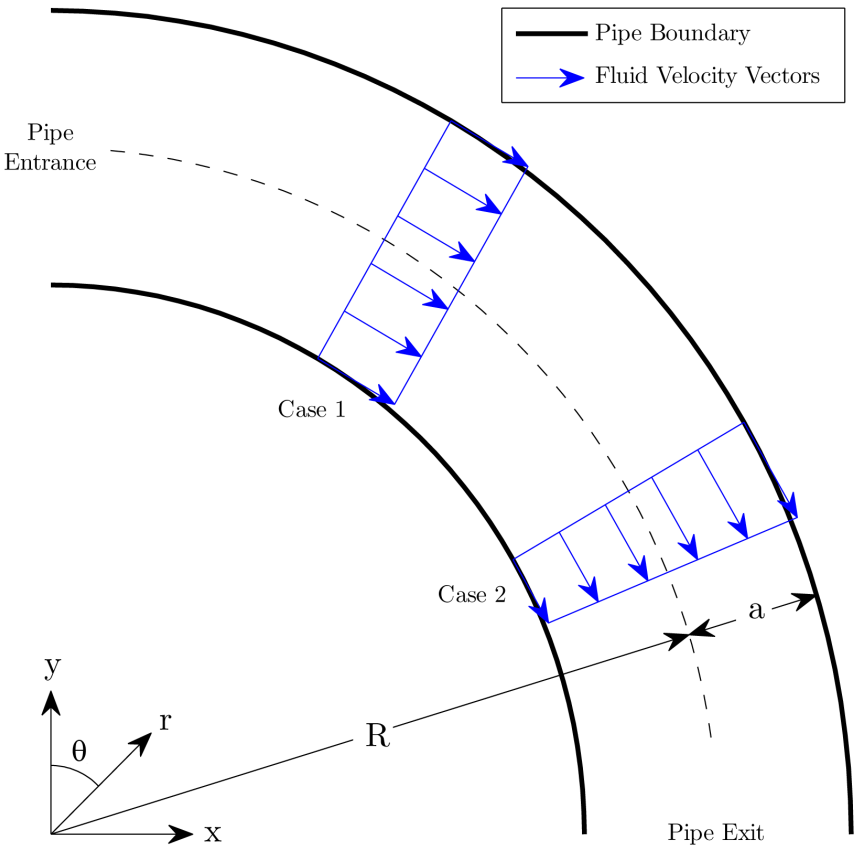

Figure 3.1: *The curved pipe geometry used in the solution of the single particle trajectory ODE; also shown shown are the two fluid velocity profiles used: constant (case 1), and linear (case 2).*

3.2.1 The Radially Constant Flow Case

For the first case, the fluid velocity field is circular with magnitude independent of radial distance from the origin; the origin is located at the center of curvature of the pipe. Polar coordinates were used in this case to simplify the flow field equations. Keeping in mind that the variable quantities are dimensionless, the

equations are

$$\ddot{r} - r\dot{\theta}^2 + \frac{1}{S}\dot{r} = 0$$
$$r\ddot{\theta} + 2\dot{r}\dot{\theta} + \frac{1}{S}\left(r\dot{\theta} - f(t)\right) = 0$$

where $f(t)$ is a time-dependent function describing the magnitude of the θ-component of the fluid velocity. A solution to this system of highly nonlinear ODEs was not readily apparent, so it was instead solved numerically in order to provide some insight into its behaviour. A plot of the equilibrium solution revealed that it had the shape of an Archimedean spiral, which has consecutive loops equally spaced apart. The spiral follows the relation $r = C_1 + C_2\theta$, where C_1 and C_2 are constants. Substituting this relation into the ODEs allowed them to be combined into one equation:

$$\left(2 + \frac{1}{{C_2}^2}r^2\right)\dot{r}^2 - \frac{1}{S}C_2 f(t) = 0$$

A constant fluid velocity function ($f(t) = 1$) was used for the first solution attempt along with the boundary conditions $r(0) = 1$ and $r'(0) = 0$. The result, provided by the symbolic toolbox of *Matlab*, is

$${C_2}^2\sinh^{-1}\left(\frac{r}{\sqrt{2}C_2}\right) + \frac{r}{2}\sqrt{2{C_2}^2 + r^2} = {C_2}^2\sinh^{-1}\left(\frac{1}{\sqrt{2}C_2}\right) + \frac{r}{2}\sqrt{2{C_2}^2 + 1} + t\sqrt{\frac{1}{S}{C_2}^3}$$

For this solution to be useful, the constant C_2 must be determined. This constant determines the distance between consecutive loops of the spiral, which is equal to $2\pi C_2$. From the numerical solution, it was determined that $C_2 = S$. The equation becomes

$$\sinh^{-1}\left(\frac{r}{\sqrt{2}S}\right) + \frac{r}{\sqrt{2}S}\sqrt{1 + \left(\frac{r}{\sqrt{2}S}\right)^2} = \sinh^{-1}\left(\frac{1}{\sqrt{2}S}\right) + \frac{r}{\sqrt{2}S}\sqrt{1 + \left(\frac{1}{\sqrt{2}S}\right)^2} + \frac{t}{S}$$

This equation cannot be solved explicitly for r, limiting its utility. Further analysis revealed that, over the relevant range of S, the first term on the LHS is much smaller than the second term and can be neglected. The corresponding term on the RHS is also neglected to preserve the boundary condition. With this simplification the equation can be solved for r, but the result is quite

long and complicated. The equation can be further simplified by noting that $S << 1$ in most cases and thus the square root terms can be approximated by

$$\sqrt{1+\left(\frac{r}{\sqrt{2}S}\right)^2} \approx \frac{1}{2}+\frac{r}{\sqrt{2}S}$$

making the final equations

$$\begin{aligned} r &= \frac{1}{2}+\frac{1}{2}\sqrt{1+8St} \\ \theta &= \frac{-\frac{1}{2}+\frac{1}{2}\sqrt{1+8St}}{S} \end{aligned}$$

This result shows the basic form of the trajectory of a particle at equilibrium moving through a radially constant circular flow field. It is not practically useful, however, since it only applies to a steady fluid flow and it cannot account for different particle initial velocities. Rather than proceed with a more complex transient case using this type of flow field, a second flow field function was introduced which more accurately represents the actual flow profile inside a curved pipe.

3.2.2 The Radially Variable Flow Case

For the second case, the fluid flow field is circular, has a value of zero at the origin, and increases linearly with distance from the origin. This field is a better representation of the real flow in a curved pipe which slows down slightly toward the inside edge and speeds up slightly toward the outside edge [18]. In polar coordinates, the radial velocity of this flow is zero and the angular component is given by $v_\theta = rf(t)$ where $f(t)$ is once again some dimensionless function of time. This velocity field is simply represented in Cartesian coordinates by $w_x = yf(t)$ and $w_y = -xf(t)$. The ODEs for this case are also more simply written in Cartesian coordinates:

$$\begin{aligned} \ddot{x}+\frac{1}{S}\left(\dot{x}-yf(t)\right) &= 0 \\ \ddot{y}+\frac{1}{S}\left(\dot{y}+xf(t)\right) &= 0 \end{aligned}$$

Since a steady-state solution was already found for the previous case, a fully transient solution was attempted straight away for the current case. The chosen $f(t)$ must be periodic to be an accurate representation of a real breathing pattern. A sine wave has been shown to be a good approximation of inhalation through a mechanical breathing device [19], while regular tidal breathing patterns approximately vary between sine, triangle, and square waves [3]. The function used, defined first in dimensioned terms and then non-dimensionalized, was

$$\begin{aligned} f(t_{dim}) &= \sin\left[\frac{2\pi}{T}t_{dim}\right] \\ \Rightarrow \quad f(t) &= \sin\left[2\pi\left(\mathrm{Str}\, t + t_0\right)\right] \end{aligned}$$

where t_{dim} is the dimensioned time, T is the period of the flow,

$$\mathrm{Str} = \frac{R}{WT}$$

is the Strouhal number of the flow, and

$$t_0 = \frac{t_{0,dim}}{T}$$

is a dimensionless time representing the point in the flow cycle at which the particle entered the pipe. Note that this time is not equivalent to the value of the dimensionless time variable at which the particle entered the pipe; in fact, that value is always zero because of the definition used for the dimensionless time. It is more helpful to view the dimensionless time as a measure of how long the particle has been in the pipe rather than an indication of the absolute time as seen by the fluid flow function.

Unfortunately, a solution was not found using this fluid function. To simplify the equation, the sine wave function was replaced by its first order Taylor series approximation:

$$\begin{aligned} f(t) &\approx f(0) + f'(0)t \\ &= \sin\left[2\pi t_0\right] + 2\pi\mathrm{Str}\cos\left[2\pi t_0\right] t \end{aligned}$$

Or, in general, $f(t) = A + Bt$, where A and B are constants. This approximation is accurate for most particles due to their short residence time in the pipe. With this form for the fluid time-function, a solution was found for the system of ODEs. To accomplish this, the complex number $s = x + iy$ was defined, which allowed the two equations to be combined into one:

$$\ddot{s} + \frac{1}{S}\left(\dot{s} + isf(t)\right) = 0$$

The solution to this equation (equation A.1.1 in Appendix A.1) is given by *Mathematica* as

$$\begin{aligned} s(t) \;=\; e^{-\frac{t}{2S}} \Bigg[& \frac{\left[(s_0 + 2SV_0)\mathrm{Bi}(C) - 2k^{1/3}Ss_0\mathrm{Bi}'(C)\right]\mathrm{Ai}(D)}{2k^{1/3}S[\mathrm{Ai}'(C)\mathrm{Bi}(C) - \mathrm{Ai}(C)\mathrm{Bi}'(C)]} \\ & - \frac{\left[(s_0 + 2SV_0)\mathrm{Ai}(C) - 2k^{1/3}Ss_0\mathrm{Ai}'(C)\right]\mathrm{Bi}(D)}{2k^{1/3}S[\mathrm{Ai}'(C)\mathrm{Bi}(C) - \mathrm{Ai}(C)\mathrm{Bi}'(C)]} \Bigg] \end{aligned} \tag{3.2.1}$$

where

$$\begin{aligned} C &= \frac{1 - 4iSA}{4k^{2/3}S^2} \\ D &= C + k^{1/3}t \\ k &= \frac{-iB}{S} \end{aligned}$$

and $s_0 = x_0 + iy_0$ and $V_0 = u_0 + iv_0$ are the complex initial conditions. Ai, Bi, Ai$'$, and Bi$'$ are the Airy functions and their derivatives. The real functions for the Cartesian coordinates are recovered from this complex solution by taking $x(t) = \mathrm{Re}[s(t)]$ and $y(t) = \mathrm{Im}[s(t)]$. The full explanation of this procedure is

provided in Appendix A.1. The solution is

$$
\begin{aligned}
&x(t) = \\
&\frac{e^{\frac{-t}{2S}}}{2(1+\alpha^2)^{9/8}(1+\gamma^2)^{1/8}}\Bigg\{\cosh f_1\Bigg[2(1+\alpha^2)^{1/4}[(g_1h_1+g_2h_2)x_0+(g_2h_1-g_1h_2)y_0]\cos f_2 \\
&+\Big[g_2\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)-g_1\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\sin f_2\Bigg] \\
&+\sinh f_1\Bigg[2(1+\alpha^2)^{1/4}[(g_2h_1-g_1h_2)x_0-(g_1h_1+g_2h_2)y_0]\sin f_2 \\
&+\Big[g_1\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)+g_2\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\cos f_2\Bigg] \\
&-\frac{2qB^{2/3}S^{4/3}}{(1+\alpha^2)}\Bigg[\Big((\sqrt{3}\alpha-1)g_1-(\sqrt{3}+\alpha)g_2\Big)\cosh f_1\sin f_2 \\
&-\Big((\sqrt{3}\alpha-1)g_2+(\sqrt{3}+\alpha)g_1\Big)\sinh f_1\cos f_2\Bigg]\Bigg\} \\
&y(t) = \\
&\frac{e^{\frac{-t}{2S}}}{2(1+\alpha^2)^{9/8}(1+\gamma^2)^{1/8}}\Bigg\{-\cosh f_1\Bigg[2(1+\alpha^2)^{1/4}[(g_2h_1-g_1h_2)x_0-(g_1h_1+g_2h_2)y_0]\cos f_2 \\
&-\Big[g_1\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)+g_2\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\sin f_2\Bigg] \\
&-\sinh f_1\Bigg[-2(1+\alpha^2)^{1/4}[(g_1h_1+g_2h_2)x_0+(g_2h_1-g_1h_2)y_0]\sin f_2 \\
&+\Big[g_2\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)-g_1\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\cos f_2\Bigg] \\
&+\frac{2qB^{2/3}S^{4/3}}{(1+\alpha^2)}\Bigg[\Big((\sqrt{3}\alpha-1)g_2+(\sqrt{3}+\alpha)g_1\Big)\cosh f_1\sin f_2 \\
&+\Big((\sqrt{3}\alpha-1)g_1-(\sqrt{3}+\alpha)g_2\Big)\sinh f_1\cos f_2\Bigg]\Bigg\}
\end{aligned}
$$

where

$$
\begin{aligned}
f_1 &= \frac{\gamma^2+2\gamma-1+(\gamma+1)\sqrt{1+\gamma^2}}{6\beta S\sqrt{\sqrt{1+\gamma^2}+\gamma}} - \frac{\alpha^2+2\alpha-1+(\alpha+1)\sqrt{1+\alpha^2}}{6\beta S\sqrt{\sqrt{1+\alpha^2}+\alpha}} \\
f_2 &= \frac{-\gamma^2+2\gamma+1+(1-\gamma)\sqrt{1+\gamma^2}}{6\beta S\sqrt{\sqrt{1+\gamma^2}+\gamma}} - \frac{-\alpha^2+2\alpha+1+(1-\alpha)\sqrt{1+\alpha^2}}{6\beta S\sqrt{\sqrt{1+\alpha^2}+\alpha}} \\
g_1 &= \sqrt{\frac{1}{2}+\frac{1}{2\sqrt{2}}\sqrt{1+\frac{-1+\sqrt{3}(\alpha+\gamma)+\alpha\gamma}{2\sqrt{(1+\alpha^2)(1+\gamma^2)}}}} \\
g_2 &= \frac{\sqrt{3}(1-\alpha\gamma)+\alpha+\gamma}{16\sqrt{(1+\alpha^2)(1+\gamma^2)}g_1(g_1^2-0.5)} \\
h_1 &= \sqrt{\frac{1}{2}+\frac{1+\sqrt{3}\alpha}{4\sqrt{1+\alpha^2}}} \\
h_2 &= \sqrt{\frac{1}{2}-\frac{1+\sqrt{3}\alpha}{4\sqrt{1+\alpha^2}}} \\
\alpha &= 4AS \\
\beta &= 4BS \\
\gamma &= \alpha+\beta t \\
q &= \mathrm{Sign}(B)
\end{aligned}
$$

The next chapter describes how this solution was numerically coded for use in particle deposition simulations.

4

Numerical Implementation

This chapter discusses the numeric implementation of the analytic particle path solution from the previous chapter. Three types of numeric simulation were coded: steady-state, which has a constant fluid flow function; quasi-steady, which also uses a constant fluid function but updates the magnitude every time step; and transient, which uses a fully time-dependent fluid function and calculates instantaneous deposition rate at any point in the pipe. Each simulation is explained in detail in this chapter.

The basic principle behind particle deposition is the same for all simulations. The particle domain is treated as a curved cylinder overlaying the pipe; any part of the cylinder that is outside the limits of the pipe represents particles that have deposited. The pipe is fixed in space, and the beginning of the par-

ticle cylinder is aligned with the pipe inlet. Further into the pipe, the cylinder diverges and warps according to the motion of the particles. Though only inertia and drag are modelled here, this framework allows for the inclusion of other flow phenomena; for example, diffusion could be modelled by a radial expansion of the particle cylinder based on an appropriate mathematical equation.

4.1 Steady Flow Simulation

A steady-state version of the deposition simulation was created despite the fact that the particle path solution is fully transient. It was created primarily for validation purposes; other analytic solutions and experimental results for the transient case are not available for comparison, whereas they are for the steady case. The steady-state simulation can also be used to approximate transient deposition results by choosing a fluid velocity magnitude that results in the same total inhaled volume as the integrated transient case. The accuracy of such approximations can vary wildly depending on the specific flow under consideration.

The steady case uses a constant fluid function, which is equivalent in the particle path function to values of $A = 1$ and $B = 0$. Setting B to zero creates indeterminacies in some terms of the particle function, so a new set of equations had to be solved for steady flow. This involved simply taking the limit as B goes to zero of each function containing it. The resulting steady particle path functions are given below. Note that "steady" here refers to the fluid function only; the particle functions still give particle position as a

function of time.

$$
\begin{aligned}
x(t) &= e^{-t/2S}\Bigg\{\cosh\tilde{f}_1\left[x_0\cos\tilde{f}_2+\frac{(2Su_0+x_0)h_2-(2Sv_0+y_0)h_1}{(1+\alpha^2)^{1/4}}\sin\tilde{f}_2\right] \\
&\qquad +\sinh\tilde{f}_1\left[-y_0\sin\tilde{f}_2+\frac{(2Su_0+x_0)h_1+(2Sv_0+y_0)h_2}{(1+\alpha^2)^{1/4}}\cos\tilde{f}_2\right]\Bigg\} \\
y(t) &= e^{-t/2S}\Bigg\{\cosh\tilde{f}_1\left[x_0\cos\tilde{f}_2+\frac{(2Su_0+x_0)h_2-(2Sv_0+y_0)h_1}{(1+\alpha^2)^{1/4}}\sin\tilde{f}_2\right] \\
&\qquad +\sinh\tilde{f}_1\left[-y_0\sin\tilde{f}_2+\frac{(2Su_0+x_0)h_1+(2Sv_0+y_0)h_2}{(1+\alpha^2)^{1/4}}\cos\tilde{f}_2\right]\Bigg\}
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{f}_1 &= \frac{1+2\alpha^3+\sqrt{1+\alpha^2}+\alpha(2+\sqrt{1+\alpha^2})+\alpha^2(1+2\sqrt{1+\alpha^2})}{4S\sqrt{1+\alpha^2}(\alpha+\sqrt{1+\alpha^2})^{3/2}}t \\
\tilde{f}_2 &= \frac{1-2\alpha^3-\sqrt{1+\alpha^2}+\alpha(-2+\sqrt{1+\alpha^2})+\alpha^2(1-2\sqrt{1+\alpha^2})}{4S\sqrt{1+\alpha^2}(\alpha+\sqrt{1+\alpha^2})^{3/2}}t
\end{aligned}
$$

and h_1, h_2 are the same as in the original solution.

This solution is implemented in the code *steadyDep.m* found in Appendix B.1. Calculating particle deposition is much simpler in the steady case than in the transient. All that is required is the deposited particle area fraction at the end of the pipe, which, multiplied by the particle inflow rate, gives the deposition rate in the pipe. The downside of this simple method is that only a total deposition rate is produced; however, the method can be applied to each CV within the pipe to calculate an approximate deposition rate per unit length of the pipe at any point along it. Figure 4.1 shows a schematic of the geometry used in the calculation of steady deposition for a typical case. As shown in Figure 4.1, a radial cross-section of the deposition appears as an ellipse (the particles) overlapping a circle (the pipe). The ellipse has a minor axis length equal to the circle radius a, and their edges are aligned in radial direction. The major axis length of the ellipse begins at a value of $b = a$ at the pipe inlet, where the particle cross-section is circular, and increases from there.

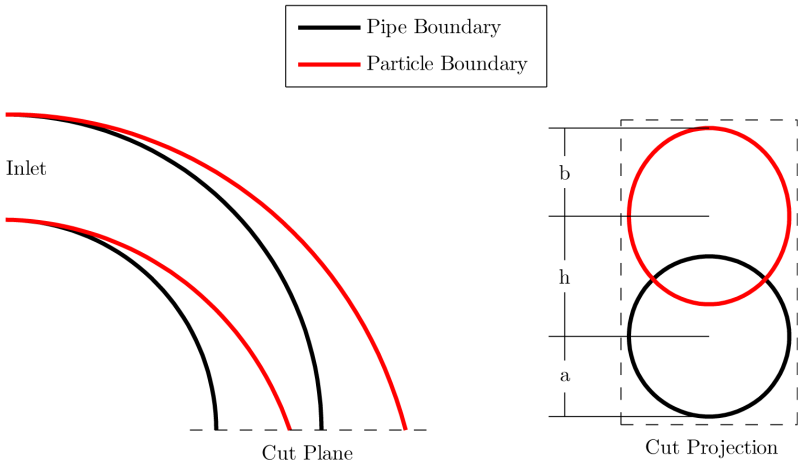

Figure 4.1: *On the left, the boundary of the bulk particle flow is compared to that of the pipe for a typical steady flow case; the cross-sectional projection of the pipe and particles in the radial plane at the pipe exit is shown on the right.*

The function used to calculate the deposited area is

$$A_{dep} = \pi ab + a\left(\frac{h\sqrt{(a+b)^2 - h^2}}{a+b} - (a+b)\text{asin}\left[\frac{\sqrt{(a+b)^2 - h^2}}{a+b}\right]\right)$$

The full derivation of this equation is found in A.3. The quantities b and h (h being the center-to-center distance of the circle and ellipse) are determined from the radially innermost and outermost points of the particle ellipse at the pipe exit. These points have known (dimensioned) initial positions of $y_0 = R - a$ and $y_0 = R + a$ respectively. The initial times of the points are solved iteratively using the bisection method with the steady path functions above, which then allows for the calculation of their radial positions at time of exit.

Once calculated, the area A_{dep} is divided by the ellipse area to get the fraction of particles which are deposited for a given flow case; this fraction is multiplied by the average inlet velocity and an arbitrary concentration to get deposition rate. The concentration can be specified in whatever units are desired, such as particles per unit volume or particle mass per unit volume. The concentration

decreases as the particles flow through the pipe due to stretching of the ellipse, but it does not affect the results because deposition is calculated as a fraction rather than an absolute amount.

4.2 Quasi-Steady Flow Simulation

There are ways to improve upon the accuracy of a steady-state simulation without actually simulating or solving the full complexity of a transient flow field. One such way is to use a quasi-steady simulation. This approach still uses the relatively simple steady-state fluid flow function, but updates its value over time to follow the magnitude of the desire transient fluid flow. This means that fluid acceleration constant remains at $B = 0$ throughout the simulation, while the magnitude constant A varies; for example,

$$A = \sin(2\pi t_0)$$

for a sine wave time profile and

$$A = 4\, t_0$$

for the acceleration portion of a triangle wave time profile ($0 < t_0 < 0.25$). The quasi-steady method accurately captures true fluid velocity; the only information lost is the fluid acceleration and the time-history of the particles. This method can greatly improve accuracy over the steady-state method while only slightly increasing the calculation complexity of the simulation.

The code for the quasi-steady simulation, *quasiDep.m*, is included in Appendix B.2. The quasi-steady deposition rate is calculated the same way as the steady deposition described above; but rather than calculate it only once, it is re-calculated at every time step of the simulation. Thus, the progression of time is tracked in this code, with t_i representing the simulation start time and t_f the end time. The instantaneous deposition rate for each time step is multiplied by

the length of the time step to get total particles deposited, and these quantities are summed, along with the particles entering the pipe at each time, to get total deposition and inflow. The former is divided by the latter to get the overall deposition fraction of the simulation.

An adaptive time step is used for the sake of efficiency, calculated by

$$dt = \frac{\pi R}{4N_{CV}} \frac{1}{WA(t)}$$

where the pipe has been divided into N_{CV} equal-length control volumes (CVs). The first term in this equation represents the arc-length of half of one CV along the central axis of the pipe, and the second term is the inverse of the instantaneous fluid velocity. The result of this calculation is that a particle at the center of the pipe moving at the fluid velocity (which is the limiting axial velocity for any particle at a particular location) will travel about halfway through a CV in any given time step. This is useful because very small time steps are needed to accurately model fast-moving fluid, but they drastically slow down the simulation for slow-moving fluids, and both speeds are present in simulations spanning a half- or full-breath cycle. The time step is also limited to a maximum value of 1% of the total simulation time to avoid extremely large time steps that arise for extremely small fluid velocities using the formula above.

The presence of CVs in the pipe is another departure from the steady flow case. They are used to calculate the time step size as mentioned previously, but more importantly, they are used to track the distance into the pipe that the particles have penetrated. This is important for cases which start with an empty pipe at $t_i = 0$. For these cases, the transient particle path function is used every time step to determine the deepest CV which has been fully infiltrated by the particles; deposition is then calculated as before except that the overlap area at the end of this CV, rather than at the pipe exit, is used. Once the entire pipe has been infiltrated, deposition calculations proceed as

normal without this additional calculation. Consequently, varying N_{CV} will affect the accuracy and computational intensity of the quasi-steady simulation, though not nearly so much as it does the transient.

4.3 Transient Flow Simulation

The basic structure of the transient deposition code *eulerDep.m*, found in Appendix B.3, is the same as the quasi-steady code. Time advances from t_i to t_f using the same adaptive time step, and deposition is calculated at each step. Pipe infiltration is also determined the same way. Beyond this, the similarity ends. Deposition must be calculated separately for every infiltrated CV at each time step, and the procedure for doing so is more complicated as a result of the time-dependent particle history. Concentration changes must also be taken into account; transient flow means that axial compression or expansion of the particle field is possible. The procedures for calculating concentration and deposition are explained in the following sections, as well as that for calculating outflow.

Before the flux quantities of interest can be calculated, the values on which they depend must be solved for. First, the minimum and maximum radial position values of the particles at each CV boundary must be calculated. With these positions the shape of the particle cylinder is fully described on the discretized domain, because the particle cross-section always remains an ellipse. Since initial position and current θ-position are known for these limiting particles, initial (injection) time and current radial position remain to be solved for. Unfortunately, the complexity of the particle path solutions precludes their inversion, which would allow for direct solution of the desired variables. Instead, they must be solved numerically; this is done using the bisection method to find the t_0 value which gives the desired value of θ to an acceptable degree of accuracy. Once the initial times for all limiting particles are known, they are

fed back into the path equations to solve for current radial position.

Once the upper and lower limiting points are known at each CV boundary, they are used to calculate the ellipse parameters b and h as well as other required geometric quantities. They are also used to calculate particle radial velocity. This is done using a simple first order approximation by adding a small time increment (*i.e.* $1/100^{th}$ of the current time step) to the current time, calculating the change in radial position for each particle, and dividing this change by the time increment. With all of these prerequisite values solved for, the simulation moves on to calculation of the deposition solution.

4.3.1 Concentration

In contrast to the previous two simulation types, transient deposition is calculated as an absolute quantity rather than a fraction. Consequently, the transient simulation must account for the particle concentration change in the radial direction which results from the changing shape of the particle domain cross-section. Furthermore, as mentioned at the beginning of this section, transient flow can also lead to density changes in the axial direction. For these reasons, a method of calculating particle concentration C at a given point in time and space is needed.

The method chosen to calculate concentration at a given point and time requires the initial conditions of the particle occupying that point. If the particle in question is not one of the limiting particles already solved for, its initial conditions can be interpolated from its nearest neighbours. From the initial conditions, a square area element is constructed at the particle's initial location and time. This requires three pairs of coordinates: first, the initial conditions themselves, (t_0, y_0); second, a point offset in space, $(t_0, y_0 + s_0)$; and third, a point offset in time, $(t_0 - dt_0, y_0)$. The element side length s_0 must be small relative to the pipe radius, and the offset time $dt_0 = s_0/u_0(t_0, y_0)$ is chosen so

that both sides of the area element are equal.

The three pairs of initial conditions forming the area element are input into the path functions to determine their new relative positions at the current simulation time. The area of the new stretched element is calculated using the formula

$$A_{new} = s_1 s_2 \sin\phi$$

where s_1 and s_2 are the lengths of the element sides and ϕ is the angle between them. Relative concentration is then simply the ratio of the original element area to the current element area. This procedure is shown graphically in Figure 4.2. Absolute concentration is found by multiplying the relative value by the absolute value at the pipe inlet defined in the desired units.

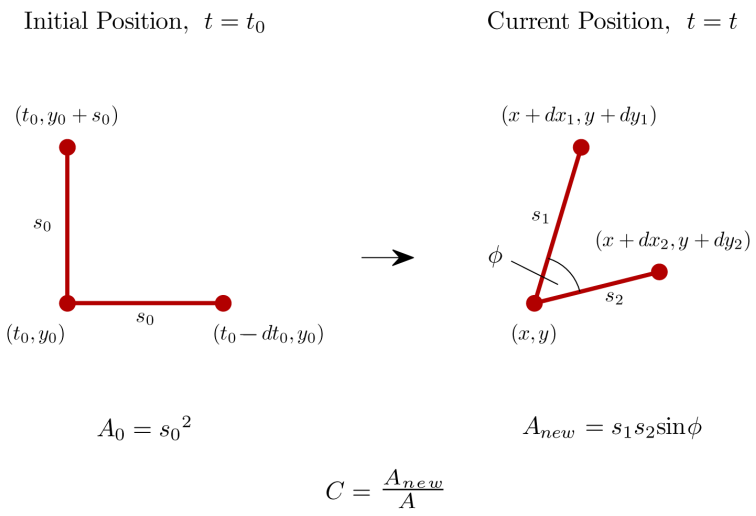

Figure 4.2: *The concentration C at a given point (x, y, t) is calculated by measuring the relative size change of the transformed area element at that point; note that the coordinates on the left refer to initial conditions (t_0, y_0) while those on the right refer to current spatial position (x, y).*

4.3.2 Deposition

A deposition rate function was determined for a given CV by integrating the particle mass flux over the surface of intersection between the pipe and the particle cylinder. The mass flux is simply the multiple of the particle volume flow rate and the relative concentration from the previous section. Figure 4.3 shows a three-dimensional visualization of the surface of intersection for a CV spanning (θ_1, θ_2). The end result of the integration process, which is shown in detail in Appendix A.5, is the following function for deposition rate $\dot{Q}$:

$$\begin{aligned}\dot{Q} \;=\; & \frac{a}{12k_1}\Big[3[3a^2c_1+4R(c_1R+c_2)](k_1\theta+k_2)\sqrt{1-(k_1\theta+k_2)^2} \\ & +2a\sqrt{1-(k_1\theta+k_2)^2}\big[(2c_1R+c_2)[-8+2(k_1\theta+k_2)^2] \\ & -ac_1[(k_1\theta+k_2)+(k_1\theta+k_2)^3]\big] \\ & +12a(2c_1R+c_2)(k_1\theta+k_2)\mathrm{acos}(k_1\theta+k_2) \\ & +3[3a^2c_1+4R(c_1R+c_2)]\mathrm{asin}(k_1\theta+k_2)\Big]_{\theta_1}^{\theta_2}\end{aligned}$$

The angles θ_1 and θ_2 locate the boundaries of the CV. The constants c_1 and c_2 define the radial variation of the particle radial velocity in the CV, and the constants k_1 and k_2 define the variation of the ellipse variables b and h with θ.

4.3.3 Outflow

Particle outflow is not an essential part of the transient simulation as it is unnecessary for the calculation of deposition; however, it was still included so that it could be used in the mass balance equation to verify global mass conservation. The calculation of outflow is relatively simple compared to deposition; it is merely the particle mass flux through the pipe exit. As in the steady flow simulations, the overlap area of the pipe cross-section and the particle ellipse is considered. This area cannot simply be multiplied by particle

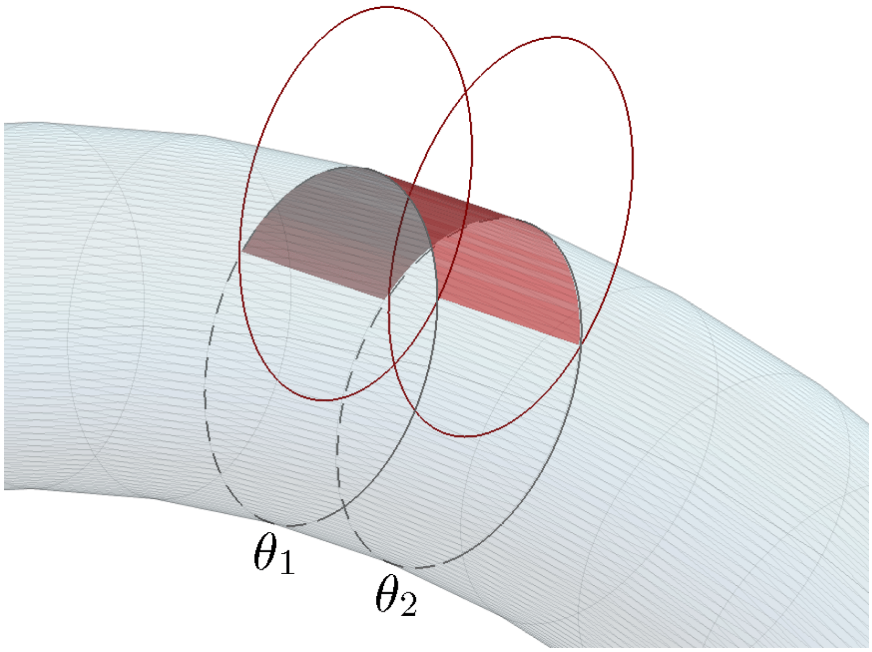

Figure 4.3: *The pipe/particle intersection surface (shown as the shaded red area) is plotted on a single CV for typical case; the red ellipses are the particle cross-sections at the CV boundaries, and θ_1 and θ_2 are the CV limits.*

exit velocity, because the exit velocity varies in the radial direction. Instead, another integral must be performed where the integrand is the multiple of the outlet area element, the local exit velocity, and the local concentration. The solution of this integral is found in Appendix A.4.

5

Results and Validation

In this section, the results generated by the deposition code are validated and interpreted.

5.1 Validation

Two approaches were taken to validation of the particle deposition code described in Chapter 4. First, the conservation of mass of the code was verified by comparing the total mass of particles in to the sum total mass of particles out, particles deposited, and particles resident at the end of the simulation. Second, the code was externally verified by comparing results with another

published analytic deposition model and available experimental results.

Three noteworthy approximations were made in the process of generating the final deposition code, and thus some degree of inaccuracy is expected. The sources of error and their anticipated effect are briefly discussed here. First, the transient fluid flow function was replaced by its first order Taylor series approximation. For sinusoidal fluid flow, this substitution will only have a significant effect for relatively long particle residence times. Such long times occur only near the beginning of simulations which start with zero velocity. For triangle wave flow, which, while farther from an actual breath pattern than sinusoidal flow, is preferentially used to study the effects of flow acceleration on deposition due to its constant acceleration rate, there is no error due to the Taylor approximation because it is identical to the original triangle function.

Second, the Airy functions which appeared in the original solution to the particle path ODE were replaced by asymptotically equivalent (for large arguments) exponential functions. The error contribution of these substitutions is difficult to ascertain due to the number of Airy functions replaced and the complex nature of their arguments. It is clear that the error will increase with Stokes number, which appears in the denominator of all of the Airy function arguments. Thus, there will be some maximum value of S beyond which the asymptotic approximations break down and the numerical results are invalid.

Third, geometric approximations that were used in the numerical code itself will create error in the calculation of flux and other quantities. Included in this category are discretization approximations common to all numerical simulations (i.e. finite CV and time step lengths) as well as averages and other approximations of length, area, and volume which were made necessary by the complicated nature of the curved pipe domain. Finer grid and time spacing will reduce the effect of both of these types of approximations to a point; the latter type will also be affected by the relative magnitudes of fluid velocity, flow period, and pipe ratio.

It should be noted that only the third type of approximation will affect the conservation of mass; the first two types relate to the accuracy of the physical model itself and are independent of the numerical implementation of the model. They will potentially lead to disparity when the simulation results are externally validated. Though they will not affect conservation of mass, the first two approximations could still lead to erroneous simulation results due to the particle path functions returning nonphysical values. Also, as they are unaffected by the third type of approximation, the steady and quasi-steady simulations do not need to be checked for conservation of mass.

5.1.1 Internal Validation

The sources of error described above suggest that once CV size and time step have been sufficiently reduced to create grid independence of the solution, the simulation error will depend only on the dimensionless quantities S, Str, and R_p. Indeed, it was found that these three quantities determine the results of the simulation in general. Dimensioned parameters such as particle diameter, pipe radius, and fluid magnitude can be varied without affecting the results, as long as the dimensionless quantities are maintained. One exception to this rule is that extreme values of the parameters may lead to the failure of the simulation for purely numeric reasons.

The pipe ratio is the least interesting of the dimensionless quantities, so it will be held at a constant value of $R_p = 10$ for all of the simulations testing mass conservation. This value is roughly equal to that of a typical mouth/throat bend. For the remaining two quantities, S and Str, a series of simulations were run over a wide range of values of each. The Stokes number was varied from 5E-3 to 5E-1, which corresponds to a particle size range of 5.8μm to 58μm for $W = 5$m/s and $R = 0.1$m. The Strouhal number was varied from 1.25E-2 to 5E-2, which corresponds to a flow period range of 1.6s to 0.4s for the same

values of W and R.

Mass loss magnitude is plotted against S and Str in Figure 5.1. As expected, the error increases with both the Stokes and Strouhal numbers. For a given Stokes number, there exists a maximum Strouhal number beyond which the corresponding fluid acceleration is sufficient to cause early particles to be surpassed by other particles which enter the pipe later. Overlapping particles lead to mass conservation error because the simulation cannot distinguish between them, and the larger the acceleration, the greater the effect.

Three points have been identified at which the mass loss is approximately 5%; these points were used to generate a limit of Str as a function of S, shown in black:

$$Str < 1.5\text{E-}3 \left(9.85 + \frac{1}{3.7\text{E-}3 + S}\right)$$

Keeping S and Str within this limit should avoid any problems due to nonphysical flow behaviour, and the conservation of mass error will be kept roughly below 5%.

5.1.2 External Validation

The ideal validation of the transient numerical simulation code is comparison with appropriate experimental results. Unfortunately, such results are unavailable at the present time for time-dependent particle deposition. Instead, results from the steady-state version of the code will be compared with analytic and experimental results from the literature.

An analytic solution was obtained for particle deposition in steady curved-pipe flow by Cheng and Wang [1] using a similar approach to the present work. Specifically, the same differential equation was solved using the same spatial fluid velocity profile but without the transient component. The actual solution process used was different, but deposition was calculated the same

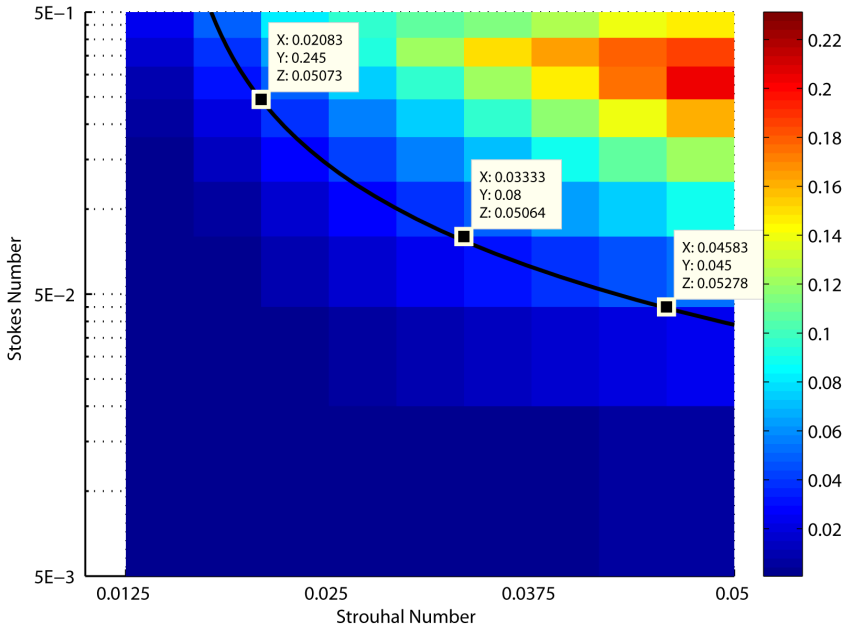

Figure 5.1: *Magnitude of mass lost (as a fraction of total mass in) is plotted versus a range of Stokes and Strouhal numbers; the black line indicates a rough limit beyond which error is > 5%.*

way. This solution has been implemented numerically in *chengDep.m*, found in Appendix B.4. Since it models steady flow, this solution depends on S and R_p but not Str, which is the same as *steadyDep.m*. The deposition fractions over a range of S and with $R_p = 10$ for the two steady solutions are plotted in Figure 5.2. The experimental results of [2] are also included in this plot. In that experiment, monodisperse aerosols were generated using a spinning disc atomizer. The aerosols were passed through various bends and the deposited fraction was measured using fluorimetry.

Figure 5.2 shows only a minor difference between the two analytic solutions. This reflects well on the accuracy of the process used to create the analytic functions in *steadyDep.m*; furthermore, it also lends confidence to the accuracy of the transient analytic solutions of *eulerDep.m*. This is because the

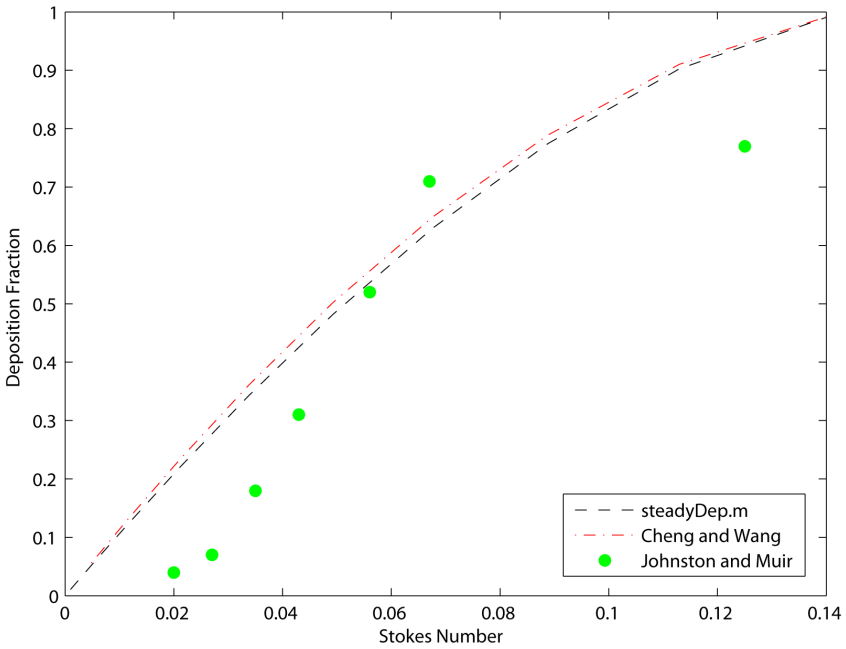

Figure 5.2: *The steady-state deposition predictions of steadyDep.m are compared against the analytic results of [1] the experimental results of [2]*

steady deposition functions of the present work were obtained *a posteriori* from the previously solved transient differential equations, rather than from steady differential equations as was the case for Cheng and Wang. However, this favourable comparison says nothing about the accuracy of the physical model underlying the analytic equations in terms of real-world deposition, as both solutions have a purely mathematical basis. To test the model itself, its results must be compared with those of physical experiments.

The goal of the present work was never to create an accurate and complete Eulerian analytic deposition model; rather, as was previously mentioned, the goal was to identify functional forms on which such a deposition model could be based. For this reason, the only idea that needs to be taken from the comparison with the experimental results in 5.2 is that the deposition results are indeed within a reasonable range suggested by experiment. Deposition proceeds from

zero to near one hundred percent over the range of Stokes numbers roughly predicted by experiment. Any conclusions more specific than this would be fruitless, as there are many more factors affecting real-world deposition (such as diffusion and sedimentation, turbulence, particle shape and hygroscopicity, etc.) than are taken into account in a simple analytic model like the one presented here. Thus, with a fair degree of certainty that the physical principles which underlie the deposition simulations are sound, and that they have been properly implemented in the code, the results are examined.

5.2 Results

Since the accuracy of the transient results cannot be guaranteed, the discussion of results is centered around comparisons of the quasi-steady and transient results. The qualitative results should be reliable even if the quantitative results are not. Conclusions can be drawn about which flow regimes demand Eulerian modelling, and which can do without. Three different transient flow profiles are examined below: the acceleration and deceleration phases of a triangle wave, and a complete sine wave. Standard values of $R = 0.1$m and $a = 0.01$m are used in the interpretation of all results; these values approximate a curved-pipe analogue for a human throat. Also, all percentages reported below refer to the *relative* difference between the two simulation types, so for a transient result of 30% deposition and a quasi-steady result of 36% deposition, it would be reported that the quasi-steady code predicted 20% more deposition than the transient code, and not 6% more.

5.2.1 Constant Acceleration

Constant, positive flow acceleration is the simplest case, so it was tested first. The fluid velocity was started at zero and accelerated to the maximum value

of W, and the pipe began empty of particles. Quasi-steady and transient simulations were run using the same flow parameters, time step, and number of CVs. The flow parameters were varied over a wide range of Stokes and Strouhal numbers, but kept within the validity limit established in the previous section. The results are summarized in Figure 5.3.

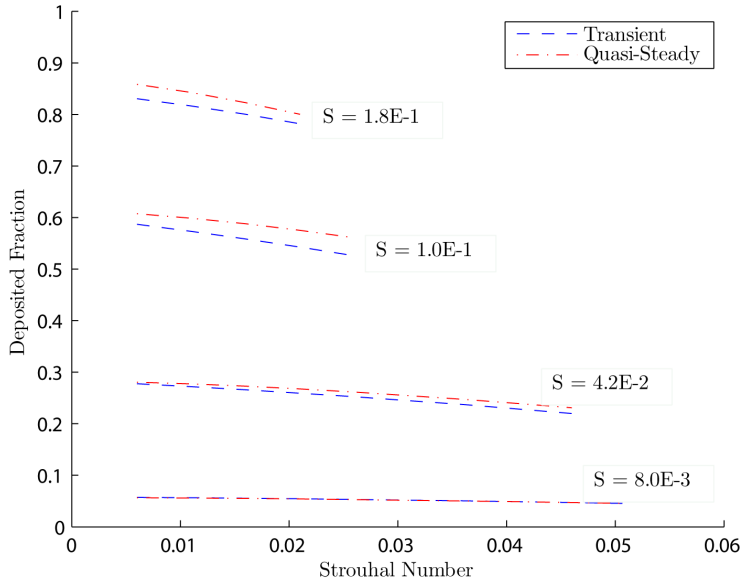

Figure 5.3: *Deposition results for transient and quasi-steady simulations are plotted versus Strouhal number for a range of Stokes numbers; these simulations used constant fluid acceleration.*

In all cases, the quasi-steady code predicted equal or larger deposition than the transient code; the reason for this is explained below. The relative difference increases with Stokes number along with the absolute deposition fraction. Deposition is effectively equal for the two methods at the lowest Stokes number, it increases with Strouhal number for the middle two values of S, and decreases slightly with Str for the highest S tested. This last result is likely due to the flow parameters being near the limit of the range of validity previously established and not some physical effect.

The largest difference was 6.6%, occurring at values of $S = 0.1$ and $Str = 0.026$. This is equivalent to a peak inhalation speed of 8m/s and an inhalation time of 0.24s for a 20μm particle, with a pipe Reynold's number of about 5E3 (the pipe Reynold's number is defined as Wa/ν). This is a fairly extreme case that is unlikely to occur in a normal breathing situation, but it is possible during a sharp intake of breath such as a gasp, and if the particle size increases just slightly, the corresponding inhalation speed and time become much more reasonable. This case was examined in more detail by plotting the instantaneous deposition rate as a fraction of inflow rate versus time for both the quasi and transient simulations over their full duration, Figure 5.4. Note that while deposition rate may differ, both simulations have the same inflow rate.

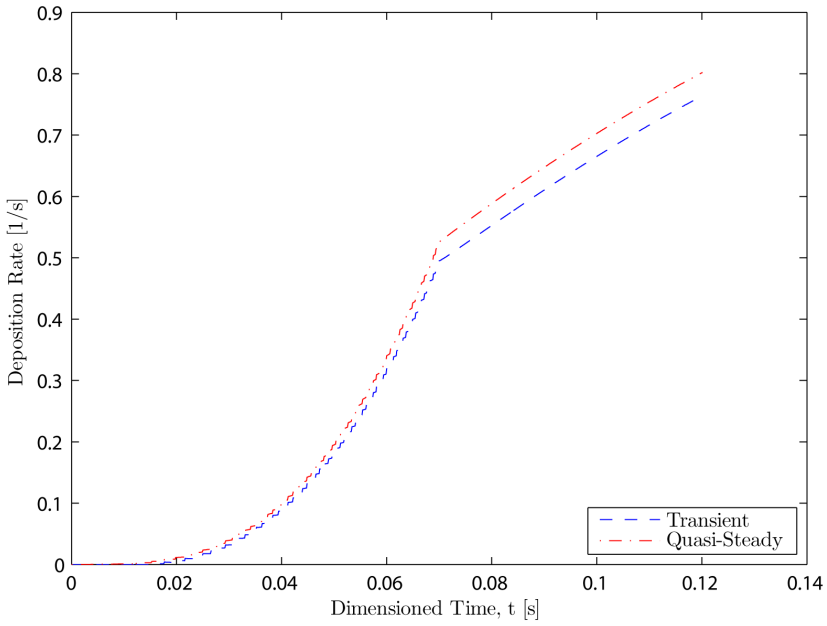

Figure 5.4: *Quasi-steady and transient instantaneous deposition rates are plotted for the set of parameters corresponding to the largest total deposition prediction difference in the constant acceleration case.*

Figure 5.4 shows that both simulation types have the same deposition rate

profile, with the quasi-steady simulation having a slightly higher magnitude throughout. The deposition rates start at zero when the pipe is empty, and begin curving upward. The curved portions of the plots appear bumpy; this is a result of the discretized pipe domain and the fact that deposition is not calculated for a given CV until it is fully infiltrated by the particles. This is also the reason why there is a small period of zero deposition at the start of the simulations. There is a discontinuity in the plots which coincides with the time when the pipe becomes fully infiltrated; after this point deposition increases approximately linearly with the fluid velocity.

The quasi-steady deposition is slightly higher because of the lack of time-history of the particles. Because each new time step is treated as a new slightly faster steady flow, the velocity change is felt instantly throughout the pipe by all of the particles. By contrast, the particles in the transient simulation have inertia and time-history, and thus there is a time lag before the particles reach the new higher velocity that is reflected in the slightly lower deposition rate.

5.2.2 Constant Deceleration

Next, the second half of a triangle wave inhalation was simulated. The fluid velocity started at W and steadily decelerated down to zero, and once again the pipe started empty. The same ranges of flow parameters as in the acceleration case were used. The results are summarized in Figure 5.5.

As in the previous case, the quasi-steady simulation predicted as much or more deposition than the transient for all cases. Also as before, total deposition decreased with Strouhal number. For both simulations, the deposition result was slightly lower than the corresponding run in the acceleration case. The maximum deposition result difference was 5%, for $S = 0.3$ and $Str = 0.016$. This case is equivalent to a peak inhalation speed of 15m/s and an inhalation period of 0.21s for a 26μm particle, and a pipe Reynold's number of about

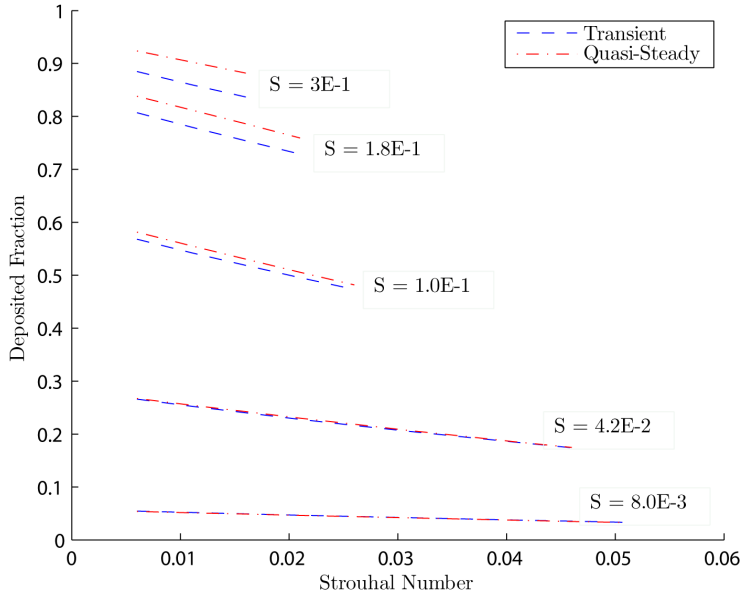

Figure 5.5: *Deposition results for transient and quasi-steady simulations are plotted versus Strouhal number for a range of Stokes numbers; these simulations used constant fluid deceleration.*

9E3. Instantaneous deposition is plotted versus time for this case in Figure 5.6.

This plot is slightly more interesting than the last deposition rate profile. Both rates are roughly the same as the pipe begins to fill up with particles, but once all of the initial particles have deposited, the profiles diverge somewhat. This point is represented by the first discontinuity in each of the plots, and it occurs slightly earlier for the transient run. Note that the point of complete deposition occurs for both simulations *before* the pipe is fully infiltrated; the dotted black line denotes the time of full infiltration.

After complete deposition is reached, the deposition rate of the quasi-steady run levels off at unity for about half of the remaining simulation time. During this period, the fluid velocity is constantly decreasing, but still large enough

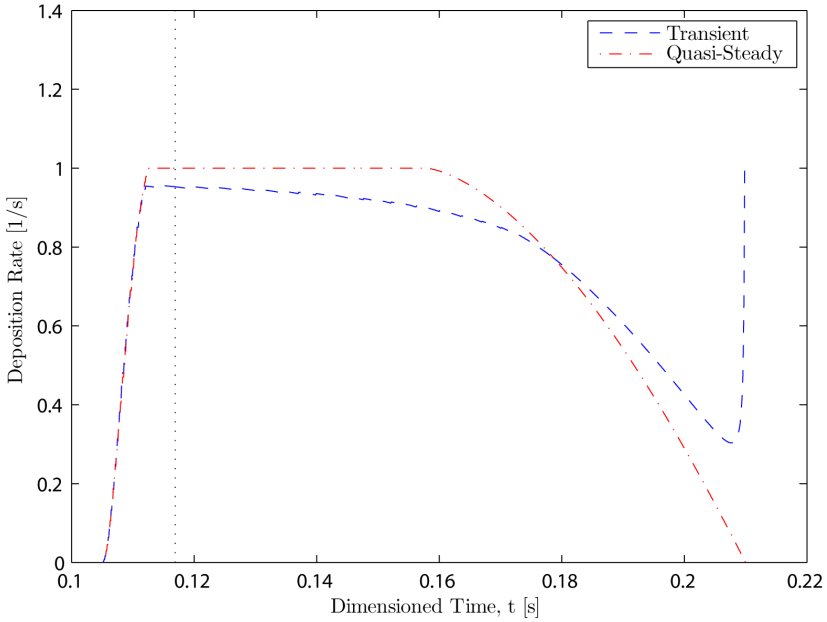

Figure 5.6: *Quasi-steady and transient instantaneous deposition rates are plotted for the set of parameters corresponding to the largest total deposition prediction difference in the constant deceleration case; the dotted black line denotes the point at which the pipe is fully infiltrated.*

for complete deposition to occur in the corresponding steady flow case. Once the fluid velocity has decreased sufficiently to allow some particles to escape the pipe, the quasi rate begins to drop off, reaching a value of zero at the end of the simulation when the fluid velocity is also zero.

To understand the behaviour of the transient deposition rate, inertia and the time-history of the particles must once again be taken into account. Before depositing, particles entering the pipe must spend some time in it. In the quasi-steady case, these resident particles simply appear in the pipe because equilibrium is instantly achieved at each new time step. The result is that the inflow rate will always equal the sum of the deposition and outflow rates. For the transient case, however, two fluxes must be considered: the inflow of particles adding to the resident particle mass, and the sum of deposition

and outflow subtracting from it. Since the resident particle mass increases throughout the constant deceleration simulation, the normalized deposition rate must always be something less than unity. That said, the time-lag which caused the transient rate to trail the quasi slightly in the constant acceleration case has the opposite effect here. This means that the transient deposition rate decreases at a slower rate than the quasi, eventually overtaking it near the end of the simulation. Finally, the time-lag also leads to the inflow rate reaching zero before the deposition rate, which causes the singular spike at the end of the simulation for the transient case.

5.2.3 Sinusoidal Acceleration and Deceleration

The final flow profile tested was a sine curve which began and ended at zero velocity, with the peak equal to W. Thus, for this flow, both the acceleration and deceleration phases are included in the simulation. Other than the flow profile, no other parameters were modified from the previous two cases. The results are shown in Figure 5.7.

For the top three values of Stokes numbers, the quasi-steady simulations predicted more deposition than the transient as in the constant acceleration cases. For the lower two values, the quasi-steady results can actually be seen to dip slightly below the transient. However, the difference is very small and could easily be due to simulation error. It can also be seen that the deposition difference is actually decreasing with Strouhal number for the top two Stokes numbers. This is unexpected and could be due to these cases being outside the range of validity of the simulation. The limiting function derived above used the constant acceleration case, and may not apply as well to the sinusoidal flow. The maximum deposition difference is 3%, seen for $S = 0.18$ and $Str = 0.006$. This is equivalent to an inhalation speed of 10m/s and period of 0.8s for a 25μm particle, and a pipe Reynold's number of 6E3. The deposition

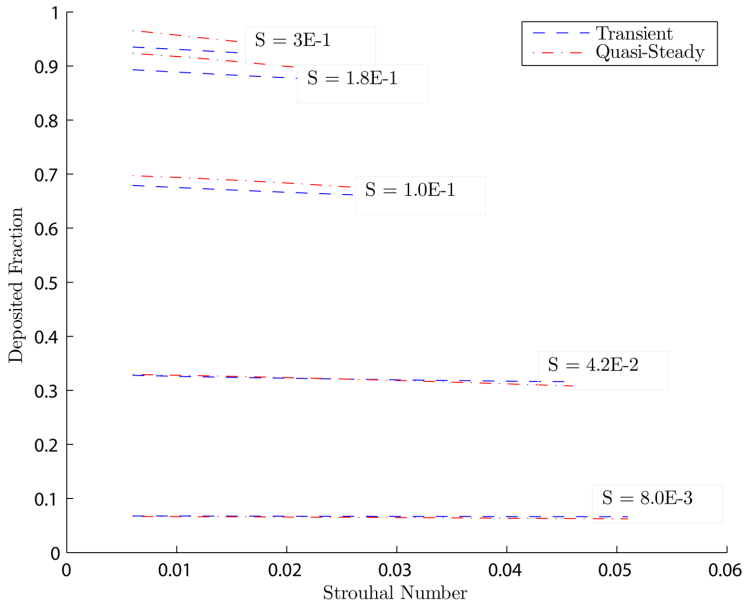

Figure 5.7: *Deposition results for transient and quasi-steady simulations are plotted versus Strouhal number for a range of Stokes numbers; these simulations used sinusoidal fluid acceleration and deceleration.*

rates for this case are plotted in Figure 5.8.

Here, the characteristics of the two deposition rate profiles for the sinusoidal flow case can be seen to be a combination of the previous two cases. At first, the rates increase from zero as more of the pipe is infiltrated; next, there is a discontinuity when full infiltration occurs, after which the deposition rates continue to increase with fluid velocity. Once complete deposition is achieved, the quasi-steady rate levels off at unity while the transient rate peaks at a slightly lower value and begins to decrease. At the halfway point in time, the flow begins to decelerate; soon the particles are once again moving slow enough that some can exit the pipe. At this point, the deposition rates begin to decrease more rapidly, with the transient rate overtaking the quasi rate and spiking up at the end of the simulation. The positive and negative slopes

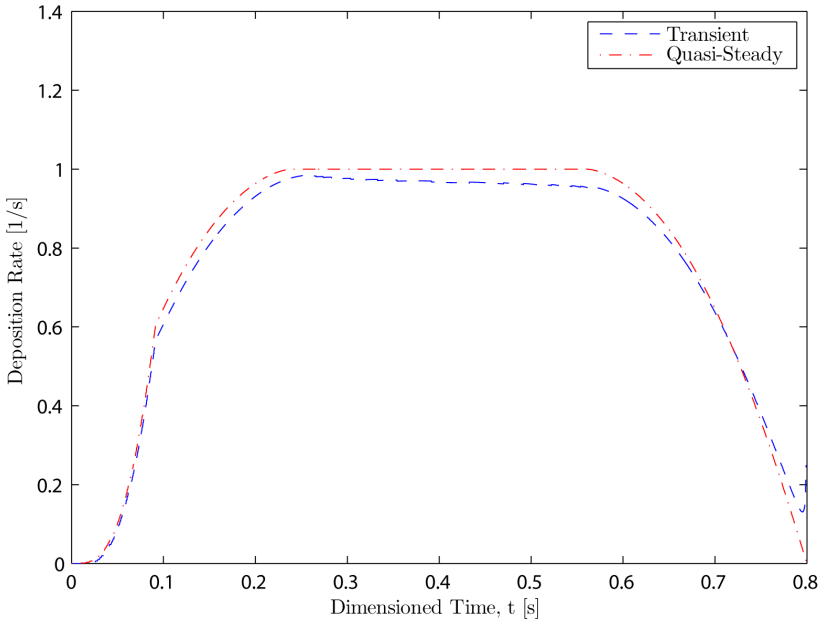

Figure 5.8: *Quasi-steady and transient instantaneous deposition rates are plotted for the set of parameters corresponding to the largest total deposition prediction difference in the sinusoidal flow case.*

are steeper than the corresponding regions of the constant acceleration cases. This is a result of the sine wave being steeper than the triangle wave at early and late times. If lighter particles were used and peak deposition was not 100%, the slopes of the sinusoidal case would be shallower than those of the triangular cases near the middle of the deposition profiles.

Finally, the cumulative deposition of both simulation types is plotted versus time for the case above, Figure 5.9. The cumulative deposition for each type has been normalized by the total deposition of that type. As seen in the figure, the normalized deposition profiles are effectively identical for the two methods. This means that, for this flow case at least, the quasi-steady simulation is a very accurate substitute for the instantaneous deposition rate of the transient simulation, as long the relative difference of the total deposition is known.

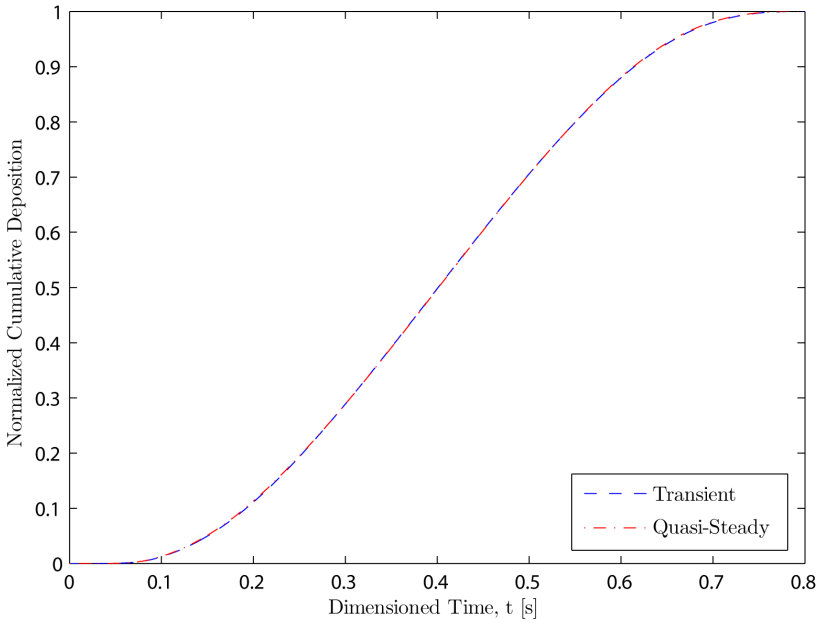

Figure 5.9: *Normalized cumulative quasi-steady and transient deposition fractions are plotted versus time for a sinusoidal flow case.*

The specific deposition cases described here involve larger particles, higher inhalation velocities, and shorter inhalation periods than those found in most real-life aerosol inhalation and breathing scenarios. However, they also represent the highest predicted differences between the quasi-steady and transient simulations. Differences of 1-2% could be found in more realistic cases; the consideration of Eulerian acceleration effects could be used to refine an already fairly accurate deposition model.

6

Conclusions

6.1 Summary

Several mathematical techniques were tested to solve transient equations of particle motion so that the solutions could be used to create an analytic Eulerian deposition model. The model would be used to predict aerosol deposition in the human lung using simplified representative geometry. Initial attempts used a continuum assumption for the particle field; this assumption was abandoned when the time-dependent Navier-Stokes equations proved too difficult to solve even in simplified forms. Success was found in the path solution of a single particle using an ODE based on Newton's second law and Stokes' law

for drag.

The transient particle path solution was implemented numerically in three different types of simulation: steady, quasi-steady, and transient. A dearth of experimental results for transient particle deposition limited external validation to the steady flow simulation; the transient simulation was internally validated using conservation of mass.

The results of the quasi-steady simulation, which models time-behaviour in a similar fashion to most deposition models currently in use, were compared to those of the fully transient (Eulerian) simulation. Using these two methods, deposition by impaction was predicted for particle-laden flow in a curved pipe. It was found that the difference in total deposition results for the two varied increasingly with the Stokes and Strouhal numbers of the flow, corresponding to heavier particles and shorter flow periods respectively.

Although the total deposition fractions varied, it was found that the instantaneous deposition rates predicted by the two methods were quite similar in character, and varied only slightly in magnitude. It was also found that the normalized instantaneous cumulative deposition fractions of the two methods were nearly identical, which suggests that the quasi-steady formulation could be used to determine an Eulerian instantaneous deposition rate as long as the ratio of the total deposition predictions of the two methods is known.

6.2 Future Work

The next step of the present work is to account for flow phenomena which were neglected in the particle path solution. The most important of these is secondary flow of the fluid, which will cause particle motion in the pipe circumferential direction. The Dean number of a typical flow case is approximately 2000, which suggests that secondary flow effects due to pipe curvature will

greatly affect the pipe flow field and thus deposition. Indeed, it was shown by [20] that the inclusion of this motion causes significant variation in the plot of deposition fraction versus Stokes number as compared to the simplified flow used here. In the simulation framework created here, secondary flow could be accounted for by adding a term for circumferential concentration changes in the particle ellipse, which would be tracked separately from the bulk particle motion and then included in the deposition calculation.

The effects on deposition of particle motion due to sedimentation and diffusion should also be included; the former is caused by gravity and the latter by turbulence, particle-particle collisions, and, for small particles, fluid-particle collisions. Sedimentation could be included in the simulation by adding an extra velocity term to the particles in the direction of gravity; it also might be possible to solve the particle motion ODE with the additional gravity term. Diffusion could be modelled by a combination of an area increase in the particle ellipse and a change in the concentration profile similar to that for secondary flow.

It is also important to obtain direct validation of the transient deposition model created here. If experimental data for instantaneous transient deposition are not forthcoming, an experiment could be devised to obtain them. Alternately, comparisons with other transient numerical simulations could be made.

Finally, separate from any additions to the model, the utility of the particle path solutions could be increased through simplification. The primary goal of this project was to identify functional forms for transient particle deposition. The particle path solutions presented here are quite complex, and as a result they require numerical solution before they can be used to calculate deposition. If they could be sufficiently simplified to allow inversion (without an unacceptable loss of accuracy) then they could be combined with the geometry-based deposition functions derived here to create a transient, fully analytic deposition model which does not require any numerical solution. Whatever loss of accu-

racy is required for inversion could be offset by the use of fitting coefficients based on empirical results.

References

[1] Y Cheng and C Wang. Inertial deposition of particles in a bend. *Journal of Aerosol Science*, 6(2):139–145, 1975. ISSN 0021-8502.

[2] JR Johnston, KD Isles, and DC Muir. Inertial deposition of particles in human branching airways. *Inhaled particles*, 4:61, 1975. ISSN 0301-1577.

[3] WH Finlay. *The mechanics of inhaled pharmaceutical aerosols: An introduction*. Academic Pr, 2001. ISBN 0122569717.

[4] AA Rostami. Computational Modeling of Aerosol Deposition in Respiratory Tract: A Review. *Inhalation toxicology*, 21(4):262–290, 2009. ISSN 0895-8378.

[5] CF Lange and WH Finlay. A Fully Eulerian Approach to the Simulation of Volatile/ Hygroscopic Aerosols. In G. E. Schneider, editor, *Proceedings of the Ninth Annual Conference of the CFD Society of Canada*, pages 279–283. CFD Society of Canada, July 2001.

[6] DB Taulbee and CP Yu. A theory of aerosol deposition in the human respiratory tract. *Journal of Applied Physiology*, 38(1):77, 1975. ISSN 8750-7587.

[7] MJ Egan and W Nixon. A model of aerosol deposition in the lung for use in inhalation dose assessments. *Radiation Protection Dosimetry*, 11(1):5, 1985. ISSN 0144-8420.

[8] D.A. Edwards. The macrotransport of aerosol particles in the lung: Aerosol deposition phenomena. *Journal of Aerosol Science*, 26(2):293–317, 1995. ISSN 0021-8502.

[9] MI Rahman. Lagrangian-to-Eulerian Conversion Scheme For Aerosol Deposition In Human Respiratory Tract. Master's thesis, University of Alberta,

2007.

[10] W. Stahlhofen, G. Rudolf, and AC James. Intercomparison of experimental regional aerosol deposition data. *J. Aerosol Med*, 2(3):285–308, 1989.

[11] AC James, W. Stahlhofen, G. Rudolf, MJ Egan, W. Nixon, P. Gehr, and JK Briant. The respiratory tract deposition model proposed by the ICRP Task Group. *Radiation Protection Dosimetry*, 38(1-3):159, 1991. ISSN 0144-8420.

[12] Computer-optimised pulmonary delivery in humans of inhaled therapies (cophit). http://www.ist-world.org/ProjectDetails.aspx?ProjectId=0002f719bfbe460e9c9ddb88bd7ae2a5, 2002.

[13] P.R. Byron, M. Hindle, C.F. Lange, P.W. Longest, D. McRobbie, M.J. Oldham, B. Olsson, C.G. Thiel, H. Wachtel, and W.H. Finlay. In Vivo–In Vitro Correlations: Predicting Pulmonary Drug Deposition from Pharmaceutical Aerosols. *Journal of Aerosol Medicine and Pulmonary Drug Delivery*, 23(S2):59–69, 2010. ISSN 1941-2711.

[14] CM Bender and SA Orszag. *Advanced mathematical methods for scientists and engineers: Asymptotic methods and perturbation theory*. Springer Verlag, 1999. ISBN 0387989315.

[15] J He. A new approach to nonlinear partial differential equations. *Communications in Nonlinear Science and Numerical Simulation*, 2(4):230–235, 1997. ISSN 1007-5704.

[16] E. Cunningham. On the velocity of steady fall of spherical particles through fluid medium. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 83(563):357–365, 1910. ISSN 0950-1207.

[17] GG Stokes. *On the effect of the internal friction of fluids on the motion of pendulums*. Pitt Press, 1851.

[18] SA Berger, L. Talbot, and LS Yao. Flow in curved pipes. *Annual Review of*

Fluid Mechanics, 15(1):461–512, 1983. ISSN 0066-4189.

[19] AP Roth, CF Lange, and WH Finlay. The effect of breathing pattern on nebulizer drug delivery. *Journal of Aerosol Medicine*, 16(3):325–339, 2003. ISSN 0894-2684.

[20] YS Cheng and CS Wang. Motion of particles in bends of circular pipes. *Atmospheric Environment (1967)*, 15(3):301–306, 1981. ISSN 0004-6981.

A

Appendix A: Derivations

A.1 The Complex $s(t)$ Solution

The solution to the time-dependent particle flow equation (equation 3.2.1 in Section 3) is given by Mathematica as

$$\begin{aligned} s(t) &= e^{-\frac{t}{2S}}\left[\frac{\left[(s_0+2SV_0)\mathrm{Bi}(C)-2k^{1/3}Ss_0\mathrm{Bi}'(C)\right]\mathrm{Ai}(D)}{2k^{1/3}S[\mathrm{Ai}'(C)\mathrm{Bi}(C)-\mathrm{Ai}(C)\mathrm{Bi}'(C)]}\right. \\ &\quad \left.-\frac{\left[(s_0+2SV_0)\mathrm{Ai}(C)-2k^{1/3}Ss_0\mathrm{Ai}'(C)\right]\mathrm{Bi}(D)}{2k^{1/3}S[\mathrm{Ai}'(C)\mathrm{Bi}(C)-\mathrm{Ai}(C)\mathrm{Bi}'(C)]}\right] \end{aligned} \tag{A.1.1}$$

where

$$\begin{aligned} C &= \frac{1-4iSA}{4k^{2/3}S^2} \\ D &= C+k^{1/3}t \\ k &= \frac{-iB}{S} \end{aligned}$$

Ai, Bi are the Airy functions and Ai′, Bi′ are their derivatives, S is the Stokes number, and $f(t) \approx A(t_0) + B(t_0)t$. The solution is first simplified using the asymptotic

behaviour of the Airy functions as $x \to \infty$:

$$\begin{aligned}
\mathrm{Ai}(x) &\sim \frac{e^{-\frac{2}{3}x^{3/2}}}{2\sqrt{\pi}x^{1/4}} \\
\mathrm{Bi}(x) &\sim \frac{e^{\frac{2}{3}x^{3/2}}}{\sqrt{\pi}x^{1/4}} \\
\mathrm{Ai}'(x) &\sim \frac{-e^{-\frac{2}{3}x^{3/2}}}{2\sqrt{\pi}}\left[\frac{x^{-5/4}}{4} + x^{1/4}\right] \\
\mathrm{Bi}'(x) &\sim \frac{-e^{\frac{2}{3}x^{3/2}}}{\sqrt{\pi}}\left[\frac{x^{-5/4}}{4} - x^{1/4}\right]
\end{aligned}$$

The first term of the numerator in the parentheses of equation A.1.1:

$$\begin{aligned}
&\left[(s_0 + 2SV_0)\mathrm{Bi}(C) - 2k^{1/3}Ss_0\mathrm{Bi}'(C)\right]\mathrm{Ai}(D) \\
\approx\ &\left[(s_0 + 2SV_0)\frac{e^{\frac{2}{3}C^{3/2}}}{\sqrt{\pi}C^{1/4}}\right. \\
&\left. +\ 2k^{1/3}Ss_0\frac{e^{\frac{2}{3}C^{3/2}}}{\sqrt{\pi}}\left(\frac{C^{-5/4}}{4} - C^{1/4}\right)\right]\frac{e^{-\frac{2}{3}(C+k^{1/3}t)^{3/2}}}{2\sqrt{\pi}(C+k^{1/3}t)^{1/4}} \\
=\ &\left[(s_0 + 2SV_0)C^{-1/4}\right. \\
&\left. +\ 2k^{1/3}Ss_0\left(\frac{C^{-5/4}}{4} - C^{1/4}\right)\right]\frac{e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]}}{2\pi(C+k^{1/3}t)^{1/4}} \qquad \text{(A.1.2)}
\end{aligned}$$

The second term of the numerator:

$$\begin{aligned}
&\left[(s_0 + 2SV_0)\mathrm{Ai}(C) - 2k^{1/3}Ss_0\mathrm{Ai}'(C)\right]\mathrm{Bi}(D) \\
\approx\ &\left[(s_0 + 2SV_0)\frac{e^{-\frac{2}{3}C^{3/2}}}{2\sqrt{\pi}C^{1/4}}\right. \\
&\left. +\ 2k^{1/3}Ss_0\frac{e^{-\frac{2}{3}C^{3/2}}}{2\sqrt{\pi}}\left(\frac{C^{-5/4}}{4} + C^{1/4}\right)\right]\frac{e^{\frac{2}{3}(C+k^{1/3}t)^{3/2}}}{\sqrt{\pi}(C+k^{1/3}t)^{1/4}} \\
=\ &\left[(s_0 + 2SV_0)C^{-1/4}\right. \\
&\left. +\ 2k^{1/3}Ss_0\left(\frac{C^{-5/4}}{4} + C^{1/4}\right)\right]\frac{e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]}}{2\pi(C+k^{1/3}t)^{1/4}} \qquad \text{(A.1.3)}
\end{aligned}$$

The denominator:

$$\begin{aligned} & 2k^{1/3}S[\mathrm{Ai}'(C)\mathrm{Bi}(C) - \mathrm{Ai}(C)\mathrm{Bi}'(C)] \\ \approx\ & 2k^{1/3}S\left[\frac{-e^{-\frac{2}{3}C^{3/2}}}{2\sqrt{\pi}}\left(\frac{C^{-5/4}}{4} + C^{1/4}\right)\frac{e^{\frac{2}{3}C^{3/2}}}{\sqrt{\pi}C^{1/4}}\right. \\ & \left. + \frac{e^{-\frac{2}{3}C^{3/2}}}{2\sqrt{\pi}C^{1/4}}\frac{e^{\frac{2}{3}C^{3/2}}}{\sqrt{\pi}}\left(\frac{C^{-5/4}}{4} - C^{1/4}\right)\right] \\ =\ & \frac{-2k^{1/3}S}{\pi} \end{aligned} \tag{A.1.4}$$

Substituting equations A.1.2, A.1.3, and A.1.4 into equation A.1.1 gives

$$\begin{aligned} s(t) =\ & e^{\frac{-t}{2S}}\left\{\left[-\frac{(s_0+2SV_0)C^{-1/4}}{2k^{1/3}S} + s_0\left(-\frac{C^{-5/4}}{4} + C^{1/4}\right)\right]\frac{e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]}}{2(C+k^{1/3}t)^{1/4}}\right. \\ & \left. + \left[\frac{(s_0+2SV_0)C^{-1/4}}{2k^{1/3}S} + s_0\left(\frac{C^{-5/4}}{4} + C^{1/4}\right)\right]\frac{e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]}}{2(C+k^{1/3}t)^{1/4}}\right\} \\ =\ & \frac{e^{\frac{-t}{2S}}}{2C^{1/4}(C+k^{1/3}t)^{1/4}}\left\{s_0C^{1/2}\left(e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} + e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]}\right)\right. \\ & \left. - \left[\frac{s_0+2SV_0+0.5C^{-1}}{2k^{1/3}S}\right]\left(e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} - e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]}\right)\right\} \end{aligned} \tag{A.1.5}$$

Each piece of this complex equation will now be expanded to create an entirely real analytic function. First, the complex exponent of e:

$$\frac{2}{3}\left[(C+k^{1/3}t)^{3/2} - C^{3/2}\right] \tag{A.1.6}$$

Factoring the expressions C and $C+k^{1/3}t$ gives

$$\begin{aligned} C &= \frac{1-4iAS}{4k^{2/3}S^2} \\ &= \frac{(-i+\sqrt{3})(i+\alpha)}{8B^{2/3}S^{4/3}} \end{aligned} \tag{A.1.7}$$

$$\begin{aligned} C+k^{1/3}t &= \frac{(-i+\sqrt{3})(i+\alpha)}{8B^{2/3}S^{4/3}} + \left(\frac{-iB}{S}\right)^{1/3}t \\ &= \frac{(-i+\sqrt{3})(i+\alpha+\beta t)}{8B^{2/3}S^{4/3}} \end{aligned} \tag{A.1.8}$$

Here we have defined $\alpha = 4AS$ and $\beta = 4BS$. Equation A.1.6 becomes

$$\begin{aligned}\frac{2}{3}\left[(C+k^{1/3}t)^{3/2}-C^{3/2}\right] &= \frac{2}{3}\left(\frac{-i+\sqrt{3}}{8B^{2/3}S^{4/3}}\right)^{3/2}\left[(i+\alpha+\beta t)^{3/2}-(i+\alpha)^{3/2}\right] \\ &= \left(\frac{1-i}{3\sqrt{2}\beta S}\right)\left[(i+\alpha+\beta t)^{3/2}-(i+\alpha)^{3/2}\right] \quad \text{(A.1.9)}\end{aligned}$$

For further simplicity we define $\gamma = \alpha + \beta t$. Expanding the first term in the square brackets of equation A.1.9 gives

$$\begin{aligned}&(i+\gamma)^{3/2} \\ =\ &\left(1+\gamma^2\right)^{3/4}\cos\left(\frac{3}{2}\arg(i+\gamma)\right)+i\left(1+\gamma^2\right)^{3/4}\sin\left(\frac{3}{2}\arg(i+\gamma)\right) \quad \text{(A.1.10)}\end{aligned}$$

The second term in the square brackets of equation A.1.9 is expanded the same way with γ replaced by α. The function $\arg(i+\gamma)$ is equivalent to $\operatorname{atan}(1/\gamma)$. We thus define $\theta = \operatorname{atan}(1/\gamma)$, which leads to the expressions $\cos\theta = \gamma/\sqrt{1+\gamma^2}$ and $\sin\theta = 1/\sqrt{1+\gamma^2}$. Going back to equation A.1.10 above, we can create equations for $\cos\frac{3}{2}\theta$ and $\sin\frac{3}{2}\theta$ using basic trigonometric identities. They are

$$\cos\frac{3}{2}\theta = \sqrt{\frac{\cos\theta+1}{2}}(2\cos\theta-1) \quad \text{(A.1.11)}$$

$$\sin\frac{3}{2}\theta = \frac{\sin\theta(1+2\cos\theta)}{\sqrt{2(\cos\theta+1)}} \quad \text{(A.1.12)}$$

It should be noted that these equations are only valid for $-\pi < \theta < \pi$, which is sufficient for this case since $\operatorname{atan}(x)$ is bounded by $\pm\pi/2$. Combining equations A.1.11 and A.1.12 and the expressions for $\sin\theta$ and $\cos\theta$ above with equation A.1.10 and simplifying, we have

$$(i+\gamma)^{3/2} = \frac{(i+\gamma)(i+\gamma+\sqrt{1+\gamma^2})}{\sqrt{2}(1+\gamma^2)^{1/4}\sqrt{1+\frac{\gamma}{\sqrt{1+\gamma^2}}}}$$

Combining this expansion with equation A.1.9, we now have

$$\begin{aligned}
&\frac{2}{3}\left[(C+k^{1/3}t)^{3/2}-C^{3/2}\right] \\
= &\left(\frac{1-i}{6\beta S}\right)\left[\frac{(i+\gamma)(i+\gamma+\sqrt{1+\gamma^2})}{(1+\gamma^2)^{1/4}\sqrt{1+\frac{\gamma}{\sqrt{1+\gamma^2}}}}-\frac{(i+\alpha)(i+\alpha+\sqrt{1+\alpha^2})}{(1+\alpha^2)^{1/4}\sqrt{1+\frac{\alpha}{\sqrt{1+\alpha^2}}}}\right] \\
= &\frac{1}{6\beta S}\left[\frac{(\gamma^2+2\gamma-1+(\gamma+1)\sqrt{1+\gamma^2})+i(-\gamma^2+2\gamma+1+(1-\gamma)\sqrt{1+\gamma^2})}{\sqrt{\sqrt{1+\gamma^2}+\gamma}}\right. \\
&\left.-\frac{(\alpha^2+2\alpha-1+(\alpha+1)\sqrt{1+\alpha^2})+i(-\alpha^2+2\alpha+1+(1-\alpha)\sqrt{1+\alpha^2})}{\sqrt{\sqrt{1+\alpha^2}+\alpha}}\right] \\
= &f_1+if_2
\end{aligned}$$

For readability we have defined

$$\begin{aligned}
f_1 &= \frac{\gamma^2+2\gamma-1+(\gamma+1)\sqrt{1+\gamma^2}}{6\beta S\sqrt{\sqrt{1+\gamma^2}+\gamma}}-\frac{\alpha^2+2\alpha-1+(\alpha+1)\sqrt{1+\alpha^2}}{6\beta S\sqrt{\sqrt{1+\alpha^2}+\alpha}} \\
f_2 &= \frac{-\gamma^2+2\gamma+1+(1-\gamma)\sqrt{1+\gamma^2}}{6\beta S\sqrt{\sqrt{1+\gamma^2}+\gamma}}-\frac{-\alpha^2+2\alpha+1+(1-\alpha)\sqrt{1+\alpha^2}}{6\beta S\sqrt{\sqrt{1+\alpha^2}+\alpha}}
\end{aligned}$$

Returning to equation A.1.5, we can now expand the exponential functions inside the curly brackets:

$$\begin{aligned}
&e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} \\
= &e^{f_1+if_2} \\
= &e^{f_1}\cos(f_2)+ie^{f_1}\sin(f_2) \\
= &[\cosh(f_1)+\sinh(f_1)]\cos(f_2)+i[\cosh(f_1)+\sinh(f_1)]\sin(f_2)
\end{aligned}$$

$$\begin{aligned}
&e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]} \\
= &e^{-f_1-if_2} \\
= &e^{-f_1}\cos-f_2+ie^{-f_1}\sin-f_2 \\
= &[\cosh(-f_1)+\sinh(-f_1)]\cos(-f_2)+i\,[\cosh(-f_1)+\sinh(-f_1)]\sin(-f_2) \\
= &[\cosh(f_1)-\sinh(f_1)]\cos(f_2)-i\,[\cosh(f_1)-\sinh(f_1)]\sin(f_2)
\end{aligned}$$

Adding and subtracting the exponential terms:

$$\begin{aligned} & e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} - e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]} \\ = \; & [\cosh(f_1)+\sinh(f_1)]\cos(f_2) + i[\cosh(f_1)+\sinh(f_1)]\sin(f_2) \\ & -[\cosh(f_1)-\sinh(f_1)]\cos(f_2) + i[\cosh(f_1)-\sinh(f_1)]\sin(f_2) \\ = \; & 2\sinh(f_1)\cos(f_2) + 2i\cosh(f_1)\sin(f_2) \end{aligned} \tag{A.1.13}$$

$$\begin{aligned} & e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} + e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]} \\ = \; & [\cosh(f_1)+\sinh(f_1)]\cos(f_2) + i[\cosh(f_1)+\sinh(f_1)]\sin(f_2) \\ & +[\cosh(f_1)-\sinh(f_1)]\cos(f_2) - i[\cosh(f_1)-\sinh(f_1)]\sin(f_2) \\ = \; & 2\cosh(f_1)\cos(f_2) + 2i\sinh(f_1)\sin(f_2) \end{aligned} \tag{A.1.14}$$

We now consider the coefficient of the leading exponential in equation A.1.5:

$$\frac{1}{2C^{1/4}(C+k^{1/3}t)^{1/4}}$$

Expanding this term using Mathematica gives

$$\begin{aligned} \frac{1}{2C^{1/4}(C+k^{1/3}t)^{1/4}} &= \frac{\mathrm{Abs}(B)^{1/3}S^{2/3}}{[(1+\alpha^2)(1+\gamma^2)]^{1/8}}\left[\cos\left(\frac{1}{4}(\theta_1+\theta_2)\right)\right. \\ &\qquad \left. -i\sin\left(\frac{1}{4}(\theta_1+\theta_2)\right)\right] \\ &= \frac{\mathrm{Abs}(B)^{1/3}S^{2/3}}{[(1+\alpha^2)(1+\gamma^2)]^{1/8}}\left[g_1 - ig_2\right] \end{aligned} \tag{A.1.15}$$

Here we have expanded the expression using the same method as for equation A.1.10 with $\theta_1 = \mathrm{atan}[(\sqrt{3}-\alpha)/(1+\alpha)]$ and $\theta_2 = \mathrm{atan}[(\sqrt{3}-\gamma)/(1+\gamma)]$ using the quarter angle formulas

$$\cos\frac{1}{4}\theta = \sqrt{\frac{1}{2}+\frac{\sqrt{1+\cos\theta}}{2\sqrt{2}}} \tag{A.1.16}$$

$$\sin\frac{1}{4}\theta = \frac{\sin\theta}{\sqrt{(4+2\sqrt{2}\sqrt{1+\cos\theta})(1+\cos\theta)}} \tag{A.1.17}$$

and defining the functions

$$\begin{aligned} g_1 &= \sqrt{\frac{1}{2}+\frac{1}{2\sqrt{2}}\sqrt{1+\frac{-1+\sqrt{3}(\alpha+\gamma)+\alpha\gamma}{2\sqrt{(1+\alpha^2)(1+\gamma^2)}}}} \\ g_2 &= \frac{\sqrt{3}(1-\alpha\gamma)+\alpha+\gamma}{16\sqrt{(1+\alpha^2)(1+\gamma^2)}g_1(g_1^2-0.5)} \end{aligned} \tag{A.1.18}$$

Next, we move on to the constant terms inside the square brackets of equation A.1.5, starting with $s_0C^{1/2}$. Expanding this term as above gives

$$s_0C^{1/2} = \frac{(1+\alpha^2)^{1/4}}{2\text{Abs}(B)^{1/3}S^{2/3}}\left[x_0h_1-y_0h_2+i(y_0h_1+x_0h_2)\right] \tag{A.1.19}$$

where

$$\begin{aligned} h_1 &= \sqrt{\frac{1}{2}+\frac{1+\sqrt{3}\alpha}{4\sqrt{1+\alpha^2}}} \\ h_2 &= \sqrt{\frac{1}{2}-\frac{1+\sqrt{3}\alpha}{4\sqrt{1+\alpha^2}}} \end{aligned}$$

noting that $s_0 = x_0 + iy_0$ and $V_0 = u_0 + iv_0$ are the complex initial conditions.

Finally from equation A.1.5,

$$\begin{aligned} &\frac{s_0+2SV_0+0.5C^{-1}}{2k^{1/3}S} \\ =\ &\frac{2(\sqrt{3}+q\alpha)B^{2/3}S^{4/3}+(1+\alpha^2)[\sqrt{3}(x_0+2Su_0)-q(y_0+2Sv_0)]}{4(1+\alpha^2)\text{Abs}(B)^{1/3}S^{2/3}} \\ &+\ i\left(\frac{2(\sqrt{3}+q\alpha)B^{2/3}S^{4/3}+(1+\alpha^2)[\sqrt{3}(x_0+2Su_0)-q(y_0+2Sv_0)]}{4(1+\alpha^2)\text{Abs}(B)^{1/3}S^{2/3}}\right) \end{aligned} \tag{A.1.20}$$

where $q = \text{sign}(B)$ has been defined for readability.

The expansions from equations A.1.13, A.1.14, A.1.15, A.1.19, and A.1.20 are substituted into equation A.1.5, and the complex terms are multiplied out. Simplifying

the result and setting $x(t) = \mathrm{Re}[s(t)]$ and $y(t) = \mathrm{Im}[s(t)]$, we have

$$
\begin{aligned}
&x(t) = \\
&\frac{e^{\frac{-t}{2S}}}{2(1+\alpha^2)^{9/8}(1+\gamma^2)^{1/8}}\Bigg\{\cosh f_1\Bigg[2(1+\alpha^2)^{1/4}[(g_1h_1+g_2h_2)x_0+(g_2h_1-g_1h_2)y_0]\cos f_2 \\
&+\Big[g_2\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)-g_1\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\sin f_2\Bigg] \\
&+\sinh f_1\Bigg[2(1+\alpha^2)^{1/4}[(g_2h_1-g_1h_2)x_0-(g_1h_1+g_2h_2)y_0]\sin f_2 \\
&+\Big[g_1\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)+g_2\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\cos f_2\Bigg] \\
&-\frac{2qB^{2/3}S^{4/3}}{(1+\alpha^2)}\Bigg[\Big((\sqrt{3}\alpha-1)g_1-(\sqrt{3}+\alpha)g_2\Big)\cosh f_1\sin f_2 \\
&-\Big((\sqrt{3}\alpha-1)g_2+(\sqrt{3}+\alpha)g_1\Big)\sinh f_1\cos f_2\Bigg]\Bigg\} \\
&y(t) = \\
&\frac{e^{\frac{-t}{2S}}}{2(1+\alpha^2)^{9/8}(1+\gamma^2)^{1/8}}\Bigg\{-\cosh f_1\Bigg[2(1+\alpha^2)^{1/4}[(g_2h_1-g_1h_2)x_0-(g_1h_1+g_2h_2)y_0]\cos f_2 \\
&-\Big[g_1\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)+g_2\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\sin f_2\Bigg] \\
&-\sinh f_1\Bigg[-2(1+\alpha^2)^{1/4}[(g_1h_1+g_2h_2)x_0+(g_2h_1-g_1h_2)y_0]\sin f_2 \\
&+\Big[g_2\Big(\sqrt{3}(2Su_0+x_0)-q(2Sv_0+y_0)\Big)-g_1\Big(q(2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\Big)\Big]\cos f_2\Bigg] \\
&+\frac{2qB^{2/3}S^{4/3}}{(1+\alpha^2)}\Bigg[\Big((\sqrt{3}\alpha-1)g_2+(\sqrt{3}+\alpha)g_1\Big)\cosh f_1\sin f_2 \\
&+\Big((\sqrt{3}\alpha-1)g_1-(\sqrt{3}+\alpha)g_2\Big)\sinh f_1\cos f_2\Bigg]\Bigg\}
\end{aligned}
$$

A.2 The Steady Particle Path Function

The case of steady fluid flow is modelled by setting $B = 0$ in the approximated fluid velocity function $f(t) \approx A(t_0) + B(t_0)t$. Doing so in the final particle path functions results in an indeterminate answer; instead, we must begin with the asymptotic approximation of the original Airy function solution from A.1:

$$\begin{aligned} s(t) \approx & \frac{e^{\frac{-t}{2S}}}{2C^{1/4}(C+k^{1/3}t)^{1/4}} \left\{ s_0 C^{1/2} \left(e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} + e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]} \right) \right. \\ & \left. - \left[\frac{s_0 + 2SV_0 + 0.5C^{-1}}{2k^{1/3}S} \right] \left(e^{\frac{2}{3}[(C+k^{1/3}t)^{3/2}-C^{3/2}]} - e^{\frac{2}{3}[C^{3/2}-(C+k^{1/3}t)^{3/2}]} \right) \right\} \end{aligned}$$

Taking the limit of this function as $B \to 0$ and expanding out the complex ICs gives

$$\begin{aligned} \tilde{s} = & e^{\frac{-t}{2S}} \left\{ \frac{(x_0 + iy_0)}{2} \left[e^{\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} + e^{-\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} \right] \right. \\ & + \frac{\sqrt{2}(-1)^{1/6}[2S(u_0 + iv_0) + (x_0 + iy_0)]}{2\sqrt{(-i+\sqrt{3})(i+\alpha)}} \\ & \left. \left[e^{\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} - e^{-\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} \right] \right\} \end{aligned}$$

where $\tilde{s}$ has been defined as the steady particle function. Taking the same approach to complex term expansion as in the transient flow case (A.1), we first find that

$$\begin{aligned} & e^{\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} - e^{-\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} \\ = & \ 2\cos\tilde{f}_2 \sinh\tilde{f}_1 - 2i\sin\tilde{f}_2 \cosh\tilde{f}_1 \\ & e^{\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} + e^{-\frac{[(-i+\sqrt{3})(i+\alpha)]^{3/2}}{4\sqrt{2}S(i+\alpha)}t} \\ = & \ 2\cos\tilde{f}_2 \cosh\tilde{f}_1 - 2i\sin\tilde{f}_2 \sinh\tilde{f}_1 \end{aligned}$$

where

$$\begin{aligned} \tilde{f}_1 = & \frac{\alpha + \sqrt{3}(1 + \sqrt{1+\alpha^2})}{4S\sqrt{1+\sqrt{3}\alpha + 2\sqrt{1+\alpha^2}}}t \\ \tilde{f}_1 = & \frac{-1 + \sqrt{3}\alpha + \sqrt{1+\alpha^2}}{4S\sqrt{1+\sqrt{3}\alpha + 2\sqrt{1+\alpha^2}}}t \end{aligned}$$

Now the complex denominator of the coefficient of the exponential difference is expanded:

$$\frac{\sqrt{2}}{2\sqrt{(-i+\sqrt{3})(i+\alpha)}} = \tilde{g}_1 - i\tilde{g}_2$$

where

$$\begin{aligned}\tilde{g}_1 &= \frac{1}{4(1+\alpha^2)^{1/4}}\sqrt{2+\frac{1+\sqrt{3}\alpha}{\sqrt{1+\alpha^2}}} \\ \tilde{g}_2 &= \frac{1}{4(1+\alpha^2)^{1/4}}\sqrt{2-\frac{1+\sqrt{3}\alpha}{\sqrt{1+\alpha^2}}}\end{aligned}$$

Substituting these expansions into the original equation gives

$$\begin{aligned}\tilde{s} &= e^{\frac{-t}{2S}}\left\{\frac{(x_0+iy_0)}{2}\left[2\cos\tilde{f}_2\cosh\tilde{f}_1 - 2i\sin\tilde{f}_2\sinh\tilde{f}_1\right]\right. \\ &+ \left.[\tilde{g}_1 - i\tilde{g}_2](-1)^{1/6}[2S(u_0+iv_0)+(x_0+iy_0)]\left[2\cos\tilde{f}_2\sinh\tilde{f}_1 - 2i\sin\tilde{f}_2\cosh\tilde{f}_1\right]\right\}\end{aligned}$$

Finally, multiplying out all of the complex terms and setting $\tilde{x}(t) = \mathrm{Re}[\tilde{s}(t)]$ and $\tilde{y}(t) = \mathrm{Im}[\tilde{s}(t)]$ yields

$$\begin{aligned}\tilde{x}(t) &= e^{\frac{-t}{2S}}\left\{\cosh\tilde{f}_1\left[2x_0\cos\tilde{f}_2 + \left[\tilde{g}_2\left(-\sqrt{3}(2Su_0+x_0)+(2Sv_0+y_0)\right)\right.\right.\right. \\ &\qquad \left.\left.\left.+\tilde{g}_1\left((2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\right)\right]\sin\tilde{f}_2\right] \\ &\quad +\sinh\tilde{f}_1\left[2y_0\sin\tilde{f}_2 + \left[\tilde{g}_1\left(\sqrt{3}(2Su_0+x_0)-(2Sv_0+y_0)\right)\right.\right. \\ &\qquad \left.\left.+\tilde{g}_2\left((2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\right)\right]\cos\tilde{f}_2\right] \\ \tilde{y}(t) &= e^{\frac{-t}{2S}}\left\{\cosh\tilde{f}_1\left[2y_0\cos\tilde{f}_2 - \left[\tilde{g}_1\left(\sqrt{3}(2Su_0+x_0)-(2Sv_0+y_0)\right)\right.\right.\right. \\ &\qquad \left.\left.\left.+\tilde{g}_2\left((2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\right)\right]\sin\tilde{f}_2\right] \\ &\quad +\sinh\tilde{f}_1\left[-2x_0\sin\tilde{f}_2 + \left[\tilde{g}_2\left(-\sqrt{3}(2Su_0+x_0)+(2Sv_0+y_0)\right)\right.\right. \\ &\qquad \left.\left.+\tilde{g}_1\left((2Su_0+x_0)+\sqrt{3}(2Sv_0+y_0)\right)\right]\cos\tilde{f}_2\right]\end{aligned}$$

A.3 The Area Overlap Function

To calculate particle deposition in the curved pipe, an expression is needed for the area of overlap of an ellipse and a circle. As shown in 4.1, some restrictions apply: the minor axis length of the ellipse is equal to the circle radius a, the circle remains centered at $r = R$ and $z = 0$, and the ellipse moves only in the radial direction.

First, equations are written for the boundaries of the two shapes:

$$\begin{aligned} \text{Circle: } r &= \pm\sqrt{a^2 - z^2} + R \\ \text{Ellipse: } r &= \pm\frac{b}{a}\sqrt{a^2 - z^2} + R + h \end{aligned}$$

Since the ellipse moves in the positive r direction relative to the circle, only the negative half of the ellipse and the positive half of the circle need to be considered. Next, the two equations are combined to determine the z-values of the intercept points:

$$\begin{aligned} \sqrt{a^2 - z^2} + R &= -\frac{b}{a}\sqrt{a^2 - z^2} + R + h \\ \sqrt{a^2 - z^2}\left(1 + \frac{b}{a}\right) &= h \\ a^2 - z^2 &= \left(\frac{h}{1 + \frac{b}{a}}\right)^2 \\ z &= \pm\sqrt{a^2 - \left(\frac{ah}{a+b}\right)^2} \end{aligned}$$

With the intercepts known, the overlap area can be calculated by integrating the difference of the two boundary functions with the intercepts as the limits:

$$\begin{aligned} A_{ol} &= \int_{-\sqrt{a^2 - \frac{ah}{a+b}^2}}^{\sqrt{a^2 - \frac{ah}{a+b}^2}} \left[\sqrt{a^2 - z^2}\left(1 + \frac{b}{a}\right) - h\right] dz \\ &= a\left(-\frac{h\sqrt{(a+b)^2 - h^2}}{a+b} + (a+b)\text{asin}\left[\frac{\sqrt{(a+b)^2 - h^2}}{a+b}\right]\right) \end{aligned}$$

Finally, the deposited area of particles is simply the difference between the ellipse area and the overlap area,

$$A_{dep} = \pi ab + a\left(\frac{h\sqrt{(a+b)^2 - h^2}}{a+b} - (a+b)\text{asin}\left[\frac{\sqrt{(a+b)^2 - h^2}}{a+b}\right]\right)$$

The numeric deposition simulations require the rate of change of the deposited area, given by the derivative of the above formula:

$$\frac{\mathrm{d}A_{dep}}{\mathrm{d}t} = \frac{\partial A_{dep}}{\partial h}\frac{\mathrm{d}h}{\mathrm{d}t} + \frac{\partial A_{dep}}{\partial b}\frac{\mathrm{d}b}{\mathrm{d}t}$$

The rate of change of b with respect to time is much smaller than that of h, so the second term is neglected. Taking the derivative and simplifying,

$$\frac{\mathrm{d}A_{dep}}{\mathrm{d}t} = \frac{2a\sqrt{(a+b)^2 - h^2}}{a+b} v_r$$

where v_r is the radial velocity at the center of the ellipse.

A.4 The Outflow Function

Volumetric flow rate is calculated by multiplying cross-sectional area by fluid speed in the normal direction. At the outlet of the curved pipe the fluid speed is a linear function of the radial distance, $v = C_1 r + C_2$. The outlet flow rate could be approximated by multiplying the average outlet speed by the particle overlap area, but doing so would introduce error due to the asymmetry of the overlap area. Instead, a better way is to integrate the differential volume flow rate across the overlap area. The differential area element is the product of the overlap area width as a function of r multiplied by the infinitesimal height dr, which is then multiplied by the value of the fluid speed function at r. The limits of integration are the bottom of the ellipse and the top of the circle. The integral must be broken into two because of the discontinuity where the two shapes intersect. The equation and its solution are

$$
\begin{aligned}
\dot{Q}_{out} &= 2\int_{R+h-b}^{\frac{ah}{a+b}+R} \left[\frac{a}{b}\sqrt{b^2-[r-(R+h)]^2}\right](C_1 r + C_2)\mathrm{d}r \\
&+ 2\int_{\frac{ah}{a+b}+R}^{R+a} \left[\sqrt{a^2-(r-R)^2}\right](C_1 r + C_2)\mathrm{d}r \\
&= \pi\frac{a^2}{2}(C_1 r + C_2) + \frac{\pi ab}{2}[C_1(R+h)+C_2] \\
&\quad -a^2\left\{\frac{\sqrt{(a+b)^2-h^2}}{3(a+b)}\left(2C_1 a\left[\left(\frac{h}{a+b}\right)^2-1\right]+\frac{3h(C_1R+C_2)}{a+b}\right)\right. \\
&\qquad \left. +(C_1R+C_2)\mathrm{atan}\left(\frac{h}{\sqrt{(a+b)^2-h^2}}\right)\right\} \\
&\quad -ab\left\{\frac{\sqrt{(a+b)^2-h^2}}{3(a+b)}\left(2C_1 b\left[1-\left(\frac{h}{a+b}\right)^2\right]+\frac{3h(C_1(R+h)+C_2)}{a+b}\right)\right. \\
&\qquad \left. +(C_1(R+h)+C_2)\mathrm{atan}\left(\frac{h}{\sqrt{(a+b)^2-h^2}}\right)\right\}
\end{aligned}
$$

A.5 The Deposition Rate Function

The deposition rate of particles in a given CV in the curved pipe can be treated as the volume flux of particles through the pipe surface. Volumetric flow rate ($\dot{Q}$) through a surface is found by multiplying the surface area by the magnitude of the normal component of the velocity passing through the surface. If the normal velocity is not constant across the surface, as in the case of the curved pipe, $\dot{Q}$ must be found by integration. Two integrals will be performed: one around the circumference of the pipe and one in the θ-direction. The differential volume flow element is given by

$$\mathrm{d}\dot{Q} = v_{norm}\, a\mathrm{d}\phi\, r\mathrm{d}\theta$$

where $a\mathrm{d}\phi$ is the length of the surface area element in the circumferential direction and $r\mathrm{d}\theta$ is the length in the θ-direction. The angle ϕ has been defined as the angle between the r-axis and the circumferential location of the surface element. This angle is related to r by $\cos\phi = (r - R)/a$.

The circumference integral is symmetric about $\phi = 0$, so only half the integral needs to be performed with the result multiplied by two. The outer limit of the integral is the point of intersection between the pipe circumference and the particle ellipse. This point is given by

$$z = \sqrt{a^2 - \left(\frac{ah}{a+b}\right)^2}$$

or

$$\phi = \mathrm{acos}\left(\frac{h}{a+b}\right)$$

The limits of the second integral are the angular limits of the CV in question, θ_1 and θ_2. With these limits, the full double-integral can be written:

$$\dot{Q} = 2a \int_{\theta_1}^{\theta_2} \int_0^{\mathrm{acos}\left(\frac{h}{a+b}\right)} r v_{norm}\, \mathrm{d}\phi\, \mathrm{d}\theta$$

To determine v_{norm}, only the radial component of the particle velocity must be considered, since the axial component is tangential to the pipe surface. The radial velocity varies linearly with r:

$$\begin{aligned} v_r &= c_1 r + c_2 \\ &= c_1(a\cos\phi + R) + c_2 \end{aligned}$$

and the normal component of the velocity is given by

$$v_{norm} = [c_1(a\cos\phi + R) + c_2]\cos\phi$$

With these definitions, the complete equation can be written:

$$\dot{Q} = 2a \int_{\theta_1}^{\theta_2} \int_0^{\mathrm{acos}\left(\frac{h}{a+b}\right)} [a\cos\phi + R][c_1(a\cos\phi + R) + c_2]\cos\phi\, \mathrm{d}\phi\, \mathrm{d}\theta$$

Appendix A: Derivations

Evaluating the first integral gives

$$\begin{aligned}\dot{Q} &= \frac{a}{6}\int_{\theta_1}^{\theta_2}\Bigg[a\left(6\phi(2c_1R+c_2)+3(2c_1R+c_2)\sin 2\phi+ac_1\sin 3\phi\right)\\ &\qquad +3[3a^2c_1+4R(c_1R+c_2)]\sin\phi\Bigg]_0^{\operatorname{acos}\left(\frac{h}{a+b}\right)}\mathrm{d}\theta\\ &= \frac{a}{6}\int_{\theta_1}^{\theta_2}\Bigg\{a\Bigg[6(2c_1R+c_2)\operatorname{acos}\left(\frac{h}{a+b}\right)+6(2c_1R+c_2)\left(\frac{h}{a+b}\right)\sqrt{1-\left(\frac{h}{a+b}\right)^2}\\ &\qquad +ac_1\left(3\sqrt{1-\left(\frac{h}{a+b}\right)^2}-4\left[1-\left(\frac{h}{a+b}\right)^2\right]^{3/2}\right)\Bigg]\\ &\qquad +3[3a^2c_1+4R(c_1R+c_2)]\sqrt{1-\left(\frac{h}{a+b}\right)^2}\Bigg\}\mathrm{d}\theta\end{aligned}$$

The constants c_1 and c_2 for a given CV vary only a small amount over that CV and thus can be treated as constant. The ellipse parameters h and b vary linearly with θ, so the following substitution is made in order to evaluate the second integral:

$$\left(\frac{h}{a+b}\right)=k_1\theta+k_2$$

As long as the correct values of the constants k_1 and k_2 are chosen and the magnitude of $\theta_2-\theta_1$ is preserved, any limits can be used for the θ integral. To simplify the math θ_1 is set to zero and θ_2 becomes θ_{CV}, the angle transcribed by each CV. The final equation gives the deposition rate for a given CV, with the constants c_1, c_2, k_1, k_2 uniquely defined for each CV.

$$\begin{aligned}\dot{Q} &= \frac{a}{12k_1}\Big[3[3a^2c_1+4R(c_1R+c_2)](k_1\theta+k_2)\sqrt{1-(k_1\theta+k_2)^2}\\ &\quad +2a\sqrt{1-(k_1\theta+k_2)^2}\big[(2c_1R+c_2)[-8+2(k_1\theta+k_2)^2]\\ &\quad -ac_1[(k_1\theta+k_2)+(k_1\theta+k_2)^3]\big]\\ &\quad +12a(2c_1R+c_2)(k_1\theta+k_2)\operatorname{acos}(k_1\theta+k_2)\\ &\quad +3[3a^2c_1+4R(c_1R+c_2)]\operatorname{asin}(k_1\theta+k_2)\Big]_{\theta_1}^{\theta_2}\end{aligned}$$

B

Appendix B: Code

B.1 *steadyDep.m*

```
function depFrac = steadyDep(D,W,R,a)
% This function approximates the results of the fully transient eulerDep.m
% using a fully steady method of deposition calculation. The same inputs and
% output are used. The simulation assumes a steady fluid flow at each time
% step, with the magnitude of the flow updated according to the transient
% flow function.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Initialize simulation parameters  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S = calcS(D,W,R);                % Stokes number
tRes = pi/2*R/W;                 % approximate particle residence time
t0 = [0 0];                      % initial time of limiting particles
y0 = [1-a/R 1+a/R];              % initial position of limiting particles
r = [0 0];                       % current radial position of limiting particles
% The radial position of the limiting (upper- and lower-most) particles at
% the pipe exit are determined using the bisection method.
for i = 1:2
    error = 1;
    tLower = 0;
    tUpper = 1.5*tRes;
    while error > 1E-4
        tMid = (tLower+tUpper)/2;
        [x,y] = AirySolnSteady(S,y0(i),y0(i),0,W/R*(tMid),1);
        error = abs(y);
        if y > 0
            tLower = tMid;
        else
            tUpper = tMid;
        end
    end
    r(i) = R*sqrt(x^2+y^2);
    t0(i) = tMid;
end
% The particle ellipse parameters are caculated and then used to determine
% the deposited area and area fraction.
b = (r(2)-r(1))/2;
h = (r(2)+r(1))/2 - R;
if h < a+b
    A = a*((a+b).*asin(sqrt(1 - h.^2./(a+b).^2))...
          - h.*sqrt((a+b).^2 - h.^2)./(a+b));
    depFrac = 1 - A/(pi*a*b);
else
    depFrac = 1;
end
```

B.2 *quasiDep.m*

```
function depFrac = quasiDep(D,W,R,a,T)
% This function approximates the results of the fully transient eulerDep.m
% using a quasi-steady method of deposition calculation. The same inputs and
% output are used. The simulation assumes a steady fluid flow at each time
% step, with the magnitude of the flow updated according to the transient
% flow function.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Define simulation parameters  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S = calcS(D,W,R);           % particle Stokes number
v0 = 0;                     % particle initial y-velocity [m/s]
ti = 0;                     % simulation start time [s]
tf = T/2;                   % simulation end time [s]
Ncv = 64;                   % number of pipe control volumes
Np = 0;                     % number of CVs penetrated by particles

% The transient component of the fluid velocity function is defined. Forms
% not in use are commented out.
%   A sine wave:
f = @(t0) sin(2*pi*t0/T); df = @(t0) 2*pi*R/(W*T)*cos(2*pi*t0/T);
%   The acceleration section of a triangle wave:
% f = @(t0) 4*t0/T; df = @(t0) 4*R/(T*W);
%   The deceleration section of a triangle wave:
% f = @(t0) 2 - 4*t0/T; df = @(t0) -4*R/(T*W);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Initialize variables  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Qdep = 0;                   % particles deposited
Qin = 0;                    % particles in
t = ti;                     % absolute simulation time [s]
n = 0;                      % iteration counter

%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Begin main loop  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%
while t < tf
    dt = pi*R/(4*Ncv*W*f(t));       % adaptive time step
    if dt > (tf - ti)/1E3
        dt = (tf - ti)/1E3;
    end
    % The deepest infiltrated CV is determined along with its angle.
    if Np < Ncv
        [xPs,yPs] = AirySolnFull(S,(R+a)/R,(R+a)/R*f(ti),v0,W/R*(t-ti),f(ti),df(ti));
        if min(yPs) > 0
            thetaMax = atan(xPs./yPs);
            Np = floor(2*Ncv*thetaMax/pi);
            thetaTarget = Np/Ncv*pi/2;
        else
            Np = Ncv;
            thetaTarget = pi/2;
        end
    end
    if Np > 0
        % The radial position of the lower-most particle at the CV boundary
        % identified above is determined using the bisection method.
        y0s = (R-a)/R;
        u0s = y0s*f(t);
        tLower = 0;
        tUpper = 1.2*pi/2*(R-a)/(W*(R-a)/R*f(t));

        error = 1;
        while error > 1E-4
            tMid = (tLower + tUpper)/2;
            [xs,ys] = AirySolnSteady(S,y0s,u0s,0,W/R*(tMid),f(t));
            if ys > 0
                theta = atan(xs/ys);
            else
                theta = pi/2 + atan(abs(ys)/xs);
            end
            if theta < thetaTarget
                tLower = tMid;
            else
                tUpper = tMid;
            end
            error = abs(thetaTarget - theta)/thetaTarget;
        end
        rBottom = R*sqrt(xs.^2 + ys.^2);

        % The radial position of the upper-most particle is determined.
        y0s = (R+a)/R;
        u0s = y0s*f(t);
        tLower = 0;
```

Appendix B: Code

```
        tUpper = 1.2*pi/2*(R+a)/(W*(R+a)/R*f(t));
        error = 1;
        while error > 1E-4
            tMid = (tLower + tUpper)/2;
            [xs,ys] = AirySolnSteady(S,y0s,u0s,0,W/R*(tMid),f(t));
            if ys > 0
                theta = atan(xs/ys);
            else
                theta = pi/2 + atan(abs(ys)/xs);
            end
            if theta < thetaTarget
                tLower = tMid;
            else
                tUpper = tMid;
            end
            error = abs(thetaTarget - theta)/thetaTarget;
        end
        rTop = R*sqrt(xs.^2 + ys.^2);
        b = (rTop - rBottom)/2;          % calculation of ellipse parameters
        h = (rTop + rBottom)/2 - R;
        % The deposited fraction of particles is calculated.
        if h < a+b
            areaOverlap = a*((a+b).*asin(sqrt(1 - h^2/(a+b)^2))...
                          - h*sqrt((a+b)^2 - h^2)/(a+b));
            depf = 1 - areaOverlap/(pi*a*b);
        else
            depf = 1;
        end
        dQin = pi*a^2*W*f(t);        % The particle deposition rate is simply
        dQdep = depf*dQin;           % the deposited fraction multiplied by
                                     % the inflow rate.
        Qdep = Qdep + dQdep*dt;          % increment deposition
        Qin = Qin + dQin*dt;             % increment inflow
    end
    t = t + dt;                  % increment time
    n = n + 1;                   % increment iteration counter
end
depFrac = Qdep/Qin;
```

B.3 *eulerDep.m*

```
function [depFrac,lossFrac] = eulerDep(D,W,R,a,T)
% This function calculates the deposition of solid particles flowing
% through a fluid-filled curved pipe with circular cross-section. The
% radius of curvature of the pipe is constant and the bend angle is 90
% degrees.
%
% The inputs of the function are:
%   D - particle diameter [m]
%   W - maximum fluid velocity at pipe axis [m/s]
%   R - pipe bend radius [m]
%   a - pipe cross-section radius [m]
%   T - flow period [s]
%
% The fluid flow magnitude is transient and varies linearly in the radial
% direction, making the velocity function w = W*r/R*f(t), where f(t) is
% some function of time describing the flow behaviour. f(t) has a magnitude
% ≤ 1 and is periodic, usually with a sine or triangle wave shape.
% Simulations are only run on the part of the flow cycle where velocity is
% positive, and can be split into the acceleration or deceleration segments
% of the flow, which have a duration of T/4.
%
% The outputs of the function are:
%   depFrac - the fraction of particles entering the pipe which were
%             deposited
%   lossFrac - the fraction of particles entering the pipe which were
%              unaccounted for (either lost or gained)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Define simulation parameters  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A = pi*a^2;                     % area of pipe cross-section [m^2]
S = calcS(D,W,R);               % particle Stokes number
v0 = 0;                         % particle initial y-velocity [m/s]
ti = 0;                         % simulation start time [s]
tf = T/2;                       % simulation end time [s]
Ncv = 64;                       % number of pipe control volumes
Np = 0;                         % number of CVs penetrated by particles
thetaCV = pi/2/Ncv;             % angle transcribed by each CV
% The transient component of the fluid velocity function is defined. Forms
% not in use are commented out.
%   A sine wave:
f = @(t0) sin(2*pi*t0/T); df = @(t0) 2*pi*R/(W*T)*cos(2*pi*t0/T);
%   The acceleration section of a triangle wave:
% f = @(t0) 4*t0/T; df = @(t0) 4*R/(T*W);
%   The deceleration section of a triangle wave:
% f = @(t0) 2 - 4*t0/T; df = @(t0) -4*R/(T*W);
% The particle initial x-velocity is defined.
%   Equal to the fluid velocity:
u0 = @(y0,t0) W*y0/R.*f(t0);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Initialize variables  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = ti;                         % simulation time [s]
n = 0;                          % iteration counter
Qin = 0;                        % particles in
Qout = 0;                       % particles out
Qdep = 0;                       % particles deposited
Qres = 0;                       % particles resident in the pipe
maxNp = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Begin main loop  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%
while t ≤ tf
    dt = pi*R/(4*Ncv*W*f(t));   % An adaptive time step is used to speed
    if dt > (tf-ti)/1E3         % up the simulation, with a maximum value
        dt = (tf-ti)/1E3;       % imposed to prevent over-stepping.
    end
    if Np < Ncv
        % The number of CVs fully penetrated by the particles is calculated.
        [xPs,yPs] = AirySolnFull(S,(R+a)/R,u0(R+a,ti)/W,v0/W,W/R*(t-ti),f(ti),df(ti));
        if yPs > 0
            thetaMax = atan(xPs/yPs);
            Np = floor(2*Ncv*thetaMax/pi);
        else
            Np = Ncv;
        end
        % Once the first CV has been penetrated, the injection times t0 of the
        % lowermost and uppermost particles at each penetrated CV boundary
```

Appendix B: Code

```
    % are calculated by the subfunction t0solve. These are the
    % particles with initial positions y0 = R +-a; quantities
    % pertaining to the lower particles are denoted by an L, and those
    % of the upper particles by a U. These calculations appear twice
    % because different inputs are used once the pipe is fully
    % infiltrated.
    if Np > 0
        [t0L,rL] = t0solve(ti,R-a,Np,@AirySolnFull,f,df,u0,v0,S,R,W,t,Ncv);
        [t0U,rU] = t0solve(ti,R+a,Np,@AirySolnFull,f,df,u0,v0,S,R,W,t,Ncv);
    end
else
    [t0L,rL] = t0solve(t0L(Ncv),R-a,Np,@AirySolnFull,f,df,u0,v0,S,R,W,t,Ncv);
    [t0U,rU] = t0solve(t0U(Ncv),R+a,Np,@AirySolnFull,f,df,u0,v0,S,R,W,t,Ncv);
end
if Np > 0
    % The major axis radius b and center offset distance h of the
    % particle ellipse are calculated.
    b = (rU - rL)/2;
    h = (rU + rL)/2 - R;

    % A mask vector is created to negate the effects of any CV in which
    % the particle ellipse is fully deposited.
    mask = ((a+b-h) + abs(a+b-h))./(2*abs(a+b-h));
    for i = 2:Np
        mask(i) = min([mask(i) mask(i-1)]);
    end
    % The deposition within each CV depends on the line of intersection
    % between the particle ellipse and the pipe circle at each CV
    % boundary. Specifically, the initial conditions of the highest and
    % lowest radial points on this line are required, which are
    % calculated here.
    y0mn = R - a./b.*(h - a*h./(a+b));
    y0mx = R + a./b.*(a-h);
    t0mid = (t0U+t0L)/2;
    t0mn = t0mid - (t0mid-t0L).*(h - a*h./(a+b))./b;
    t0mx = t0mid + (t0mid-t0L).*(a-h)./b;

    % Two positions one time-step apart are calculated for each of the
    % points mentioned above in order to calculate the particle radial
    % velocity at those locations.
    [xI,yI] = AirySolnFull(S,y0mn/R,u0(y0mn,t0mn)/W,...
                           v0/W,W/R*(t-t0mn),f(t0mn),df(t0mn));
    [xF,yF] = AirySolnFull(S,y0mn/R,u0(y0mn,t0mn)/W,...
                           v0/W,W/R*(t+0.01*dt-t0mn),f(t0mn),df(t0mn));
    [xI2,yI2] = AirySolnFull(S,y0mx/R,u0(y0mx,t0mx)/W,...
                           v0/W,W/R*(t-t0mx),f(t0mx),df(t0mx));
    [xF2,yF2] = AirySolnFull(S,y0mx/R,u0(y0mx,t0mx)/W,...
                           v0/W,W/R*(t+0.01*dt-t0mx),f(t0mx),df(t0mx));

    rI = R*sqrt(xI.^2 + yI.^2);      % Initial and final radial positions
    rF = R*sqrt(xF.^2 + yF.^2);      % for the lower points...
    rI2 = R*sqrt(xI2.^2 + yI2.^2);   % ...and for the upper points.
    rF2 = R*sqrt(xF2.^2 + yF2.^2);
    vrmax = (rF-rI)/(0.01*dt);       % upper and lower radial velocities
    vrmin = (rF2-rI2)/(0.01*dt);
    % The subfunction calcQdep is used to calculate particle deposition
    % rate for each infiltrated CV.
    dQdep = calcQdep(a,b,R,h,vrmin,vrmax,pi/2/Ncv);
end
% Particles in:
dQin = u0(R,t)*A;
Qin = Qin + dQin*dt;

if Np > 0
    % Particle concentration is calculated for each CV.
    Ccv = concSolve(y0min,t0mn,@AirySolnFull,f,df,u0,v0,S,R,W,t,a);
    Cmid = ([1 Ccv(1:Np-1)] + Ccv)/2;

    % Particles resident:
    % Boundary values of geometric quantities are averaged to get the
    % values for each CV.
    Ares = a*((a+b).*asin(sqrt(1 - h.^2./(a+b).^2))...
               - h.*sqrt((a+b).^2 - h.^2)./(a+b));
    Amid = ([pi*a^2 Ares(1:Np-1)] + Ares.*mask)/2;
    c = centroid(a,b,h,R);
    cmid = ([R c(1:Np-1)] + c)/2;

    Qres = sum(Amid.*cmid*thetaCV.*Cmid.*mask);

    % Particles deposited:
    Qdep = Qdep + sum(dQdep.*Cmid.*mask)*dt;
end
if Np == Ncv
    % Particles out:
    y0o = [(R-a) y0max(Ncv)];
    t0o = [t0L(Ncv) t0mx(Ncv)];
```

Appendix B: Code

```
        [xE,yE] = AirySolnFull(S,y0o/R,u0(y0o,t0o)/W,...
                               v0/W,W/R*(t-t0o),f(t0o),df(t0o));
        [xE2,yE2] = AirySolnFull(S,y0o/R,u0(y0o,t0o)/W,...
                               v0/W,W/R*(t+0.01*dt-t0o),f(t0o),df(t0o));
        vE = abs(R*(yE2-yE))/(0.01*dt);
        VC = polyfit([rL(Ncv) R+a],vE*Ccv(Ncv),1);
        K1 = VC(1)*R + VC(2);
        K2 = VC(1)*(R+h(Np)) + VC(2);
        K3 = a + b(Np);
        dQout = pi*a^2/2*K1 + pi*a*b(Np)/2*K2 ...
                - a^2*(sqrt(K3^2-h(Np)^2)/(3*K3)...
                        *(2*a*VC(1)*((h(Np)/K3)^2-1) + 3*h(Np)*K1/K3)...
                       + K1*atan(h(Np)/sqrt(K3^2-h(Np)^2)))...
                - a*b(Np)*(sqrt(K3^2-h(Np)^2)/(3*K3)...
                        *(2*b(Np)*VC(1)*(1-(h(Np)/K3)^2) + 3*h(Np)*K2/K3) ...
                       + K2*atan(h(Np)/sqrt(K3^2-h(Np)^2)));
        Qout = Qout + dQout*dt*mask(Ncv);
        maxNp = max(mask(Ncv),maxNp);
    end
    t = t + dt;                 % increment time and iteration counter
    n = n + 1;
end
if maxNp == 0                   % If no particles exit the pipe, complete
    Qdep = Qin;                 % deposition is enforced.
end
depFrac = Qdep/Qin;                         % deposited fraction
lossFrac = (Qin - (Qdep+Qout))/Qin;     % mass loss fraction

function [t0LT,rLT] = t0solve(t0b,y0,Np,pathFunction,f,df,u0,v0,S,R,W,t,Ncv)
% This subfunction determines the start times of particles with initial
% positions y0 and current positions along the radial lines of the
% (penetrated) CV boundaries. The bisection method is used to find the
% roots.
t0LT = zeros(1,Np);
rLT = zeros(1,Np);
for i = Np:-1:1
    n = 0;
    t0a = t;
    phi = 1;
    while abs(phi) > 1E-4 && n < 1000
        t0c = (t0a + t0b)/2;
        [xPs,yPs] = pathFunction(S,y0/R,u0(y0,t0c)/W,...
                               v0/W,W/R*(t-t0c),f(t0c),df(t0c));
        phi = atan(yPs/abs(xPs)) - pi/2*(1-i/Ncv);
        if phi > 0
            t0a = t0c;
        else
            t0b = t0c;
        end
        n = n + 1;
    end
    t0b = t0c;
    t0LT(i) = t0c;
    rLT(i) = R*sqrt(xPs^2 + yPs^2);
    if i > Ncv/2 && i < Np
        if rLT(i) > rLT(i+1)
            t = t;
        end
    end
end

function [C,s0,xPlot,yPlot] = concSolve(y0,t0,pathFunction,f,df,u0,v0,S,R,W,t,a)
% This subfunction determines the particle concentration at the point with
% initial conditions (y0,t0) at time t.
[x,y] = pathFunction(S,(y0)/R,u0(y0,t0)/W,v0/W,W/R*(t-t0),f(t0),df(t0));
x = R*x; y = R*y;
s0 = 1E-2*a;
dt = s0./u0(y0,t0);
[xW,yW] = pathFunction(S,y0/R,u0(y0,t0+dt)/W,...
                       v0/W,W/R*(t-(t0+dt)),f(t0+dt),df(t0+dt));
xW = R*xW; yW = R*yW;
[xN,yN] = pathFunction(S,(y0+s0)/R,u0(y0+s0,t0)/W,...
                       v0/W,W/R*(t-t0),f(t0),df(t0));
xN = R*xN; yN = R*yN;
s1 = sqrt((x-xW).^2 + (y-yW).^2);
s2 = sqrt((x-xN).^2 + (y-yN).^2);
s3 = sqrt((xW-xN).^2 + (yW-yN).^2);
phi = acos((s1.^2 + s2.^2 - s3.^2)./(2*s1.*s2));
C = s0.^2./(s1.*s2.*sin(phi));
xPlot = [xW(1) x(1) xN(1)];     % These coordinates are for plotting the
yPlot = [yW(1) y(1) yN(1)];     % transformed area element.
```

Appendix B: Code

```
function dQdt = calcQdep(a,b,R,h,vrmin,vrmax,theta)
% This subfunction calculates particle deposition using a geometrically
% derived forumla.
n = length(b);
c1 = zeros(1,n+1);
c2 = zeros(1,n+1);
rmin = a*h./(a+n) + R;
c1(2:n+1) = (vrmax-vrmin)./(R+a-rmin);
c2(2:n+1) = vrmin - rmin.*(vrmax-vrmin)./(R+a-rmin);
c1 = 0.5*(c1(1:n)+c1(2:n+1));
c2 = 0.5*(c2(1:n)+c2(2:n+1));
hab1 = [0 h(1:n-1)./(a+b(1:n-1))];
hab2 = h./(a+b);
k1 = 1/theta*(hab2 - hab1);
k2 = hab1;
p = c1*R+c2;
q = 2*c1*R+c2;
s = k1*theta+k2;
dQdt = a./(12*k1).*(3*(3*a^2*c1 + 4*R*p).*s.*sqrt(1-s.^2)...
        + 2*a*sqrt(1-s.^2).*(-8*q - a*c1.*s + 2*q.*s.^2 + a*c1.*s.^3)...
        + 12*a*q.*s.*acos(s) + 3*(3*a^2*c1 + 4*R*p).*asin(s)...
        - (3*(3*a^2*c1 + 4*R*p).*k2.*sqrt(1-k2.^2)...
        + 2*a*sqrt(1-k2.^2).*(-8*q - a*c1.*k2 + 2*q.*k2.^2 + a*c1.*k2.^3)...
        + 12*a*q.*k2.*acos(k2) + 3*(3*a^2*c1 + 4*R*p).*asin(k2)));
dQdt = (dQdt + abs(dQdt))/2;

function c = centroid(a,b,h,R)
% This subfunction calculates the centroid of the overlap area of a circle
% and ellipse using a derived formula.
c = ((a+b).*(3*a*pi*R + 3*b*pi.*(h+R)...
 - 2*(a*sqrt(1 - h.^2./(a+b).^2).*(a*(-2 + (2*h.^2)./(a+b).^2)...
 + (3*h*R)./(a+b)) + 3*a*R*atan(h./sqrt(a^2 + 2*a*b + b.^2 - h.^2)))...
 + 2*(-b.*sqrt(1 - h.^2./(a+b).^2).*(2*b + (h.*(3*a*(h+R)...
 + b.*(h+3*R)))./(a+b).^2) - 3*b.*(h+R).*atan(h./sqrt(a^2 + 2*a*b + b.^2 - h.^2)))))...
 ./(3*(-2*h.*sqrt(a^2 + 2*a*b + b.^2 - h.^2) + (a+b).^2*pi...
 - 2*(a+b).^2.*atan(h./sqrt(a^2 + 2*a*b + b.^2 - h.^2))));
```

B.4 *chengDep.m*

```
function depFrac = chengDep(S,R0)
K = 4*S/R0;
A = ((1 + (1+K.^2).^0.5)/2).^0.5;
B = K./(2*A);
tz = zeros(1,length(S));
for i = 1:length(S)
    t = 0;
    dt = 1;
    error = 1;
    e1 = cos(B(i)*t).*(cosh(A(i)*t) + A(i)^3/(A(i)^2+B(i)^2)*sinh(A(i)*t));
    e2 = B(i)^3/(A(i)^2+B(i)^2)*sin(B(i)*t).*cosh(A(i)*t);
    direction = sign(e1+1E-8-e2);
    while error > 0.001 || changed == 0
        t = t + dt;
        e1 = cos(B(i)*t).*(cosh(A(i)*t) + A(i)^3/(A(i)^2+B(i)^2)*sinh(A(i)*t));
        e2 = B(i)^3/(A(i)^2+B(i)^2)*sin(B(i)*t).*cosh(A(i)*t);
        if direction ≠ sign(e1-e2)
            changed = 1;
            direction = -direction;
            dt = -0.1*dt;
            error = abs((e1-e2)/e1);
        end
    end
    tz(i) = t;
end
n = sin(B.*tz) .* (sinh(A.*tz) + A.^3/(A.^2+B.^2).*cosh(A.*tz)) ...
    + B.^3/(A.^2+B.^2).*cos(B.*tz).*sinh(A.*tz);
z = (1 - (R0^2*(n - exp(tz)).^2)./(n + exp(tz)).^2).^0.5;
depFrac = 1 - 1/(pi*R0) * ((exp(2*tz)./n.^2 - 1).*((R0^2+1)*z - z.^3/3) ...
                        + R0*(exp(2*tz)./n.^2 + 1).*(z.*sqrt(1-z.^2) + asin(z)));
```