

University of Nevada, Reno

Dynamics of Motile Microbes and Model Development

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Hydrogeology

by

Xueke Yang

Dr. Rishi Parashar/ Thesis Advisor

August 2017

Copyrights by Xueke Yang 2017

All Rights Reserved



THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

Xueke Yang

Entitled

Dynamics of Motile Microbes and Model Development

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Rishi Parashar, Advisor

Rina Schumer, Committee Member

Chen Li, Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

August, 2017

Abstract

Accurate numerical models of microbial transport are needed to support design and evaluation of bioremediation implementations. The characteristics associated with the motion dynamics of motile bacteria is thought to play an important role in bioremediation processes. By stimulating the activity of natural microorganisms it is possible to manipulate the redox state of contaminants and greatly decrease their solubility, rendering them immobile and therefore reducing the risk of human exposure. Traditional transport models, such as Advection-Diffusion Equation (ADE), are generally found to be inadequate to describe the transport of bacteria in groundwater as bacterial motility often exhibits non-Fickian properties owing to the complex nature of their run-and-tumble motion and chemotaxis behavior. In this study, *Geobacter* and *Pelosinus*, which are motile species important to bioremediation of certain heavy metals, are used in micromodel experiments for microscopic examination of their motility characteristics. Experiments to observe two-dimensional motion in unobstructed medium were conducted at the Environmental Molecular Sciences Laboratory (EMSL) of the Pacific Northwest National Laboratory (PNNL). EMSL's Cell Isolation and Systems Analysis (CISA) facility provides advanced capabilities for microbial cell isolation and microscopic studies. Statistical properties of bacterial transport are measured from recorded trajectories. Individual tracks on the order of several seconds to a few minutes in duration are analyzed to provide information on 1) the length (distance in microns) of microbial runs, 2) velocity distributions along individual trajectories, and 3) the angle between the directions of sequential runs. A Continuous Time Random Walk (CTRW) model to simulate transport and predict breakthrough plots of motile microbes is developed based on the investigated statistical properties of their motion dynamics. The results of the CTRW model are compared with the ADE model and experimentally obtained trajectories.

Acknowledgements

I would first like to thank my thesis advisor Dr. Rishi Parashar of the Division of Hydrologic Sciences at Desert Research Institute. Dr. Parashar is not only a great professor who guided me in my research, but also a great tutor who provided plenty of help outside of my research project. As a group partner and mentor, Dr. Nicole Sund helped me tremendously in coding and thesis editing.

I would also like to thank Dr. Rina Schumer and Dr. Li Chen for serving on my thesis committee and for providing valuable comments.

This work would not have been possible without the huge amount of help provided by PNNL scientists Mr. Andy Plymale, Dr. Dehong Hu, and Dr. Ryan Kelly during the experiments and data collection stage of the work.

Table of Contents

List of Tables	8
1. Introduction.....	1
1.1 Background	1
1.2 Hypotheses, Objectives, and Assumptions	7
1.3 Scope of Work.....	9
2. Transport Models	11
2.1 The Advection Dispersion Equation	11
2.2 Alternative Models	12
2.3 CTRW Model.....	14
3. Methodology.....	16
3.1 Bacteria Cultivation.....	17
3.2 Laboratory Experiments.....	18
3.3 Code Developments	22
3.3.1 Trajectories Extraction	23
3.3.2 Statistics Analysis.....	27
3.3.3 Model Developments and Breakthrough Curves.....	28
4. Results.....	32
4.1 Jump Length and Waiting Time.....	32
4.2 Real Path	32
4.3 Model Comparison.....	40
5. Discussions and Conclusions.....	46
References.....	53
Appendix: MATLAB Code	58

A1. Video files extraction and processing section	58
A1.1 Main loop.....	58
A1.2 Position determination function.....	60
A1.3 Path determination function.....	63
A1.4 Jump length, waiting time and statistics calculation and output function	67
A2. Data merge and interpolation section.....	70
A2.1 Jump length and waiting time merge.....	70
A2.2 Statistics interpolation	71
A3. Model stimulations, normalization and graphing.....	73
A3.1 Real path	73
A3.2 ADE	76
A3.3 CTRW.....	77
A3.4 Normalization	82

List of Figures

- Figure 1:** Depiction of flagella driven motion and run-and-tumble trajectories of motile microbes (adapted from Watari and Larson, 2010) 6
- Figure 2:** Comparison of experimentally observed bacterial tracks and trace of Levy motion (adapted from Berg, 2000)..... 6
- Figure 3:** Continuous Time Random Walk model sketch. The trajectory of bacteria is 1-A-2-B, where jump-1 transports it from original location to A, waiting for a time; then jump-2 transports it from A to B followed by another waiting time (adapted from L. Pickman, M. S. Thesis, UNR, 2015)..... 10
- Figure 4:** Flowchart of main Procedures 16
- Figure 5:** Experimental cell sketch. The black dots at left-end and right-end represent inlet and outlet. The main body of chamber is consisted of two parts: pillar area (hollow dots) and open area (black). 18
- Figure 6:** Fluorescent *Geobacter* under microscope and in presence of periodic pore structure. White dots show the location of individual *Geobacter* cells. 21
- Figure 7:** Flowcharts of extracting positions and paths from video files. Determination of bacteria position is shown in (a) where σ = standard deviation, d_{\max} = high end of bacteria diameter range, d_{\min} = low end of bacteria diameter range. In (b), the process of connecting positions to construct paths is demonstrated. 25
- Figure 8:** Projected bacteria points in a video frame. (a) location of bacteria after application of temporal and spatial filter to separate the background noise. (b) The white

circles are recognized as moving bacteria in a video frame after application of a size and distance filter. (c) Each red line represents one bacteria trajectory extracted from a video file. (d) A trajectory of a bacterium in units of pixel. 26

Figure 9: Illustration of control plane settings and microbial trajectories in random directions. The concentric circles represent control planes located at 10, 20, 30, 40, 50, and 60 microns away from point source (0,0). For *Pelosinus*, breakthrough curves are obtained for all 6 control plane, and for *Geobacter*, only 4 nearest control planes are used. All trajectories start from the origin and their first passage time are recorded for individual control planes to obtain breakthrough plots.....31

Figure 10: (a) Jump length and (b) waiting time probability densities s for *Geobacter* and *Pelosinus*. 33

Figure 11: First- arrival time curves of *Geobacter*.(a), (b), (c), (d), (e), and (f) corresponds to $L = 10, 20, 30, 40, 50, 60$ microns. 34

Figure 12: First- arrival time curves of *Pelosinus*.(a), (b), (c), (d), (e), (f) corresponds to $L = 10, 20, 30, 40, 50, 60$ microns. 35

Figure 13: Normalized first-arrival - time plots of *Geobacter* and *Pelosinus*. 37

Figure 14: δ^2 over time for *Geobacter* and *Pelosinus*. 38

Figure 15: Mean Displacements of (a) *Geobacter* over time, and (b) *Pelosinus*..... 39

Figure 16: Comparison of Models for *Geobacter* when $t = 1000s$. (a), (b), (c), (d) corresponds to $L = 10, 20, 30, 40$ microns. 41

Figure 17: Comparison of Models and Real Path for *Pelosinus* when $t = 2500s$.(a), (b), (c), (d), (e), (f) corresponds to $L = 10, 20, 30, 40, 50, 60$ microns. 43

Figure 18: Jump length and waiting time correlations at step n and step $n+1$ for

Geobacter and *Pelosinus*. 49

List of Tables

Table 1: Mean and Standard Deviations of each model for <i>Geobacter</i>	44
Table 2: Mean and Standard Deviations of each model for <i>Pelosinus</i>	45

1. Introduction

1.1 Background

Groundwater contamination is a serious challenge around the world as a result of agricultural, industrial, domestic, and nuclear activities. Site specific complexities and challenges are evaluated to strategies remediation techniques focusing on biological, chemical, and/or physical treatment of contaminated water. In-situ bioremediation is an efficient and affordable method to treat a variety of contaminants including various heavy metals (Juwarkar et al., 2010). Heavy metals, however, present unique challenges for remediation because they cannot be degraded to innocuous substances. Additionally, metals in subsurface environments are usually unreachable for treatment by the many methods available to decontaminate surface soils.

The *in situ* removal of toxic metals involves both hydrological and complex biogeochemical processes. Remediation of metal and radionuclide contaminants in soil and groundwater systems is challenging because of their strong chemical interactions with mineral surfaces and low concentration limits for human health requirements. One of the major emphases of metal bioremediation research is on the microbially-mediated reduction of metals and radionuclides. Metal-reducing microorganisms play a critical role in the mobility of several groundwater contaminants including metals such as chromium (e.g., Chirwa and Wang, 2000) and radionuclides such as uranium (e.g., Anderson et al., 2003) and technetium (e.g., Lloyd et al., 2000). Oxidized states of heavy metals, such as Cr(VI), Tc(VII), and U(VI), are generally toxic and mobile (with high solubility); while the reduced states of heavy metals, such as Cr(IV), Tc(IV), and U(IV), are innocuous and immobile (with low solubility, or tend to form minerals). By stimulating the activity of

natural microorganisms it is possible to manipulate the redox state of contaminants and greatly decrease their solubility, rendering them immobile and therefore reducing the risk of human exposure.

Because microbes are living organisms, their transport in porous media is far more complex than abiotic colloids. Zhao et al. (2011) developed a numerical model of uranium bioremediation that includes terms describing the bulk movement (passive advection) of planktonic bacteria as well as their attachment to solid surfaces (transition from planktonic to attached phase), and demonstrated through sensitivity analyses that these processes play a significant role in determining the overall rate of contaminant reduction. Field experiments have observed a strong correlation between the rate of metal reduction and the abundance of planktonic (free-swimming) cells (Vrionis et al., 2005).

It is also known that many metal reducing microorganisms exhibit flagellar motility and some can temporarily store electrons on the external surface of cytochromes, thus allowing continued oxidation of soluble electron donors during a motile phase. The net result of these processes imparts a self-propagating character to a microbe. However, most applications of numerical models of uranium and other metal contaminant bioremediation (e.g., Fang et al., 2009; Yabusaki et al., 2011) treat microbial biomass as an immobile constituent, and do not consider the effects of microbial motility or transport on the efficacy of bioremediation practices.

The motile microbes provide an example of transport that is both nonlinear and random (Selmeczi et al., 2005; Taktikos et al., 2013; Gharasoo et al., 2014). The motility pattern has been observed to exhibit enhanced diffusion (Kim and Breuer, 2004; Licata et al., 2016) with mean-square displacement growing faster than linear in time. Recently,

microfluidics devices have been successfully used in experiments to quantify microbial chemotaxis as reviewed by Wu et al. (2013). Notably, Ford and co-workers (e.g., Kusy and Ford, 2007; Wang et al., 2012) observed non-diffusive motility effects and incorporated them into cell-scale and porous media models. However, the approach in most of these studies has been to include motility effect by means of effective parameters in generalized conventional transport equations (Bradford et al., 2014):

$$\frac{\partial \theta C}{\partial t} = \frac{\partial}{\partial z} \left(\theta D \frac{\partial C}{\partial z} \right) - \frac{\partial qC}{\partial z} - E_{sw} - E_{aw} + B_w \quad (1)$$

where z [L] is distance in direction of flow, C [NL^{-3}] is the concentration of microbe in aqueous phase, q [LT^{-1}] is Darcy velocity, θ [$\text{L}^3 \text{L}^{-3}$] is the volumetric water content, D [$\text{L}^2 \text{T}^{-1}$] is the hydrodynamic dispersion coefficient for microbes, B_w [$\text{NL}^{-3} \text{T}^{-1}$] is the rate of survival of the microbes in the water phase, and E_{sw} [$\text{NL}^{-3} \text{T}^{-1}$] and E_{aw} [$\text{NL}^{-3} \text{T}^{-1}$] are the exchange terms between the aqueous phase and surface or air phase. The above expression is an extension of the classical Advection-Diffusion Equation (ADE) where the entity undergoing transport is thought to exhibit Brownian motion resulting in a plume variance that grows linearly with time. Several researchers have used the classical ADE approach to study microbial transport processes (Ford and Harvey, 2007; Wang and Ford, 2009; Liu et al., 2011; Singh and Olson, 2011; Singh and Olson, 2012). The application of a classical ADE-based model to study bacterial transport in many studies have resulted in unrealistic values of fitted parameters such as values of retardation coefficients of less than 1 (Wang and Ford, 2009; Liu et al., 2011). Though modifications to ADE-based models have been developed to better address the problem of microbial transport, Cates (2012) indicates that the food-chasing motion of bacteria represents a far-from-equilibrium process. Murphy and Ginn (2000) and Ginn *et al.*

(2002) provide reviews and discussion of microbial transport processes and approaches to modeling microbial transport. They note that there is generally lack of accepted quantitative treatment for biologically driven active attachment/detachment, and the process of chemotaxis (movement of a motile cell in a direction corresponding to increasing or decreasing concentration of certain chemical) lacks a validated model for use in porous media environments. Though chemotaxis increases bacterial migration in response to microscale gradients and can enhance bacterial residence time near a nutrient source, well controlled studies on the role of motility in porous media transport are scarce (Tufenkji, 2007). A question thus arises: what advancements in bioremediation and related fields can be realized by conducting controlled experiments to study microbial motility and by developing robust mathematical and numerical framework to model the observed behavior?

Past research has shown that in stagnant fluid microbial motility shows ‘run-and-tumble’ nature (see Fig. 1) with flagellar movement resulting in trajectories of bacteria resembling the typical trace of a Levy flight (Vishwanathan et al., 1999; Berg, 2000).

Figure 2 shows the comparison of long flights interspersed with shorter jumps of experimentally obtained bacterial tracks and trace of typical Levy motion random walks.

The Levy flight patterned motility dynamics of microbes exhibits a super-diffusion feature as the mean-square displacement growing faster than linear in time. Analysis of breakthrough curves of motile bacteria has also shown characteristics of super-diffusion (Sherwood et al., 2003; Kim and Breuer, 2004).

Several researchers have studied the run and tumbling motion of motile microbes to quantify their transport. Costanzo et al. (2012) developed a 2-D model to stimulate the

motion of self-propelling bacteria in micro-channel flow. They investigate the model dependence on particle velocity profiles, particle flux, and the particle density. Licata et al. (2016) developed model based on cell's intrinsic tumbling mechanism as a biophysical process giving rise to directional reorientation. Their model suggests the tumbling process can be characterized by pore size, tumbling frequency, cell speed, and directional persistence. Some researchers are using microscopy to provide insights of cells motility. Selmeczi et al. (2005) mathematically describe movement of human dermal cells as an Ornstein-Uhlenbeck (a stochastic process describing the velocity of Brownian particles under the influence of friction) (OU) process based on individual cell trajectories. Davis et al. (2011) use microscopy to track motility of *Pseudomonas putida* *KT2440*, which can degrade aromatic compounds and are considered to be a primary candidate for bioremediation projects. Microscopic videos have also been used to track the transverse mixing enhancement due to bacterial random motility at the microscale (Singh and Olsen, 2011). This thesis presents details of a series of experiments conducted with microfluidic devices using motile bacteria to quantify the fundamental character of bacterial motions during active swimming. These experiments form the basis for development and testing of new computational models to simulate microbial transport.

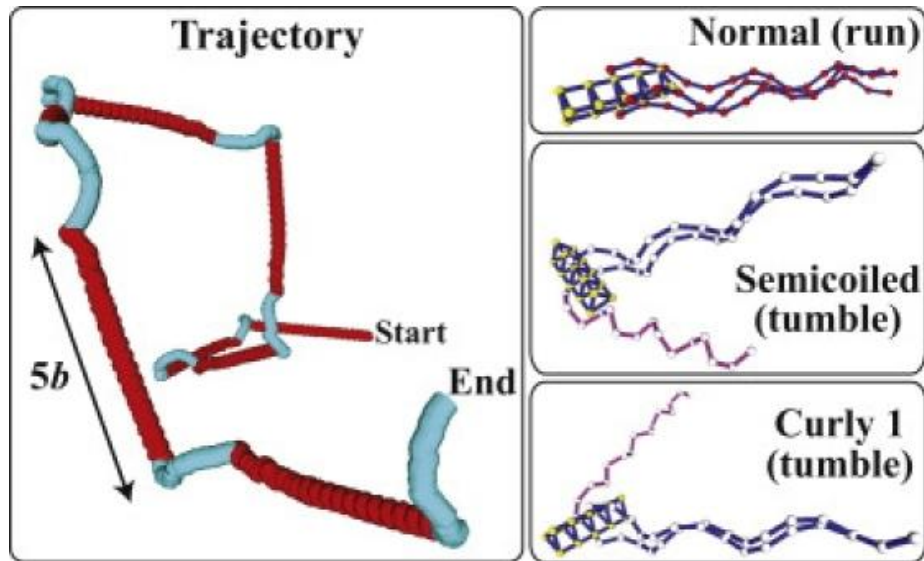


Figure 1: Depiction of flagella driven motion and run-and-tumble trajectories of motile microbes (adapted from Watari and Larson, 2010)



Figure 2: Comparison of experimentally observed bacterial tracks and trace of Levy motion (adapted from Berg, 2000).

1.2 Hypotheses, Objectives, and Assumptions

This work is focused on developing a mechanistic understanding of the dynamics of microbial motion. A sequence of experiments was conducted at the Environmental Molecular Sciences Laboratory (EMSL) of the Pacific Northwest National Laboratory (PNNL) to record the movement trajectories of selected microorganisms and to quantify their motion patterns in unobstructed medium and in presence of simple pore network structure. By examining the suitability of generalized random walk transport processes in modeling the movement of motile microbes, the project aims at addressing role of diffusive transport processes for migrating microbes at a fundamental level. We hypothesize that inclusion of mechanistic description of microbial motility characteristics will significantly impact model predictions of bacterial transport.

The experimental data of microbial motion was analyzed to extract various statistical attributes of microbial motion. Individual tracks on the order of several seconds to a few minutes in duration are analyzed to provide information on 1) the length (distance in microns) of microbial runs, 2) velocity distributions along individual trajectories, and 3) the angle between the directions of sequential runs. The distributional properties of these microbial motion attributes are investigated, and are further used to develop a mathematical model for simulating microbial transport based on a Continuous Time Random Walk (CTRW) framework. The CTRW modeling approach examines the behavior of an ensemble of particles undergoing instantaneous jumps of varied lengths followed by randomly distributed waiting time periods.

The research goal of analyzing bacterial motion dynamics and developing a random-walk based model to predict their transport is achieved by completing the following tasks:

- 1) Record bacteria movements in small micro-model chambers under microscope in laboratory. Magnification factor of microscopes and other attributes of the experiments are chosen based on the size of species and their average speed. For fluorescence species, the trajectories are also recorded under laser
- 2) Extract trajectories and analyze features of statistical distributions of microbial motion.
- 3) Analyze the trajectory data to examine the extent of non-Fickian behavior.
- 4) Develop a CTRW model of bacteria transport using the statistical distributions obtained from previous analysis.
- 5) Compare and evaluate the CTRW model against the raw experimental data and a simpler predictive model based on an ADE approach.

To above tasks are carried out under the assumptions that:

- 1) The comparatively small depth in the vertical direction of the micro-models allows for relatively accurate experimental observation of microbial trajectories in 2D.
- 2) The microbial solution is adequately diluted such that the interaction between microbes is negligible;
- 3) Hundreds of trajectories are sufficient to describe the statistical nature of observed quantities including the tailing features.
- 4) Sealing of micro-models after the injection of bacterial solution, prevented entry of air or any pressure difference to cause a background flow in the devices.

1.3 Scope of Work

The following chapters of this thesis contains a brief summary of some particle transport models, the methodology adopted for experimental and numerical study, results of the model development, discussions of the results, and conclusions. The work presented can be viewed as a prelude to the development and testing of new models of microbial transport during bioremediation. The choice of using a Continuous Time Random Walk (CTRW) approach to model the transport behavior of motile microbes is a first step at developing mechanistic models honoring the fundamental features of microbial motion. In CTRW models, statistical properties of jump length and waiting time are evaluated to simulate motion of a random walker from one site to another, followed by a waiting period on a site (see Fig. 3). More sophisticated random-walk models can be developed by examining correlation structure of transport processes. The scope of this thesis is however limited to development of simpler CTRW models and comparing their performance with conventional ADE-based approaches.

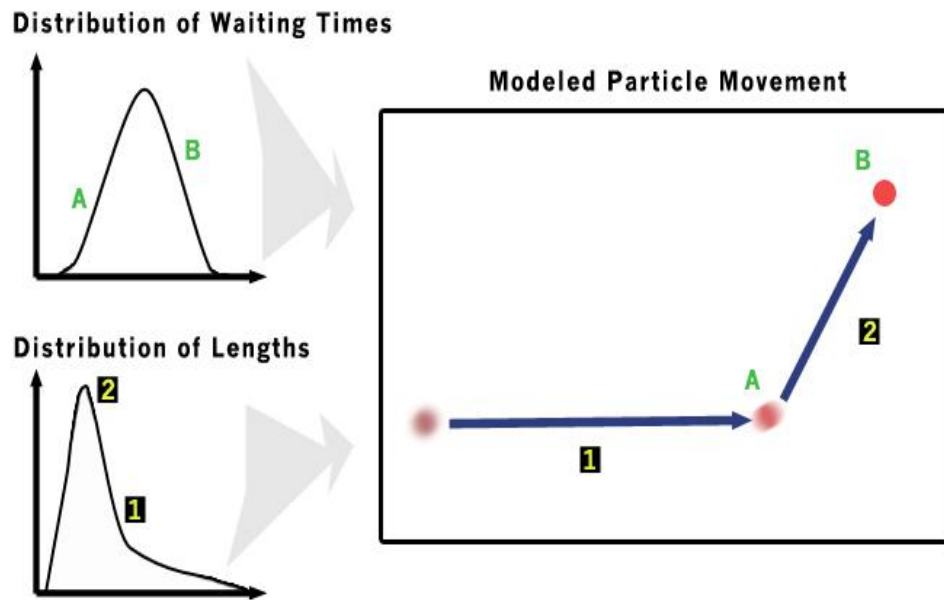


Figure 3: Continuous Time Random Walk model sketch. The trajectory of bacteria is 1-A-2-B, where jump-1 transports it from original location to A, waiting for a time; then jump-2 transports it from A to B followed by another waiting time (adapted from L. Pickman, M. S. Thesis, UNR, 2015).

2. Transport Models

2.1 The Advection Dispersion Equation

The Advection-Dispersion Equation (ADE) is a classical equation for studying transport of solute and contaminant and is widely used in areas of subsurface hydrology and other scientific disciplines (Bear 1972; Fetter 1993). The advection component of the equation accounts for bulk movement of particles with the advecting fluid (water), and the dispersion component of the equation describes the spreading of migrating plume of particles because of molecular diffusion and tortuosity in transport pathways. The general form of ADE can be expressed as:

$$\frac{\partial C}{\partial t} = -\nabla \cdot (\mathbf{v}C) + \nabla \cdot (D\nabla C) \quad (2)$$

where C is concentration of particles, \mathbf{v} is flow velocity, and D is dispersion tensor. At the foundation of ADE lies the assumption that the variance of migrating particles grows linearly with time, a characteristic of Fickian diffusion. In many circumstances however, the structure of the porous media (e.g., high degree of heterogeneity) or the inherent dynamics of the particles itself can either suppress or enhance the rate of diffusion to give rise to sub-Fickian or super-Fickian phenomenon. With its self-propelling motion characteristics, it is intuitive to expect bacteria to exhibit non-Fickian behavior. All microbial species are however not alike in their motion dynamics and it is possible that the growth of variance with respect to time may span a wide spectrum for various microbial species. This thesis focuses on two metal-reducing species, namely *Geobacter* and *Pelosinus*, and explores the suitability of an ADE to model their transport. For the

class of microbial species that doesn't conform well to modeling using an ADE approach, more sophisticated methods built using the statistical features of microbial run-and-tumble behavior can possibly lead to the development of improved predictive tools. A brief overview of alternative models available to model non-Fickian transport is presented followed by description of Continuous Time Random Walk (CTRW) Model --- a modeling approach selected in this work as a first step to providing a framework for studying bacterial transport processes.

2.2 Alternative Models

In transport scenarios where long range correlations are present (so that individual particles moving in a direction or resting are likely to continue the same for a long time), or particle motion consists of non-identical displacement that can be fitted to a PDF with non-finite mean or variance (possibly because of arbitrarily large displacement occurring with a certain frequency), the migrating plume of particles in ensemble exhibits either sub- or super-Fickian diffusion. Eulerian models such as Fractional Advection-Diffusion Equation (FADE) provide a generalized tool to study these kinds of systems by employing the concepts of Levy motion and Levy distributions (Benson et. al., 2001; Schumer et. al., 2001). Brownian motion and Gaussian distributions, which forms the basis of classical ADE, can be thought as a subset of the more general Levy motion and distributions. The FADE approach is well-suited to study transport scenarios where variance of displacements does not converge and therefore is a promising alternative model to study microbial species exhibiting extremely wide distribution of jump lengths. The governing equation for a FADE model is similar to the classical ADE except that the order of the highest derivative is fractional. In real space the fractional derivatives are

integrodifferential operators and describe a spatially nonlocal process. A spatial fractional derivative therefore describes particles that move with long-range spatial dependence or high velocity variability (Benson, 1998; Schumer et al., 2001). If a one-dimensional random walk is considered as a motion on an infinite lattice, then a larger-order fractional derivative places more weight on nearer cells and probability of jumps to distant cells decrease very quickly with distance. Similarly, lower-order fractional derivatives place relatively less weight on nearer cells and probability of jumps to distant cells decrease slowly with distance. As the order of derivative $\alpha \in (0,2]$ decreases, long flights are interspersed with shorter jumps, resulting in fractal paths. The densities are scale invariant with $t^{1/\alpha}$, and the sample variance grows proportional to $t^{2/\alpha}$, i.e. always equal to or faster than Fickian growth. Parashar and Cushman (2008) presented a model of microbial motility based on α -stable Levy motion, a Lagrangian model that can be used to represent “run-and-tumble” behavior. A method, based on tracking the evolution of separation distance between a pair of particles, was developed to determine the parameters of a Levy process (Parashar and Cushman, 2007). Development and application of FADE model however has its own challenges associated with defining boundary conditions for integrodifferential operators, and setting up numerical solution technique for fractional derivatives. Though FADE models are a potentially strong technique for modeling bacterial transport, in this thesis we focus on the simpler methodology of a CTRW approach which is designed to capture the distributional features of run-and-tumble behavior without investigating the correlation structure that may exist in the dynamics of bacterial motion.

2.3 CTRW Model

Considering the ‘run-and-tumble’ nature of microbe movement (Berg, 2000), the Continuous Time Random Walk (CTRW) approach is a promising mathematical framework to model and predict the motion of microbes in groundwater. The CTRW approach assumes the number of jumps (n) and their magnitude during any given time interval $(0, t)$ is random. A microbe starting from the location \mathbf{r}_0 will jump to \mathbf{r}_1 , and then wait for time τ_1 before the next jump. In this study, displacement (\mathbf{r}_i) and waiting time (τ_i) are treated as both coupled and uncoupled variables. We consider individual microbial trajectories to be made up of single displacements whose length is a random variable and are also separated in time by random waiting periods. The $n+1$ step displacement can be expressed as:

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \boldsymbol{\varepsilon}_n \quad (3)$$

where $\boldsymbol{\varepsilon}_n$ is the displacement increment at step $n+1$. The $n+1$ step time can be expressed as:

$$t_{n+1} = t_n + \tau_n \quad (4)$$

where τ_n is the time increment at step $n+1$. The concept of waiting time generalized the idea of classical random walks by providing a PDF for of time intervals between successive jumps compared to use of discrete intervals. Every jump and waiting time increment is extracted from modeled or observed distributions of those quantities, so CTRW can not only model Fickian motion but also anomalous transport. Let $P(t; n)$ be the probability for n jump events in time t . The goal of CTRW is to calculate $P(\mathbf{r}, t)$, the probability of finding the particles at \mathbf{r} at time t , which can be expressed as:

$$P(\mathbf{r}, t) = \sum_{n=0}^{\infty} P(t; n)P_n(\mathbf{r}) \quad (5)$$

Where $P_n(\mathbf{r})$ is the probability of finding the particle at \mathbf{r} after n jumps. The identification of $P_n(\mathbf{r})$ with mechanistic understanding of bacterial motion is at the core of the CTRW formulation of transport modeling (Berkowitz et. al., 2000). The series of jump lengths recorded for each individual trajectory, the turn angle between successive jumps, and the time required to complete a linear jump were analyzed to study statistical distributions of bacterial motion dynamics. Through convolution of distributions of both microbe displacement (jump) lengths and their waiting time (time required to complete a linear displacement) distributions, transport of bacteria was modeled on a continuum of spatial scales via the CTRW technique thus providing a method to model and upscale bacterial transport. Movement patterns of individual microbes (of metal-reducing species *Geobacter* and *Pelosinus*) were found by a series of laboratory experiments. Analysis of experimental data of microbial trajectories provides the necessary information to construct a CTRW model for bacterial transport.

3. Methodology

A schematic describing the sequential steps of the experimental and modeling procedures are shown below in Figure 4. The experimental part of the work was conducted at the Environmental Molecular Science Laboratory (EMSL) located at the Pacific Northwest National Lab (PNNL), Richland, WA. Analysis of experimental data and model construction were mostly carried out using MATLAB. Both CTRW and ADE models were investigated for the selected species of bacteria, and model results were compared against transport metrics computed directly using the real experimental trajectories.

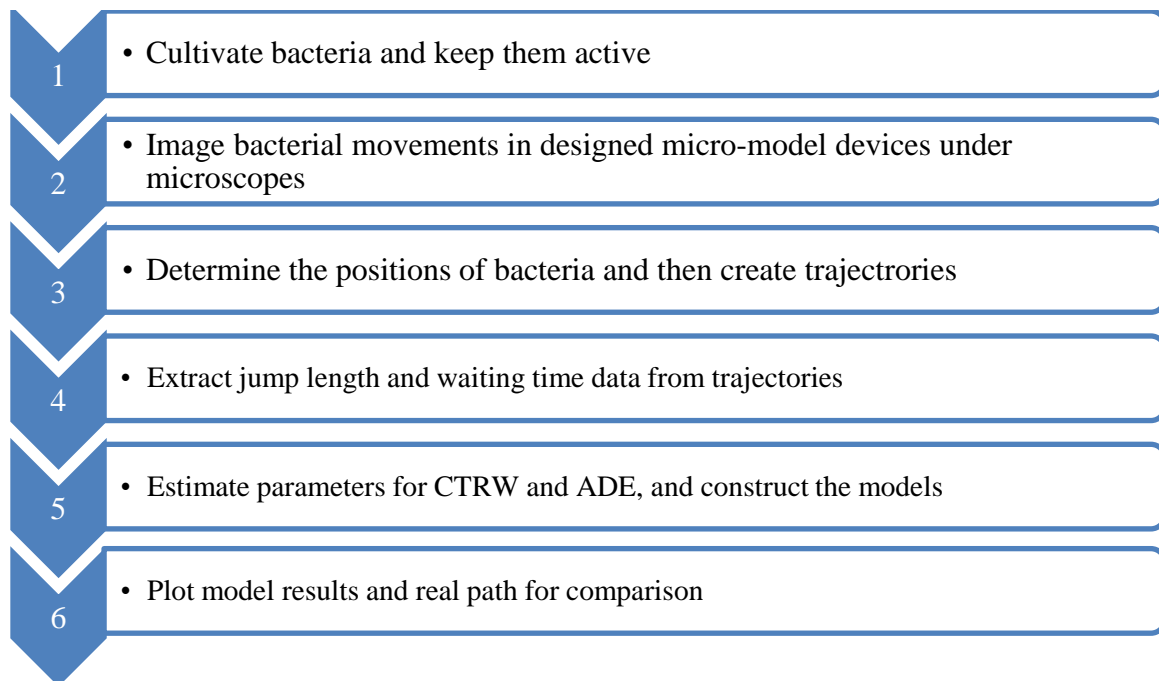


Figure 4: Flowchart of main Procedures

3.1 Bacteria Cultivation

The microorganism studied include the model metal-reducing bacteria *Geobacter sulfurreducens* strain DL-1 and strain PCA, and *Pelosinus* strain JHL-11, an organism isolated from the groundwater of the 300 Area of the heavily contaminated Hanford Site. This set of bacteria encompasses strictly anaerobic (*Geobacter*) and fermentative (*Pelosinus*) species. *Geobacter sulfurreducens* strain DL-1, expressing GFP (Franks et al. 2012), received from the Derek Lovley lab at the University of Massachusetts, was cultured anaerobically in a NBAF medium (acetate and fumarate as electron donor and acceptor, respectively), reduced with cysteine and containing 150 µg/ml spectinomycin to maintain the plasmid encoding of the GFP Late log to stationary phase (phases of bacterial growth) cells were examined without dilution, using a 1 cc syringe and 22-gauge needle sparged with 80% N₂ and 20% CO₂ to remove the cell suspension from the Balch tube or serum vial, and a microfluidic device that had been degassed overnight in a Coy glovebag (~95% N₂, ~5% H₂) and kept in an anaerobic jar (~95% N₂, ~5% H₂) until immediately before use.

Geobacter sulfurreducens strain PCA (Caccavo et al.) was cultured anaerobically in *Geobacter* Medium (ATCC medium # 1957), with sodium acetate at 10 mM and sodium fumarate at 50 mM (headspace of 80% N₂ and 20% CO₂). Cells were used either without dilution or after diluting in phosphate-buffered saline (PBS) containing 10 mM sodium fumarate.

Pelosinus strain JHL-11, isolated from sands incubated from Hanford 300 Area groundwater (see Lee et al. 2015), was cultured anaerobically in TSB (without dextrose) with 5 mM potassium nitrate (KNO₃) added as electron acceptor, at either 30°C or room

temperature. The culture was examined, without dilution, using an N₂-sparged 1-cc syringe and 22-gauge needle to remove the cell suspension from the Balch tube, and a microfluidic device that had been degassed overnight in a Coy glovebag (~95% N₂, ~5% H₂) and kept in an anaerobic jar (95:5 N₂:H₂) until immediately before use.

3.2 Laboratory Experiments

Individual tracks of cells on the order of several seconds to few minutes in duration were recorded to provide information of bacterial motility in two dimensions. The approach is similar to some other studies where the relatively small value of vertical dimension compared to the extent of horizontal plane allows for projecting the bacterial motion onto two-dimensional planes (e.g., Davis et. al., 2011). ; The experiments were repeated several times to collect sufficient data for statistical analysis. Micro-model chambers for easy injection and viewing of the cells were constructed, the schematic sketch of which is shown in Figure 5.



Figure 5: Experimental cell sketch. The black dots at left-end and right-end represent inlet and outlet. The main body of chamber is consisted of two parts: pillar area (hollow dots) and open area (black).

The chambers are thin in the vertical direction (20 microns), and on the order of 1-2 cm in the other two directions. Microfluidic fabrication capabilities at EMSL were utilized to construct the non-flowing chambers which were made out of Polydimethylsiloxane (PDMS) for easy design adaptation. The micro-model chambers were divided in half to simultaneously provide an open unobstructed medium, and a simple region of periodic pore structures. The periodic pore structures were created using staggered rows of fixed diameter cylindrical pillars to monitor changes in motion of microbes in presence of obstructions that are invariably encountered in porous media. For the two species of microorganisms studied in this thesis the designed pore throat diameter was too large to have any significant impact on the motion dynamics. Hence the results presented are based on the recording of trajectories in the open unobstructed medium. For introduction of solution containing new microorganisms, an inlet and outlet are constructed. After injection of the microbial solution, the inlet and outlet were sealed to eliminate the possibility of airflow from affecting active bacterial swimming. The circuitous path leading from the inlet/outlet point to the micro-model chamber was also designed to reduce the chances of any small perturbation in pressure gradient to cause flow in the chamber affecting the bacterial motility pattern. The laboratory experiments were conducted under sterile environment at EMSL with help from collaborators and laboratory staff. Advanced microfluidic imaging techniques for biological and chemical systems developed under PNNL's Microbial Communities Initiative (Grate et al., 2013) at the Environmental Molecular Sciences Laboratory (EMSL) is applied to this research. The motion of microbes were viewed under various microscopic magnifications and videos were recorded in the format of *.fits files using a camera with a sensor pixel size

of 13 microns x 13 microns. Each recorded frame on the video is of the size of 1024 pixels x 1024 pixels. The physical size of each pixel on the video file is easily determined by dividing the sensor size by the microscope magnification. For example, with the microscope magnification of 20X, each pixel on the video file has a size of $13/20 = 0.65$ microns. The total size of the viewing window then becomes $1024 \times 0.65 = 665$ microns. The choice of the microscope magnification is mainly guided by the average body length of the cells being studied. Each individual cell should ideally occupy at least 2-3 pixels of the video frames in order to minimize the numerical complexities associated with tracking in MATLAB. A good choice of magnification to observe bacteria with average body length of 0.5 microns is 50X (each pixel on the video file is then $13/50 = 0.26$ microns) or more. Increasing the magnification too much has the drawback of reducing the total size of viewing window as well as reduction in depth of field causing the cells to frequently go in and out the focal plane as they change location in the vertical direction. In this research, we used magnification of 20X, 32X, 40X and 64X to study the two species of bacteria. In addition to setting up the magnification of microscopes, the recording of videos also requires to set the frame capturing speed (frequency) of the camera. Higher average swimming speed of bacteria requires a higher frequency to allow for easy tracking of the cells from one frame to the next. Recording frequencies ranging from about 4 Hz to 8 Hz were used in this research. A snapshot of an example video file is shown in Figure 6.

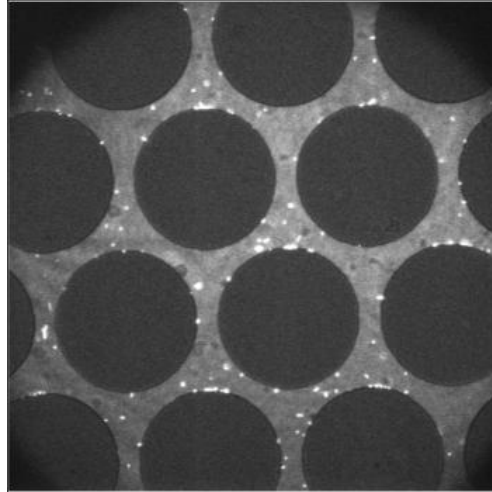


Figure 6: Fluorescent *Geobacter* under microscope and in presence of periodic pore structure.

White dots show the location of individual *Geobacter* cells.

Hundreds of videos under various combinations of microscope magnifications and frame capturing frequencies were recorded for both species of bacteria. The small body size of *Geobacter* warrants larger magnification of 40X or 64X and use of laser (when possible) to observe and record fluorescent cells. Videos of *Pelosinus* were mostly recorded under bright light with 32X and 20X magnification factor.

In addition to the experimental layout described above, a separate device was constructed to record motion of motile cells in the presence of chemical gradient. Two major channels were added at each side of the main chamber, and then they were connected to the chamber by very thin transverse channels. The thin transverse channels do not allow any advection and aid in establishing a relatively constant chemical gradient between the two major channels by means of diffusion only. However, after solutions at different concentrations were injected in the two major channels, the concentration in the main chamber continuously changed thus preventing an equilibrium condition. Even though

the design to study transport under the influence of chemical gradient (chemotaxis) failed, it is noted here to motivate further research.

3.3 Code Developments

A series of video file extraction and processing codes were written in MATLAB (see Appendix A.1) analyze the *.fits files. Using these codes, the distributions of jump length and waiting time were obtained (codes available in Appendix A.2) and then incorporated into the framework of a CTRW model for bacterial transport (codes available in Appendix A.3).

Each video file (in *.fits format) consists of 1000 frames with every frame being 1024 pixel x 1024 pixel in size. A frame capturing frequency of 8 Hz for example gives us a total video duration of $1000/8 = 125$ seconds. The time interval separating two frames is easily found by inverting the frequency. Every frame of a video file consists of little more than a million pixels ($1024 \times 1024 = 1,048,576$). Each of these pixels appears as a shade of black, white, or gray and is associated with a pixel intensity value ranging from a few hundred to few thousand. The data of pixel intensity is recorded and is read by MATLAB and treated as a $1024 \times 1024 \times 1000$ matrix. Generally, the bacteria are darker than their surrounding under bright light, but brighter under laser for the fluorescence species thus giving the cells distinct pixel intensity value from its surroundings. The background pixels tend to have similar intensity values throughout time while the pixels occupied by bacteria change intensity over time.

Numerical challenges linked to distinguishing bacteria from the background and determination of the correct path in light of changes in swimming speed and possible

trajectory intersection makes the processing of video files a non-trivial task. To accurately determine the positions of bacteria and create paths, spatial and temporal filters are introduced to eliminate the noise of background. A radius around each path end is defined, outside of which a bacterium is considered to be ‘new’. This method is also applied to solve the problem of bacteria moving in and out of camera views. In addition, an appropriate searching radius needs to be applied which does not ignore or overlap bacteria. After each step, a search radius is constructed and the new location of bacteria is determined within that zone. We assume bacteria take exactly the same jump and then search around that location instead of the end of the path. This allows us to use a smaller search radius without losing track of many bacteria. Furthermore, different bacteria exhibit different motility behavior: many just sit in one location and twitch, while others are highly motile. Different categories are defined for different behavior, with each category having a different search radius, because faster bacteria are easier to lose track of.

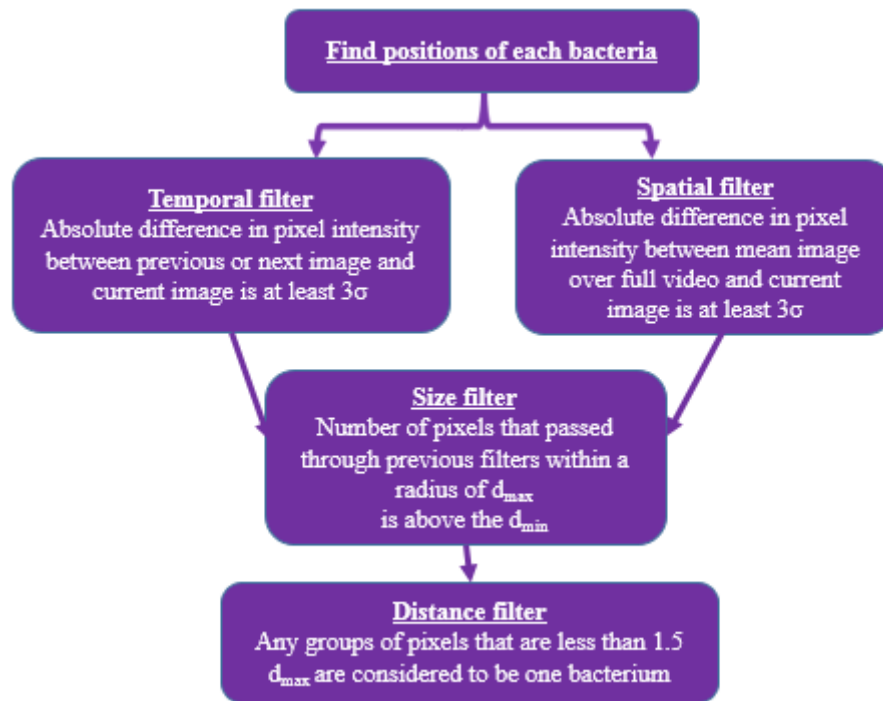
3.3.1 Trajectories Extraction

The trajectories extraction process is demonstrated in Figure 7. A “moving bacteria” is recognized as an assembly of points, where 1) the intensity of these points is significantly different (darker or lighter) from surroundings in a frame, and 2) the center of these points are changing though time. Part (a) of the figure shows the application of a set of numerical filters to separate a cell from the surrounding noise in intensity values, and then application of a size and distance filter to pin-point new location of a moving cell based on search radii tied to cell size and average speed. Part (b) of the figure

demonstrates the process of connecting locations in successive frames to form complete trajectory of a motile cell.

Figure 8 (a) shows a video frame after application of temporal and spatial filter to separate the background noise. Figure 8 (b) shows bacteria position as a white circle after application of all filters listed in Fig. 7(a). The search radius d_{min} and d_{max} are determined by taking the swimming speeds and size (Caccavo et al., 1994; Shelobolina et al., 2007) into consideration. Each red line on Figure 8 (c) represents one trajectory from a single video file. In most situations a moving cell occupies more than one pixel on a video frame (i.e., the body length covers multiple pixels). The x- and y-coordinates are found by locating the centroid of all occupied pixels. Figure 8 (d) displays a trajectory recorded in the units of pixels in x- and y-directions. More details are shown in Appendix A1.

(a)



(b)

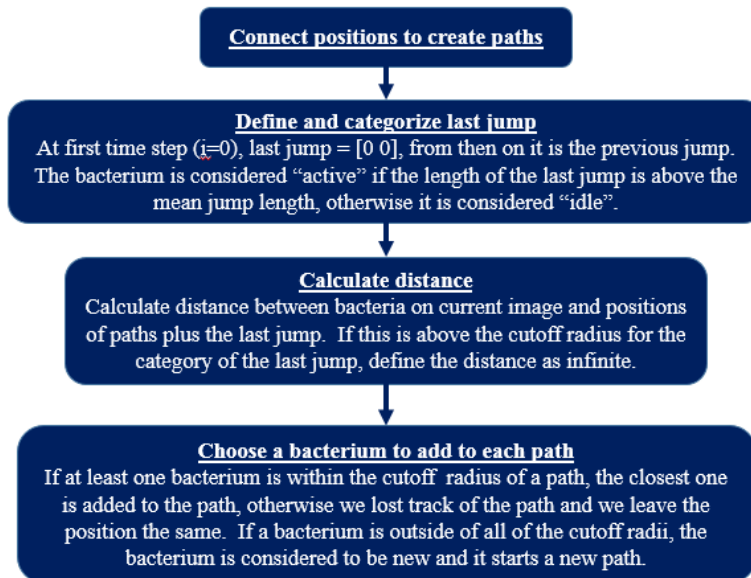
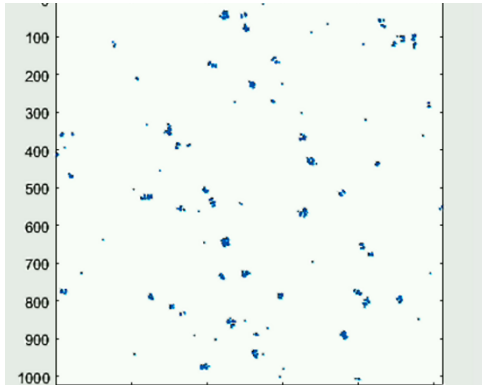
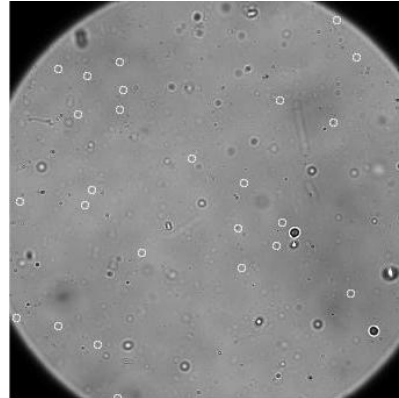


Figure 7: Flowcharts of extracting positions and paths from video files. Determination of bacteria position is shown in (a) where σ = standard deviation, d_{\max} = high end of bacteria diameter range, d_{\min} = low end of bacteria diameter range. In (b), the process of connecting positions to construct paths is demonstrated.

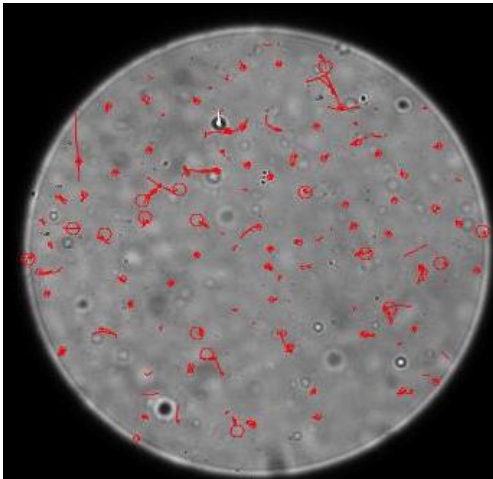
(a)



(b)



(c)



(d)

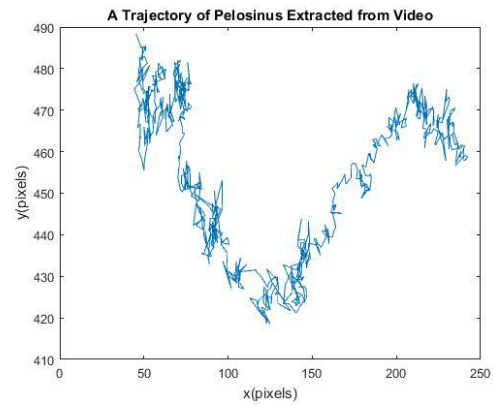


Figure 8: Projected bacteria points in a video frame. (a) location of bacteria after application of temporal and spatial filter to separate the background noise. (b) The white circles are recognized as moving bacteria in a video frame after application of a size and distance filter. (c) Each red line represents one bacteria trajectory extracted from a video file. (d) A trajectory of a bacterium in units of pixel.

3.3.2 Statistics Analysis

After path files for each bacterium are created, data was further filtered to only include those paths that have meaningful time existence (10 frames or more) and at least one jump during the duration of its total recording time (i.e., bacterium that were completely idle during their recording time are excluded from further analysis). A cell is assumed to be in “waiting state” if it moves less than the average body length for its species. Continuous motion of a cell in a certain direction is treated as “long” jump unless the direction of motion at some instance changes by more than 5° . A new jump is registered when at least one of the two conditions is met: either 1) the bacterium moves from a waiting state; or 2) the change in turn angle in the direction of motion is larger than $\pm 5^\circ$ (1 pixel change in perpendicular direction over 10 pixels of displacement in the direction of travel).

As individual trajectories are assumed to be independent of each other, the initial location (i.e., start point of a trajectory) can be arbitrarily changed to make all trajectories start from a common location (the origin of the coordinate system). This is to say that the exact start point and end point of a trajectory within the viewing window doesn't matters; what matters is the change in pixel location and the time elapsed for those changes to happen. After transferring the time and distance from units of frame and pixels to units of seconds and microns by equations:

$$\text{Time (sec)} = \text{Number of Frames} \times (1 / \text{Frequency})$$

$$\text{Distance (microns)} = \text{Number of Pixels} \times (13 \text{ microns} / \text{Magnification Factor})$$

each individual trajectory of a species (*Geobacter* or *Pelosinus*) is merged into one master file which contains information for all jumps including x-increment, y-increment, and waiting time (See MATLAB codes in Appendix A2). This master file is then used to analyze distributional properties of waiting time and jump length for each species.

In addition to recording the statistics of jump length and waiting time, we also continuously monitor the number of individual cells in the system, the mean travel distance as a function of time, and the variance in cell locations as a function of time. The relationships between variance and diffusion coefficient (for ADE) can be presented as:

$$D = \frac{1}{2} \times \frac{\partial \delta^2}{\partial t} \quad (6)$$

where D is diffusion coefficient and δ is standard deviation of distance traveled.

The master file containing information of all trajectories (longer than or equal to 10 frames) progressively becomes sparse as only a few trajectories are continuously recorded for a long duration (say more than 100 frames). This is because majority of trajectories go in and out of the viewing window turning them into discontinuous pieces of data. Statistical quantities such as mean and variance are computed only up to the point where we have at least 50 unique trajectories in the master file. After the data becomes less dense than 50 unique trajectories, we assume the variance to linearly increase with time for the purpose of ADE-model calculations.

3.3.3 Model Developments and Breakthrough Curves

The jump length and waiting time data allows for construction of their respective statistical distributions which lays the foundation for development of the CTRW model.

We investigate both a coupled and an uncoupled CTRW model by treating waiting time as dependent variable (i.e., waiting time increment is computed by first conditioning the random walk process on the jump length) or independent variable (i.e., waiting time and jump length are treated as unrelated quantities) respectively. For the coupled model, specific waiting time values that corresponds to known jump lengths are used in the CTRW process. For the uncoupled model, all possible values of waiting time are treated equally regardless of the magnitude of jump length. The computation of mean travel distance and mean centered variance (first and second moment of breakthrough plots) allows for construction of an ADE-based model. The raw trajectory data, without any statistical investigation of its distributional properties, can be replicated to reproduce the real observed transport. The goal of the model development is to compare the prediction of breakthrough curves obtained from the CTRW (both coupled and uncoupled) and ADE model with that of the observed transport. All particles (individual cells) are assumed to start from the origin and diffuse radially to concentric rings of control planes located at fixed radial distance (see Figure 9). The breakthrough curves, which in essence are the first passage time densities as microbes being living organisms can move back and forth multiple times across a control plane, are computed and compared.

The duration of the longest recorded video is about 250 seconds (1000 frames recorded at a low frequency of approximately 4 Hz), hence the breakthroughs are limited in time by that value. The modeled breakthrough curves obtained for CTRW and ADE model can however continue up to the desired value in time. For the CTRW model, random values are generated for each step for magnitude of jump length and waiting time from the pre-determined probability density functions. For the coupled model, the jump length and

waiting time are dependent. For the uncoupled model, jump length and waiting time are selected independently of each other. Breakthrough curves at $L= 10, 20, 30, 40, 50,$ and 60 microns are obtained by both coupled and uncoupled model for *Pelosinus*, and only at the first four control planes for *Geobacter*. To obtain breakthrough curves for the ADE model, we consider the solution to the 1-D ADE equation (Sauty, 1980) as the radial model of diffusion and control planes (as described above) is not biased towards a preferential direction. Hence the following analytical solution, based on the linear distance to the control planes, can be used to compute breakthrough curves:

$$\frac{C}{C_0} = \frac{1}{2} \times \operatorname{erfc} \left(\frac{L - V \times t}{2 \times \sqrt{D \times t}} \right) \quad (7)$$

where C is the number of bacteria reaching distance L in time t , C_0 is the initial number of bacteria, L is distance to the control plane, V is the average velocity, which is obtained from slope of mean displacement over time, and D is diffusion coefficient which can be obtained using Eq. (6). Note that the value of D changes with time as the rate of growth of variance is not uniform. For time periods exceeding the maximum duration of recorded trajectories, we assume the diffusion coefficient to be a constant (equal to the value of D computed for the longest recorded trajectories). By using $L= 10, 20, 30, 40, 50,$ and 60 microns, ADE solution (Eq. 7) yields breakthrough curves for each species.

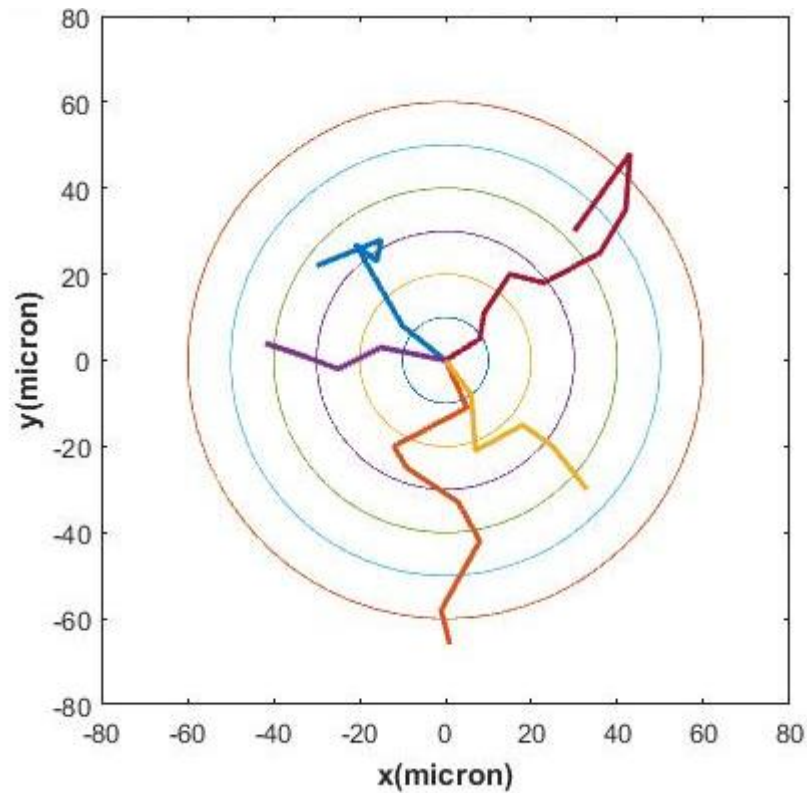


Figure 9: Illustration of control plane settings and microbial trajectories in random directions. The concentric circles represent control planes located at 10, 20, 30, 40, 50, and 60 microns away from point source (0,0). For *Pelosinus*, breakthrough curves are obtained for all 6 control plane, and for *Geobacter*, only 4 nearest control planes are used. All trajectories start from the origin and their first passage time are recorded for individual control planes to obtain breakthrough plots.

4. Results

4.1 Jump Length and Waiting Time

The empirical probability density functions of jump length and waiting time for the two species are shown in Figure 10. The probability densities of the jump length show the same trend for the two species: the probability increases until it reaches the peak (5-6 microns), after which it decreases with a maximum recorded jump value of about 100 microns. The longest jump of 100 microns is about 30 to 40 times of bacteria body length. The jump length probability density function of *Geobacter* spans a shorter range and shows a comparatively higher probability associated with longer jumps. In contrast, the jump length probability density function of *Pelosinus* shows a wider range and lower probability associated with longer jumps when compared to *Geobacter*. The waiting time probability density function for *Geobacter* has a high and constant probability for low values of waiting time (< 10 seconds) after which the probability gradually declines with increasing waiting time. The waiting time probability density function for *Pelosinus* continuously declines and shows a higher probability associated with shorter waiting periods and lower probability associated with longer waiting period compared to *Geobacter*. In other words, *Pelosinus* is more likely to make fast jumps and less likely to make slow jumps when compared with *Geobacter*. The longest waiting time can be over 200s for both species.

4.2 Real Path

The first-arrival-time curves of *Geobacter* and *Pelosinus* are shown in Figure 11 and 12 based on tracking the real trajectories until the control planes are reached.

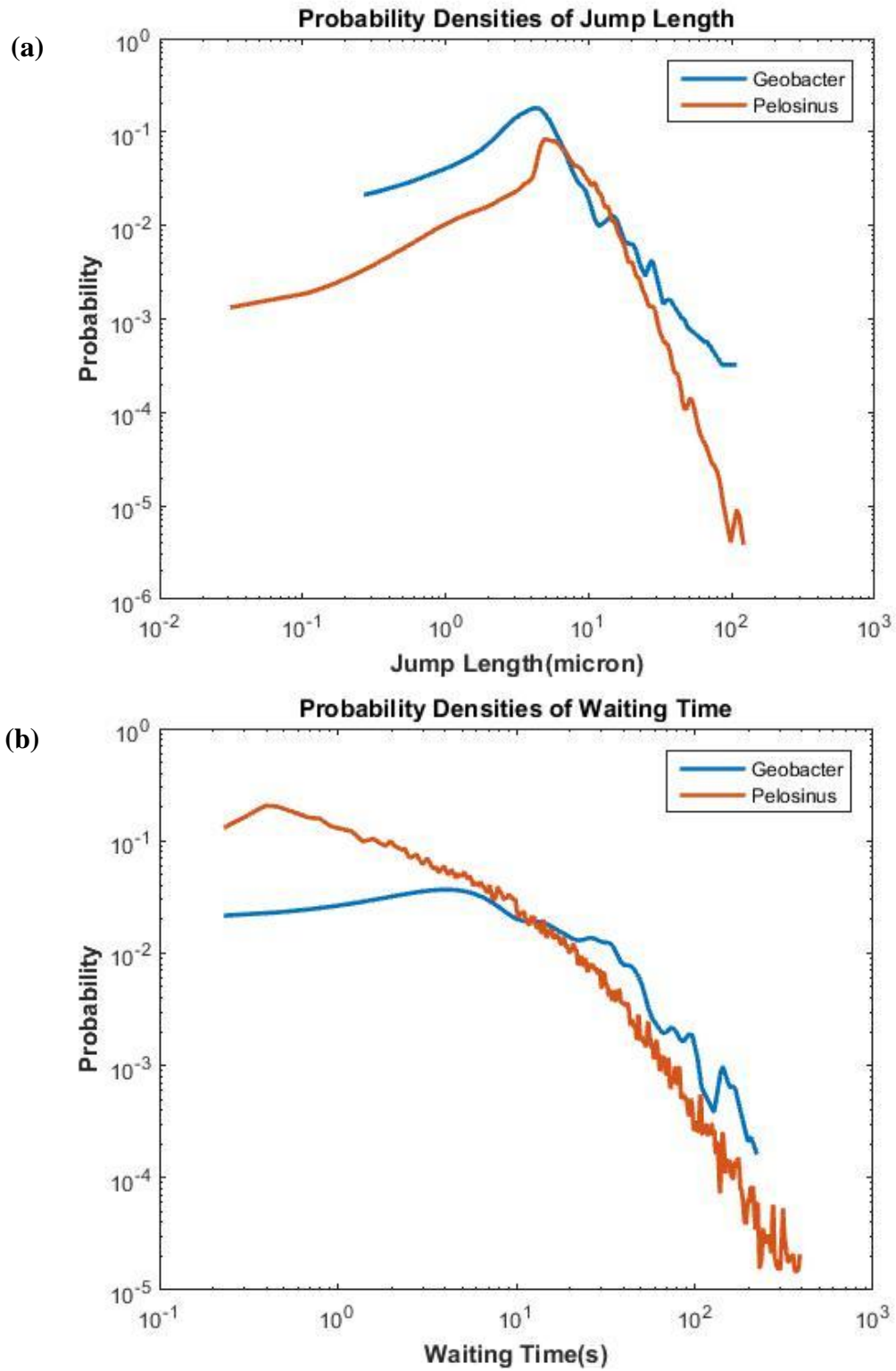
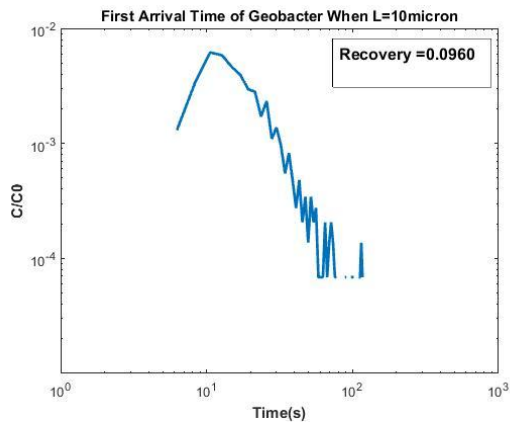
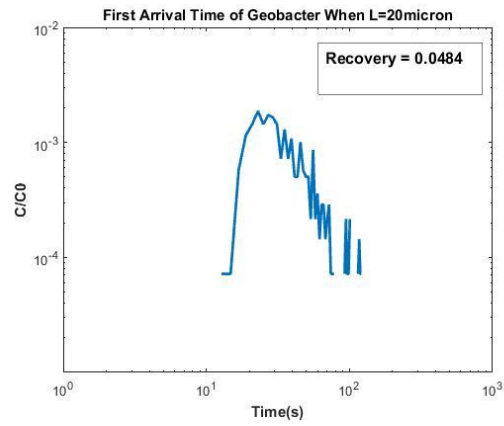


Figure 10: (a) Jump length and (b) waiting time probability densities for *Geobacter* and *Pelosinus*.

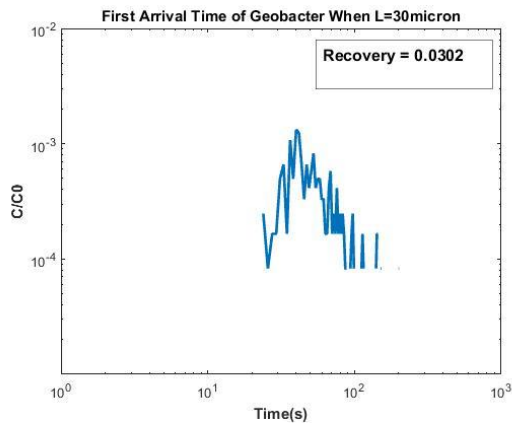
(a)



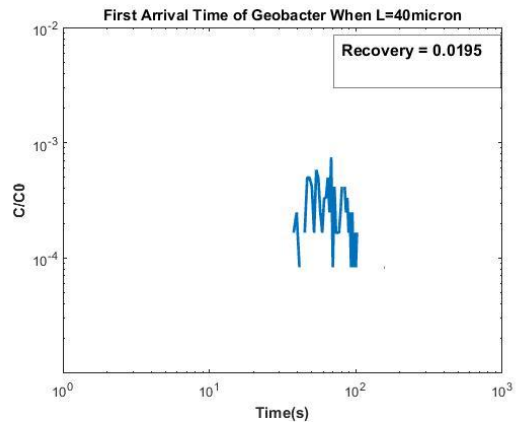
(b)



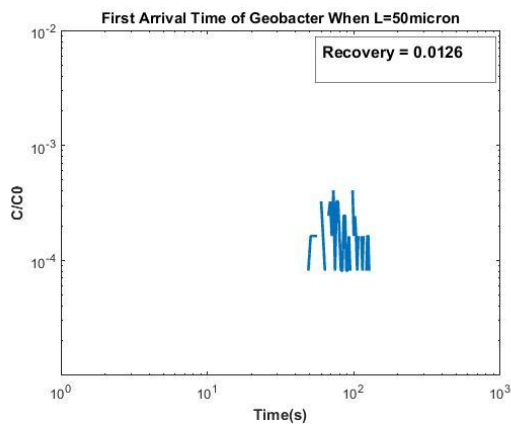
(c)



(d)



(e)



(f)

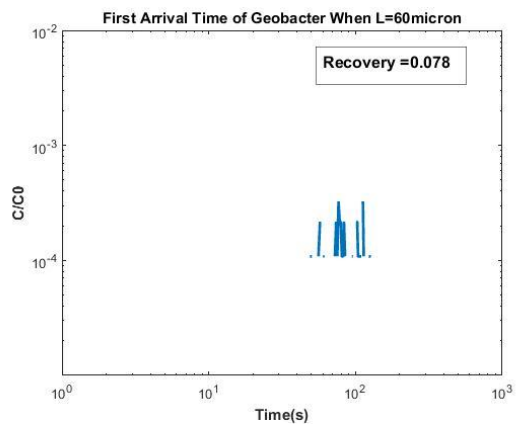


Figure 11: First- arrival time curves of *Geobacter*.(a), (b), (c), (d), (e), and (f) corresponds to L =10, 20, 30 ,40 ,50, 60 microns.

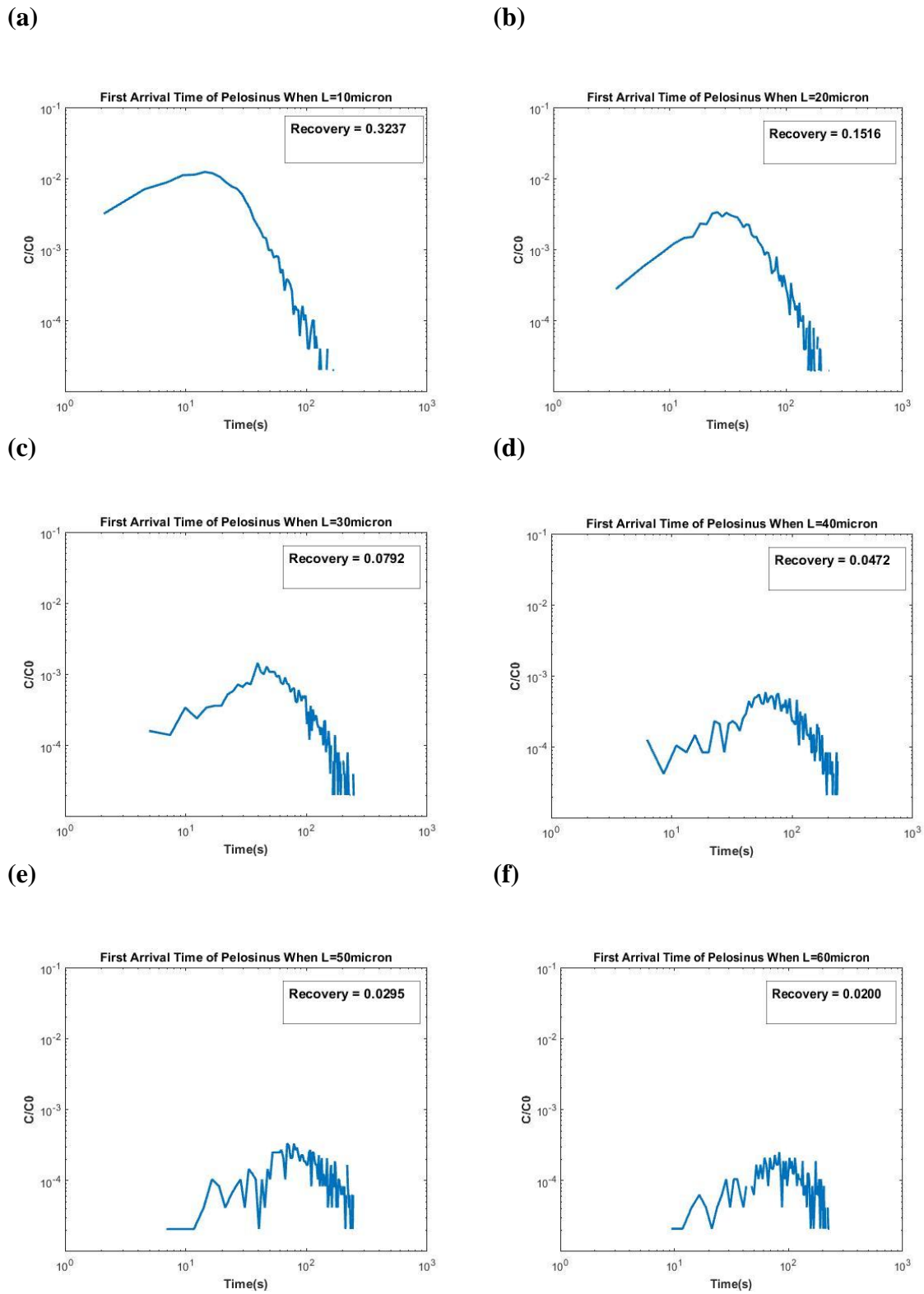


Figure 12: First- arrival time curves of *Pelosinus*.(a), (b), (c), (d), (e), (f) corresponds to $L = 10, 20, 30, 40, 50, 60$ microns.

The box in the top right corner shows the fraction of individual trajectories that are able to reach the control planes in 250 seconds. For *Geobacter*, only about 10% bacteria reach control plane when $L = 10$ microns, and the recovery number decreases as L becomes larger. In the case of *Pelosinus*, about 30% of the trajectories reaches control plane where $L = 10$ microns. The recovery rates of *Pelosinus* reaching every control plane are higher than that number of *Geobacter*, which indicates that *Pelosinus* is generally more “active”. When $L = 50$ and 60 microns, the recovery of *Geobacter* curve is lower than 2% and it does not display a clear profile. For this reason, the breakthrough curves for *Geobacter* are not computed for the cases of $L = 50$ or 60 microns. The normalized first-arrival time plots are shown in Figure 13. The wiggles of these plots have been smoothed for an easier visual comparison. As control planes become farther, the peak time of curves migrate to higher values and the magnitude of peak concentration decrease. The rate of decrease in the magnitude of peak concentration slows down dramatically for farther control planes. For *Pelosinus*, the curves have a larger spread and a longer rising limb, while *Geobacter* has narrower spread and a steeper rising limb.

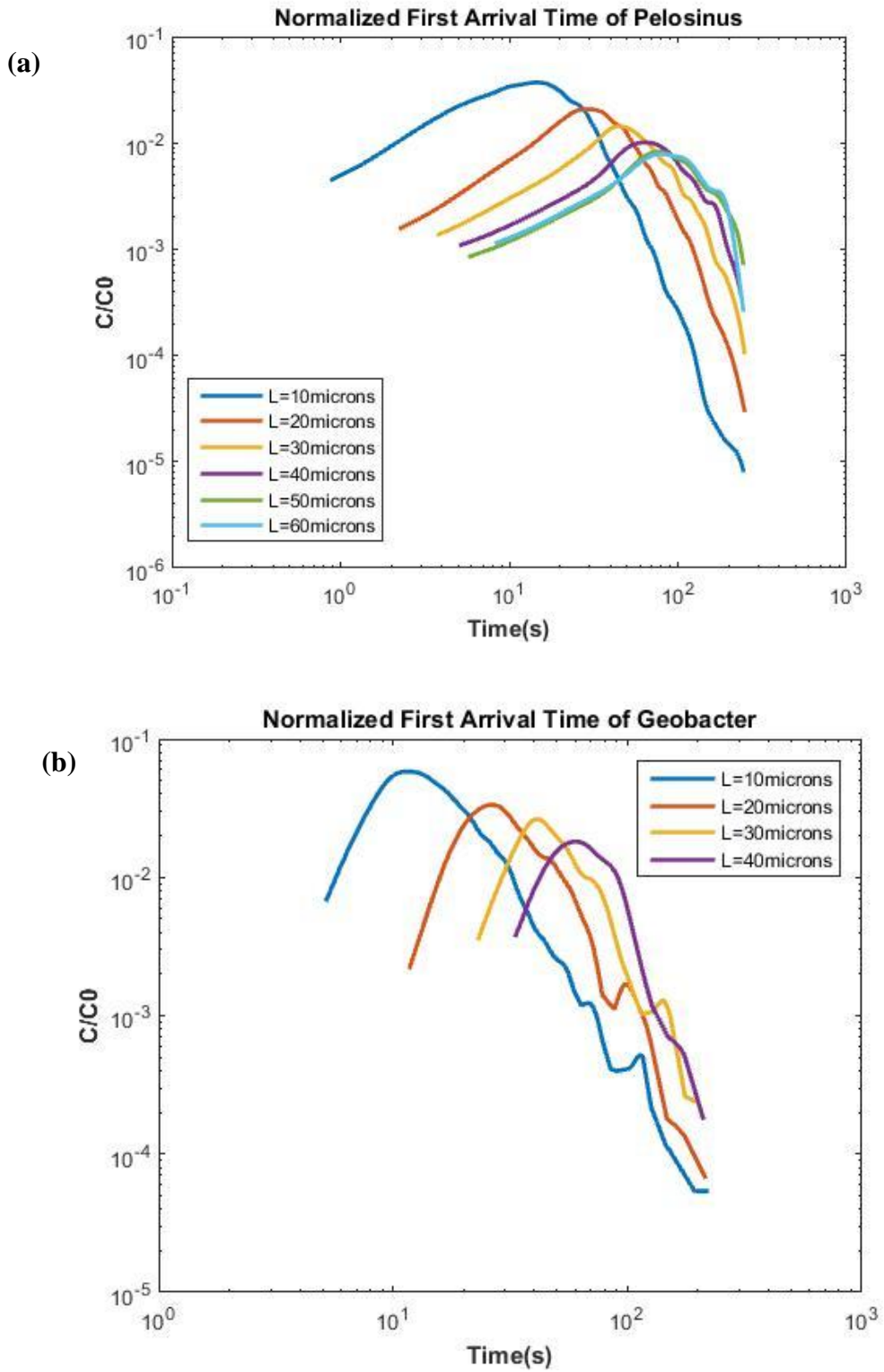


Figure 13: Normalized first-arrival - time plots of *Geobacter* and *Pelosinus*.

Figure 14 and 15 shows the plots of δ^2 and mean displacements over time. These quantities are necessary for construction of an ADE model and provide valuable insights into the possibility of non-Fickian transport behavior, Recall that variance and mean displacements are computed only up to the point when there are a minimum of 50 cells present in the system.

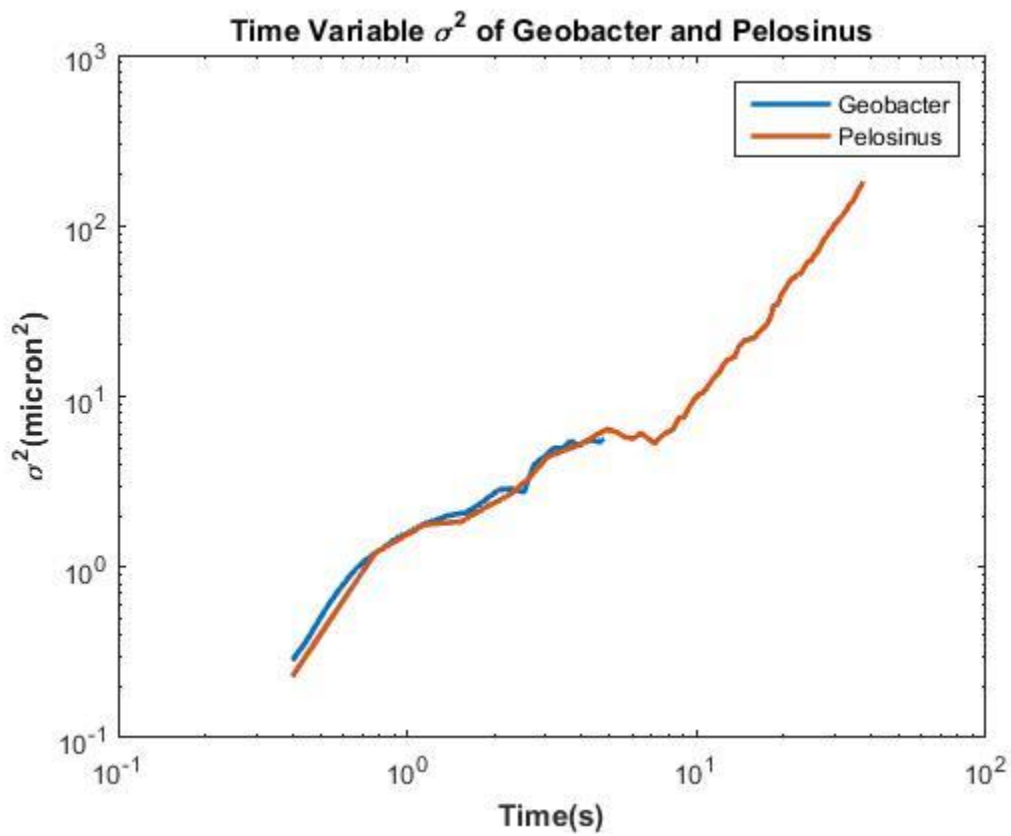


Figure 14: δ^2 over time for *Geobacter* and *Pelosinus*.

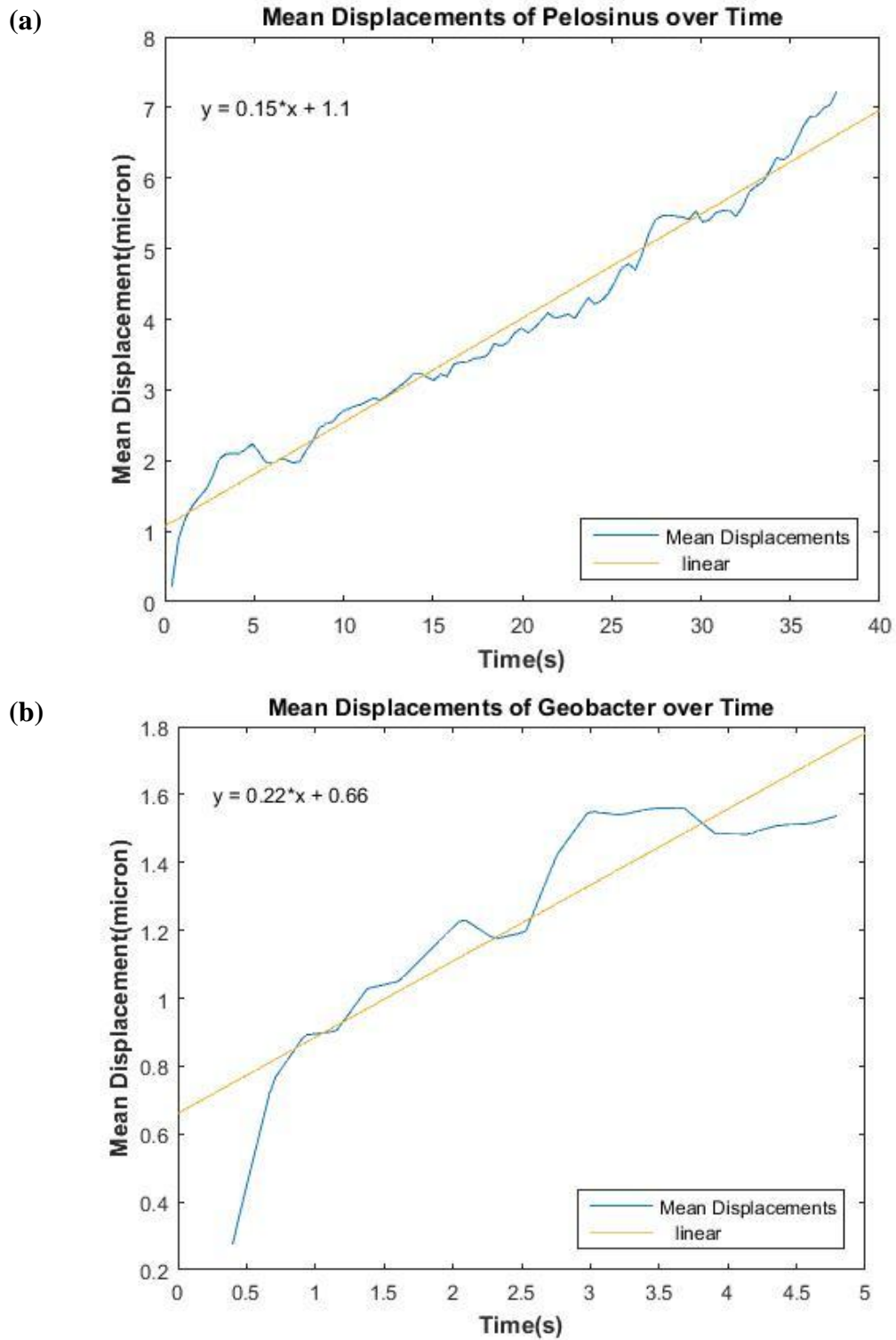


Figure 15: Mean Displacements of (a) *Geobacter* over time, and (b) *Pelosinus*

On the log-log plots of Figure 14, it appears that δ^2 shows a linearly increasing relationship with respect to time for *Geobacter*, and for *Pelosinus* the linearly increasing relationship transitions into a faster than linearly increasing plot of variance (i.e., super-Fickian behavior) after about 10 seconds. This raises that possibility that an ADE based model might perform better in the case of *Geobacter* than in the case of *Pelosinus*.

Though great care was taken to prohibit any movement of fluid inside the micro-model chambers (by designing circuitous inlet and outlet paths and by sealing the chamber after injection of solution as discussed in chapter 3) our analysis shows that the mean position of the ensemble of particles does move at a very gradual pace (Figure 15), This could be caused by an improper sealing of the chamber or by a minute tilt or vibration of the experimental apparatus. The mean displacement of *Geobacter* moves at a speed of 0.22 microns/s and that of *Pelosinus* moves at a speed of 0.15 micron/s. These values, though very small, needs to be included in the ADE model computations.

4.3 Model Comparison

Figure 16 displays breakthrough results from real path, ADE model, coupled CTRW model, and uncoupled CTRW model for *Geobacter* at $L = 10, 20, 30,$ and 40 microns. The uncoupled CTRW results do not show a clear peak, has a very wide spread, and predicts a high concentration for very low values of time. The uncoupled CTRW results are also less sensitive to changes in distance to the control planes. The coupled CTRW model and the ADE model perform better in approximating the “real breakthrough” in all of these aspects. The fit of the modeled results to the real breakthrough can however be generally categorized as poor for all control planes in the case of *Geobacter*. As control planes move farther, the breakthrough curves also shift along the time axis as expected.

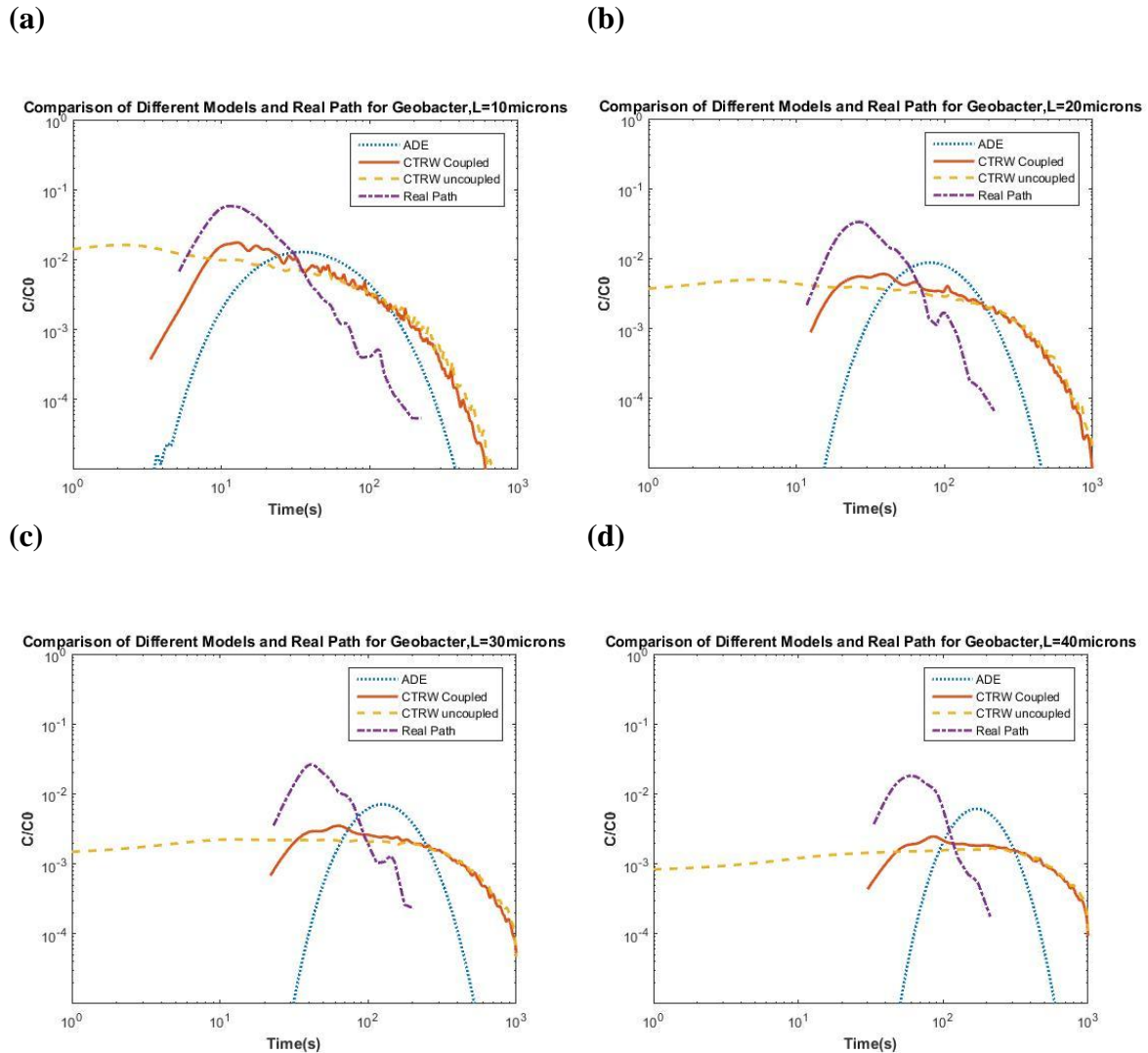


Figure 16: Comparison of Models for *Geobacter* when $t = 1000$ s. (a), (b), (c), (d) corresponds to $L = 10, 20, 30, 40$ microns.

For real path, the breakthrough curve grows faster than any other models and the magnitude of peak concentration are also higher. After peak, the concentrations of real path declines rapidly. The breakthrough curves of real path show a reduced spread and rapid rising and falling limb when compared to the modeled results. Coupled CTRW does a relatively good job at predicting early transport and matching the time to peak. However, when it comes to late time, coupled and uncoupled CTRW models yield

similar curves, which decline significantly slower than real path breakthrough curves. The ADE results in fact show a better match with the real path for late time concentrations. The better performance for ADE (at least for late time) is not totally unexpected as the plot of variance with respect to time for *Geobacter* shows an approximate linear relationship (see Fig. 14).

In Figure 17, breakthrough curves from different models and real path are compared for *Pelosinus*. According to the plots, real path breakthrough curves decline rather quickly after the peak concentration is reached. The magnitude of the peak concentration is higher for the real path plot when compared to the CTRW or ADE model predictions. The ADE model results perform very poorly on all aspects of breakthrough data (i.e., shape, spread, peak magnitude, or rate of concentration decline). Coupled CTRW model performs well in matching the real path breakthrough plots for $L = 10$ and 20 microns. The performance of the coupled CTRW model gradually becomes poorer for longer control plane distances. The coupled and uncoupled CTRW curves are very close at late time and have slow decline rates than real path. The uncoupled CTRW model mimics the coupled CTRW results for smaller values of control plane distances, but becomes widely different in the early time behavior for breakthroughs of longer distance control planes (the couple and uncoupled CTRW model still match fairly well at late times for all control planes). Though not really a good match with the real path breakthrough plots, the improved performance of CTRW model over the ADE model for *Pelosinus* highlights the importance of constructing non-local models for studying systems that shows signs of non-Fickian behavior (as demonstrated by the variance plot of Fig. 14).

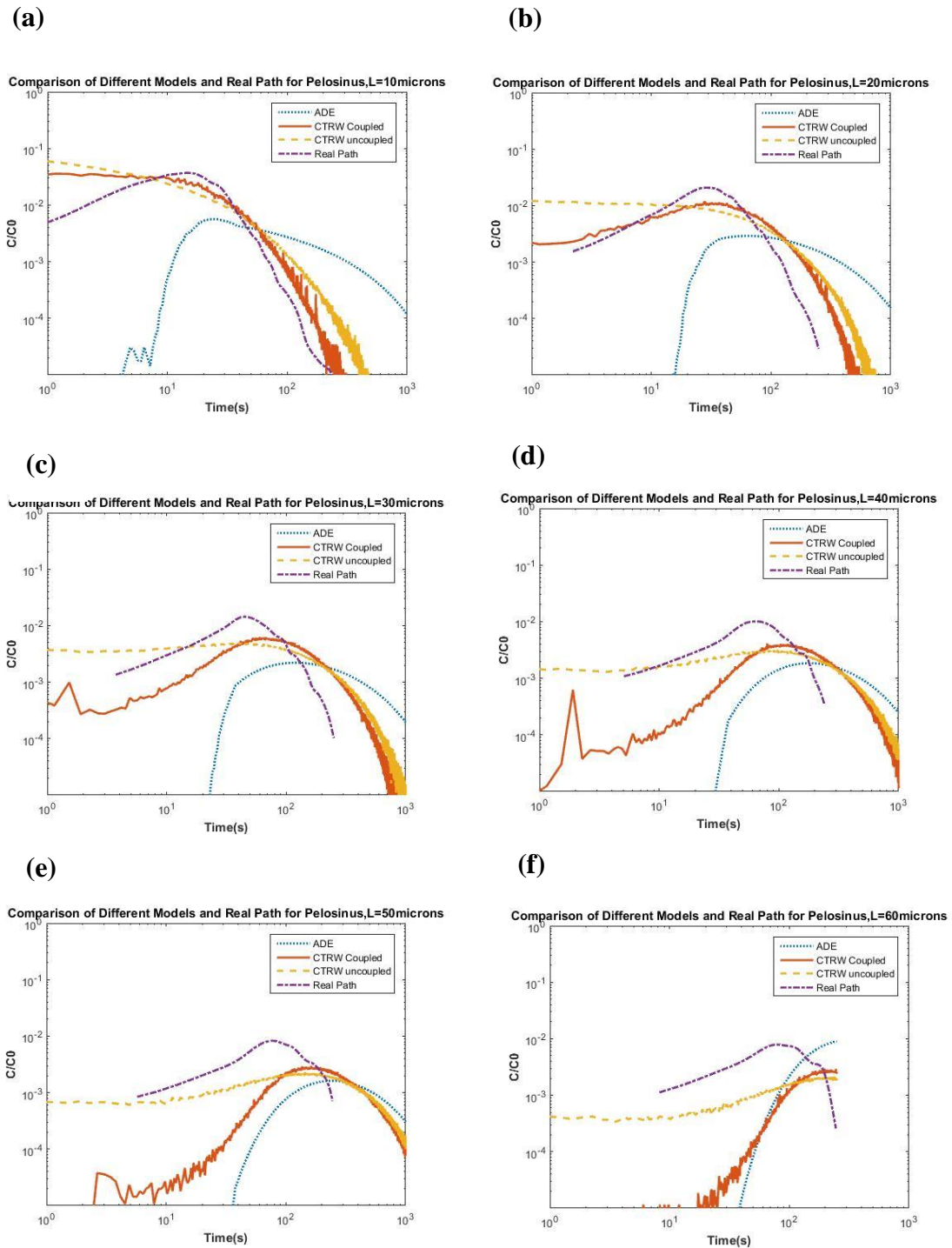


Figure 17: Comparison of Models and Real Path for *Pelosinus* when $t = 2500s$. (a), (b), (c), (d), (e), (f) corresponds to $L = 10, 20, 30, 40, 50, 60$ microns.

Table 1 displays the mean values and standard deviations of real path and models for *Geobacter*. The mean values of all three models (coupled and uncoupled CTRW, and ADE) are 3 to 5 times of mean values of real path. The mean centered standard deviations of CTRW models are 4 to 8 times higher than that of real path. The mean centered standard deviations of ADE models are significantly lower, pointing to relative suitability of an ADE-based model (compared to CTRW models) for studying transport of *Geobacter*. All three models however yield much larger mean values and standard deviations values than real path does. Coupled and uncoupled CTRW models are close to each other at late time.

For *Pelosinus*, the ADE model performs very poorly and the coupled CTRW and uncoupled CTRW model match the real path mean and standard deviation to a much higher degree than in the case of *Geobacter* (comparing Table 1 to Table 2). The match of CTRW models to the real path is slightly better for lower L values (see Table 2).

Table 1: Mean and Standard Deviations of each model for *Geobacter*.

	Real Path		ADE		Coupled CTRW		Uncoupled CTRW	
	Mean ¹	STD ²	Mean	STD	Mean	STD	Mean	STD
L=10microns	23.03	21.57	69.67	47.69	84.45	86.36	95.15	100.05
L=20microns	41.43	25.68	115.13	58.09	187.80	161.99	199.53	177.21
L=30microns	58.59	29.91	160.58	66.90	279.87	209.28	291.75	223.56
L=40microns	72.44	29.44	206.04	74.68	353.97	233.85	363.17	247.06

¹ In unit of sec, ² In unit of sec

Table 2: Mean and Standard Deviations of each model for *Pelosinus*.

	Real Path		ADE		Coupled CTRW		Uncoupled CTRW	
	Mean ¹	STD ²	Mean	STD	Mean	STD	Mean	STD
L=10microns	22.33	17.81	286.18	318.03	24.95	26.17	30.10	38.47
L=20microns	45.90	31.15	350.90	336.62	69.67	51.21	69.80	59.30
L=30microns	68.17	41.57	412.66	354.01	109.37	58.87	100.94	66.10
L=40microns	88.32	47.72	476.74	370.32	137.34	57.37	122.44	66.73
L=50microns	105.48	54.57	541.68	385.20	157.41	54.24	137.47	65.45
L=60microns	104.85	51.76	606.50	398.72	173.20	49.00	148.60	63.64

¹ In unit of sec, ² In unit of sec

5. Discussions and Conclusions

The jump length and waiting time probability density functions of *Geobacter* and *Pelosinus* reveals the nature of transport of these two species. Both the jump length and waiting time probability densities have heavy tails, which suggests that these two species of bacteria departs significantly from the Brownian motion type random walk processes. Both the jump length and the waiting time probability density functions span multiple orders of magnitude. The jump length can be over 100 microns, which is 30-40 times of the body length of these two species of bacteria (Caccavo et al., 1994; Shelobolina et al., 2007). The waiting time can be longer than 200 seconds. In this study, we collected data from 83 *Geobacter* video files and 81 *Pelosinus* video files. Each of the video files consisted of information on multiple independent trajectories. From our observations and data collected, *Pelosinus* was found to be more mobile than *Geobacter*. MATLAB analysis of the video files recorded in *.fits format resulted in 6827 independent *Geobacter* trajectories and 20226 independent *Pelosinus*, trajectories. To fully represent the movement statistics and to compute breakthrough plots using ADE models, the mean displacements, variance, and diffusion coefficients were calculated when the number of cells in the data set were 50 or more. For *Geobacter*, the real path data consisted of trajectory runs ranging from 0.4s to 4.8s, and for *Pelosinus*, the real path data consisted of trajectory runs ranging from 0.4s to 37.6s. Both *Pelosinus* and *Geobacter* exhibits a linear relationship between δ^2 and time in the initial phase ($t < 10$ sec). *Pelosinus* shows a strong super-Fickian behavior after $t=10$ s. First-arrival time curves (breakthrough curves) of real path have relatively sharp peaks and narrower spread. The ADE results show that the model may perform better for *Geobacter* but is not adequate to capture the features of

Pelosinus transport. The mean values of the ADE breakthrough curves are much larger than mean values of real path for *Pelosinus*. The variance $(\delta)^2$ of the ADE is several times higher than the values obtained from the real path plots. Breakthroughs resulting from coupled CTRW model performs well at early time and for short control plane distances, especially for *Pelosinus*. However, it fails to match long distance travel and late time breakthrough concentration. The coupled and the uncoupled CTRW model tends to resemble each other for late-time concentration values. In summary, for *Geobacter*, none of the three models perform well on matching the real path breakthrough data; and for *Pelosinus*, the coupled CTRW model match real path breakthrough plots well for short control plane distance (i.e., when L is 10 or 20 microns in our study). The performance of ADE for *Geobacter* is however slightly better than the performance of CTRW models, highlighting the promise that ADE based models holds for systems with variance showing linear increase with time. The performance of CTRW models is significantly better than ADE for *Pelosinus*. The fact that none of the models used in this study was able to predict the real path breakthrough plots well points to the need of developing more sophisticated modeling tools for studying bacterial transport, possibly by constructing random walk processes that also takes correlation structures of motion dynamics into account.

The complexity of bacterial transport and behavior limits the performances of models. Only about 10% of *Geobacter* and 30% of *Pelosinus* passed the first control plane (located at 10 microns from the origin) in 250 seconds. A longer imaging time may be required to record more bacteria trajectories. Unlike conservative contaminants, the recovery of bacteria is significantly lower likely because some cells may stay dormant for

a long duration. It is challenging to obtain the overall motion characteristics of bacteria. Secondly, the transport features of bacteria may not be constant over a time period. *Pelosinus* shows linear relationship between δ^2 and time when t is smaller than 10s, and it displays super-Fickian behavior after $t = 10$ s. This phenomenon may be caused by groups of bacteria acting in widely different ways. Most bacteria rest during the imaging process, while a small group of bacteria moves at a low speed to short distances. A small fraction of highly active cells can travel at high speed to farther locations. CTRW and ADE models assume that every bacterium has a finite probability to carry long and short jumps and that each jump is independent from each other. However, our results suggest that this may not be true. Figure 18 show correlation of jump length and waiting time increments at step n and step $n+1$ for *Geobacter* and *Pelosinus*. The data are collected only from trajectories containing 2 or more consecutive jumps.

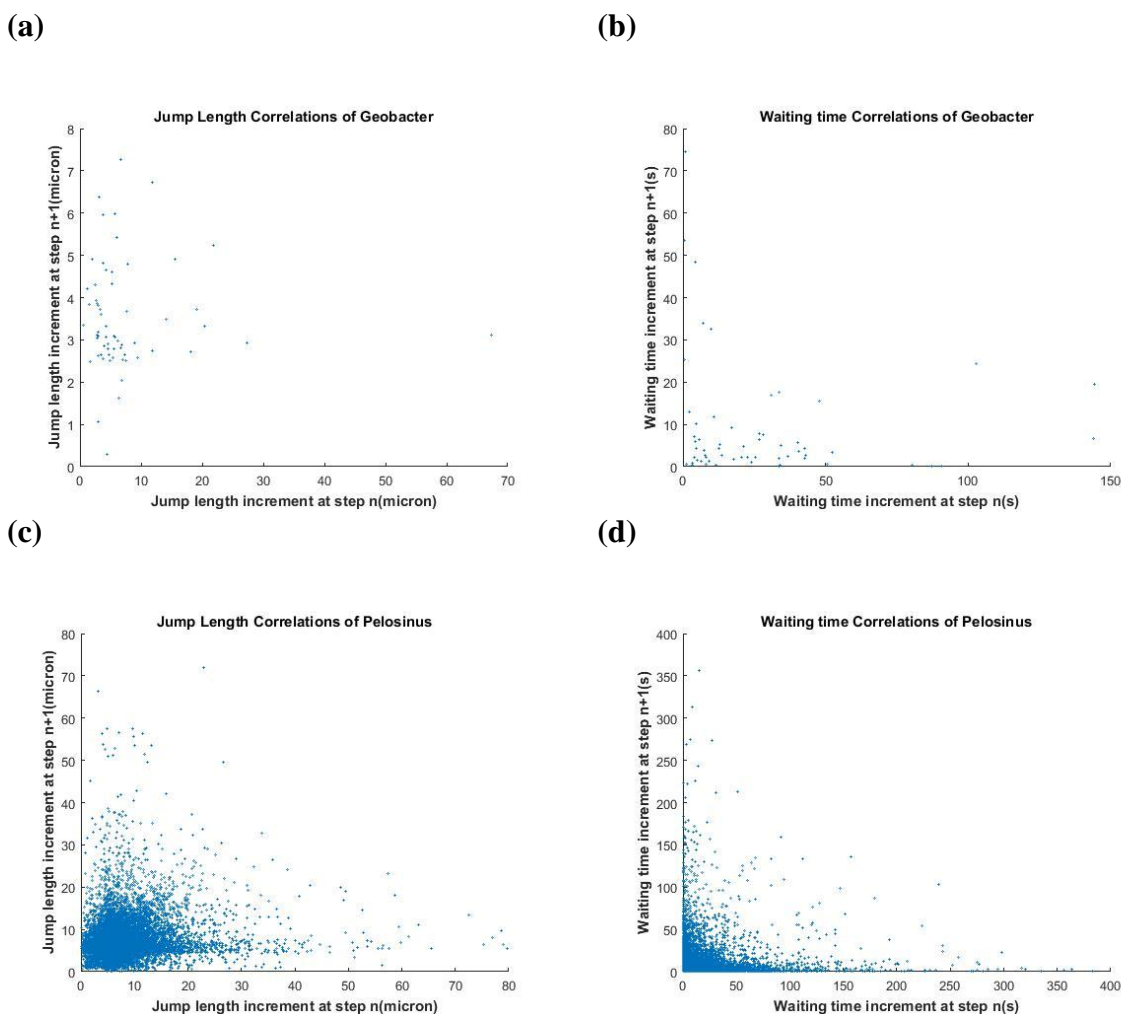


Figure 18: Jump length and waiting time correlations at step n and step $n+1$ for *Geobacter* and *Pelosinus*.

As Figure 18 suggests, for both *Geobacter* and *Pelosinus*, the correlation in jump length and waiting time increments is weak. The number of data points available for *Geobacter* is too sparse to make any definite conclusions. For *Pelosinus*, there are signs of positive correlation in jump lengths and negative correlation in waiting times (a longer waiting time leads to a shorter waiting time in the next step, and vice versa). It should also be noted here that the limited duration of recoding of trajectories may preclude collection of statistics of multiple successive long duration jumps.

A multi population model could probably aid in improving the model results by subdividing a species into multiple sub-groups based on their levels of relative activity. When distance to control plane (L) is small, both fast moving and slow moving groups are likely to reach the control planes making the coupled CTRW model yield good match with real path and peak values. However, when L is large, only the fast group of bacteria passes the control plane. In the CTRW model however, the waiting time and jump length probability densities are applied evenly to all groups of bacteria, which makes the model perform unsatisfactorily for farther control planes. Modelling bacterial transport is challenging which also requires better understanding of culturing protocols to ensure full motility of cells when injected in micro-models and observed under laser and microscopes. To further improve the models proposed in this research, we recommend classifying motile bacteria into multiple groups based on some of their motion attributes and construction of random walk processes by applying statistics of various groups separately. Examining the correlation structure of bacterial motion and incorporating them into random walk models should also help strengthen the modeling tools. In conclusion we found that:

1. Careful design of micro-model experiments to study motion dynamics of metal-reducing motile microorganisms allows us to process experimental data in MATLAB and determine statistical properties of motion attributes related to transport of bacterial cells.
2. The jump length and waiting time of *Geobacter* and *Pelosinus* follow distributions spread over several orders of magnitude. Both distributions show

- heavy tail behavior where the jump length can possibly exceed 100 microns and waiting time can possibly exceed 200 seconds.
3. The values of area under the breakthrough curves suggests that *Pelosinus* is more active than *Geobacter*.
 4. The plot of variance (δ^2) with respect to time shows that *Geobacter* and *Pelosinus* exhibit linear relationship during early times. The nature of this plot changes to super-Fickian behavior for *Pelosinus* in late time. Data for *Geobacter* trajectories only exists from 0.4s to 4.8s, hence the late time behavior of variance is unknown.
 5. ADE based model performs poorly on modeling *Geobacter* and *Pelosinus* transport. The breakthrough curves generated by these models have much larger mean and standard deviations when compared to the real path breakthrough data. The ADE model however appears to be a better prospect than CTRW models for the case of *Geobacter*.
 6. Uncoupled CTRW also lacks severely when modeling *Geobacter* breakthrough plots (especially at early time) and performs only marginally better at late time for *Pelosinus*.
 7. Coupled CTRW model is relatively good at modeling breakthroughs at short distances and at early times for both *Geobacter* and *Pelosinus*, but not at late time and long distances. Furthermore, the coupled CTRW model result matches *Pelosinus* real path plots better than *Geobacter*.
 8. Modeling transport of bacteria is a complicated topic of research as a species may consist of a mixture of multiple groups differing in their motion characteristics. Strong correlation structure in jump length and waiting time also likely exists that

were not considered in the CTRW and ADE models presented in this thesis.

Three-dimensional imaging techniques also have potential improving the model performance by recording true jump length and not their projections onto two-dimensional planes.

References

- Anderson, R. T., Vrionis, H. A., Ortiz-Bernad, I., Resch, C. T., Long, P. E., Dayvault, R., Karp, K., Marutzky, S., Metzler, D. R., Peacock, A., White, D. C., Lowe, M., and Lovley, D. R. (2003). Stimulating the in situ activity of *Geobacter* species to remove uranium from the groundwater of a uranium-contaminated aquifer. *Applied and environmental microbiology*, 69(10), 5884-5891.
- Berg, H. C. (2000). Motile behavior of bacteria. *Physics Today*, 53(1), 24-30.
- Berkowitz, B., Scher, H., and Silliman, S. E. (2000). Anomalous transport in laboratory-scale, heterogeneous porous media. *Water Resources Research*, 36(1), 149-158.
- Bradford, S. A., Wang, Y., Kim, H., Torkzaban, S., and Šimůnek, J. (2014). Modeling microorganism transport and survival in the subsurface. *Journal of environmental quality*, 43(2), 421-440.
- Caccavo, F., Lonergan, D. J., Lovley, D. R., Davis, M., Stolz, J. F., and McInerney, M. J. (1994). *Geobacter sulfurreducens* sp. nov., a hydrogen-and acetate-oxidizing dissimilatory metal-reducing microorganism. *Applied and environmental microbiology*, 60(10), 3752-3759
- Cates, M. E. (2012). Diffusive transport without detailed balance in motile bacteria: does microbiology need statistical physics? *Reports on Progress in Physics*, 75(4), 042601.
- Chirwa, E. N., and Wang, Y. T. (2000). Simultaneous chromium (VI) reduction and phenol degradation in an anaerobic consortium of bacteria. *Water Research*, 34(8), 2376-2384.
- Costanzo, A., Di Leonardo, R., Ruocco, G., and Angelani, L. (2012). Transport of self-propelling bacteria in micro-channel flow. *Journal of Physics: Condensed Matter*, 24(6), 065101.
- Davis, M. L., Mounteer, L. C., Stevens, L. K., Miller, C. D., and Zhou, A. (2011). 2D motility tracking of *Pseudomonas putida* KT2440 in growth phases using video microscopy. *Journal of bioscience and bioengineering*, 111(5), 605-611.

- Fang, Y., Yabusaki, S. B., Morrison, S. J., Amonette, J. P., and Long, P. E. (2009). Multicomponent reactive transport modeling of uranium bioremediation field experiments. *Geochimica et Cosmochimica Acta*, 73(20), 6029-6051.
- Franks, A. E., Glaven, R. H., and Lovley, D. R. (2012). Real-Time Spatial Gene Expression Analysis within Current-Producing Biofilms. *ChemSusChem*, 5(6), 1092-1098.
- Ford, R. M., and Harvey, R. W. (2007). Role of chemotaxis in the transport of bacteria through saturated porous media. *Advances in Water Resources*, 30(6), 1608-1617.
- Gharasoo, M., Centler, F., Fetzer, I., and Thullner, M. (2014). How the chemotactic characteristics of bacteria can determine their population patterns. *Soil Biology and Biochemistry*, 69, 346-358.
- Ginn, T. R., Wood, B. D., Nelson, K. E., Scheibe, T. D., Murphy, E. M., and Clement, T. P. (2002). Processes in microbial transport in the natural subsurface. *Advances in Water Resources*, 25(8), 1017-1042.
- Grate, J. W., Zhang, C., Wilkins, M., Warner, M.G., Anheier, N.C., Suter, J., Kelly, R. and Oostrom, M. (2013). Chemical sensing and imaging in microfluidic pore network structures relevant to natural carbon cycling and industrial carbon sequestration. In *Proc. of SPIE Vol* (Vol. 8725, pp. 872522-1).
- Haggerty, R., Wondzell, S. M., and Johnson, M. A. (2002). Power-law residence time distribution in the hyporheic zone of a 2nd-order mountain stream. *Geophysical Research Letters*, 29(13).
- Holmes, D. E., Finneran, K. T., O'neil, R. A., and Lovley, D. R. (2002). Enrichment of members of the family Geobacteraceae associated with stimulation of dissimilatory metal reduction in uranium-contaminated aquifer sediments. *Applied and Environmental Microbiology*, 68(5), 2300-2306.
- Holmes, D. E., O'neil, R.A., Vrionis, H.A., N'guessan, L.A., Ortiz-Bernad, I., Larrahondo, M.J., Adams, L.A., Ward, J.A., Nicoll, J.S., Nevin, K.P. and Chavan, M.A. (2007). Subsurface clade of Geobacteraceae that predominates in a diversity of Fe (III)-reducing subsurface environments. *The ISME journal*, 1(8), 663.

- Juwarkar, A. A., Singh, S. K., and Mudhoo, A. (2010). A comprehensive overview of elements in bioremediation. *Reviews in Environmental Science and bio/technology*, 9(3), 215-288.
- Kim, M. J., and Breuer, K. S. (2004). Enhanced diffusion due to motile bacteria. *Physics of Fluids (1994-present)*, 16(9), L78-L81.
- Kusy, K., and Ford, R. M. (2007). Monte Carlo simulations derived from direct observations of individual bacteria inform macroscopic migration models at granular porous media interfaces. *Environmental science & technology*, 41(18), 6403-6409.
- Lee, J. H., Fredrickson, J. K., Plymale, A. E., Dohnalkova, A. C., Resch, C. T., McKinley, J. P., and Shi, L. (2015). An autotrophic H₂-oxidizing, nitrate-respiring, Tc (VII)-reducing *Acidovorax* sp. isolated from a subsurface oxic-anoxic transition zone. *Environmental microbiology reports*, 7(3), 395-403.
- Lloyd, J. R., Sole, V. A., Van Praagh, C. V. G., and Lovley, D. R. (2000). Direct and Fe (II)-mediated reduction of technetium by Fe (III)-reducing bacteria. *Applied and environmental microbiology*, 66(9), 3743-3749.
- Licata, N. A., Mohari, B., Fuqua, C., and Setayeshgar, S. (2016). Diffusion of bacterial cells in porous media. *Biophysical journal*, 110(1), 247-257.
- Liu, J., Ford, R. M., and Smith, J. A. (2011). Idling time of motile bacteria contributes to retardation and dispersion in sand porous medium. *Environmental science & technology*, 45(9), 3945-3951.
- Mishra, S., Jyot, J., Kuhad, R. C., and Lal, B. (2001). In situ bioremediation potential of an oily sludge-degrading bacterial consortium. *Current microbiology*, 43(5), 328-335.
- McLean, J. S., Pinchuk, G.E., Geydebekht, O.V., Bilskis, C.L., Zakrajsek, B.A., Hill, E.A., Saffarini, D.A., Romine, M.F., Gorby, Y.A., Fredrickson, J.K. and Beliaev, A.S. (2008). Oxygen-dependent autoaggregation in *Shewanella oneidensis* MR-1. *Environmental microbiology*, 10(7), 1861-1876.
- Murphy, E. M., and Ginn, T. R. (2000). Modeling microbial processes in porous media. *Hydrogeology Journal*, 8(1), 142-158.
- Pearce, C. I., Wilkins, M. J., Zhang, C., Heald, S. M., Fredrickson, J. K., and Zachara, J. M. (2012). Pore-scale characterization of biogeochemical controls on iron and

- uranium speciation under flow conditions. *Environmental science & technology*, 46(15), 7992-8000.
- Pickman, L. (2015). *Development of a Continuous Time Random Walk Model for Fractured Media—Site Characterization and Comparison with Discrete Fracture Network Method*. University of Nevada, Reno.
- Sauty, J. P. (1980). An analysis of hydrodispersive transfer in aquifers. *Water Resources Research*, 16(1), 145-158.
- Selmeçzi, D., Mosler, S., Hagedorn, P. H., Larsen, N. B., and Flyvbjerg, H. (2005). Cell motility as persistent random motion: theories from experiments. *Biophysical journal*, 89(2), 912-931.
- Shelobolina, E. S., Nevin, K.P., Blakeney-Hayward, J.D., Johnsen, C.V., Plaia, T.W., Krader, P., Woodard, T., Holmes, D.E., VanPraagh, C.G. and Lovley, D.R. (2007). *Geobacter pickeringii* sp. nov., *Geobacter argillaceus* sp. nov. and *Pelosinus fermentans* gen. nov., sp. nov., isolated from subsurface kaolin lenses. *International Journal of Systematic and Evolutionary Microbiology*, 57(1), 126-135.
- Sherwood, J. L., Sung, J. C., Ford, R. M., Fernandez, E. J., Maneval, J. E., and Smith, J. A. (2003). Analysis of bacterial random motility in a porous medium using magnetic resonance imaging and immunomagnetic labeling. *Environmental science & technology*, 37(4), 781-785.
- Singh, R., and Olson, M. S. (2011). Transverse mixing enhancement due to bacterial random motility in porous microfluidic devices. *Environmental science & technology*, 45(20), 8780-8787.
- Singh, R., and Olson, M. S. (2012). Transverse chemotactic migration of bacteria from high to low permeability regions in a dual permeability microfluidic device. *Environmental science & technology*, 46(6), 3188-3195.
- Taktikos, J., Stark, H., and Zaburdaev, V. (2013). How the motility pattern of bacteria affects their dispersal and chemotaxis. *PloS one*, 8(12), e81936.
- Tufenkji, N. (2007). Modeling microbial transport in porous media: Traditional approaches and recent developments. *Advances in water resources*, 30(6), 1455-1469.

- Vidali, M. (2001). Bioremediation. an overview. *Pure and Applied Chemistry*, 73(7), 1163-1172.
- Viswanathan, G. M., Buldyrev, S. V., Havlin, S., Da Luz, M. G. E., Raposo, E. P., and Stanley, H. E. (1999). Optimizing the success of random searches. *Nature*, 401(6756), 911-914.
- Vrionis, H. A., Anderson, R.T., Ortiz-Bernad, I., O'Neill, K.R., Resch, C.T., Peacock, A.D., Dayvault, R., White, D.C., Long, P.E. and Lovley, D.R. (2005). Microbiological and geochemical heterogeneity in an in situ uranium bioremediation field site. *Applied and environmental microbiology*, 71(10), 6308-6318.
- Wang, M., and Ford, R. M. (2009). Transverse bacterial migration induced by chemotaxis in a packed column with structured physical heterogeneity. *Environmental science & technology*, 43(15), 5921-5927.
- Wang, X., Long, T., and Ford, R. M. (2012). Bacterial chemotaxis toward a NAPL source within a pore-scale microfluidic chamber. *Biotechnology and bioengineering*, 109(7), 1622-1628.
- Watari, N., and Larson, R. G. (2010). The hydrodynamics of a run-and-tumble bacterium propelled by polymorphic helical flagella. *Biophysical journal*, 98(1), 12-17.
- Wu, J., Wu, X., and Lin, F. (2013). Recent developments in microfluidics-based chemotaxis studies. *Lab on a Chip*, 13(13), 2484-2499.
- Yabusaki, S. B., Fang, Y., Williams, K.H., Murray, C.J., Ward, A.L., Dayvault, R.D., Waichler, S.R., Newcomer, D.R., Spane, F.A. and Long, P.E. (2011). Variably saturated flow and multicomponent biogeochemical reactive transport modeling of a uranium bioremediation field experiment. *Journal of contaminant hydrology*, 126(3), 271-290.
- Zhao, J., Fang, Y., Scheibe, T. D., Lovley, D. R., and Mahadevan, R. (2010). Modeling and sensitivity analysis of electron capacitance for *Geobacter* in sedimentary environments. *Journal of contaminant hydrology*, 112(1), 30-44.
- Zhao, J., Scheibe, T. D., and Mahadevan, R. (2011). Model-based analysis of the role of biological, hydrological and geochemical factors affecting uranium bioremediation. *Biotechnology and bioengineering*, 108(7), 1537-1548.

Appendix: MATLAB Code

A1. Video files extraction and processing section

A1.1 Main loop

```

clear; clc; close all;

position_video = 0; %do you want videos of positions? 1=yes 0=no
path_video = 0; %do you want videos of paths? 1=yes 0=no

pathname = '*'; % files path
foldername = {'geobacter', 'pelosinus'}; % folders where store files
under path

bact_len = {[.15 .23], [.15 .54]}; %cell with range of bacterial
lengths for each species
vel_vec = [1.2 .8]; %average velocity of species
alpha = [0.7 .4]; %ratio of radius for below average velocity bacteria
to radius for above average velocity bacteria
rad_active = [3.9 4.0]; %radius for above average velocity bacteria
rad_idle = alpha. *rad_active; %radius for below average velocity
bacteria

for species_iter = 1:2
    addpath([pathname foldername{species_iter}]) %add folder to path

    %find which files are in the folder
    files = fopen('zfiles.txt'); %open text file which stores all file
names
    C = textscan(files, '%s%s%s%s', 'headerlines', 7); %read text file
with filenames
    fclose(files); %close the text file
    filenames = C{5}; %just the names of the files

    %go through all of the files

    for file_iter = 1:length(filenames)
        if strcmp(filenames{file_iter}, 'zfiles.txt') %only read video
files
            break
        end

        C =
textscan(filenames{file_iter}, '%s%s%s%s%s', 'Delimiter', '_', 'E
mptyValue', -Inf); %read filename{f}
        str = char (C {1}); %convert magnification to string
        len = length(str); %find length of string
        m = str2double (str(1: len-1)); %magnification
        str = char (C {3}); %convert frequency to string
        freq = str2double(str(1:3)); %frequency
        freq = freq/100;
        filename = filenames{file_iter}; %filename used to save data
        len = length(filename); %length of filename
        filename = filename (1: len-5); %cut of '.fits'

```

```

        diameter_range = bact_len{species_iter}*m; %diameter range in
pixels
        diameter = mean(diameter_range); %minimum distance between
bacteria
        body_length=diameter;
        avg_vel = vel_vec(species_iter); %average velocity of species
        radius_active = rad_active(species_iter); %radius for above
average velocity bacteria
        radius_idle = rad_idle(species_iter); %radius for below average
velocity bacteria

get_positions_new(filename,bact_len{species_iter}*m,position_video)
    % Call osition determination function
get_paths_kinematic_active_newest(filename,diameter,avg_vel,radius_acti
ve,radius_idle,freq,path_video);
    % Call path determination function
get_jump_length_waiting_time_distribution(filename,m,freq,body_length)
    % Call Jump length, waiting time extraction and statistics
calculation function

end
    rmpath([pathname foldername{species_iter}]) %remove folder from
path

end

```

A1.2 Position determination function

```
function get_positions_new(filename, len, vid)

im_data = fitsread([filename '.fits']); %load video
if vid
    V = VideoWriter([filename '_new.avi']); %create video object
    V.FrameRate = 8; %set frame rate
    open(V); %open video object
end

pos = cell(1, length(im_data(1,1,:))-2); %create a cell to hold particle
positions at each timestep
old_im = im_data(:,:,1); %image from last time step
sz = size(old_im); %size of a single image

%main loop
mean_im = sum(im_data,3)/1000; %average intensity of video

for t_iter = 2:length(im_data(1,1,:))-1

    %filter images
    current_im = im_data(:,:,t_iter); %image from current time step
    next_im = im_data(:,:,t_iter+1); %image at next time step
    time_diff_fwd = next_im-current_im; %forward difference in time
    time_diff_bkwd = current_im-old_im; %backward difference in time
    eps_f = 3*sqrt(var(time_diff_fwd(:))); %forward difference cutoff
in time
    eps_b = 3*sqrt(var(time_diff_bkwd(:))); %backard difference cutoff
in time
    f_filter = abs(time_diff_fwd-mean(time_diff_fwd(:)))>eps_f; %filter
out noise in forward difference
    b_filter = abs(time_diff_bkwd-mean(time_diff_bkwd(:)))>eps_b;
%filter out noise in backward difference

    mean_diff = current_im-mean_im; %difference between the current
image and the average intensity
    eps_m = 3*sqrt(var(mean_diff(:))); %intensity difference cutoff in
space
    m_filter = abs(mean_diff-mean(mean_diff(:)))>eps_m; %filter out
noise in space
    t_filter = f_filter | b_filter; %passes through one of the two time
filters

    m_neigh = m_filter + [zeros(1,sz(2)); m_filter(1:end-1,:)] +
[m_filter(2:end,:); zeros(1,sz(2))] +...
    [zeros(sz(1),1) m_filter(:,1:end-1)] + [m_filter(:,2:end)
zeros(sz(1),1)]; %sum of the number of neighboring pixels that pass
through spatial filter

    t_neigh = t_filter + [zeros(1,sz(2)); t_filter(1:end-1,:)] +
[t_filter(2:end,:); zeros(1,sz(2))] +...
    [zeros(sz(1),1) t_filter(:,1:end-1)] + [t_filter(:,2:end)
zeros(sz(1),1)]; %sum of the number of neighboring pixels that pass
through time filter
```

```

    [I,J] = find(m_neigh>min(ceil(len(1)/3),2) &
t_neigh>min(ceil(len(1)/3),2)); %this sum must be at least 2 for both
filters

    %group pixels into bacteria
    iter = 1; %iteration index for while loop
    x_loc = -1000; y_loc = 1; n_used = 1; %initialize variables
assigned in while loop

for i = 1:length(I)
    ind = find(abs(I(i)-I)<= len(2) & abs(J(i)-J)<= len(2));
    if length(ind)< len(1) %make sure there are enough pixels to take
up smallest bacteria area
        I(i) = NaN;
    end
end
J(isnan(I)) = [];
I(isnan(I)) = [];
while ~isempty(I)
    ind = find(abs(I(1)-I)<= len(2) & abs(J(1)-J)<= len(2));
    if length(ind)>= len(1) %make sure there are enough pixels to
take up average bacteria area
        x_loc(iter) = mean(J(ind)); %x location of center of
bacteria
        y_loc(iter) = mean(I(ind)); %y location of center of
bacteria
        n_used(iter) = length(ind); %number of pixels in location
        iter = iter+1; %continue loop
    end
    I(ind) = []; %pixels have already been accounted for
    J(ind) = []; %pixels have already been accounted for
end
%ensure only one x-y location exists for each bacterium
for i = 1:length(x_loc)
    dist = sqrt((x_loc-x_loc(i)).^2+(y_loc-y_loc(i)).^2); %distance
between positions
    ind = find(dist<=1.5*len(2));
    weights = n_used(ind)/sum(n_used(ind)); %weights to be used
when averaging positions
    x_loc(i) = sum(x_loc(ind).*weights); %merged x location of
center of bacteria
    y_loc(i) = sum(y_loc(ind).*weights); %merged y location of
center of bacteria
    ind(ind==i) = []; %get rid of the index of the current (merged)
position
    x_loc(ind) = -1000; %assign position outside of the domain for
positions that got merged (and need to go away)
end
y_loc(x_loc<0) = []; %kill locations outside of domain
x_loc(x_loc<0) = []; %kill locations outside of domain

    pos{t_iter-1} = struct('x',x_loc,'y',y_loc); %place positions
inside cell

    old_im = current_im; %get ready for next iteration

```

```
    if vid && t_iter<101
        %write video
        f = figure(1);
        pcolor(current_im); shading interp; axis square; axis off;
    colormap gray
        hold on; scatter(x_loc,y_loc,25,'w'); hold off;
        fr = getframe(f);
        writeVideo(V,fr);
    end
end
if vid
    close(V) %close video object
end
save([filename 'pos_mean.mat'],'pos'); %save particle positions
```

A1.3 Path determination function

```

function
get_paths_kinematic_active_newest(filename,diameter,vel,radius_active,r
adius_idle,freq,vid)

load([filename 'pos_mean.mat']); %load position data
if isempty(pos); return; end %if filename wasn't a video
full = cellfun('isempty',pos);
if vid
    im_data = fitsread([filename '.fits']); %load video
    V = VideoWriter([filename 'paths_active.avi']); %create video
object
    V.FrameRate = 8; %set frame rate
    open(V); %open video object
end
x_loc = pos{1}.x; %x locations for the current timestep
y_loc = pos{1}.y; %y locations for the current timestep
n_path = length(x_loc); %number of paths
x_paths = NaN(length(pos),n_path); %x coordinates for each path
y_paths = NaN(length(pos),n_path); %y coordinates for each path
x_paths(1,:) = x_loc; %starting x location of each path
y_paths(1,:) = y_loc; %starting y location of each path
dx = zeros(size(x_loc)); %last x step
dy = zeros(size(y_loc)); %last y step

%main loop
for t_iter = 2:length(pos)-sum(full)
    x_loc = pos{t_iter}.x; %x locations for the current timestep
    y_loc = pos{t_iter}.y; %y locations for the current timestep
    x_plus_dx = x_paths(t_iter-1,:)+dx; %x location at last timestep
plus dx
    y_plus_dy = y_paths(t_iter-1,:)+dy; %y location at last timestep
plus dy
    D = zeros(length(x_loc),length(x_plus_dx)); %initialize distance
    last_jump = sqrt(dx.^2 + dy.^2); %length of last jump
    for i = 1:length(x_loc)
        D(i,:) = sqrt((x_loc(i)-x_plus_dx).^2+(y_loc(i)-y_plus_dy).^2);
%distance between new and predicted positions
    end
    is_idle = last_jump<=diameter*vel/freq;% & last_jump>0; %particles
that have been around for at least a step and have low velocity (below
mu + 3sigma)
    D(D>diameter*radius_active/freq) = NaN; %indices of positions
outside of active search radius
    D(D(:,is_idle)>diameter*radius_idle/freq) = NaN; %indices of
positions outside of idle search radius
    boo = ~isnan(D); %boolean variable telling if particles are within
acceptable distance
    num_i = sum(boo,2); %number of old particles within range of each
new particle
    num_j = sum(boo,1); %number of new particles witing range of each
old particle

    %case 1: only one old particle in range

```

```

    ind = find(num_i==1); %indices of particles with only one old
particle in range
    ind2 = find(sum(boo(ind,:))>1); %indices of old particles with over
one new particle in range
    for i = 1:length(ind2)
        ind_new = find(D(ind,ind2(i))>min(D(ind,ind2(i)))); %only the
closest particle is matched with the old particle
        num_i(ind(ind_new)) = 0; %other particles are considered to be
new
        D(ind(ind_new),ind2(i)) = NaN; %so they are out of range
        num_j(ind2(i)) = NaN;%num_j(ind2(i))-1;
        boo(ind(ind_new),ind2(i)) = 0;
        ind(ind_new) = []; %and they will be dealt with in case 3
    end
    old_boo = boo(ind,:); %boolean variable telling if new particles
are within acceptable distance
    for i = 1:length(ind)
        x_paths(t_iter,old_boo(i,:)) = x_loc(ind(i)); %assign location
        y_paths(t_iter,old_boo(i,:)) = y_loc(ind(i)); %assign location
        num_j(old_boo(i,:)) = NaN;
    end
    D(ind,:) = NaN; %treat old particle as out of range

    %case 2: only one new particle in range
    ind = find(num_j==1); %indices of particles with only one new
particle in range
    ind2 = find(sum(boo(:,ind),2)>1); %indices of new particles with
over one old particle in range
    for i = 1:length(ind2)
        ind_new = find(D(ind2(i),ind)>min(D(ind2(i),ind))); %only the
closest particle is matched with the new particle
        num_j(ind(ind_new)) = 0; %other particles are considered to
have disappeared
        D(ind2(i),ind(ind_new)) = NaN; %so they are out of range
        num_i(ind2(i)) = NaN;
        boo(ind2(i),ind(ind_new)) = 0;
        ind(ind_new) = []; %and they will be dealt with in case 5
    end
    new_boo = boo(:,ind); %boolean variable telling if old particles
are within acceptable distance
    for j = 1:length(ind)
        x_paths(t_iter,ind(j)) = x_loc(new_boo(:,j)); %assign location
        y_paths(t_iter,ind(j)) = y_loc(new_boo(:,j)); %assign location
    end
    D(:,ind) = NaN; %treat new particle as out of range

    %case 3: new_i is a new particle
    ind = find(num_i==0); %indices of new particles
    len = length(ind); %number of new particles
    x_paths(t_iter,n_path+(1:len)) = x_loc(ind); %assign location
    y_paths(t_iter,n_path+(1:len)) = y_loc(ind); %assign location
    n_path = n_path+len; %now there are len more paths

    %case 4: multiple choices: sort by distance
    if ~isempty(D)
        [new,old] = find(D==min(D(:)),1); %closest new - old pair

```

```

while ~isempty(new)
    x_paths(t_iter,old) = x_loc(new); %assign location
    y_paths(t_iter,old) = y_loc(new); %assign location
    D(new,:) = NaN; %treat particle as out of range
    D(:,old) = NaN; %treat particle as out of range
    [new,old] = find(D==min(D(:)),1); %closest new - old pair
end
end

%case 5: old_j disappeared
disappeared = boolean([num_j==0 zeros(1,n_path-length(num_j))]);
%whether or not particles have disappeared
if t_iter>ceil(10*freq) %whether or not particles have been missing
for at least a 10 seconds
    ten_second_check = x_paths(t_iter-1,:)-x_paths(t_iter-
ceil(10*freq),:)==0;
else
    ten_second_check = boolean(zeros(1,n_path));
end
x_paths(t_iter,disappeared & ~ten_second_check) = x_paths(t_iter-
1,disappeared & ~ten_second_check);
y_paths(t_iter,disappeared & ~ten_second_check) = y_paths(t_iter-
1,disappeared & ~ten_second_check);
if sum(ten_second_check)>0
    x_paths((t_iter-ceil(10*freq)+1):t_iter,ten_second_check) =
NaN; %the particle disappeared for over 10 seconds, so cut off the
position it has been waiting at
    y_paths((t_iter-ceil(10*freq)+1):t_iter,ten_second_check) =
NaN; %the particle disappeared for over 10 seconds, so cut off the
position it has been waiting at
end
%assign new dx and dy
x_paths(x_paths==0) = NaN;
y_paths(y_paths==0) = NaN;
dx = [dx zeros(1,n_path-length(dx))];
dy = [dy zeros(1,n_path-length(dy))];
dx(~disappeared) = x_paths(t_iter,~disappeared)-x_paths(t_iter-
1,~disappeared);
dy(~disappeared) = y_paths(t_iter,~disappeared)-y_paths(t_iter-
1,~disappeared);
dx(isnan(dx)) = 0;
dy(isnan(dy)) = 0;

if vid && t_iter<101
    lag = min(t_iter-1,ceil(10*freq));
    %write video
    f = figure(1);

    pcolor(im_data(:,:,t_iter+1)); shading interp; axis square;
axis off; colormap gray
    hold on; h = scatter(x_plus_dx(~is_idle &
~disappeared(1:length(is_idle))),y_plus_dy(~is_idle &
~disappeared(1:length(is_idle))), 'r');
    currentunits = get(gca, 'Units');
    set(gca, 'Units', 'Points');
    axpos = get(gca, 'Position');

```

```

        set(gca, 'Units', currentunits);
        s = diameter*radius_active/freq;
        markerWidth = s/diff(xlim)*axpos(3); % Calculate Marker width
in points
        set(h, 'SizeData', markerWidth^2)
        h = scatter(x_plus_dx(is_idle &
~disappeared(1:length(is_idle))),y_plus_dy(is_idle &
~disappeared(1:length(is_idle))), 'r');
        currentunits = get(gca, 'Units');
        set(gca, 'Units', 'Points');
        axpos = get(gca, 'Position');
        set(gca, 'Units', currentunits);
        s = diameter*radius_idle/freq;
        markerWidth = s/diff(xlim)*axpos(3); % Calculate Marker width
in points
        set(h, 'SizeData', markerWidth^2)
        plot(x_paths(t_iter-lag:t_iter,:),y_paths(t_iter-
lag:t_iter,:), 'r');
%         scatter(x_paths(t_iter,1:n_path-len),y_paths(t_iter,1:n_path-
len),3,'filled','y');
        scatter(x_paths(t_iter,(n_path-
len+1):end),y_paths(t_iter,(n_path-len+1):end),3,'filled','y'); hold
off
        fr = getframe(f);
        writeVideo(V,fr);
    end
end
if vid
    close(V) %close video object
end
save([filename
'_paths_active.mat'],'x_paths','y_paths')%, 'disappeared','D_orig','D_1'
,'D_2','D_3','num_i','num_j')
end

```

A1.4 Jump length, waiting time and statistics calculation and output function

```

function
get_jump_length_waiting_time_distribution(filename,m,freq,body_length)
load([filename '_paths_active.mat']);
[r,c]=size(x_paths);
% get size of x_path, c is the number of trajectories
D=zeros(r-1,1);%distance matrix
A=zeros(r-1,1);%turn angle matrix
jump_step=1; % number of jump
waiting_interval=1; %number of waiting interval;
delta_x=0;delta_y=0;%jump length of each jump step
max_turning_A=pi/36;% max turning angle that is considered as 1 jump(10
degree)
D_Max=0; % record number of frames of longest trajectories

for i=1:c % different trajectories

    for j=1:(r-1) % from first to last frame
D(j,1:2,i)=[x_paths(j,i),y_paths(j,i)];
% distance bewteen each two frames
A(j,1,i)=atan2((y_paths(j+1,i)-y_paths(j,i)),(x_paths(j+1,i)-
x_paths(j,i)));
    % angle bewteen each two frame
    end

D_2=D(:, :, i);
D_2=D_2(all(~isnan(D_2),2), :);
%remove NaN in D matrix and get distance array for one trajectory
A_2=A(~isnan(A(:,1,i)),1,i);
%remove NaN and get Angle array for one trajectory
[r_D,c_D]=size(D_2);

for l=1:(r_D-1)
    D_2(l,3)=sqrt((D_2(l+1,1)-D_2(l,1)).^2+(D_2(l+1,2)-D_2(l,2)).^2);
end
% calculate distance of each jump over time

if r_D<=10
    ;
    % make sure trajectory is long enough
else
    for h=1:(r_D-1)
        if D_2(h,3)<=body_length;
            D_2(h,4)=0;
            D_2(h,3)=0;
            A_2(h,1)=0;
            A_2(h,2)=0;
        else
            D_2(h,4)=1;
            A_2(h,2)=1;
        end
    end
    % use 0/1 as indicator of stay or moving. If the move distance is
smaller

```

```

    % than one body length, bacteria is considered as staying.

a=1; %position variable.

for k=1:(r_D-2)
if ((A_2(k+1,2)-A_2(k,2))==0)&&(A_2(k+1,2)==1)
    if abs(A_2(k+1,1)-A_2(k,1))<=max_turning_A;
        ;
        % if two frame are both moving and turning angle is smaller than
        % max_turning_angle, they are in same trajectory
    else
        delta_x=D_2(k+1,1)-D_2(a,1);
        delta_y=D_2(k+1,2)-D_2(a,2);
        J(jump_step,:)= [jump_step,delta_x,delta_y];
        jump_step= jump_step + 1;
        W(waiting_interval,:)= [waiting_interval,k+1-a];
        waiting_interval= waiting_interval + 1;
        a=k+1;
    end
    % if the turning angle is bigger than maximum turning angle, then
    % they are considered as a different jump, jump length and waiting time
    % are restored in J and W matrix
end

    if ((A_2(k+1,2)-A_2(k,2))==1);
        W(waiting_interval,:)= [waiting_interval,k+1-a];
        waiting_interval= waiting_interval + 1;
        delta_x=D_2(k+1,1)-D_2(a,1);
        delta_y=D_2(k+1,2)-D_2(a,2);
        J(jump_step,:)= [jump_step,delta_x,delta_y];
        jump_step= jump_step + 1;
        a=k+1;
    end
    % if the turning angle is changing from 0 to 1, it means a jump
    % happens. Jump length and waiting time are restored in j and W.
end
end
end

if (exist('W'))
W(:,2)=W(:,2).*(1/freq); %transfer waiting time to sec
J(:,2:3)=J(:,2:3).*(13/m); % transfer jump length to micron
J(:,4)=sqrt(J(:,2).^2+J(:,3).^2); %calculate each jump length
M(:,1:2)=J(:,2:3);
M(:,3)=W(:,2);
save([filename '_M.mat'], 'M');

N=zeros(D_Max,c+1);
N(:,1)=1:D_Max;
% N is used to store information of each trajectories

stat=zeros(D_Max,5);
stat(:,1)=N(:,1)*(1/freq);
% stat is used to store statistics

for i=1:c

```

```
D_2=D(:, :, i);
D_2=D_2(all(~isnan(D_2), 2), :);
[r, co]=size(D_2);
for j=1:r
N(j, i+1)=sqrt((D_2(j, 1)-D_2(1, 1)).^2+(D_2(j, 2)-D_2(1, 2)).^2);
end
end
%calculate mean displacement of each time step

for i=1:D_Max
stat(i, 2)=nnz(N(i, 2:(c+1)));
stat(i, 3)=mean(nonzeros(N(i, 2:(c+1))));
stat(i, 4)=std(nonzeros(N(i, 2:(c+1))));
stat(i, 5)=stat(i, 4)/(2*stat(i, 1));
end

save([filename '_sigma.mat'], 'stat');
end
end
```

A2. Data merge and interpolation section

A2.1 Jump length and waiting time merge

```

clear
clc
pathname = '*';
foldername = {'geobacter_M', 'pelosinus_M'};
%you will still need to make the outer loop that iterates through the
files
file = cell(0); %placeholder for list of filenames
M_merged = []; %merged M matrix

for species_iter = 1:2
addpath([pathname foldername{species_iter}]) %add folder to path

    %find which files are in the folder
    file = fopen('zfiles.txt'); %open text file which stores all
filenames
    C = textscan(file, '%s%s%s%s%s', 'headerlines', 7); %read text file
with filenames
    fclose(file); %close the text file
    filenames = C{5}; %just the names of the files

for file_iter = 1:length(filenames)
    if strcmp(filenames{file_iter}, 'zfiles.txt') %only read video files
        break
    end
    current = load(filenames{file_iter}); %current file that is being
merged
    M_merged = [M_merged; current.M]; %concatenate current M matrix
end

save([foldername{species_iter}], 'M_merged');
end

```

A2.2 Statistics interpolation

```

clear
clc
pathname = 'D:\3rd visit\';
foldername = {'geobacter_stat','pelosinus_stat'};
%you will still need to make the outer loop that iterates through the
files
file = cell(0); %placeholder for list of filenames
sigma_sq_n = zeros(100,1); %sigma squared times n for all files
interpolated to a time vector t
mean_n = zeros(100,1); %mean x position times n for all files
interpolated to a time vector t
n = zeros(100,1); %number of particles for each time
t(:,1) = % create time vector as needed

for species_iter = 1:2
addpath([pathname foldername{species_iter}]) %add folder to path
    %find which files are in the folder
    file = fopen('zfiles.txt'); %open text file
    C = textscan(file, '%s%s%s%s%s%s%s', 'headerlines',7); %read text
file with filenames
    fclose(file); %close the text file
    filenames = C{5}; %just the names of the files

for file_iter = 1:length(file)

if strcmp(filenames{file_iter}, 'zfiles.txt') %only read video files
        break
    end

%(must start before lowest time you have data for and end after highest
time for interpolation to work)

    current = load(filenames{file_iter}); %current file that is being
merged
    %this is where you will put the times that you want sigma squared at
    t_current = current.stat(:,1); %where you will put in the time
vector from the current file
    n_current = current.stat(:,2); %where you will put in the number of
particles at each time from the current file
    mean_current = current.stat(:,3); %mean times n
    sigma_current = current.stat(:,4).^2; %where you will put in the
sigma squared vector from the current file
    sigma_n_interp = interp1(t_current, sigma_current.*n_current, t);
%interpolate sigma squared times n
    mean_n_interp = interp1(t_current, mean_current.*n_current, t);
%interpolate mean times n
    n_interp = interp1(t_current, n_current, t); %interpolate n
    mean_n = mean_n + mean_n_interp; %add interpolated mean
    sigma_sq_n = sigma_sq_n + sigma_n_interp; %add interpolated sigma
squared times n
    n = n + n_interp; %add interpolated n
end

mean_position = mean_n./n; %the merged mean at each time t
sigma_sq = sigma_sq_n./n; %the merged sigma squared at each time t

```

```
stat(:,1)=t;  
stat(:,2)=n;  
stat(:,3)=mean_position;  
stat(:,4)=sigma_sq;  
stat(:,5)=stat(:,4)./(2*stat(:,1));  
save([foldername{species_iter}], 'stat');  
end
```

A3. Model stimulations, normalization and graphing

A3.1 Real path

```

clear
clc
pathname = 'D:\3rd visit\';
foldername = {'geobacter_path', 'pelosinus_path'};
%you will still need to make the outer loop that iterates through the
files
file = cell(0); %placeholder for list of filenames
n=0;
t_10=[];t_20=[];t_30=[];t_40=[];t_50=[];t_60=[];

for species_iter = 1:2
addpath([pathname foldername{species_iter}]) %add folder to path

    %find which files are in the folder
    file = fopen('zfiles.txt'); %open text file
    C = textscan(file, '%s%s%s%s', 'headerlines', 7); %read text file
with filenames
    fclose(file); %close the text file
    filenames = C{5}; %just the names of the files

for file_iter = 1:length(filenames)

if strcmp(filenames{file_iter}, 'zfiles.txt') %only read video files
    break
end

C =
textscan(filenames{file_iter}, '%s%s%s%s%s', 'Delimiter', '_', 'EmptyVa
lue', -Inf); %read filename{f}
    str = char(C{1}); %convert magnification to string
    len = length(str); %find length of string
    m = str2double(str(1:len-1)); %magnification
    str = char(C{3}); %convert frequency to string
%    len = length(str); %find length of string
    freq = str2double(str(1:3)); %frequency
    freq = freq/100;
    current = load(filenames{file_iter}); %current file that is being
merged
    [r,c]=size(current.x_paths);
% get size of x_path, r is the time duration,c is the number of
% trajectories
D=zeros(r-1,1);%distance matrix

for i=1:c % different trajectories
    for j=1:(r-1) % frame
D(j,1:2,i)=[current.x_paths(j,i), current.y_paths(j,i)];
end

D_2=D(:, :, i);
D_2=D_2(all(~isnan(D_2), 2), :);
%remove NaN in D matrix and get distance array for one trajectory
[r_D,c_D]=size(D_2);

```

```

%get size of D_2, r_D is the number of frame
if r_D<=10
    continue
else %concatenate current M matrix
    D_2(:,3)=(13/m)*sqrt((D_2(:,1)-D_2(1,1)).^2+(D_2(:,2)-
D_2(1,2)).^2);
    n=n+1;
t_10(n,1)=250;
t_20(n,1)=250;
t_30(n,1)=250;
t_40(n,1)=250;
t_50(n,1)=250;
t_60(n,1)=250;
% initial first arrival time where 250 is the longest video duration

for k=2:r_D

if D_2(k,3)>=10 & D_2(k-1,3)<10
    t1=k*(1/freq)+1*(1/freq)*(1-(D_2(k,3)-10)/(D_2(k,3)-D_2(k-
1,3)));
    if t1<t_10(n,1)
        t_10(n,1)=t1;
    end
end

if D_2(k,3)>=20 & D_2(k-1,3)<20
    t2=k*(1/freq)+1*(1/freq)*(1-(D_2(k,3)-20)/(D_2(k,3)-D_2(k-
1,3)));
    if t2<t_20(n,1)
        t_20(n,1)=t2;
    end
end

if D_2(k,3)>=30 && D_2(k-1,3)<30
    t3=k*(1/freq)+1*(1/freq)*(1-(D_2(k,3)-30)/(D_2(k,3)-D_2(k-
1,3)));
    if t3<t_30(n,1)
        t_30(n,1)=t3;
    end
end

if D_2(k,3)>=40 && D_2(k-1,3)<40
    t4=k*(1/freq)+1*(1/freq)*(1-(D_2(k,3)-40)/(D_2(k,3)-D_2(k-
1,3)));
    if t4<t_40(n,1)
        t_40(n,1)=t4;
    end
end

if D_2(k,3)>=50 && D_2(k-1,3)<50
    t5=k*(1/freq)+1*(1/freq)*(1-(D_2(k,3)-50)/(D_2(k,3)-D_2(k-
1,3)));
    if t5<t_50(n,1)
        t_50(n,1)=t5;
    end
end

```

```
if D_2(k,3)>=60 && D_2(k-1,3)<60
    t6=k*(1/freq)+1*(1/freq)*(1-(D_2(k,3)-60)/(D_2(k,3)-D_2(k-
1,3)));
    if t6< t_60(n,1)
        t_60(n,1)=t6;
    end
end
    % check each trajectory frame by frame to obtain first arrival
time(minimal)reaching each control plane.
end
end
end
end
save([foldername{species_iter}], 't_10', 't_20', 't_30', 't_40', 't_50', 't_6
0');
end
```

A3.2 ADE

```
clear;
c10t = exp(-(10-0.22*stat(:,1)).^2./(2*sigmasq))./sqrt(2*pi*sigmasq);
a=trapz(stat(:,1),c10t);
c10t=c10t/a;
loglog(stat(:,1),c10t)
```

```
% take Geobacter when L=10 microns as example, c10t stores
fractionations and sigmasq is time-variable  $\delta^2$ . stat(:,1) stores time
vector.
```

A3.3 CTRW

```

for file_iter =1:2
if file_iter==1
load('geobacter_M.mat');
filename='geobacter';
end
if file_iter==2
load('pelosinus_M.mat');
filename='pelosinus';
end

[r_M,c_M]=size(M_merged);
rng('shuffle');
t_initial=;
% initialize time pointer for random number pick, and give model
duration in variabl 't_initial'

% coupled model
for i=1:100*r_M
    C_cp=[];D=[];
    j=0;
    sum_x=0;
    sum_y=0;
    sum_t=0;

% C_cp is used to store results, D is used to store distance and j is
used to position jump steps

    t_10_cp(i,1)=t_initial;
    t_20_cp(i,1)=t_initial;
    t_30_cp(i,1)=t_initial;
    t_40_cp(i,1)=t_initial;
    t_50_cp(i,1)=t_initial;
    t_60_cp(i,1)=t_initial;

while sum_t<=t_initial
    a=randi([1 r_M]);
    % pick one random number for both jump and time increments

    sum_x=sum_x+M_merged(a,1);
    sum_y=sum_y+M_merged(a,2);
    sum_tp=sum_t;
    sum_t=sum_t+M_merged(a,3);
    j=j+1;
    C_cp(j,:)=[sum_x,sum_y,sum_t];
    D(j,1)=sqrt(C_cp(j,1)^2+C_cp(j,2)^2);
    end
% adding step until t exceed model duration

if (sum_t>t_initial)

    if j==1;
    sum_x=0+(t_initial-sum_tp)/(sum_t-sum_tp)*M_merged(a,1);
    sum_y=0+(t_initial-sum_tp)/(sum_t-sum_tp)*M_merged(a,2);

```

```

sum_t=t_initial;
C_cp(j,:)=sum_x,sum_y,sum_t];
D(j,1)=sqrt(C_cp(j,1)^2+C_cp(j,2)^2);
else
sum_x=C_cp(j-1,1)+(t_initial-sum_tp)/(sum_t-sum_tp)*M_merged(a,1);
sum_y=C_cp(j-1,2)+(t_initial-sum_tp)/(sum_t-sum_tp)*M_merged(a,2);
sum_t=t_initial;
C_cp(j,:)=sum_x,sum_y,sum_t];
D(j,1)=sqrt(C_cp(j,1)^2+C_cp(j,2)^2);
end
% if model reaches maximum time at first step or not, interpolate sum_x
and sum_y by assuming last jump linear

end
if (D(1,1)>=60) && (60/D(1,1)*C_cp(1,3)<=t_initial);
t_10_cp(i,1)=10/D(1,1)*C_cp(1,3);
t_20_cp(i,1)=20/D(1,1)*C_cp(1,3);
t_30_cp(i,1)=30/D(1,1)*C_cp(1,3);
t_40_cp(i,1)=40/D(1,1)*C_cp(1,3);
t_50_cp(i,1)=50/D(1,1)*C_cp(1,3);
t_60_cp(i,1)=60/D(1,1)*C_cp(1,3);
elseif D(1,1)>=50;
t_10_cp(i,1)=10/D(1,1)*C_cp(1,3);
t_20_cp(i,1)=20/D(1,1)*C_cp(1,3);
t_30_cp(i,1)=30/D(1,1)*C_cp(1,3);
t_40_cp(i,1)=40/D(1,1)*C_cp(1,3);
t_50_cp(i,1)=50/D(1,1)*C_cp(1,3);
elseif D(1,1)>=40;
t_10_cp(i,1)=10/D(1,1)*C_cp(1,3);
t_20_cp(i,1)=20/D(1,1)*C_cp(1,3);
t_30_cp(i,1)=30/D(1,1)*C_cp(1,3);
t_40_cp(i,1)=40/D(1,1)*C_cp(1,3);
elseif D(1,1)>=30;
t_10_cp(i,1)=10/D(1,1)*C_cp(1,3);
t_20_cp(i,1)=20/D(1,1)*C_cp(1,3);
t_30_cp(i,1)=30/D(1,1)*C_cp(1,3);
elseif D(1,1)>=20;
t_10_cp(i,1)=10/D(1,1)*C_cp(1,3);
t_20_cp(i,1)=20/D(1,1)*C_cp(1,3);
elseif D(1,1)>=10;
t_10_cp(i,1)=10/D(1,1)*C_cp(1,3);
end

% if the first time reaches 10,20,30,40,50 or 60-micron control plane,
interpolate time

for k=2:j
if D(k,1)>=10 && D(k-1,1)<10
t1=C_cp(k-1,3)+(C_cp(k,3)-C_cp(k-1,3))*(1-(D(k,1)-10)/(D(k,1)-
D(k-1,1)));
if t1<t_10_cp(i,1)
t_10_cp(i,1)=t1;
end
end
if D(k,1)>=20 && D(k-1,1)<20

```

```

        t2=C_cp(k-1,3)+(C_cp(k,3)-C_cp(k-1,3))*(1-(D(k,1)-20)/(D(k,1)-
D(k-1,1)));
        if t2<t_20_cp(i,1)
            t_20_cp(i,1)=t2;
        end
    end
    if D(k,1)>=30 && D(k-1,1)<30
        t3=C_cp(k-1,3)+(C_cp(k,3)-C_cp(k-1,3))*(1-(D(k,1)-30)/(D(k,1)-
D(k-1,1)));
        if t3<t_30_cp(i,1)
            t_30_cp(i,1)=t3;
        end
    end
    if D(k,1)>=40 && D(k-1,1)<40
        t4=C_cp(k-1,3)+(C_cp(k,3)-C_cp(k-1,3))*(1-(D(k,1)-40)/(D(k,1)-
D(k-1,1)));
        if t4<t_40_cp(i,1)
            t_40_cp(i,1)=t4;
        end
    end
    if D(k,1)>=50 && D(k-1,1)<50
        t5=C_cp(k-1,3)+(C_cp(k,3)-C_cp(k-1,3))*(1-(D(k,1)-50)/(D(k,1)-
D(k-1,1)));
        if t5<t_50_cp(i,1)
            t_50_cp(i,1)=t5;
        end
    end
    if D(k,1)>=60 && D(k-1,1)<60
        t6=C_cp(k-1,3)+(C_cp(k,3)-C_cp(k-1,3))*(1-(D(k,1)-60)/(D(k,1)-
D(k-1,1)));
        if t6<t_60_cp(i,1)
            t_60_cp(i,1)=t6;
        end
    end
    % check each trajectory frame by frame to obtain first arrival
    time(minimal)reaching each control plane.

end
end

%uncoupled model
for i=1:100*r_M
    C_uncp=[];D=[];
    j=0;
    sum_x=0;
    sum_y=0;
    sum_t=0;
    t_10_uncp(i,1)=t_initial;
    t_20_uncp(i,1)=t_initial;
    t_30_uncp(i,1)=t_initial;
    t_40_uncp(i,1)=t_initial;
    t_50_uncp(i,1)=t_initial;
    t_60_uncp(i,1)=t_initial;

while sum_t<=t_initial
    a=randi([1 r_M]);

```

```

b=randi([1 r_M]);
% pick two random number, one of time increment and one for jump
increment
sum_x=sum_x+M_merged(a,1);
sum_y=sum_y+M_merged(a,2);
sum_tp=sum_t;
sum_t=sum_t+M_merged(b,3);
j=j+1;
C_uncp(j,:)=[sum_x,sum_y,sum_t];
D(j,1)=sqrt(C_uncp(j,1)^2+C_uncp(j,2)^2);
end

if (sum_t>t_initial)
    if j==1;
        sum_x=0+(t_initial-sum_tp)/(sum_t-sum_tp)*M_merged(a,1);
sum_y=0+(t_initial-sum_tp)/(sum_t-sum_tp)*M_merged(a,2);
sum_t=t_initial;
        C_uncp(j,:)=[sum_x,sum_y,sum_t];
D(j,1)=sqrt(C_uncp(j,1)^2+C_uncp(j,2)^2);
    else
        sum_x=C_uncp(j-1,1)+(t_initial-sum_tp)/(sum_t-
sum_tp)*M_merged(a,1);
sum_y=C_uncp(j-1,2)+(t_initial-sum_tp)/(sum_t-
sum_tp)*M_merged(a,2);
sum_t=t_initial;
        C_uncp(j,:)=[sum_x,sum_y,sum_t];
D(j,1)=sqrt(C_uncp(j,1)^2+C_uncp(j,2)^2);
    end
end

    if (D(1,1)>=60) && (60/D(1,1)*C_uncp(1,3)<=250);
        t_10_uncp(i,1)=10/D(1,1)*C_uncp(1,3);
t_20_uncp(i,1)=20/D(1,1)*C_uncp(1,3);
t_30_uncp(i,1)=30/D(1,1)*C_uncp(1,3);
t_40_uncp(i,1)=40/D(1,1)*C_uncp(1,3);
t_50_uncp(i,1)=50/D(1,1)*C_uncp(1,3);
t_60_uncp(i,1)=60/D(1,1)*C_uncp(1,3);
    elseif D(1,1)>=50;
        t_10_uncp(i,1)=10/D(1,1)*C_uncp(1,3);
t_20_uncp(i,1)=20/D(1,1)*C_uncp(1,3);
t_30_uncp(i,1)=30/D(1,1)*C_uncp(1,3);
t_40_uncp(i,1)=40/D(1,1)*C_uncp(1,3);
t_50_uncp(i,1)=50/D(1,1)*C_uncp(1,3);
    elseif D(1,1)>=40;
t_10_uncp(i,1)=10/D(1,1)*C_uncp(1,3);
t_20_uncp(i,1)=20/D(1,1)*C_uncp(1,3);
t_30_uncp(i,1)=30/D(1,1)*C_uncp(1,3);
t_40_uncp(i,1)=40/D(1,1)*C_uncp(1,3);
    elseif D(1,1)>=30;
t_10_uncp(i,1)=10/D(1,1)*C_uncp(1,3);
t_20_uncp(i,1)=20/D(1,1)*C_uncp(1,3);
t_30_uncp(i,1)=30/D(1,1)*C_uncp(1,3);
    elseif D(1,1)>=20;
t_10_uncp(i,1)=10/D(1,1)*C_uncp(1,3);
t_20_uncp(i,1)=20/D(1,1)*C_uncp(1,3);
    elseif D(1,1)>=10;
t_10_uncp(i,1)=10/D(1,1)*C_uncp(1,3);

```

```

end
for k=2:j
    if D(k,1)>=10 && D(k-1,1)<10
        t1=C_uncp(k-1,3)+(C_uncp(k,3)-C_uncp(k-1,3))*(1-(D(k,1)-
10)/(D(k,1)-D(k-1,1)));
        if t1<t_10_uncp(i,1)
            t_10_uncp(i,1)=t1;
        end
    end
    if D(k,1)>=20 && D(k-1,1)<20
        t2=C_uncp(k-1,3)+(C_uncp(k,3)-C_uncp(k-1,3))*(1-(D(k,1)-
20)/(D(k,1)-D(k-1,1)));
        if t2<t_20_uncp(i,1)
            t_20_uncp(i,1)=t2;
        end
    end
    if D(k,1)>=30 && D(k-1,1)<30
        t3=C_uncp(k-1,3)+(C_uncp(k,3)-C_uncp(k-1,3))*(1-(D(k,1)-
30)/(D(k,1)-D(k-1,1)));
        if t3<t_30_uncp(i,1)
            t_30_uncp(i,1)=t3;
        end
    end
    if D(k,1)>=40 && D(k-1,1)<40
        t4=C_uncp(k-1,3)+(C_uncp(k,3)-C_uncp(k-1,3))*(1-(D(k,1)-
40)/(D(k,1)-D(k-1,1)));
        if t4<t_40_uncp(i,1)
            t_40_uncp(i,1)=t4;
        end
    end
    if D(k,1)>=50 && D(k-1,1)<50
        t5=C_uncp(k-1,3)+(C_uncp(k,3)-C_uncp(k-1,3))*(1-(D(k,1)-
50)/(D(k,1)-D(k-1,1)));
        if t5<t_50_uncp(i,1)
            t_50_uncp(i,1)=t5;
        end
    end
    if D(k,1)>=60 && D(k-1,1)<60
        t6=C_uncp(k-1,3)+(C_uncp(k,3)-C_uncp(k-1,3))*(1-(D(k,1)-
60)/(D(k,1)-D(k-1,1)));
        if t6<t_60_uncp(i,1)
            t_60_uncp(i,1)=t6;
        end
    end
end

end

end
%uncoupled model
save([filename
'_CTRW_2500'], 't_10_cp', 't_20_cp', 't_30_cp', 't_40_cp', 't_50_cp', 't_60_c
p', 't_10_uncp', 't_20_uncp', 't_30_uncp', 't_40_uncp', 't_50_uncp', 't_60_un
cp')

end

```

A3.4 Normalization

```

function [fhat_universal, h0] = universal_adaptive_kernel(Tau)
%smooths the breakthrough curve made by particle tracking time
distribution
%Tau

%tuning parameters
c = .75;
b = 1;
q = 2/35;
p = 1/5;
istar = 3;

n = length(Tau); %number of particles
h = zeros(1,istar+1); %bandwidth used to estimate f at each step
h(1) = b*n^-p; %initial bandwidth used to estimate f
alpha = zeros(1,istar); %bandwidth used to estimate f'' at each step
normK0 = 1/(2*sqrt(pi)); %L2 norm of the Gaussian kernel K
m2K0 = 1; %Second moment of the Gaussian kernel K
m6convK2 = 720; %Sixth moment of the convolution of K'' with itself
convK2of0 = 3/8/sqrt(pi); %the convolution of K'' with itself evaluated
at 0
convK3of0 = 15/16/sqrt(pi); %the convolution of K''' with itself
evaluated at 0
sqrtpi = sqrt(pi);
for iteration = 2:istar+1
    alpha(iteration-1) = c*h(iteration-1)*n^q; %bandwidth used to
estimate f''
    convK2 = n*convK2of0; %convK2 is the sum over all i and j of the
convolution
                                %of K'' with itself evaluated at
                                %(Xi-Xj)/alpha(iteration-1), which is
symmetric. We
                                %initialize the sum with the diagonal
elements
                                %and then add twice the terms where i~=j
    convK3 = n*convK3of0; %same as convK3 except for K'''
    for j = 2:n
        XX = (Tau(1:j-1)- Tau(j))/alpha(iteration-1); %location where
the convolutions are evaluated
        XX2 = XX.^2;
        XX4 = XX2.^2;
        XX6 = XX2.*XX4;
        convK2 = convK2 + 1/16/sqrtpi*sum((12 - 12*XX2 +
XX4)./exp(XX2/4));
        convK3 = convK3 - 1/64/sqrtpi*sum((-120 + 180*XX2 - 30*XX4 +
XX6)./exp(XX2/4));
    end
    normfdp = 1/n^2/alpha(iteration-1)^5*convK2; %estimated norm of f''
    normftp = 1/n^2/alpha(iteration-1)^7*convK3; %estimated norm of
f'''
    h(iteration) = (normK0/(m2K0^2)/normfdp/n)^(1/5); %new bandwidth to
estimate f
    c =
(factorial(6)*convK2of0/m6convK2/normftp)^(1/7)*((m2K0^2)/normK0*normfd
p)^(1/5); %update tuning parameter c

```

```

end
h0 = h(end);
fhat = zeros(1,n); %estimate of f using global bandwidth h0
Cum_bt = zeros(1,n); %cumulative breakthrough of particles
for iteration = 1:n
    fhat = fhat + 1/n/h0/(sqrt(2*pi)).*exp(-(((Tau-
Tau(iteration))/h0)/(sqrt(2))).^2);
    Cum_bt(iteration) = length(find(Tau <= Tau(iteration)))/n;
end

a = .5; %sensitivity parameter - tells how much h0 affects adaptive
bandwidth
% 0<= a <= 1
G = exp(1/n*sum(log(fhat))); %normalization constant
h1 = h0*(fhat/G).^(-a; %adaptive bandwidth that is proportional to
fhat^-a (when a>0 the bandwidth is
%small where there are a lot of particles and large where there are
fewer particles)
h2 = (1-Cum_bt)*h0 + Cum_bt.*h1; %universal adaptive bandwidth weighted
by the cumulative breakthrough
%of the particles - varies between global
%bandwidth which captures the peak well
%and the adaptive bandwidth which captures
%the tail well
fhat_universal = zeros(1,n); %estimate of f with universal adaptive
bandwidth
for i = 1:n
    fhat_universal = fhat_universal + 1/n/h2(i)/sqrt(2*pi).*exp(-
(((Tau-Tau(i))/h2(i))/(sqrt(2))).^2);
end
end
end

```