SANDIA REPORT

SAND2008-8127
Unlimited Release
Printed September 2008

Decontamination of Water Networks: A Numerical and Experimental Investigation of Biofilms

Bart van Bloemen Waanders, Susan Altman, Brian Carnes,
Judith Hill, Kevin Long, Lucas McGarth, Youssef Marzouk

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2008-8127
Unlimited Release
Printed September 2008

Decontamination of Water Networks: A Numerical and Experimental Investigation of Biofilms

Bart van Bloemen Waanders, Susan Altman, Brian Carnes,
Lucas McGrath, Youssef Marzouk [1] Judith Hill [2], Kevin Long [3],

Abstract

This research effort was motivated by ongoing developments to secure water distribution networks which are extremely vulnerable to contamination events. Although research efforts are being devoted to early warning systems, the mitigation process needs more attention. Regardless of the efficiency of a detection system, controlling, neutralizing or flushing have to be eventually addressed to restore water quality after a contamination event. These processes however suffer from uncertainties associated with unknown contaminant characteristics, changing demand patterns, questionable pipe characteristics, modeling approximations, and unpredictable behavior of biofilms. It is the uncertainty of biofilms that motivates this research with a specific goal of developing a simulation foundation and experimental procedures from which field deploy-able numerical tools can eventually be developed.

Biofilms are a combination of bacteria and polymeric substance adhered to an external surface and in a continuous state of flux. They are ubiquitous in aqueous environments, such as water distribution networks, but also important to many other applications ranging from the food industry to

[1]Sandia National Laboratories, Albuquerque, NM, 87109, USA Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

[2]Oakridge National Laboratories

[3]Texas Tech University

human physiology. Biofilms are essentially collections of bacteria intertwined in a polymeric matrix that exhibit complicated behavior at various spatial and temporal scales. The numerical goals consisted of developing flexible software capabilities at different spatial scales but also develop algorithms that could help calibrate our implementations with experimental observations.

This research project produced fundamental simulation capabilities to help characterize biofilms for water distribution systems. Algorithms were developed at multiple spatial scales. At the molecule-scale, Density Functional Theory was investigated to characterize density variations of bacteria attached to external surfaces. The intent was to help calibrate convection-diffusion-migration dynamics in which the migration operator represents a balance of electrostatic and vanderWaal forces. Effective diffusion could then be up-scaled to biomass growth models at larger spatial scale. Significant effort was devoted on biomass growth models including verification procedures using experimental observations. A level set approach combined with diffusion-reaction dynamics was developed to predict the growth of interfaces between biofilm and bulk fluid. To verify model parameters with confocal microscope images, we developed adjoint capabilities to invert for initial conditions. Significant algorithmic challenges had to be addressed, including handling of nonlinear inversion, addressing renormalization of level set interfaces, and finite element stabilization of convective processes.

New laboratory protocols and procedures were designed in an attempt to grow biofilms in capillary tubing as part of a continuous confocal microscope monitoring setup. Significant variation in the behavior of the bacteria and equipment precluded consistent replication of local biofilm characteristics within acceptable statistics. Instead, coarse characteristics such as average roughness coefficients, biomass, and thickness, were quantified and compared to numerical results.

At the water distribution network scale, Navier Stokes coupled to convection-diffusion-reaction equations were applied to predict the behavior of contaminant and neutralizing chemical transport in simple and complicated geometries. Adjoint based error estimation methods were developed to help refine meshes for computational efficiency and accuracy. Additional experimental studies were conducted at this spatial scale using an annular reactor.

Given the range of uncertainties and the modeling complexities which force approximations, stochastic inversion methods were developed. Bayesian theory was used to calculate posterior distribution of reaction coefficients given observations. A Markov Chain Monte Carlo algorithms solved this stochastic problem. A newly funded LDRD project will continue this part of the project and investigate efficient algorithms to apply Bayesian theory to large multi-scale parameter inference.

Even though this project fell short of producing a final numerical tool ready for field deployment, we helped develop software tools that enable efficient development and rapid deployment. Furthermore, uncertainty quantification methods were implemented to help address experimental variabilities and errors associated with numerical approximations.

Contents

Appendix

List of Figures

List of Tables

Chapter 1

Introduction

Significant research has recently been devoted to develop technologies for protection of water distribution networks against contamination events, which can be devastating as past events have demonstrated. Most notably in Milwaukee with an outbreak of cryptosporidiosis, an estimated 403,000 persons became ill, of whom 4,400 were hospitalized. The number of deaths were estimated at about 100. The cost associated with decontamination, equipment upgrades and legal suites were on the order of $100M. Although the Milwaukee incident was one of the worst incidents in recent history, smaller events pose similar risks and are equally difficult to mitigate. Considering the relatively frequent occurrence of contamination events, it is surprising how little effort is being devoted to the development of numerical tools for guidance in mitigation procedures. Currently public utility companies have an overwhelming task of controlling further spread, neutralizing with Chlorine, and manually sampling for water quality. Without numerical tool support, such as accurate simulators, optimization algorithms, uncertainty quantification methods, and other analysis capabilities, managing such a process becomes time-consuming, expensive and is error-prone. The goal of this research was to develop algorithms, methods, and software to serve as a foundation from which real time, multi-scale, multi-physics, inversion, control, uncertainty quantification capabilities can eventually be built in support of decontamination of water distribution networks.

Nearly unlimited access points render water distribution systems vulnerable to intentional and accidental contamination events. Although physical security protects large components, such as storage tanks and pumping stations, common households and fire hydrants remain accessible. In recognition of this vulnerability, a range of technologies have been developed over the last five years to support the possibility of detecting contaminants through the use of general water quality sensors as part of an early warning system. Only a sparse set of sensors can be installed however throughout a water distribution system because of limited installation possibilities, expanse of the network, and high cost associated with installation and maintenance. This limitation therefore requires careful sensor placement for optimal detection performance which has been the subject of several research efforts [21, 20, 26]. Current sensing technology is limited to detecting fluctuations in standard water characteristics (Ph, Chlorine, Oxygen, Phosphates, etc) and in combination with numerical algorithms, anomalous intrusions can be identified [46, 45]. Furthermore optimization algorithms have been developed comparing sparse sensor measurements to numerical predictions to determine the location and magnitude of the contamination source [15, 42, 47]. This approach

has demonstrated accurate inversion results with real-time performance and robustness even when subjected to sparse and error induced measurements.

Unfortunately, installing sensors in every water distribution network will not occur anytime soon and even if real-time early warning system existed, complete prevention is still not possible. Consequently, the mitigation of contamination is critically important and therefore additional research and development is necessary to support the mitigation phase. The primary goals of a mitigation process consist of minimizing the spread and ensuring that all contaminants have been removed. The first goal can be achieved by locating and removing the source of the contamination and then manipulating the velocity field of a network to control the spread, although instantaneously halting the spread by simply closing valves is not a realistic and practical option. Besides the time-consuming logistics of manipulating any mechanical part of the network, valves may not exist in certain locations or they may not have been recently (or ever) exercised. Controlling the spread will therefore involve complicated manipulations of the velocity field to achieve a desired affect. A combination of optimization algorithms and numerical simulation tools, validated with field measurements need to be considered to efficiently execute control procedures. The second goal of ensuring that all contaminant have been removed can be accomplished by again manipulating the velocity field to divert the contaminants to an appropriate disposal site. In addition, Chlorine or other appropriate agents can be injected to help neutralize the contaminants. Accordingly, numerical algorithms are essential in support of a cleanup process which will involve a range of complicated choices to manipulate the velocity field and to administer the neutralizing agents.

Another major challenge is that numerical tools will be plagued by uncertainties from unknown contaminant characteristics, changing demand patterns, questionable pipe characteristics, and unpredictable behavior of biofilms. Over time, many of these uncertainties may be resolved though a combination of repeated application of numerical tools with validation of manual testing of water quality however the usefulness of any numerical tool will be determined by its ability to verify results or quantify the uncertainties with predictions. Biofilms will likely impose significant and unpredictable behavior. Biofilms are essentially collections of bacteria intertwined in a polymeric matrix that can potentially act as repositories for contaminants with the unknown behavior one of which could be the release of contaminants at a later time. As such, contaminants can find refuge in a biofilm, protected from any flushing or neutralization process, and then later to be released when portions of biofilms are sheared off. To further complicate matters, the dynamics of biofilms are extremely variable and depend on many factors, ranging from pipe material, bulk fluid flow characteristics, chemistry of species, corrosion, multiple bacterial species, and availability of nutrients. Central to this research therefore was to help diagnose the role of biofilms during a contamination event using numerical modeling tools validated through laboratory experiments.

The scope of characterizing biofilms however is complicated and depends on multi-physics coupled with chemical processes at several orders of spatial and temporal scales. At the smallest-scales, the deposition of bacteria requires molecular theory to predict the variation in certain parameters that at higher spatial scales are most commonly set to constant. These parameters then need to be upscaled to be incorporated into transport dynamics at higher spatial scales. Once bacte-

ria are deposited, multiple processes are responsible for biomass growth including cell replication, excretion of extracellular polymer substance, and variable nutrient consumption.

Although multiple approaches could be considered including discrete methods such as cellular automata and agent based algorithms, continuum approaches provide a mechanism from which phenomena can be explained from first principles in addition to providing a foundation that can be used with large scale optimization and embedded uncertainty quantification algorithms. The topology of the biomass which depends strongly on reaction coefficients, will be indicative of how solutes are transported into and retained by biofilms. Finally at the network spatial scale, biofilms can be represented as reactive boundary conditions, but flow and transport need to be resolved for complex flow geometries. Even though the majority of flow occurs in straight pipes, it the flow junctions, storage tanks, and processing facilities where complicated flow dominate. Biofilm deposition and interaction with the bulk fluid will likely be difficult to predict and require accurate resolution of flow and transport phenomena. This project has touched development of simulation at all the above mentioned spatial scales.

A critical component of simulation development is the need to validate algorithms and implementation. Unfortunately, very little work has been conducted in the area of biofilms. Most of the calibration work has relied on general and qualitative observations. This clearly is an extremely difficult problem, requiring sophisticated numerical predictions, large scale optimization methods, quantification of error and variability at multiple spatial and temporal scales and finally, repeatable and robust experimental capabilities to grow and measure certain features of biofilms. In this project, we have developed new experimental procedures to grow and visualize/measure certain features of biofilm with the goal to help calibrate numerical simulations. In addition, we have developed large scale optimization methods to efficiently reconcile the differences between observations and predictions.

The remainder of the report is organized as follows. We first present some background information on biofilms; how they are formed, what physics and chemistry is involved, and what numerical tools have been developed to help characterize them. As part of the background section we discuss some preliminary work in which we investigated density functional analysis to determine the variation in density of biofilm molecules. Our goal was to develop effective diffusion coefficients at various spatial scales that could then be upscaled to a continuum fluid flow model. Unfortunately, that work was never completed as a result of a loss of personnel and expertise. Next is a discussion of a level set methodology to predict the transport of the interface between biofilm mass and the bulk fluid. In this part, coupled equations for diffusion-reaction and convection combined with a level set approach was used to predict biofilm growth and in particular demonstrate the mushrooming behavior that is typical of biofilms. To validate the reaction coefficients and initial conditions of the numerical model, large inversion was developed to determine appropriate initial conditions of bacterial deposition. The next chapter describes experimental procedures to grow biofilms in a capillary tube under a confocal microscope. One of the main conclusions from the experimental work was that biofilms exhibit tremendous variability. Accordingly, a stochastic inference capability to address the many uncertainties from experimental and numerical models is discussed next.

Baseyian theory is leveraged to calculate a posterior distribution from prior information and sparse observations. Macro-scale modeling using the Sierra-Aria simulator is presented. The subject of this portion of the research is the resolution of flow and transport in the context of representing biofilms as a reaction term on the boundaries. Adaptivity and optimization are combined through adjoints to demonstrate computational efficiency gains in comparison to calculating sensitivities via finite differences and in comparison to using uniform refinement low and transport of solute is resolved. An experimental chapter follows which ideally should have driven the validation of the macro-scale modeling. This was unfortunately beyond the scope of the project. Finally, our desire to develop many different simulation capabilities and incorporate complex embedded algorithms has motivated the use of high level abstraction software. In this chapter we discuss some of our development and use of Trilinos and the Sundance software tools.

1.1 Biofilm Characterization Background

Biofilms are structured communities of microorganisms encapsulated within a self-developed extracelluar polymeric substance (EPS) and are attached to surfaces. These biofilms are ubiquous in nearly any aquous system and therefore affect a wide range of applications, including food processing, cooling systems, marine vessels, human physiology, sewage treatment, petroleum recovery, and water distribution systems. Estimating location, predicting mass, minimizing growth, controling interactions with chemicals and biological agents, are all examples of critically important information required to determine the impact on performance, health, efficiency of any affected application. A complete characterization of biofilms however entails an understanding of multi-physics, chemistry, and biology for multi-species bacteria at different spatial and temporal scales. As such, a comprehensive review of biofilms including all the associated technical aspects are clearly beond the scope of this report. The purpose of this chapter is to provide the reader with some background on biofilms. Additional background information is provided in each subsequent chapter but more specific to algorithms, computational science and experimental procedures.

Biofilms consist of an accumulation of bacteria adhered to a surface encapsulated by exopolysaccharides (a.k.a. EPS). Multiple phases make up the life of a biofilm: 1) adhesion to a surface, 2) growth through duplication, 3) consumption of nutrients, 4) excretion of a polysacchride that acts as a cement 5) additional collection of bacteria, 6) death of bacteria from lack of nutrients, and 7) seperation of biofilms as shear forces of the bulk fluid exceed the integrity of the biomass. Figure 1.1 shows a simple cartoon depiction of different processes in biofilms. In each phase, complex phenomena occur at possibly multiple spatial and temporal scales. For examples, adhesion of bacteria to a surface is dominated by a balance of electrostatic and van der Waals forces whereas growth of biofilms depend on flow, transport and chemical reactions. Within the growth process diffusive behavior is instantaneous compared to a much temporal range for convection.

Most of the biofilm physical properties come from the EPS and is probably the most important feature to characterize. There are at least three conceptual models for the structure of biofilms: het-

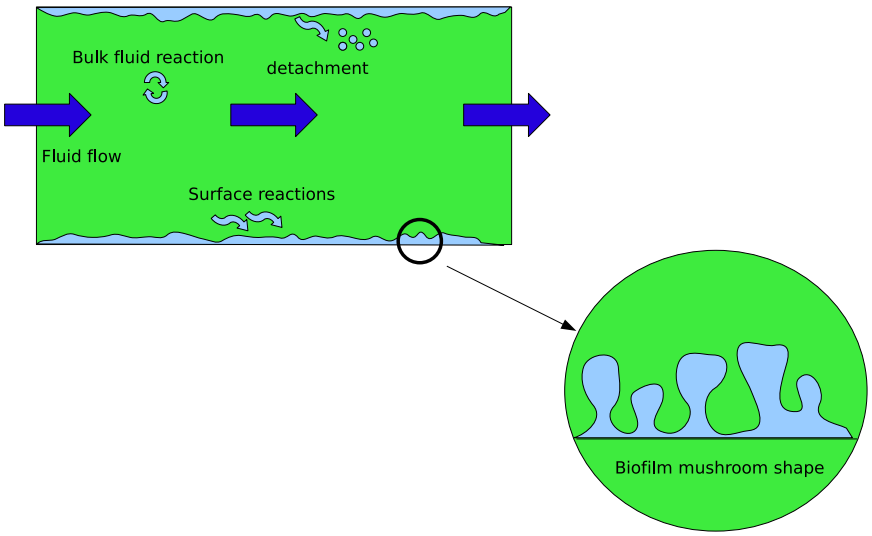


Figure 1.1. Schematic biofilm processes

erogeneous mosaics, structures penetrated by water channels, and dense confluent biofilms [56]. The exopolysaccharides (EPS) synthesized by microbial cells vary greatly and have different chemical and physical properties. It has been shown that different regions can be hydrophobic and others hydrophyllic, resulting in drastic different transport behavior. In addition EPS contributes to the structural integrity of the biofilm enabling a capability to withstand considerable shear forces. It is possible that either highly viscous solutions or localized gels are formed allowing plastic like deformation properties under shear stresses. The presence of different chemicals can determine the quality of the bacterial adhesion to external surfaces. Furthermore, these chemicals also dictate how biofilm interact with different molecules. Despite these large range of varaible properties, different numerical simulation approaches have been attempted [25]

The EPS contribute to the properties of biofilms because they can bind enormous amounts of water. For example, hyaluronic acid can bind up to 1000 times its weight in water. On the other hand, some can exclude water, such as cellulose. The EPS also contribute to mechanical stability by allowing the biofilm to withstand shear forces. Activity within a biofilm increases with the thickness up to a determined level after which the nutrients cannot diffuse far enough citeLaMa95.

There are three distinct phases to the formation of biofilms, consisting of attachment, colonization, and mature growth. Each phase involves complex physical and chemical phenomena making the numerical prediction of biofilm dynamics a challenging task.

1.1.1 Molecular Theory

Our research started with an investigation of molecular theory in an attempt to understand the behavior of colloid stability, bacteria adhesion onto external surfaces and the density variation at equilibrium for a biofilm. However due to loss of key personnel, this work was discontinued and we have no substantial results to report. However, the deposition phase is important and we have developed large scale optimization algorithms to invert for the initial deposition condition. Therefore we include a brief discussion of key phenomena.

A variety of technologies can be applied to help understand the deposition physics of bacteria, including Derjaguin, Landau, Verwey and Overbeek (DVLO) algorithms, convection-diffusion-migration (CDM) equations, Possion-Boltzmann and Density Functional Theory (DFT). These mechanisms play a key role in the ultimate characterization of biofilms from the deposition onto an external surface to the material properties that control the growth patterns. DVLO is used to predict the stability of colloids and can therefore help explain why bacteria do not adhere to each other and why they adhere to certain external surfaces. CDM incorporates convective and diffusive properties to predict the deposition onto surfaces. The migration term encapsulate DVLO like properties. Finally, DFT predicts the variation of density of fluids that are adhered to external surfaces. In the biofilm case it can predict the variation of material properties although these continuously change as the biofilm thickens and DFT is only appropriate at equilibrium. Although our initial phase of our research was focussed on the development of molecular simulation capabilities and some numerical capabilities were developed, our efforts were designed for the purposes of establishing a basic understanding for the development of numerical capabilities at the next larger spatial scales. Accordingly this part of the background section is relatively brief and can be skipped without loss of understanding for the remainder of the report.

Derjaguin, Landau, Verwey and Overbeek (DVLO) theory describes the stabiliy of a particle in solution being dependent on total potential energy essentially consisting of a balance between van der Waal attractive and electrical double layer repulsive forces [61]. DVLO is the foundation to predict colloid dispersion characteristics and applies to particles at a length scale from $10E-9$ to $10E-6$ meters. Industries range from food processing to electronic manufacturing where mixing of particle is a critical phenomena. A balance of attractive forces, known as van der Waals forces, and repulsive forces, such as those encountered in electrostatics, determine to adhesion between particles and possibly surfaces. Das et al. [27] discuss double layer forces between spherical particles and planar substrates. The convection-diffusion-migration equation provides a mechansim to combine transport and deposition/adhesion to surfaces [54].

Density Functional Theory (DFT) provides an approach to the study of materials that is intermediate between macroscopic thermodynamic approaches and truly microscopic simulation-based methods. They incorporate molecular-level detail but are simple enough that calculation time is modest and physical insight is retained even in complex situations. Density functional theory (DFT) can be applied both to equilibrium problems (phase coexistence lines, interfacial structure and free energy, effects of walls and external fields) and to dynamic problems (rates of nucleation

and growth of new phases).

DFT concepts are based on statistical mechanics ([8], [37]) and fundamental variational principles which states that at equilibrium the functional derivative of the free energy with respect to the density is zero. This gives a nonlinear integrodifferential equation that can be solved for the density. This material is from Evans . DFT is based on the idea that the free energy of the inhomogenous fluid can be expressed as a functional of $\rho(r)$ [34]. From the knowledge of this functional all the relevant thermodynamic properties can be calculated so that tensions can be computed for the interface problems, solvation forces can be determined from confined fluids, and phase transitions can be investigated for various types of inhomogeneity. The exact free energy is equivalent to solving the statistical mechanics for the particular fluid. DFT however offers functional integration and differentiation to calculate formally exact results for correlation functions and thermodynamic functions. Statistical methods that use partition functions are less elegant and less efficient.

One key result of DFT is that the Helmholtz free energy $F(p)$ is a unique functional for a given interatomic or intermolecular potential energy. That part of the free energy not associated directly with the external $V(r)$ producing inhomogeneity, has the same dependence on $\rho(r)$ for all $V(r)$. Thus the $F(r)$ should be valid for a large range of problems.

Chapter 2

Numerical Modeling of Biofilms Using Level Set Methods

2.1 Introduction

Simulation to predict biofilm dynamics can be classified into two types of categories. The first consist of discrete methods such as agent-based and cellular automata approaches. These approaches offer considerable flexibility and adaptability to produce a range of topological characteristics that at least qualitatively resemble laboratory and field observations. However, heuristics and mathematically non-rigorous approaches often are used, which do not lend themselves to a fundamental understanding of the underlying processes. In addition our final goal is to calibrate models with respect to material properties and initial condition, which both demand large inversion spaces, and therefore require efficient embedded optimization algorithms. These approaches depend on the equations of the forward simulation to be differentiable, which is not the case for discrete methods. The second category is based on continuum mechanics and therefore relies on differential equations and other differentiable mathematical principals to achieve numerical representations. Although the ability for this method to match all detailed observations from the laboratory and the field is less flexible than discrete methods, a continuum approach allows for the use of large scale optimization, error estimation, and potentially embedded uncertainty quantification methods.

Continuum modeling have been applied to estimate bacteria adhesion, EPS characterization, transport of solutes into biofilms, representation of biofilm transport in porous media flow, and growth of biomass. In this chapter we focus on developing fundamental capabilities to support the prediction of biomass growth which is considered one of the basic mechanisms indicative of material propertied within a biofilm. Although the grow process depends on many things, it possible as a first cut, to encapsulate the general behavior with a set of partial differential equations. In particular, a range of chemical-physico processes can be captured by convective, diffusive and reactive operators. The simulation of biomass therefore can be regarded as an interface propagation or front tracking exercise between biomass and bulk fluid. Front tracking and simulation of interfaces have been the subject of research to address a number of important engineering problems, ranging from viscous fingering to foam injection. Techniques such as Lagrangian methods that use marker particles and Riemann problem solutions, have been applied to a number of applications. The use of

level sets is probably one of the more popular and flexible approaches [55] and has been applied to combustion, medical imaging, fluid flow, and various biological problem, including biofilms [30], [28], [4], [5], [41].

Dockery and Klapper [29] consider the modeling of biofilm as a viscous fluid. Darcy's law is used with a growth function which essentially represents a source term. This work was motivated by the recognition that biofilms have complicated internal structures which require accurate resolution for forward predictions. Understanding these heterogeneities is important for transport predictions within the biofilm. The model comprises of Darcy's for pressure, steady state diffusion equation for substrate transport and an evolution equation for the interface. An analytic solution is presented for a 1D linear form of the equation. A level set implementation is presented for a 2D nonlinear equation. Alpkvist et al. [5] present a level set approach to predict biofilm interface growth. Conservation of mass is expressed in terms of a continuity equation and the flow field is assumed to be in the direction of the nutrient gradient. Klapper [41] discusses the use of level set methods and finite difference discretization methods to predict the movement of a single species biomass interface. Alpkvist and Klapper [4] present a multidimensional model with multiple species and multiple substrates. Conservation of mass equations are used to describe biomass growth and substrate concentration. The time scale of the substrate transport is argued to be steady state relative to the biomass dynamics. A volume fraction for different species is incorporated in the biomass flux term.

Substrate exchange from bulk fluid to biomass depends on a variety of chemical-physico phenomena including convection and diffusion. One of the potentially dominant forces is convection and as pointed out by Picioreanu et al. [53], convection is absent in the valley but dominant at the peaks. These topological variations exhibited by mushrooms shapes is referred to the general roughness metric. In this chapter, we present a level set method to predict the movement of the interface between biomass and bulk fluid. Although our techniques are similar to previous work, we differentiate our work in two ways. Instead of structured finite differences discretization, we use the more flexible and theoretically more robust finite element unstructured discretization with Streamline Upwind Petrov Galerkin (SUPG) stabilization. In addition, in the next chapter we derive the inverse problem in which the interface becomes an inversion parameter and can therefore be used to calibrate the numerical model with experimental observations.

2.1.1 Formulation

Biofilm growth is represented as an interface between biomass and bulk fluid in which the biomass starts from a collection of bacteria adhered to an external surface. The growth process is driven by the consumption of nutrients by bacteria that then duplicate and produce polymeric material. These dynamics are formulated as a set of PDEs consisting of diffusion-reaction and convection. The level set function represents the biomass density and therefore the interface is equal to the contour of the level set function is equal to zero. Figure 2.1.1 show the the computational domain.

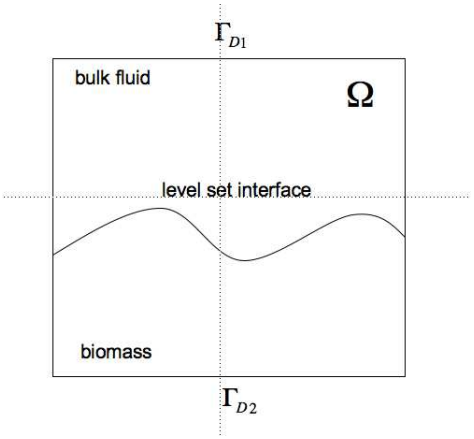


Figure 2.1. Computational domain

We assume that we have a problem domain Ω that can be decomposed into Ω_f, a subdomain containing only the bulk fluid, and Ω_b, a subdomain containing only the biofilm. Within this system, we assume there is a single nutrient of concentration c available for consumption by the biofilm. The distribution of the nutrient within Ω is represented by a diffusion-reaction model:

$$\begin{aligned} \frac{\partial c}{\partial t} - D\Delta c &= r && \in \Omega \\ c &= c_{D1} && \in \Gamma_{D_1} \\ c &= c_{D2} && \in \Gamma_{D_2} \\ c &= c_0 && \in \Omega \times (0, T) \end{aligned}$$

where the reaction term r represents Monod kinetics and can be defined as

$$r = \frac{\mu_g}{y_o} m_c c \tilde{\rho} k_2 + c \tag{2.1}$$

In our formulation, maximum biomass growth rate, μ_g, yield of biomass on oxygen y_o, and biomass m_c are lumped together into the maximum oxygen consumption rate k_1. D is the diffusion coefficient of the nutrient in the bulk fluid, k_1 is the maximum rate at which the nutrients are consumed, and k_2 is the nutrient half-saturation constant. We define $\tilde{\rho}$ as

$$\tilde{\rho}(\mathbf{x}, t) = \begin{cases} \rho_b & \text{for } \mathbf{x} \in \Omega_b(t) \\ 0 & \text{for } \mathbf{x} \in \Omega_f(t) \end{cases} .$$

where ρ_b is the density of the biofilm. Assuming that all consumed nutrients are transformed

into biomass, conservation of mass dictates that within the domain Ω,

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= f(c, \rho) \in \Omega \\ \rho &= \rho_{D1} && \in \Gamma_{D_1} \\ \rho &= \rho_{D2} && \in \Gamma_{D_2} \\ \rho &= \rho_0 && \in \Omega \times (0, T) \end{aligned}$$

where $\mathbf{u}$ is the velocity of the biofilm front and $f(c, \rho)$ is the reaction term chosen for the nutrient consumption, in this case the Monod kinetics term. Furthermore, if we assume that the biofilm grows in the direction of increasing nutrient concentration, then we can conclude that the velocity of the biofilm front should be

$$\mathbf{u} = \frac{D}{\rho_b} \nabla c.$$

to maintain the conservation of mass, simplifying 2.2 to

$$\frac{\partial \rho}{\partial t} + \frac{D}{\rho_b} \nabla \rho \cdot \nabla c = 0.$$

For additional details of this derivation, see [6].

2.1.2 Discretization and Implementation Details

Level Set Method

Many numerical techniques have been suggested for representing the motion of a dynamic interface. In general, these methods can be distinguished by the following characteristics.

- *The definition of the interface and its deformation.* The interface is either defined explicitly, also known as front-tracking, or implicitly, sometimes called front-capturing. An explicit method, through the use of marker particles or grid points, maintains the interface as a sharp discontinuity and *explicitly* tracks its motion. An implicitly scheme, which does not explicitly locate the interface, solves an additional set of field equations describing the motion of the interface.
- *The coordinate framework and discretization of the domain.* Generally, a choice is made between a Lagrangian (material) framework and an Eulerian (spatial) framework. In the Lagrangian framework, motion is observed by a set of particles embedded in the material motion. Most often, the topology of the dynamic interface is embedded in the discretization of the domain. As the interface deforms, the material points move with that deformation, maintaining a sharp interface. However, from a finite element perspective, large deformations may result in degenerate elements necessitating re-discretization of the domain. In an Eulerian framework, motion is observed from a set of fixed spatial points, eliminating the need for remeshing. The resulting challenge is to resolve the dynamic interface which is generally not embedded in the domain discretization.

In this work, we capture the dynamic interface between the biofilm and the bulk fluid using a level set method, a front-capturing method in an Eulerian framework. The level set variable, $\phi(\mathbf{x}, t)$ is defined for this system as

$$\phi(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \Omega_b(t) \\ = 0 & \mathbf{x} \in \Gamma_b(t) \\ < 0 & \mathbf{x} \in \Omega_f(t) \end{cases}.$$

Realizing that the *material time* derivative of $\phi(\mathbf{x}(t), t)$ is zero yields

$$\frac{\partial \phi(\mathbf{x}(t), t)}{\partial t} + \nabla \phi(\mathbf{x}(t), t) \cdot \mathbf{x}'(t) = 0. \tag{2.2}$$

Note that the Heaviside function $\mathcal{H}$ of ϕ can be related to the biofilm density as $\rho(\mathbf{x}, t) = \rho_b \mathcal{H}(\phi(\mathbf{x}, t))$.

Finally, we summarize the equations for the forward modeling:

$$\begin{array}{rll} \frac{\partial c}{\partial t} - D\Delta c & = \frac{k_1 \tilde{\rho} c}{k_2 + c} \hat{c} & \in \Omega \\ \frac{\partial \rho}{\partial t} + \frac{D}{\rho_b} \nabla \rho \cdot \nabla c & = 0 & \in \Omega \\ c & = c_D 1 & \in \Gamma_{D_1} \\ c & = c_D 2 & \in \Gamma_{D_2} \\ c & = c_0 & \in \Omega \times (0, T) \end{array}$$

The convective nature of the level set equation and the use of finite element discretizations require stabilization. We implement the Streamline Upwind Petrov Galerkin (SUPG) [22] by expressing the solution to the convective equation of the biofilm set of equations defined by (2.3) in weak form. To do this, we define the function spaces $V \equiv \{v \in H^1(\Omega) : v|_{\Gamma_{\text{in}}} = c_{\text{in}}\}$ and $W \equiv \{v \in H^1(\Omega) : v|_{\Gamma_{\text{in}}} = 0\}$. The weak solution is then defined by: find $c \in V$:

$$B(c, w) \equiv (\dot{\rho}, w) + (\frac{D}{\rho_b} \nabla \rho \cdot \nabla c, w) \tag{2.3}$$

to derive the SUPG stabilized version of (2.3):

$$B_\tau(c_h, w_h) = 0, \quad w_h \in W_h. \tag{2.4}$$

Here the stabilized bilinear form is defined by

$$B_\tau(c, w) \equiv B(c, w) + (\dot{\rho} + \frac{D}{\rho_b} \nabla \rho \cdot \nabla c, \tau \frac{D}{\rho_b} \nabla c \cdot \nabla w) \tag{2.5}$$

2.2 Numerical Results

The level-set approach is implemented using high level abstraction methods and additional details are provided in Chapter 8. Our primary goal was to replicate Klapper's results [41] so that we could then formulate an inversion problem and evaluate the sensitivity of the reaction and diffusion coefficients. Figure 2.2 shows an example simulation with reactions coefficients $k1$ and $k2$ set at 1000 and 1, respectively. A polynomial equation is used to prescribe initial conditions consisting of four simple domes:

$$0.944133 + 20.0787 * x - 388.802 * pow(x, 2.) - 444.205 * pow(x, 3.) + 30580.6 * pow(x, 4.) - 181036.0 * pow(x, 5.) + 445133. * pow(x, 6.) - 348158. * pow(x, 7.) - 647104.0 * pow(x, 8.) + 1.87644e6 * pow(x, 9.) - 1.964e6 * pow(x, 10.) + 987296.0 * pow(x, 11.) - 198341. * pow(x, 12.)$$

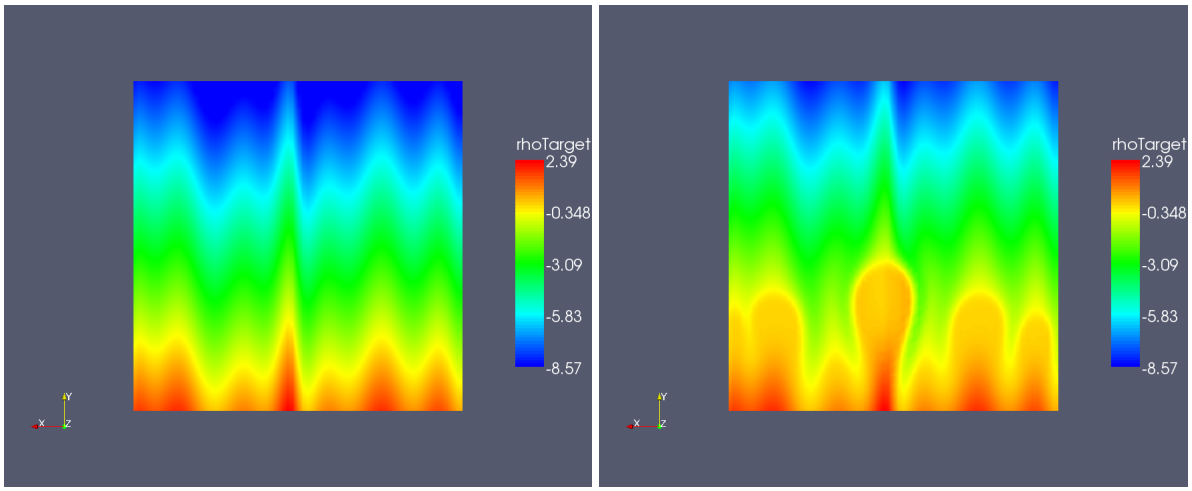


Figure 2.2. 2D biofilm interface for a) t=0 and b) t=20 timesteps

A three dimensional simulation shows similar features as the 2D dataset (Figure 2.3) but perhaps demonstrates a more realistic depiction of the heterogeneity of the biofilm topography, which ultimately needs to be calibrated with experimental observations. Although the simulation is considerably more computationally expensive, our implementation has been parallelized and has demonstrated efficient convergent properties. A three dimensional dataset can therefore be considered to drive an optimization or perhaps even an uncertainty quantification problem.

Parameter studies show different levels of sensitivities to reaction coefficients $k1$, $k2$, and the diffusion coefficient D. A two parameter study was conducted by perturbing the reaction coefficients. Figure 2.4 shows an array of interface figures in which k_1 and k_2 have been perturbed from 1 to 1000. As the figures show high $k1$ values result in more pronounced mushrooming behavior whereas high values of $k2$ counteract this behavior. This appears the be general inline with the

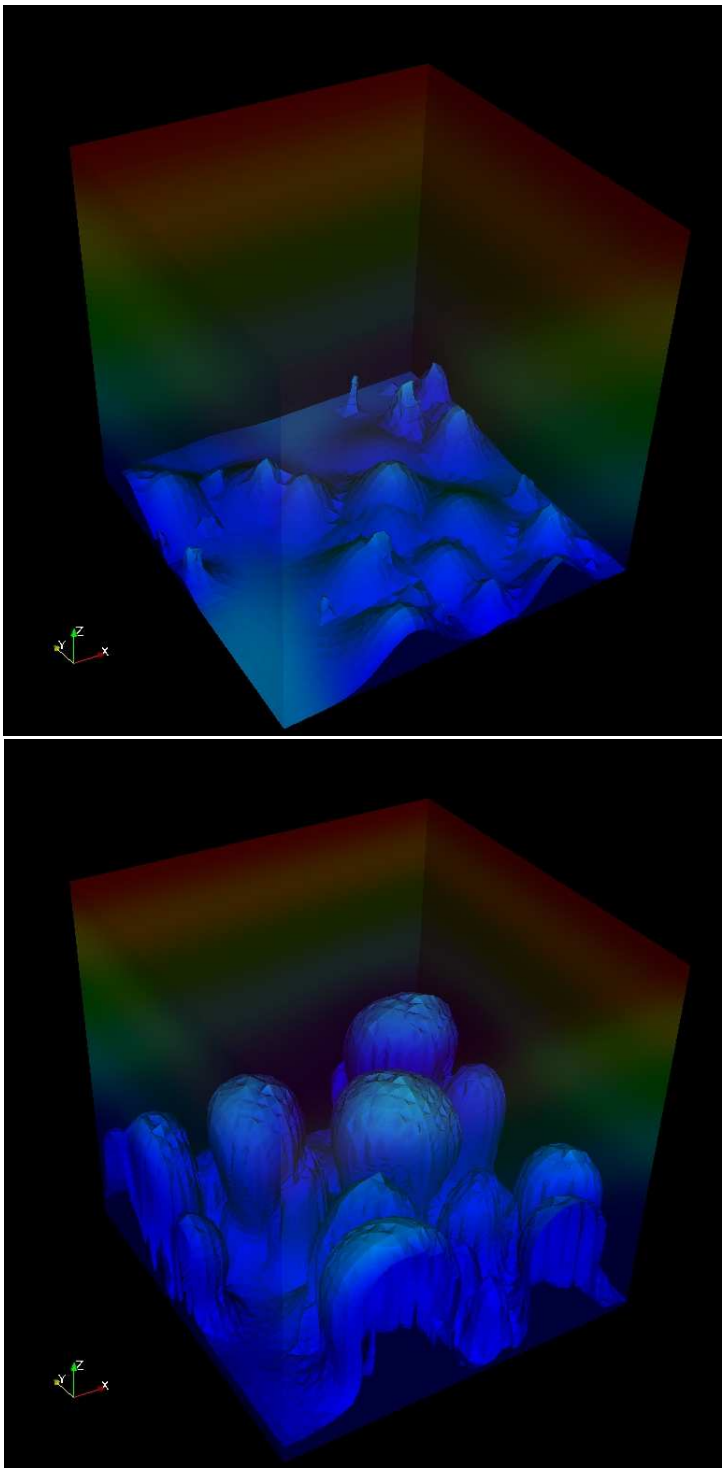


Figure 2.3. 3D Biomass transport at a) initial conditions, b) time step 70.

reaction term in the equations. These experiments were run on a 64x64 grid for 30 timesteps with a 0.005 delta timestep.

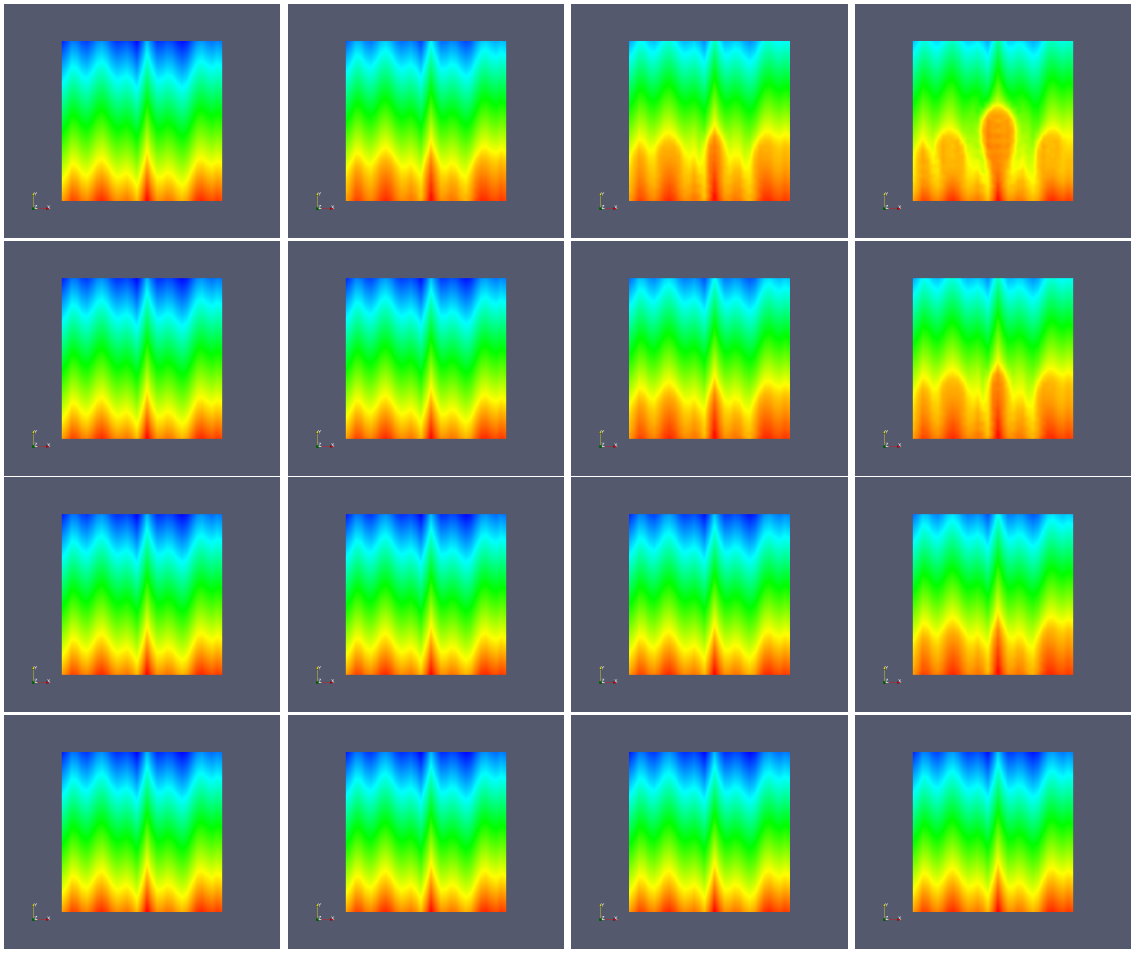


Figure 2.4. Parameter study of biofilm reaction coefficients - $k1$ varies horizontally 1, 10, 100, 100; $k2$ varies vertically 1, 10, 100, 1000

The diffusion coefficient also has a significant effect on the interface behavior. Figure 2.5 shows different interface behavior for diffusion coefficients set at 1,2, and 3.

The variation in interface behavior is large and manually perturbing parameters and initial conditions is not a tractable strategy. Accordingly, the next chapter discusses a large scale optimization derivation and implementation to address the calibration of our numerical model using potentially laboratory observations.

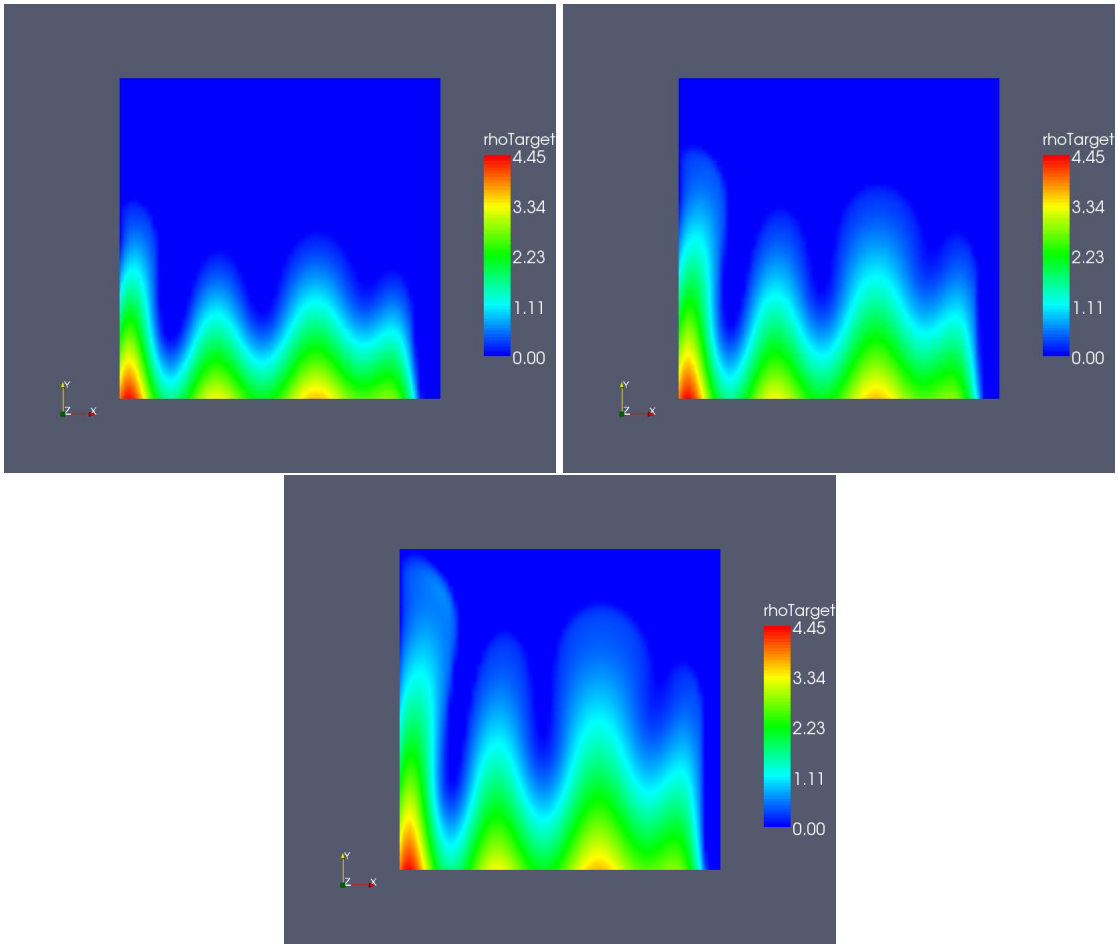


Figure 2.5. Parameter study of biofilm diffusion coefficients a) 1.0, b) 2.0, c) 3.0

2.3 Conclusions

A forward modeling capability in 2D and 3D using level set methods has been developed and demonstrates the unique mushrooming shape at certain selection of reaction coefficients. Unfortunately, there an infinite number of parameter combination each of which results in a different biofilm topology. A parallel SUPG stabilized finite element approach was developed to eventually enable large scale optimization for model calibration purposes. This is the subject of the next few chapters consisting of experimental results for capillary tube experiments, optimization techniques for initial condition inversion and stochastic inversion for material properties.

Chapter 3

Large Scale Inversion for Material Properties and Initial Conditions for Biofilms

3.1 Introduction

Next, we discuss the development of an inversion capability to help calibrate the numerical model. The intent of this work is to infer initial conditions and material properties of a biofilm given experimental observations. In our foward simulation, several parameters control the behavior of the biomass interface. First, the reaction coefficient k_1 controls the rate at which nutrients are consumed and if sufficient nutrients are available has substantial impact on the growth of the interface. The diffusion coefficient is equally important because it influences the extent that nutrients are transported to the interface. The higher the diffusion, the faster and more pronounced the growth. The second reaction parameter k_2 in the reaction term is referred to as the half saturation coefficient and it corresponds to the concentration at which the entire Monod reaction term is one-half of its maximum. In the previous chapter, all these parameters are homogeneous throughout the computational domain. However, at least both reaction coefficients should be considered as anisotropic variables in the biomass controlled by a variety of processses such as the birth and death of bacterial cells. The inversion problem therefore requires large scale algorithms since potentially different materials coefficients could exist at each discretization point. Furthermore, the initial conditions to the level set simulation that dictate the number of mushroom shaped growths, also motivate large scale optimization methods since initial conditions live in the entire computaional domain. This then makes an inversion problem computationally challenging and requires the use of efficient and scalable algorithms. We appeal to partial differential equation (PDE) constrained optimization methods in which a least squares functional is constrained by the dynamcs – the level set based simulator in this case – and make use of appropriate optimization methods to solve the resulting optimality conditions. To the best of our knowledge, an initial condition inversion for a biofilm growth has not been investigated.

Large scale inversion has been investigated for a variety of problem including contamination events for different fluid flow problems [2, 3, 42].

PDE-constrained optimization offers computationally efficiency at the cost of signficant implementation costs. Sensitivity information typically is required from the forward model and for large scale inversion problems this motivates the use of adjoints. Most forward simulator are not developed with the appropriate infrastructure which typically means a complete refactor or significant implementation to access and manipulate the linear algebra of the forward model. In this work, our implementation leverages high level abstraction concepts which is described in chapter 8. The requirement for using this approach is to write the dynamics in the weak form. The software understands differentiable operators and test functions. The adjoint equation therefore also needs to be derived in the weak form and is consequently a continuous version and only equivalent to the discrete adjoint as the mesh refinement variable h goes to zero.

Our apporach follows the general algorithmic strategy of initial condition inversion methods described in [3], although instead of implementing the multigrid preconditioning for CG, a QuasiNewton Sequential Quadratic Programming solver is used. This chapter first presents the inversion formulation and associated optimality conditions, followed by a description of the implementation and a numerical example using the 2D bioflm simulator.

3.2 Large Scale Inversion Formulation

In this work, we are primarily interested in recovering an earlier state of the biofilm based on sparse observations, or measurements. This problem can be posed as a partial differential equation (PDE) constrained optimization problem where we seek a solution that minimizes the least-squares functional

$$\min_{\rho,\rho_0} \ \mathcal{F}(\rho,\rho_0) = \frac{1}{2}\sum_{i=0}^{N}\int_0^T\int_\Omega (\rho-\rho^*)^2\delta(\mathbf{x}-\mathbf{x}_i)\,\mathbf{dx}\,dt + \frac{\beta}{2}\int_\Omega \rho_0^2\,\mathbf{dx} \tag{3.1}$$

subject to the physical constraints

$$\begin{aligned} \tfrac{\partial c}{\partial t} - D\Delta c &= \tfrac{k_1 c\tilde{\rho}}{k_2+c} && \in \Omega \\ c &= c_D && \in \Gamma_D \\ \tfrac{\partial \rho}{\partial t} + \tfrac{D}{\rho_b}\nabla\rho\cdot\nabla c &= 0 && \in \Omega\times t \\ \rho &= \rho_0 && \in \Omega\times(t=0)\,. \end{aligned}$$

In this formulation, the first term in the functional $\mathcal{F}$ minimizes the difference between the observed values ρ^* measured at locations $\mathbf{x}_i$ and the predicted values $\rho(\mathbf{x})$. The second term is a regularization term with a regularization parameter β that forces the solution to be unique. The constraints in this problem are just the original physical model for biofilm evolution and nutrient distribution, combined with appropriate boundary and initial conditions.

The inverse problem is then to recover an earlier state of the biofilm such that the time-evolution of the biofilm matches with the observed measurements. To solve this optimization problem, we

construct a Lagrangian functional $\mathcal{L}$ such that

$$\begin{aligned}\mathcal{L} = \quad & \frac{1}{2}\sum_{i=0}^{N}\int_0^T\int_\Omega(\rho-\rho^*)^2\delta(x-x_i)\,\mathbf{dx}\,dt + \frac{\beta}{2}\int_\Omega\rho_0^2\,\mathbf{dx} \\ & + \int_\Omega\lambda_1\,(D\Delta c - \frac{k_1 c\tilde{\rho}}{k_2+c}) + \int_{\Gamma_D}\lambda_1\,(c-c_D) \\ & + \int_\Omega\int_T\lambda_2\,(\frac{\partial\rho}{\partial t} + \frac{D}{\rho_b}\nabla c\cdot\nabla\rho) + \int_\Omega\lambda_2\,(\rho-\rho_0)|_{t=0}\end{aligned}$$

where λ_1 is the Lagrange multiplier associated with the nutrient concentration and λ_2 is the Lagrange multiplier associated with the biofilm.

Taking variations with respect to the nutrient concentration c and the biofilm density ρ, the Lagrange multipliers λ_1 and λ_2, and the initial state of the biofilm ρ_0, respectively, yields the first-order optimality conditions

The forward problem

$$\begin{aligned} D\Delta c &= \tfrac{k_1 c\tilde{\rho}}{k_2+c} && \in\Omega \\ c &= c_D && \in\Gamma_D \\ \tfrac{\partial\rho}{\partial t} + \tfrac{D}{\rho_b}\nabla\rho\cdot\nabla c &= 0 && \in\Omega\times t \\ \rho &= \rho_0 && \in\Omega\times(t=0)\end{aligned} \tag{3.2}$$

The adjoint problem

$$\begin{aligned} D\Delta\lambda_1 &= \tfrac{k_1k_2\tilde{\rho}}{(k_2+c)^2}\lambda_1 && \in\Omega \\ \lambda_1 &= 0 && \in\Gamma_D \\ D\nabla\lambda_1\cdot n &= 0 && \in\Gamma_N \\ \lambda_2\Delta\rho + \nabla\rho\cdot\nabla\lambda_2 &= 0 && \in\Omega\times t \\ \tfrac{D}{\rho_b}\lambda_2(\nabla\rho\cdot n) &= 0 && \in\Gamma_N\times t\end{aligned} \tag{3.3}$$

The inverse problem

$$\beta\rho_0 - \lambda_2 = 0. \tag{3.4}$$

The equivalent weak forms are

The forward problem

$$\int_\Omega -D\nabla c\cdot\nabla\hat{c} - \frac{k_1\tilde{\rho}c}{k_2+c}\hat{c} = 0 \tag{3.5}$$

$$\int_{\Gamma_N} \hat{c}(\nabla c\cdot n) = 0 \tag{3.6}$$

$$\int_{\Gamma_D} \hat{c}(c-c_D) = 0 \tag{3.7}$$

$$\int_\Omega\int_T (\frac{\partial\rho}{\partial t} + \frac{D}{\rho_b}\nabla c\cdot\nabla\rho)\hat{\rho} = 0 \tag{3.8}$$

$$\int_\Omega (\rho-\rho_0)\hat{\rho}\Big|_{t=0} = 0 \tag{3.9}$$

The adjoint problem

$$\int_\Omega \left(-D\nabla\hat{\lambda_1}\cdot\nabla\lambda_1 - \frac{k_1k_2\rho}{(k_2+c)^2}\lambda_1\hat{\lambda_1}\right) + \int_\Omega\int_T \frac{D}{\rho_b}\lambda_2(\nabla\hat{\lambda_1}\cdot\nabla\rho) = 0 \tag{3.10}$$

$$\int_{\Gamma_D}\int_T \lambda_1\hat{\lambda_1} = 0 \tag{3.11}$$

$$-\frac{\partial\lambda_2}{\partial t}\hat{\lambda_2} + \frac{D}{\rho_b}\lambda_2(\nabla\hat{\lambda_2}\cdot\nabla c) - \int_\Omega \frac{k_1c\lambda_1}{k_2+c}\hat{\lambda_2}\frac{(\partial\tilde{\rho}}{\partial\rho} = -\int_\Omega\int_T\sum_{i=0}^{N}(\rho-\rho^*)\delta(x-x_i)\hat{\lambda_2} \tag{3.12}$$

$$\int_\Omega \lambda_2\hat{\lambda_2}|_{t=T} = 0 \tag{3.13}$$

The inverse problem

$$\int_\Omega \beta\rho_0 - \lambda_2 = 0 \tag{3.14}$$

3.3 Numerical Results

To again leverage our high level abstraction software, a continuous adjoint was implemented. Although our software is capable of extracting Jacobian operators, additional work on the boundary

conditions would be required to get the correct matrix. We checked out our adjoint by finite differencing the the objective gradient and comparing it to the objective function calculated by the innner product of the adjoint and the parameter sensitivities.

A large range of experiments were conducted and were responsible for many bug fixes, formulation modifications, and implementation changes. Although the continuous adjoint is easily derived and implemented –essentially mimicing the forward solver implementation– one has to careful with initial conditions and boundary conditions. Although the adjoint can be checked with finite difference techniques, it is not always clear what finite difference steps to use, whether the finite difference algorithms has been implemented properly, and what an acceptable tolerance is between the finite difference and adjoint based gradients.

A numerical inversion experiment was conducted by forward simulating with a particular initial guesses, followed by extracting a set of density values which in turn are used as observations in the inversion problem. The forward simulation were executed with nx=80, ny=40, Nts=20 D=1.0 deltaT=0.00025 k1=100.0 k2=1.0 rhoB=1.0 The inversion was performed with a regularization coefficient of 0.00001. Figure 3.1 shows the inversion results on a 80x80 grid for different number of sensors. Using sensors at all grid points, the optimizer recovers the target solution very well. However, using only 121 evenly distributed sensors, the optimizer still locates the main amplitude locations of the target solution but produces a blurred interface. We note that to avoid "inversion crimes" we should run the truth model on a finer grid. However, our efforts were dedicated to building efficient inversion software and we simply ran out of time to perform such inversion tests.

There are some additional implementation issues that need to be resolved. In particular the interface diffuses as a result of the level set algorithm and no longer admits a signed distance function. The diffusive behavior creates problems for the inversion algorithm which performs optimally for convective dynamics. A special algorithms has been developed to renormalize the interface after a certain number optimization iterations, but has not yet been implemented.

3.4 Conclusions and Future Work

Large scale inversion capabilities have been developed to infer initial conditions for biomass transport given experimental observations. Our software and implementation can easily be applied to material property inversion or modified to include more complicated physics.

The inversion results indicate several areas that need to be improved upon. We speculate that one of the reasons for the oscilatory inverse solution with sparse sensors is that the level set interface needs to be normalized to maintain a signed distance function. Currently the interface becomes very diffuse, which is problematic for the inversion.

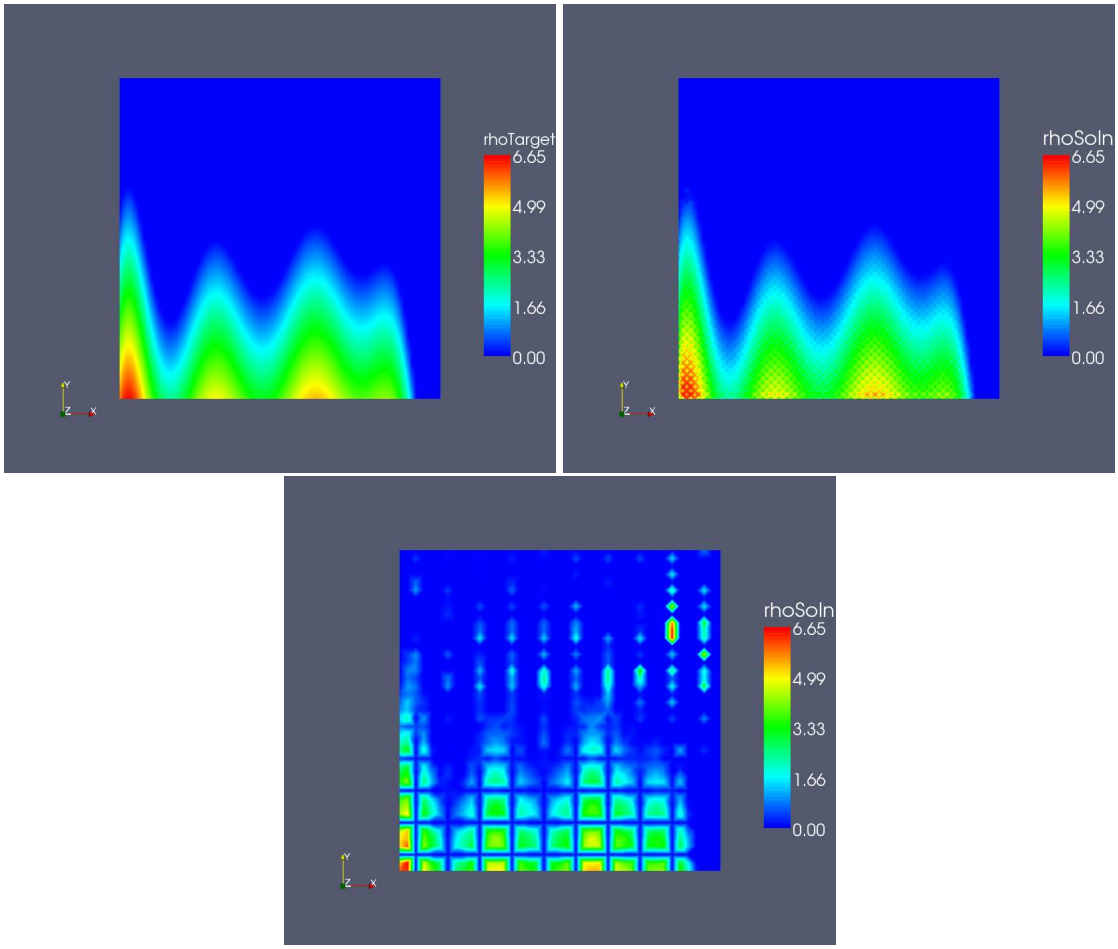


Figure 3.1. Inversion Results for a 80x80 grid a) target solution, b) inversion using all sensors, c) inversion using 121 sensors evenly distributed

Chapter 4

Biofilm Growth Experiments

4.1 Introduction

Biofilm growth is known to be variable and unpredictable, making modeling of biofilms complicated. Our premise is that biofilm modeling should be approached in a statistical fashion with probabilistic sampling of parameters needed to model biofilm growth. To do this, a quantitative assessment of the variability of biofilm structure is needed. The aim is to define metrics for biofilm structure and link these metrics to the critical model paramaters to define biofilm growth. This section describes a series of experiments where biofilms were grown under identical conditions. Images of the biofilms were taken over the duration of the experiments. Rates of change for specific parameters used to describe the biofilms were calculated and assessed. Suggestions are made on how to use these data in biofilm growth numerical models.

4.2 Methods

4.2.1 Biofilm Growth

Biofilms were grown in square, glass capillary tubes with an inside dimension of 1 mm 1 mm and a wall thickness of 0.15 mm (Friedrich & Dimmock, Inc. BMC-1-15-100). The flow system is presented in Figure 4.1. A syringe pump (Harvard Apparatus PHD 2000) was used to control the flow rate and allow for a continuous feed of bacteria. The flow rate was set at 0.05 ml/min. This converts a linear velocity of 8.3 10-3 m/s and a Reynolds number of 9 (laminar flow). A bubble trap was placed between the syringe pump and the capillary tube to minimize bubbles in the tube that could disrupt the biofilm growth. The system was placed directly on the laser scanning confocal microscope (LSCM) (Zeiss LSM-510) so that images could be taken without disturbing the system.

The fluid transported through the capillary tube consisted of 1:100 (by volume) Trypticase Soy Broth (TSB) and green fluorescent protein (GFP) labeled Pseudomonas fluorescens mut3. The

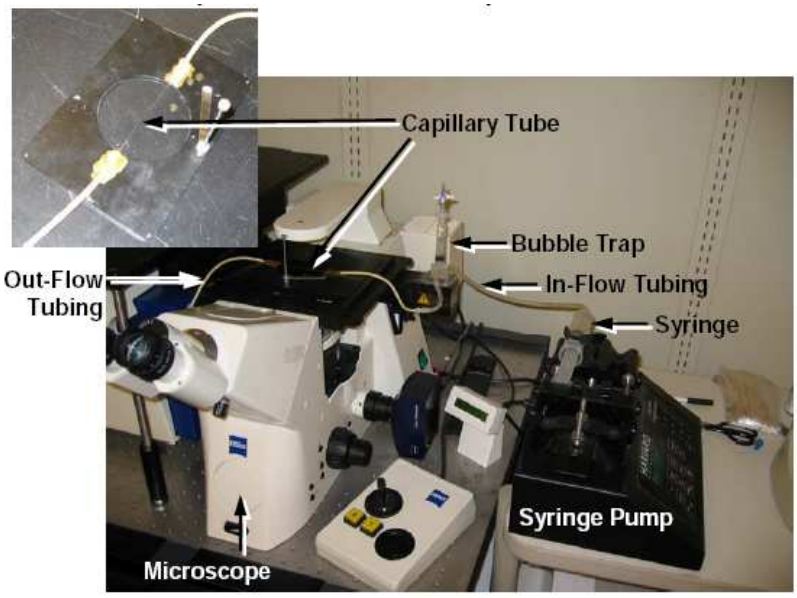


Figure 4.1. Photograph of biofilm growth experiment set-up

inoculum (0.5 ml) were mixed with 60 ml of the TSB in the 60 ml syringe. Inoculum counts for the 8 experiments that were run are summarized in Table 1. A more detailed description of the biofilm growth protocol is included in Appendix A. Once the capillary tube was filled, image acquisition started. Images were acquired using the Carl Zeiss AIM software. Images were obtained approximately every 15 minutes for up to 6 hours. X-Y slices were obtained every 1.31 m using a 20 objective for images of dimensions 642 642 m with voxel sizes of 1.26 1.26 m in the X-Y dimensions.

4.2.2 Data Analysis

Data acquired from the LSCM was analyzed using COMSTAT (**??**). The metrics used to quantify the biofilm growth were biovolume, average thickness, roughness coefficient and surface area to biovolume area. Biovolume is defined as follows:

$$Biovolume\left[\frac{\mu m^3}{\mu m^2}\right] = \frac{(number\ of\ biomass\ pixels\ in\ images)}{(voxel\ volume)substratum\ area} \tag{4.1}$$

To calculated the average thickness, the highest point with biomass above each x,y pixel is determined and the height of these points are averaged. Pores and voids within the biofilm are

Date Experiment Run	Inoculum Concentration (CFU/ml)
07/11/2008	1.32 109
07/18/2008	1.57 109
07/29/2008	1.45 109
08/28/2008	1.42 109
09/10/2008	1.64 109
09/11/2008	1.50 109
09/16/2008	2.89 109
09/17/2008	1.68 109

Table 4.1.

ignored. Average thickness is a metric of the spatial size of the biofilm.

The roughness coefficient is defined as:

$$R_a^* = \frac{1}{N} \sum_{i=1}^{N} \frac{|L_{fi} - \bar{L}_f|}{\bar{L}_f} \tag{4.2}$$

where L_{fi} is the ith individual thickness measurement, $\bar{L}$ is the mean thickness, and N is the number of thickness measurements. The roughness coefficient provides a measure of how much the thickness of the biofilm varies and thus is an indicator of the biofilm heterogeneity. Finally, the surface area to biovolume area reflects the portion of the biofilm that is exposed to nutrient flow. The surface area is the summation of all biomass voxels that have a surface exposed to the background (not adjacent to another biomass voxel). This value is divided by the total volume of biomass. Biomass, average thickness, roughness coefficient and surface area to biovolume area were plotted against time for qualitative comparisons of trends of the metrics. Rates of change were calculated by fitting the data with linear regressions (Kaleidagraph 4.03). Data at each time point were also averaged to determine trends in mean and variance of the data as a function of time. Cumulative probability plots for the rate constants were calculated by Kaleidagraph 4.03.

4.2.3 Results and Discussion

Quantitative analysis of the biofilms show that both the biovolume and average thickness of the biofilm increases over the duration of the growth experiment, indicating that the biofilm is still in a growth phase (Figure 4.2). The roughness coefficient decreased over the duration of the experiments, indicating that the biofilm is becoming more homogeneous (Figure 4.3). Finally, for four of the experiments the surface area to biovolume ratio decreased, indicating decreasing access of nutrients to the biofilm, or that the biofilm is filling in (Figure 4.3). However, for the other four

experiments the surface area to biovolume ratio appeared to stay relatively constant, despite the decrease in roughness coefficient. Trends were relatively linear for seven of the eight experiments indicating a constant rate of change for the given metric (Figures 4.2 and 3).

The rates for each metric were variable between experiments. When the data for the experiments were averaged together, this linear trend became even more pronounced. The linear model for both biovolume and average thickness, can lead to a unphysical negative intercept for some of the experiments. Thus, a different rate might exist for very early times. The exception to the constant rate was for the experiment run on July 18th, 2008. For this experiment, biovolume, average thickness and roughness coefficient stayed relatively constant for the first three hours of the experiment. During this same time, the surface area to biovolume ratio decreased. Thus, it appears at earlier time the growth of the biofilm is nutrient limited. After three hours, the decrease in surface area to biovolume ratio starts to level off and the other three parameters start following the trends of the other three experiments. Images of the biofilms are presented in Appendix B. For the experiment run on July 18th, there are clear areas at early times where there are much less bacteria than other areas. The biofilm appears to be filling in around the time that the trends in the metrics match those of the other experiments (Figures 4.2 and 4.3). Experiments run on July 29th, August 28th, and September 17th appear to have less dense biofilms than the other two experiments. This observation correlates to the higher surface area to biofilm volume calculated for these biofilms (Figure 4.3). Bacteria of biofilms traveling in the fluid phase can be observed at early times for experiments runs on September 10, 11, and 16, 2008. This is most likely biofilms sloughing of from upstream of where the images were collected. Analyses indicate that there is not a significant change in the metrics when the areas are excluded from the calculations. Given that a linear model fits most of the experimental results, it was assumed that a constant rate could be used in the numerical modeling for which these data are intended. As the intercepts calculated for biovolume and average thickness for many of these experiments assuming a constant rate are negative and thus non-physical, the model for these parameters are forced through the origin. Probability graphs were generated based on the experimental data (Figure 4.4). If the distributions were Gaussian, the lines shown on the probability graphs would be linear. The rates for biovolume appear to be bimodal. More experiments need to be run in order to generate smoother trends and determine whether or not the distributions are Gaussian, bimodal, or another distribution.

4.3 Recommendations for Model Input

The intention of these experiments is to provide experimental input to assist in biofilm growth numerical models. Linear models have been used to describe the change in metrics for characterizing biofilms as a function of time. Note that these models are only for early time growth (up to 6 hours) under very specific experimental conditions. Yet, they do describe the variability in growth for these specific experiments.

The first step in using these experimental data as input to biofilm growth models is to link the

metrics measured in these experiments to relevant model parameters. What parameters are used will be numerical-model dependent. We suggest running an optimization routine to determine which model parameters are sensitive to biofilm growth and to determine a numerical relationship between these parameters and the experimental metrics described above. The optimization problem should be able to provide parameter values with confidence intervals that in turn can be used for forward, probabilistic simulations to model biofilm growth. As we do not have the information from the optimization runs, for further discussion, it is assumed that the rates presented in Figure 4.4 are direct inputs into the biofilm growth model.

The probability distributions presented in Figure 4.4 can be used to randomly sample rates for the four different metrics. Low probability rates would have a lower chance of being selected than rates with a 50randomly selected, the model can be run in forward mode. For the modeling to be significant, many realizations are needed so that a representative sampling of the parameter space is made. Note that an average biofilm could also be run using the rates measured from the averages of the experiments shown in Figures 4.2 and 4.2.

One thing that has not been examined is correlation between the different metrics. To more accurately model biofilm growth it is important to determine whether or not the metrics are correlated with each other. If a correlation is found, it must be taken into account. These experiments were run for a short duration, using one microorganism under one set of environmental conditions. In reality, biofilm growth is much more complicated as many organisms can be involved and environmental conditions can vary. However, we believe that this set of experiments is a good starting point for testing a biofilm growth model.

4.4 Conclusions

The analysis of these experiments demonstrates how trends in biofilm growth can be observed from experiment to experiment and how these trends and the variability in these trends can be quantified. Quantification of the metrics characterizing biofilm growth can be used in numerical modeling as described above. It should be noted that this quantification was only performed on eight experiments. To develop a more realistic statistical model, many more experiments need to be run. The next step to better quantify biofilm growth would be to change variables in the experiment (e.g., flow rate, nutrient concentration, inoculant concentration) to determine how the metrics vary with changes in experimental conditions. The time span the experiments are run could also be increased in order extrapolate the model over longer time spans.

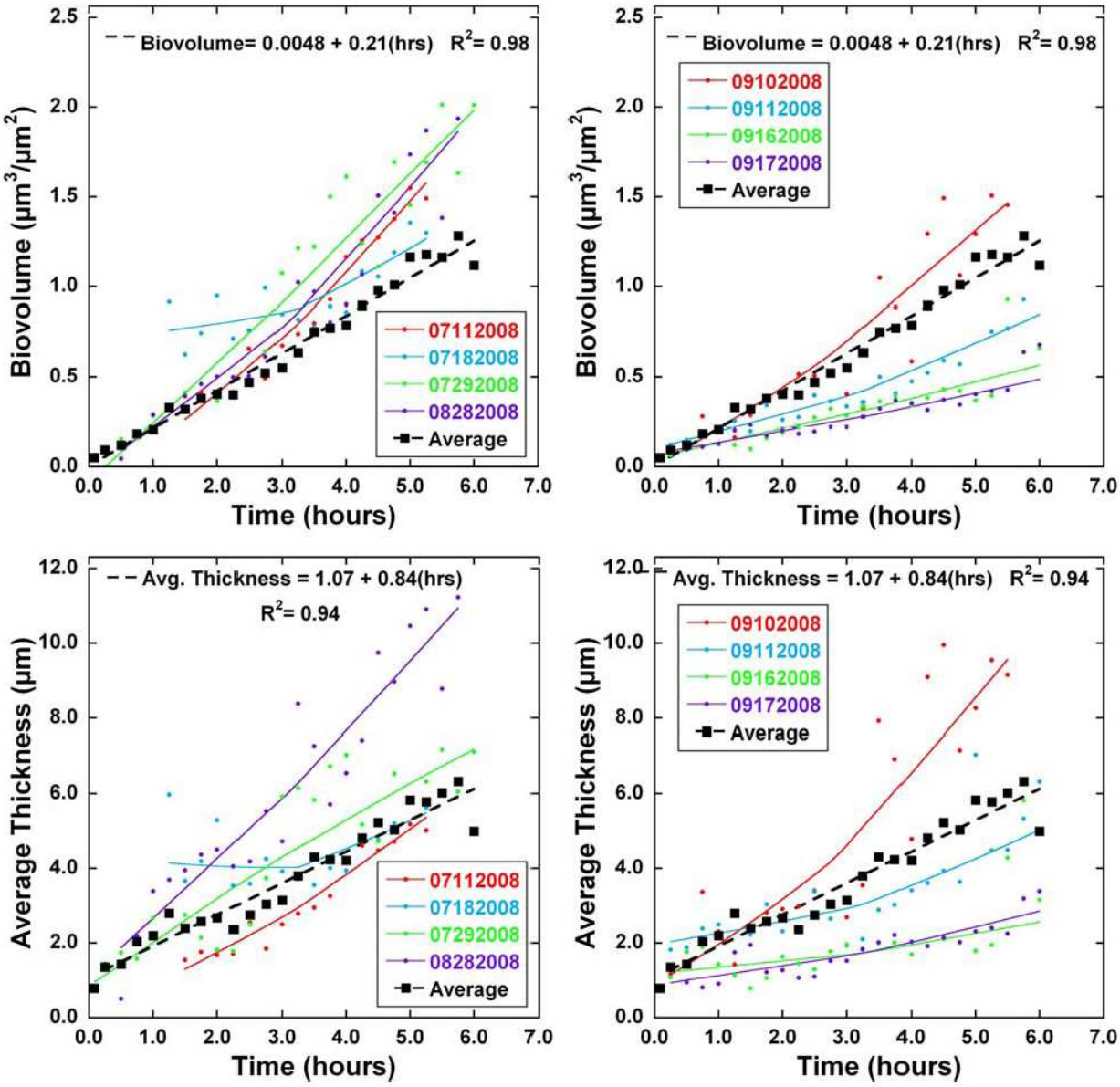


Figure 4.2. Trends in biovolume (top) and average thickness (bottom) as calculated by COMSTAT for 8 different biofilms. Linear fit of the average values for the eight experiments are shown in black

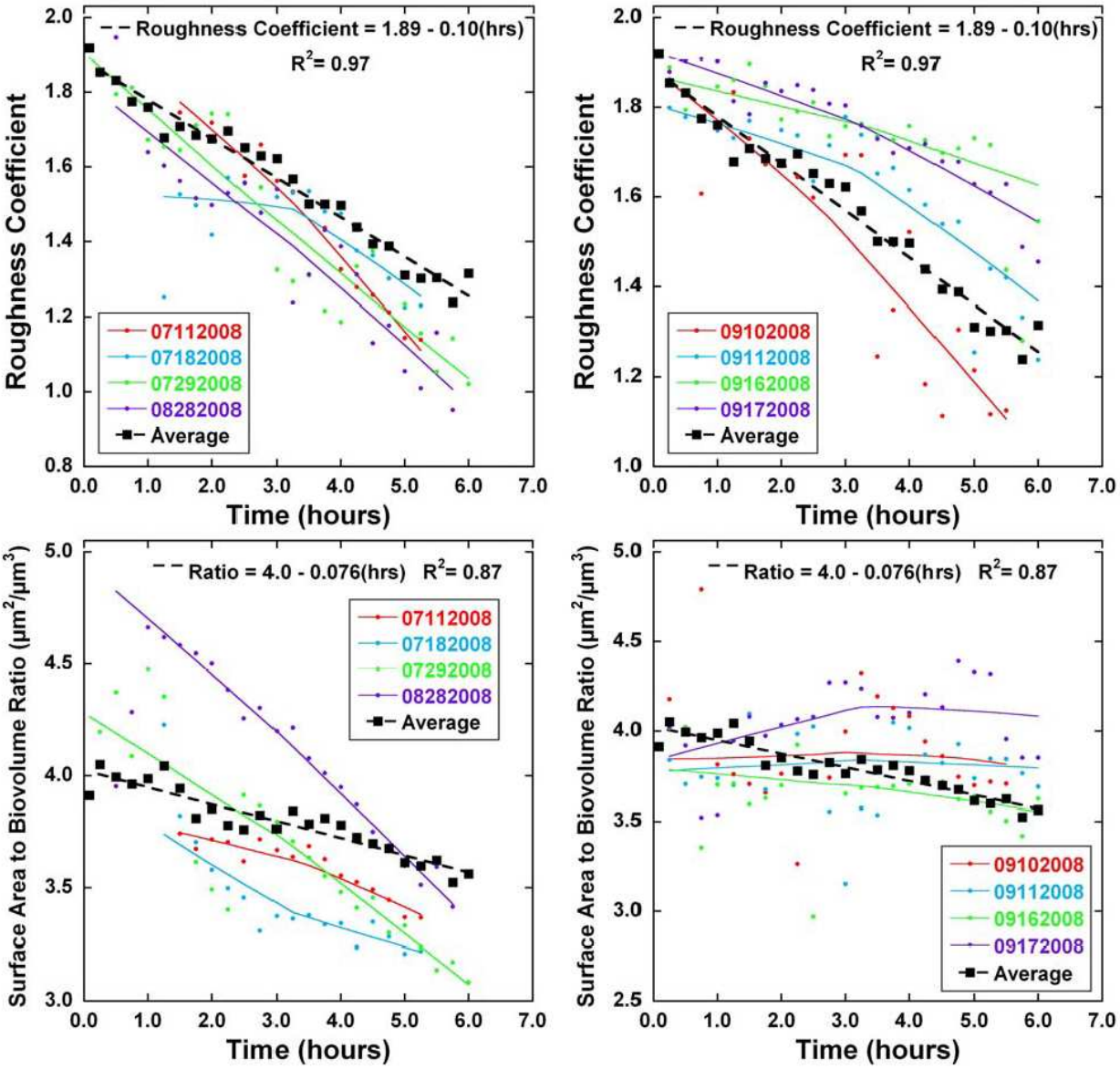


Figure 4.3. Trends in biovolume (top) and average thickness (bottom) as calculated by COMSTAT for 8 different biofilms. Linear fit of the average values for the eight experiments are shown in black

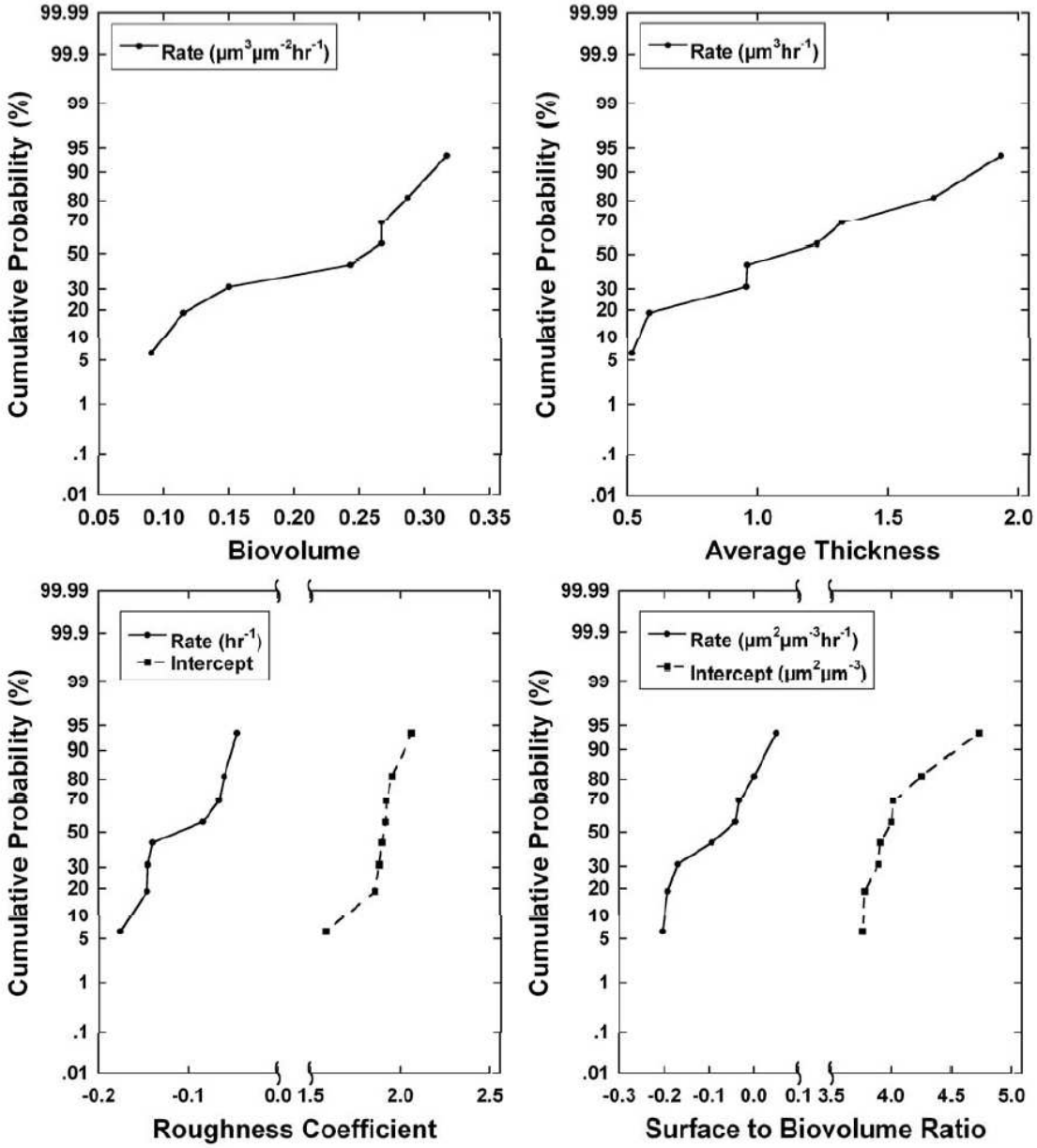


Figure 4.4. Probability distribution functions describing the slope and intercepts of the linear model to describe the metrics for biofilm growth.

Chapter 5

Stochastic Inversion for Reaction Coefficients

5.1 Introduction

As discussed in previous chapters, the goal is to ultimately calibrate numerical models with laboratory observations. However, our inversion strategy is plagued by considerable uncertainty as a result of the complexities associated with accurately predicting biofilm dynamics and with the variability of experimental biofilm growth. Multi-physics phenomena, chemical and electrostatic behavior at multi-spatial and temporal scales, and undefined biological phenomena force simplifications to enable practical simulation development with tractable computational requirements. In addition, growing biofilms in the laboratory present difficult challenges in particular achieving repeatable results, despite efforts to simplify and robustify the process. Given these sources of uncertainty, it seemed appropriate at some level to address the quantification of uncertainty as part of our numerical development even though UQ was outside the scope of the ldrd.

Uncertainty quantification is a difficult topic encompassing a large range of algorithms, techniques and strategies. However, a logical step to apply UQ to this project would be to maintain the general theme by building basic infrastructure from which further developments can be made in the future. Leveraging as much from the deterministic froward simulation and in line with our inversion theme, we have developed the basic infrastructure to perform stochastic inversion using Bayesian theory, adaptive Markov Chain Monte Carlo (AMCMC) and our level set simulator. The goal was to develop basic capabilities to infer certain stochastic properties of the level set simulator given information from experimental observations. Instead however of inferring for initial conditions which would be computationally intractable in the stochastic setting, reaction coefficients are selected as target inference parameters.

Perhaps the biggest challenge with Bayesian methods is the computational expense for large numbers of inversion parameters. Several methods to reduce the cost of a posterior evaluation can be considered such as model reduction of the forward simulation including coarse grids, surrogate models, proper orthogonal decomposition, and stochastic finite elements. In this project, we explored reduced order modeling for the deterministic inversion case and we implemented a

prototype stochastic finite element capability into Sundance. These capabilities have not been applied to the biofilm problem, but have been tested on basic prototype problem (heat equation and convection-diffusion) and we refer the reader to [17, 24] for additional details of our reduced order modeling approach and to code examples in the Sundance repository [43].

Bayesian inference methods have received much attention recently [48, 7, 40], with applications ranging from geophysics [36, 44] and climate modeling [38] to heat transfer [63, 64]. For additional details we refer the interested reader to several excellent references [40, 57, 48]. Our solution approach consists of the adaptive Markov Chain Monte Carlo method and an overview of this and related techniques can be reviewed in [1].

5.2 Formulation

Consider a forward problem defined as follows:

$$\mathbf{d} \approx \mathbf{G}(\mathbf{m}) \tag{5.1}$$

Here $\mathbf{m}$ is a vector of model parameters or inputs and $\mathbf{d}$ is a vector of observable quantities, or data; for simplicity, we let both be real-valued and finite-dimensional. The forward model $\mathbf{G}$ yields predictions of the data as a function of the parameters. In the Bayesian setting, $\mathbf{m}$ and $\mathbf{d}$ are random variables. We use Bayes' rule to define a posterior probability density for $\mathbf{m}$, given an observation of the data $\mathbf{d}$:

$$p(\mathbf{m}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{m})p_m(\mathbf{m}) \tag{5.2}$$

In the Bayesian paradigm, probability is used to express knowledge about the true values of the parameters. In other words, prior and posterior probabilities represent *degrees of belief* about possible values of $\mathbf{m}$, before and after observing the data $\mathbf{d}$.

Data thus enters the formulation through the likelihood function $p(\mathbf{d}|\mathbf{m})$, which may be viewed as a function of $\mathbf{m}$: $L(\mathbf{m}) \equiv p(\mathbf{d}|\mathbf{m})$. A simple model for the likelihood assumes that independent additive errors account for the deviation between predicted and observed values of $\mathbf{d}$:

$$\mathbf{d} = \mathbf{G}(\mathbf{m}) + \eta \tag{5.3}$$

where components of η are i.i.d. random variables with density p_η. The likelihood then takes the form

$$L(\mathbf{m}) = p_\eta\left(\mathbf{d} - \mathbf{G}(\mathbf{m})\right) = \prod_i p_\eta\left(d_i - G_i(\mathbf{m})\right). \tag{5.4}$$

In this simple model, η may encompass both measurement error (e.g., sensor noise) and the extent to which forward model predictions may differ from "true" values because of some unmodeled physics of the system.

Additional information on the model parameters may enter the formulation through the prior density, $p_m(\mathbf{m})$. Prior models may embody simple constraints on $\mathbf{m}$, such as a range of feasible values, or may reflect more detailed knowledge about the parameters, such as correlations or smoothness. In the absence of additional information, one may choose a prior that is *uninformative*. Here we will focus on uniform prior with simple bounds.

If parameters $\phi_{\mathbf{m}}$ of the prior density $p_m(\mathbf{m}|\phi_{\mathbf{m}})$ or parameters ϕ_η of the error model $p_\eta(\eta_i|\phi_\eta)$ are not known *a priori*, they may become additional objects for Bayesian inference. In other words, these *hyperparameters* may themselves be endowed with priors and estimated from data [48]:

$$p(\mathbf{m}, \phi_{\mathbf{m}}, \phi_\eta|\mathbf{d}) \propto \mathbf{p}(\mathbf{d}|\mathbf{m}, \phi_\eta)\mathbf{p}_{\mathbf{m}}(\mathbf{m}|\phi_{\mathbf{m}})\mathbf{p}(\phi_\eta)\mathbf{p}(\phi_{\mathbf{m}}). \tag{5.5}$$

The resulting joint posterior over model parameters and hyperparameters may then be interrogated in various ways—e.g., by marginalizing over the hyperparameters to obtain $p(\mathbf{m}|\mathbf{d})$; or first marginalizing over $\mathbf{m}$ and using the maximizer of this density as an estimate of the hyperparameters; or by seeking the joint maximum *a posteriori* estimate or posterior mean of $\mathbf{m}$, $\phi_{\mathbf{m}}$, and ϕ_η [48]. In the present study, our implementation can easily accommodate hyperparameters to help describe aspects of the prior covariance.

5.3 Numerical results for a single parameter prototype

We leverage our high level abstraction software for the forward problem simulation but have created an interface with python scripts that communicates with separate C programs for the adaptive MCMC process.

Our deterministic forward simulation $\mathbf{G}$ is represented by the dynamics of biofilm growth in the form of convection and diffusion-reaction PDEs:

$$\begin{aligned}
\tfrac{\partial c}{\partial t} - D\Delta c &= r && \in \Omega \\
c &= c_D 1 && \in \Gamma_{D_1} \\
c &= c_D 2 && \in \Gamma_{D_2} \\
c &= c_0 && \in \Omega \times (0, T) \\
\tfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= f(c, \rho) \in \Omega \\
\rho &= \rho_D 1 && \in \Gamma_{D_1} \\
\rho &= \rho_D 2 && \in \Gamma_{D_2} \\
\rho &= \rho_0 && \in \Omega \times (0, T)
\end{aligned}$$

where the reaction term r is defined as:

$$r = \frac{\mu_g}{y_o} m_c c \tilde{\rho} k_2 + c \tag{5.6}$$

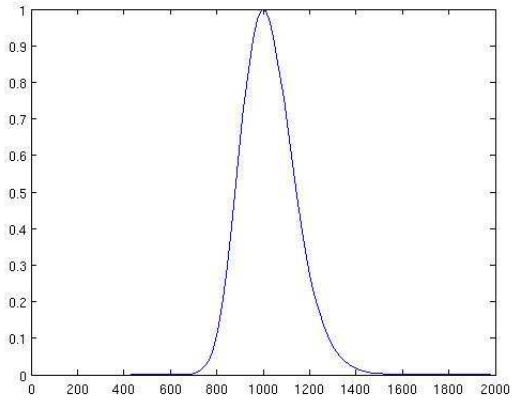


Figure 5.1. Posterior distribution from a parameter study with $k_1 = 1000$

The goal of this work is to calculate posterior distributions for a multivariate parameter space consisting of the diffusion coefficient and both reaction terms k_2 and $k1$ which is a lumping of maximum biomass growth rate, μ_g, yield of biomass on oxygen y_o, and biomass m_c. As a transition, a single parameter case is first implemented to verify the implementation by simply conducting a parameter study and plotting the solution versus iterates. This then should present the posterior distribution and serve as verification to approximate solution techniques which needs to be deployed for the multivariate case.

In the uni-variate case, a simple parameter study will provide the exact posterior distribution 5.1 and therefore can serve as the target solution for the AMCMC solution approach. Figure 5.2 shows the histogram obtained from running AMCMC 4000 iterations. Figure 5.3 shows the corresponding chain position with good mixing. These results are terribly interesting but demonstrate a fundamental capability that can be easily applied to the multivariate case.

5.4 Conclusions

A basic stochastic inversion capability was implemented to address the significant amount of uncertainty with the numerical models and experimental process. We solve a simple one parameter inversion that leverages the level set simulator and uses Bayesian inference theory. An adaptive MCMC routine is used to solve for the posterior distribution. Although this is simple example, the implementation allows for multi-variate problems and is designed to take advantage of stochastic finite elements in addition to goal oriented reduced order modeling.

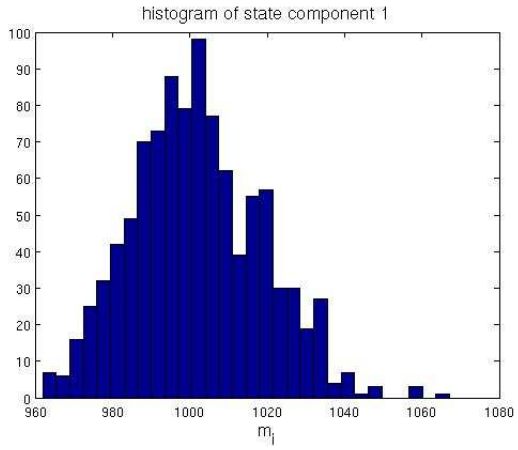


Figure 5.2. posterior histogram

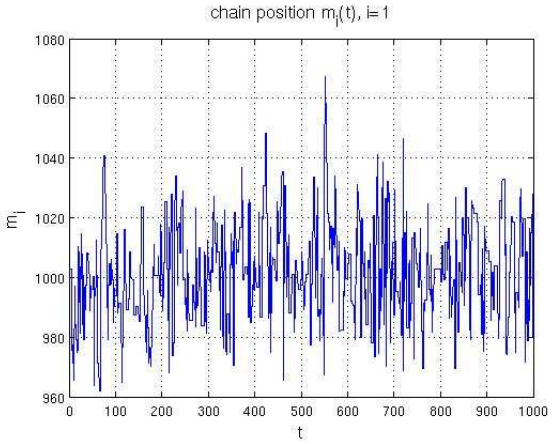


Figure 5.3. Chain position

Chapter 6

Macro Scale Modeling Using SIERRA/Aria

In this chapter macro-scale fluid flow and transport phenomena are investigated for typical and non-standard geometries of a water distribution network. Although, the majority of fluid flow occurs in straight pipe segments, problematic areas for biofilm deposition and interaction with chemicals will occur where geometries are more complicated. For instance, the flow at junctions, storage tanks, and processing facilities will exhibit recirculation patterns, diffusion dominated regions and specific convective paths. Biofilm formation, growth, detachment, and interaction with chemicals and biological agents in the bulk fluid will consequently result in unpredicatable behavior. At this scale, we simply represent biofilms as reaction terms at the boundaries and avoid multi-spatial scale phenomena. The focus here is on resolving macro-scale flow and transport for relatively complicated geometries.

Contamination events in distribution systems pose a particularly difficult challenge to utility companies. Even though contamination in the bulk fluid can be flushed or neutralized, biofilms can act as temporary repositories with possible releases of contaminants at random times. Mitigation procedures need to compensate for these release possibilities. Where biofilms are deposited, how they grow, and what interaction they might have with the bulk fluid need to be answered. The first two questions are beyond the scope of this work. Consequently, the deposition location of biofilm was predetermined based on qualitative selection criteria consiting of geometry and convective forces. In addition, the growth and interaction are represented by reaction terms.

Our primary goal of this part of the research is to resolve transport of a solute assuming that the interaction of biofilm with bulk fluid occurs via a reaction term on a subset of the boundary. Several challenges arise however, consisting of 1) resolving flow on complex geometries, 2) providing stabilization to offset high Peclet values in a finite element discretization, 3) reconciling adjoint based formulations for adaptivity using stabilization schemes, and finally, 4) developing appropriate adjoint based optimization algorithms for stabilized transport.

This chapter first presents an investigation of flow and transport with low Reynolds and Peclet numbers, less than 300 and 100, respectively. The objectives are to properly resolve fluid flow, develop reasonable solute transport, and perform optimization using adjoint based sensitivities. Secondly, a streamline upwind Petrov Galerkin (SUPG) approach is tested to investigate higher convection dominated processes. Thirdly, ajoint calculations are discussed in the context of error

estimation, adaptivity, optimization and SUPG.

6.1 Background

Significant research has been conducted in the area of adjoint based error estimation for finite element discretizations [16, 14, 9, 10, 18, 19, 35, 39, 49, 52, 59, 60]. The goal of adjoint based error estimation and adaptivity is to reduce the discretization error for a specific output functional. This dual-weighted residual approach has been applied to a range of problems including viscous fluid flow, chemically reacting flows, elasto-plasticity, and radiative transfer. Much less work however had been done on adjoints for the simultaneous use of error estimation and optimization. Bangerth [12] presents a framework in which continuous function spaces are used to formulate a nonlinear inverse problem. The Newton step and line search algorithms are formulated as continuos functions to allow for adaptation of the mesh.

An additonal implementation challenge for finite element discretizations is highly convective flows. Stabilization is typically required which also needs to be applied to the adjoint calculation. Brooks and Hughes [23] introduced streamline upwinding methods in a Petrov Galerkin formulation.

6.2 Model Formulation and Verification

In this section we present the model formulation for the flow and transport in a contact tank. We begin with the model equations and boundary conditions for both flow and transport. A simple 1D problem is used to illustrate the problem and provide a platform for basic code verification. Then we discuss the numerical stabilization using SUPG of the transport equation in the case of large Peclet numbers. We present the adjoint transport equations and define goal-oriented error estimators based on the adjoint problem.

6.2.1 Equations for the Forward Model of the Contact Tank

The mathematical model for the flow is defined by the stationary incompressible Navier Stokes equations along with appropriate boundary conditions. These can be formulated on a domain Ω as

follows.

$$\begin{aligned} \rho\, u \cdot \nabla u - \mu \Delta u + \nabla p &= \rho\, g && \text{in } \Omega, \\ \nabla \cdot u &= 0 && \text{in } \Omega, \\ u &= u_{\text{in}} && \text{on } \Gamma_{\text{in}}, \\ u &= 0 && \text{on } \Gamma_{\text{rxn}} \cup \Gamma_{\text{o}}, \\ \{-p\, I + \mu\, (\nabla u + \nabla u^t)\} \cdot n &= \mu\, n \cdot \nabla u^t && \text{on } \Gamma_{\text{out}}. \end{aligned} \tag{6.1}$$

The dominant dimensionless group for this model is the Reynolds number Re, which is defined as

$$\text{Re} \equiv \frac{\rho\, U\, L}{\mu},$$

where U and L are representative velocities and lengths.

The stationary transport of a species by convection diffusion is defined as:

$$\begin{aligned} u \cdot \nabla c - D \Delta c &= 0 && \text{in } \Omega, \\ c &= c_{\text{in}} && \text{on } \Gamma_{\text{in}}, \\ -D \nabla c \cdot n &= 0 && \text{on } \Gamma_{\text{o}} \cup \Gamma_{\text{out}}, \\ -D \nabla c \cdot n &= k\, c && \text{on } \Gamma_{\text{rxn}}. \end{aligned} \tag{6.2}$$

The dominant dimensionless group for this model is the Peclet number Pe, which is defined as

$$\text{Pe} \equiv \frac{U\, L}{D}.$$

A secondary dimensionless group is

$$\Pi \equiv \frac{k\, L}{D}.$$

For the contact tank, the boundary $\Gamma \equiv \partial\Omega$ is divided into four parts: the inflow Γ_{in}, the outflow Γ_{out}, the surface reaction Γ_{rxn}, and the remaining surface Γ_{o}.

- On Γ_{in}, we specify the fluid velocity and species concentration,
- On Γ_{out}, we specify an open flow boundary condition on the flow and a zero diffusive condition on the species concentration.
- On Γ_{rxn} we specify a first order reaction for the species The flow boundary condition is no slip.
- Finally, on the remaining surface Γ_{o}, we also specify a zero diffusive condition on the species and a no slip condition on the flow.

We can express the solution to the steady state convection diffusion problem defined by (6.2) in weak form. To do this, we define the function spaces $V \equiv \{v \in H^1(\Omega) : v|_{\Gamma_{\text{in}}} = c_{\text{in}}\}$ and $W \equiv \{v \in H^1(\Omega) : v|_{\Gamma_{\text{in}}} = 0\}$. The weak solution is then defined by: find $c \in V$:

$$B(c, w) = 0, \quad w \in W. \tag{6.3}$$

Here the bilinear form is defined by

$$B(c, w) \equiv (u \cdot \nabla c, w) + (D\nabla c, \nabla w) + \langle k\, c, w\rangle_{\Gamma_{\text{rxn}}}, \tag{6.4}$$

where we have used the usual notations for integrals $(v, w) \equiv \int_\Omega vw\, dx$, etc.

By choosing appropriate finite dimensional spaces $V_h \subset V$ and $W_h \subset W$ for the trial and test functions, respectively, we can define the Galerkin finite element approximation: find $c_h \in V_h$:

$$B(c_h, w_h) = 0, \quad w_h \in W_h. \tag{6.5}$$

In order to verify our algorithms, we now introduce a dimensionless convection-diffusion problem on a 1D domain $\Omega \equiv (0, 1)$:

$$\begin{aligned} c' - \epsilon c'' &= 0 \quad \text{in} \quad \Omega, \\ c(0) &= 0, \\ c(1) &= 1. \end{aligned} \tag{6.6}$$

where the parameter $\epsilon \equiv \text{Pe}^{-1}$ is the inverse Peclet number. This has the following analytical solution:

$$c(x) = \frac{1 - e^{x/\epsilon}}{1 - e^{1/\epsilon}} \tag{6.7}$$

This problem will be used throughout this section and has been used as a standard test problem for convection-diffusion algorithms.

6.2.2 SUPG stabilization

When the problem in (6.3) is convection-dominated, the standard Galerkin finite element formulation (6.5) is often unstable. This can be determined by calculating the local mesh Peclet number, defined by

$$\text{Pe}_h \equiv \frac{U\, h}{D}, \tag{6.8}$$

where h is the local mesh size and U is the local velocity magnitude. When $\text{Pe}_h \gg 1$, it is common to use some form of stabilization, such as Streamwise Upwind Petrov-Galerkin (SUPG) [**?**] or Galerkin Least Squares (GLS).

The SUPG stabilized version of (6.5) that we use is defined by

$$B_\tau(c_h, w_h) = 0, \quad w_h \in W_h. \tag{6.9}$$

Here the stabilized bilinear form is defined by

$$\begin{aligned} B_\tau(c, w) &\equiv (-D\triangle\, c + u \cdot \nabla c, w + \tau\, u \cdot \nabla w) + (D\nabla c, \nabla w) + \langle k\, c, w\rangle_{\Gamma_{\text{rxn}}} \\ &= B(c, w) + (-D\triangle\, c + u \cdot \nabla c, \tau\, u \cdot \nabla w). \end{aligned} \tag{6.10}$$

The function τ is mesh-dependent, decreasing with the local mesh size. We have two forms of τ available, one from the original paper of Brooks and Hughes [23], and another developed by Dr. Farzin Shakib. In the code that we are using (SIERRA/Aria), the term

$$(-D\triangle\, c, \tau\, u \cdot \nabla w) \tag{6.11}$$

has been neglected. Since $\triangle\, c$ is zero for linear elements, this can only affect higher order approximations.

Figure 6.1 shows the H1 and L2 norm of the error for different mesh sizes for a Galerkin, Classic SUPG stabilized, and Shakib SUPG stabilized discretizations. It is clear that for linear elements (Q1), optimal rates in both norms are achieved for all three discretizations. However, for quadratic elements (Q2), the rates for the two SUPG stabilized methods are suboptimal in the L2 norm. We hypothesize that this is a result of the neglected term (6.11) in the stabilization, resulting in an inconsistent method.

6.2.3 Equations for the Adjoint Model

For calculation of both parameter sensitivities and error estimators, we will need the associated adjoint problem to (6.2), along with the appropriate boundary conditions. We also need to specify the linear functional that will drive the adjoint problem. For the contact tank problem, we consider the average species concentration leaving the tank, which is characterized by the linear functional

$$J(w) \equiv \frac{1}{|\Gamma_{\text{out}}|} \int_{\Gamma_{\text{out}}} w\, ds \equiv \int_{\Gamma_{\text{out}}} \psi\, w\, ds, \tag{6.12}$$

where $\psi \equiv 1/|\Gamma_{\text{out}}|$.

For this choice of functional, the adjoint model corresponding to the convection diffusion model (6.2) is the following:

$$\begin{aligned} -u \cdot \nabla\phi - (\nabla \cdot u)\,\phi - D\Delta\phi &= 0 && \text{in } \Omega \\ \phi &= 0 \text{ on } \Gamma_{\text{in}} \\ (-D\nabla\phi - u\,\phi) \cdot n &= -\psi && \text{on } \Gamma_{\text{out}} \\ (-D\nabla\phi - u\,\phi) \cdot n &= k\,\phi && \text{on } \Gamma_{\text{rxn}} \\ (-D\nabla\phi - u\,\phi) \cdot n &= 0 && \text{on } \Gamma_{\text{o}} \end{aligned} \tag{6.13}$$

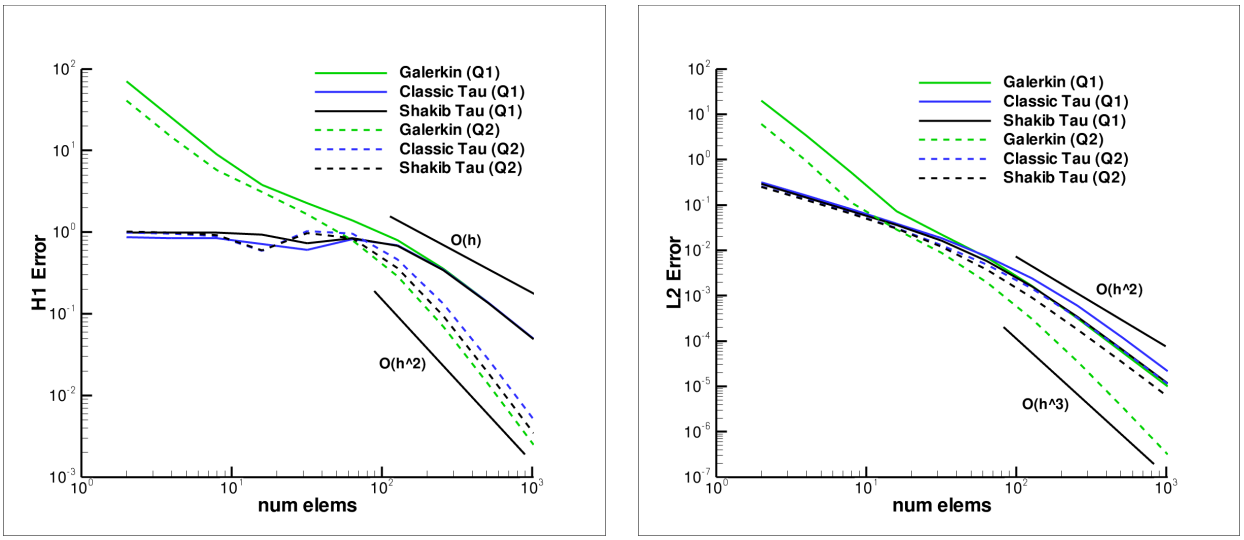


Figure 6.1. Verification of global error rates for 1D convection diffusion. (left) H1 norm of the error verus number of elements. (right) L2 norm of the error versus number of elements.

To derive the adjoint model in (6.13), we first introduce the variational form of the adjoint: find $\phi \in W$:

$$B(w, \phi) = J(w), \quad w \in W. \tag{6.14}$$

By applying integration by parts, we can rewrite this problem as

$$\begin{aligned} -(u \cdot \nabla\phi + (\nabla \cdot u)\,\phi + D\Delta\phi, w) \\ + \langle (D\,\nabla\phi + u\,\phi) \cdot n, w \rangle_{\Gamma_{\text{out}}} \\ + \langle (D\,\nabla\phi + u\,\phi) \cdot n + k\,\phi, w \rangle_{\Gamma_{\text{rxn}}} \\ + \langle (D\,\nabla\phi + u\,\phi) \cdot n, w \rangle_{\Gamma_{\text{o}}} \\ = \langle \psi, w \rangle_{\Gamma_{\text{out}}}. \end{aligned} \tag{6.15}$$

By varying the test function w we can show that the variational (6.14) and strong forms (6.13) of the adjoint are equivalent.

If we also make use of the fact the velocity is divergence free and satisfies no slip boundary

conditions on $\Gamma_{\text{rxn}} \cup \Gamma_{\text{o}}$, the adjoint problem can be reduced further to

$$\begin{array}{rcll} -u \cdot \nabla\phi - D\Delta\phi & = & 0 & \text{in } \Omega \\ \phi & = & 0 & \text{on } \Gamma_{\text{in}} \\ (-D\nabla\phi - u\,\phi) \cdot n & = & -\psi & \text{on } \Gamma_{\text{out}} \\ -D\nabla\phi \cdot n & = & k\,\phi & \text{on } \Gamma_{\text{rxn}} \\ -D\nabla\phi \cdot n & = & 0 & \text{on } \Gamma_{\text{o}} \end{array} \tag{6.16}$$

Of interest is the fact that the velocity still appears in the boundary condition on the outlet.

For the 1D model problem (6.6), the corresponding adjoint equation for the average value on Ω is:

$$\begin{aligned} -\phi' - \epsilon\phi'' = 1 \quad \text{in } \Omega, \\ \phi(0) = 0, \\ \phi(1) = 0. \end{aligned} \tag{6.17}$$

This has the following analytical solution:

$$\phi(x) = -x + \frac{1 - e^{-x/\epsilon}}{1 - e^{-1/\epsilon}}. \tag{6.18}$$

Figure 6.2 shows the H1 and L2 norm of the error in the discrete adjoint for different mesh sizes for the Galerkin, Classic SUPG stabilized, and Shakib SUPG stabilized discretizations. The error rates for the discrete adjoint are essentially the same as for the discrete forward problem.

6.3 Error Estimation

In order to estimate the error in solution responses of interest to design and optimization, we use the adjoint approach. The error in the model problem is defined by

$$e \equiv c - c_h. \tag{6.19}$$

We also make use of the error orthogonality relation

$$B(e, w_h) = 0, \quad w_h \in W_h. \tag{6.20}$$

Let J be a linear functional of interest, and consider the solution of the adjoint problem: find $\phi \in W$:

$$B(w, \phi) = J(w), \quad w \in W. \tag{6.21}$$

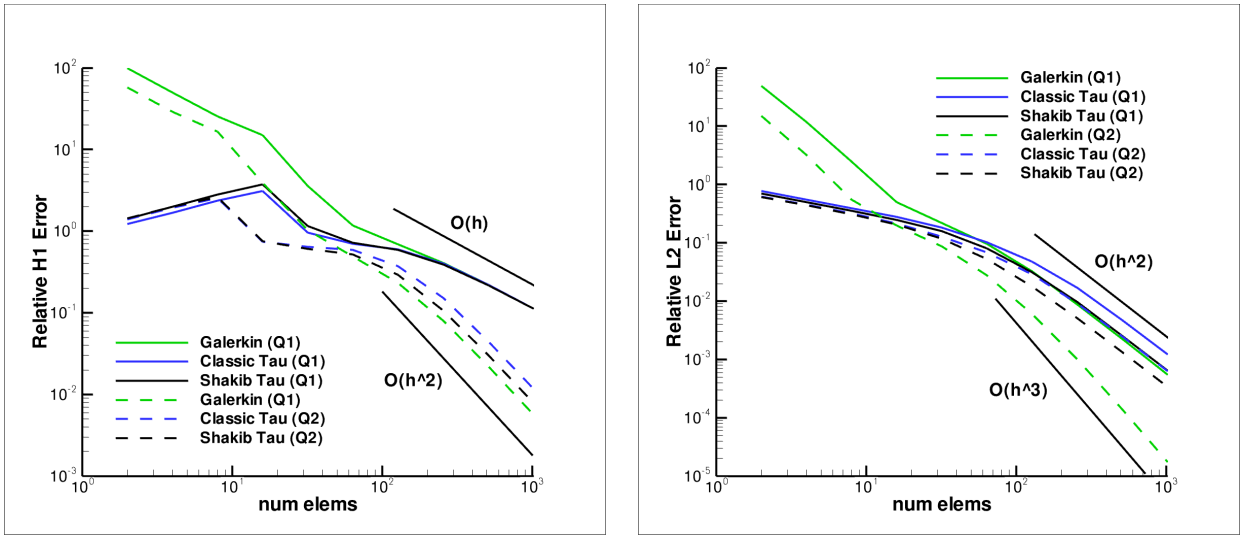


Figure 6.2. Verification of global error rates in the adjoint for 1D convection diffusion. (left) H1 norm of the adjoint error versus number of elements. (right) L2 norm of the adjoint error versus number of elements

The adjoint solution can be used to derive an error estimate for the error in the linear functional J as follows.

$$\begin{aligned} J(c) - J(c_h) &= J(e) \\ &= B(e, \phi) \\ &= B(e, \phi - \phi_h) \\ &= -B(c_h, \phi - \phi_h) \end{aligned} \tag{6.22}$$

Here $\phi_h \in W_h$ is an arbitrary finite element test function. In practice, we take ϕ_h to be the interpolant of ϕ in W. Substituting the definition of the bilinear form, we obtain the specific error formula

$$J(c) - J(c_h) = -(u \cdot \nabla c_h, \phi - \phi_h) - (D\nabla c, \nabla(\phi - \phi_h)) - \langle k\, c, \phi - \phi_h \rangle_{\Gamma_{\text{rxn}}} \tag{6.23}$$

In order to derive a computable error indicator, we must approximate the solution to the adjoint problem (6.21). To do this, we consider two approaches. The first is to compute a higher order approximation to the adjoint solution. For example, if we solve (6.5) using linear elements, then we approximate an approximate adjoint $\tilde{\phi}$ using quadratic elements. Then we compute the approximate error estimate using the adjoint weight function

$$\tilde{\phi} - I_h\, \tilde{\phi}$$

where $I_h : W \rightarrow W_h$ is the interpolant into the linear finite elements.

An alternate approach is to first compute the finite element approximation $\phi_h \in W_h$ using the same space as for the forward problem:

$$B(v_h, \phi_h) = J(v_h), \quad v_h \in W_h.$$

Then we use recovery operators $R_h$for value and gradients to compute the adjoint weight function

$$R_h(\phi_h) - \phi_h.$$

We note that for both approaches, we neglect any residuals associated with surface flux boundary conditions. In general these weighted residual contributions should be included.

For our test 1D problem, we compare the effectiveness of the two approaches for Pe $= 400$. The relative error versus number of elements is first plotted for both linear and quadratic elements. This problem exhibits superconvergence in the average value, since the error rate is greater than $O(h^{2p})$, where p is the polynomial degree. This may indicate that this is not a good test problem. We then compare the ratio of the error indicator to exact error for both methods of approximating the adjoint weights. The approach using a higher order solve (available only for linear elements) generally produces error ratioes much closer to one. In contrast, the recovery based approach intially produces somewhat reasonable ratios, but then degrades as the mesh is refined. It is not clear if this is a result of the superconvergence or the error indicator.

6.4 Numerical Results for a Prototype 2D Problem

6.4.1 Implementation Details

The numerical models were implemented in the finite element code Aria, which is part of the SIERRA Mechanics family of codes developed at Sandia. Aria is capable of first and second order finite elements on locally refined (h-adaptive) meshes.

The adjoint solver capability was achieved by using capabilities from the Trilinos solver library. This involved creating a stateless interface to the finite element model residual and Jacobian. This interface, called the Model Evaluator, was then available for the forward solve, or for use in the adjoint solve. The forcing data for the adjoint solve, which was typically the average value on a volume or surface, was generated by the SIERRA library Encore.

In the near future, the Model Evaluator interface will enable so-called intrusive analysis capabilities that require access to the application code's finite element residual vector and Jacobian matrix. These include high level code for time integration, optimization, error estimation, and uncertainty quantification.

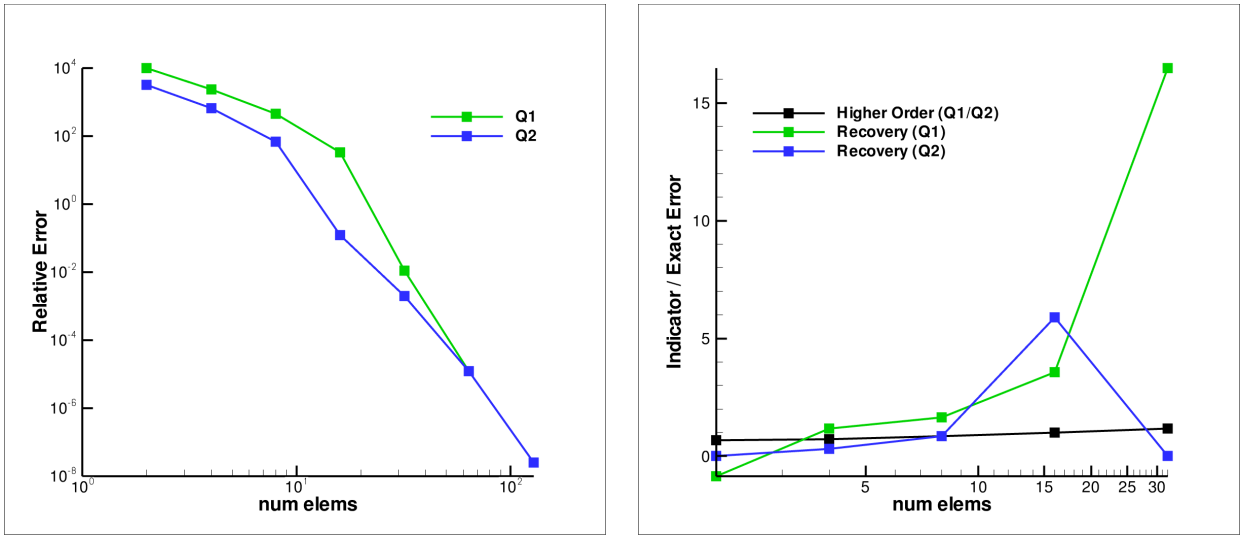


Figure 6.3. Verification of adjoint error indicator for 1D convection diffusion. (left) Exact relative error for Q1/Q2 elements. (right) Ratio of error estimator to exact error.

6.4.2 Solution Verification for a Prototype 2D Problem

In preparation of simulating a large scale 2D dataset, solution verification is first applied to a simplified 2D model. Absent an exact analytic solution, our truth model consist of a fine grid solution whereby errors are computed by subtracting interim solutions at coarse grid resolutions.

For our test problem, we considered a single channel with prescribed parabolic velocity in the positive x direction (see Figure 6.4). The intent of the prototype is to emulate a subportion of a more complicated geometry with similar dynamics. The problem is specified by (6.2), now with reactions specified on the entire channel sides ($\Gamma_o = \emptyset$).

Figures 6.5-6.6 show the forward and adjoint solutions computed on the fine (truth) mesh in the case of Re $=$ Pe $= 100$. In Figure 6.5, reactions on the boundary consume species concentration resulting in a reduction in average concentration from 1.0 to about 0.8125. The adjoint solution in Figure 6.6 is driven by the objective function which is the average value at the outlet boundary condition. As illustrated in (6.13), this objective function becomes a flux boundary condition at the outlet. The intersection of the reaction boundary condition causes large spikes in the adjoint solution.

The ideal convergence rates for finite element approximations of polynomial degree p are shown in Table 6.7. In practise and for more complicated datasets, the rates are typically lower

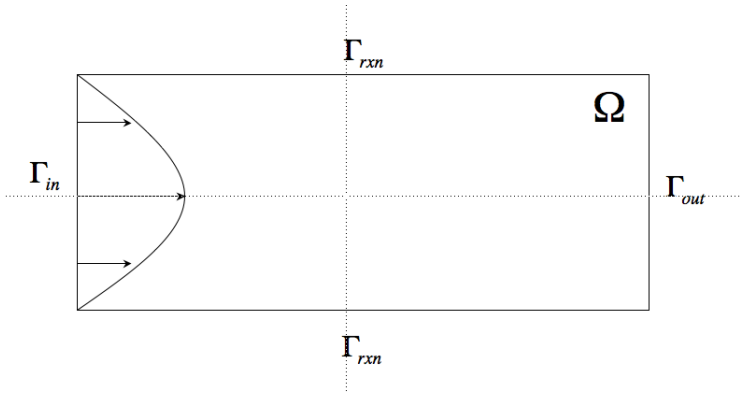


Figure 6.4. Computational domain for a 2d channel.

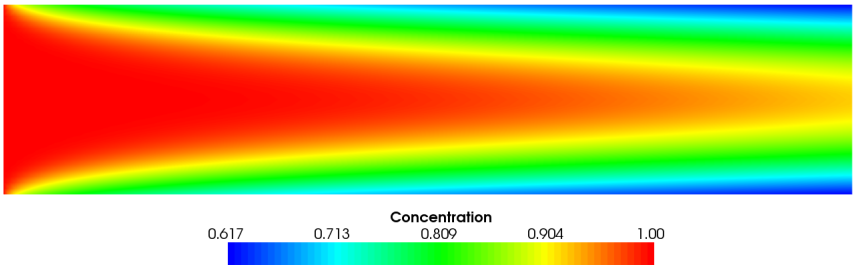


Figure 6.5. Forward solution (concentration) to prototype 2D problem with $\mathrm{Re} = \mathrm{Pe} = 100$.

then these.

Figure 6.8 demonstrates the observed convergence rates for the forward and adjoint solution in both the H^1 and L^2 norms. For linear elements, the forward solution converges at the optimal rates. However, quadratic elements exhibit anomolies that either is related to code implementation or is related to the solution regularity. The adjoint solution appears to be optimal for linear elements but also exhibits the same anomolies for quadratics.

Convergence rates for the objective function is shown in Figure 6.9. For uniform refinement the linear elements demonstrate optimal rates, quadratic elements initial show optimal behavior but then degrades. This again points to either regularity of the solution or code implementation issues. H-adaptivity clearly results in more efficient convergence in comparison to the uniform refinement with linear elements.

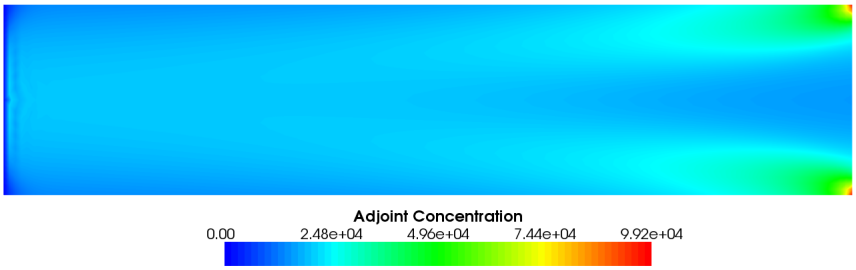


Figure 6.6. Adjoint solution to prototype 2D problem with $\mathrm{Re} = \mathrm{Pe} = 100$.

H^1 norm	L^2 norm	Obj. Fcn.
$O(h^p)$	$O(h^{p+1})$	$O(h^{2p})$

Figure 6.7. Optimal convergence rates for finite elements of degree p.

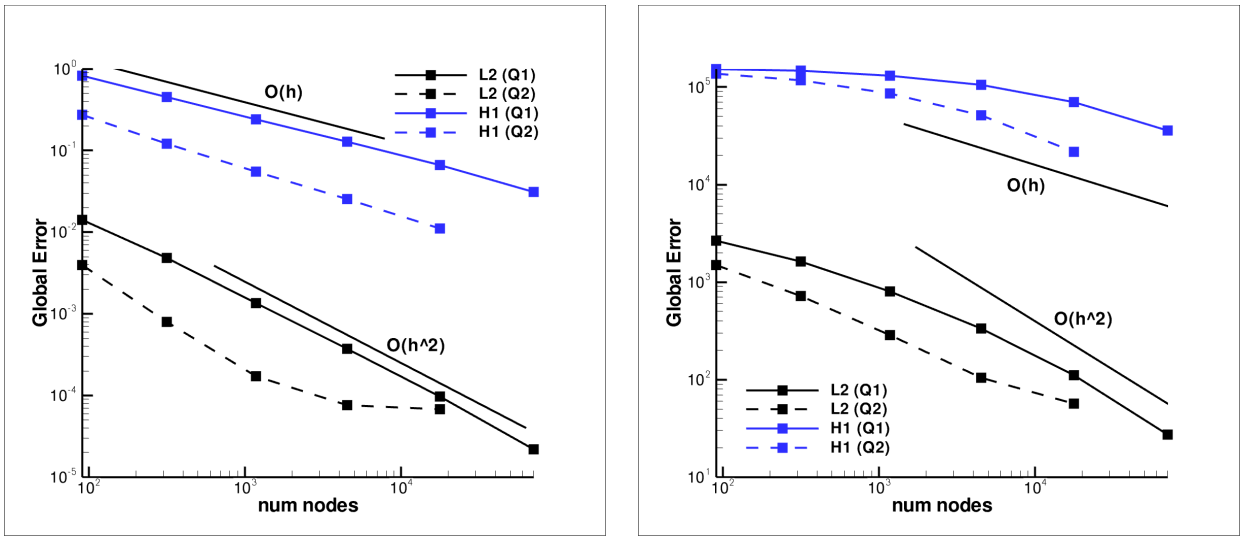


Figure 6.8. Convergence of global error norms for linear and quadratic elements. Left figure is forward solution, right figure is adjoint solution.

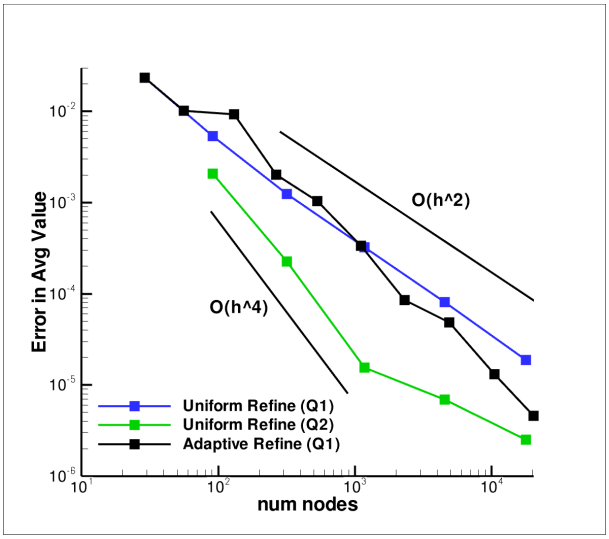


Figure 6.9. Convergence of error in average values for linear and quadratic elements.

The H-adapted meshes for this 2D channel problem are shown in Figure 6.10. The efficiency gains are a result of adaption in key local areas with coarser grid elsewhere.

6.5 Optimization and adaptivity

The optimization problem is formulated as

$$\min_{c,k} \ \mathcal{F}(c,k) = \frac{1}{2}\sum_{i=0}^{N} \int_{\Gamma_{\text{rxn}}} (c-c^*)^2 \delta(\mathbf{x}-\mathbf{x}_i)\, \mathbf{dx} + \frac{\beta}{2}\int_{\Omega} k^2\, \mathbf{dx} \tag{6.24}$$

subject to the physical constraints

$$B(c,w) = 0, \quad w \in W. \tag{6.25}$$

We solve this constrained optimization problem by deriving the first order optimality conditions. Introducing Lagrange multipliers the contrained problem can be combined into a single functional:

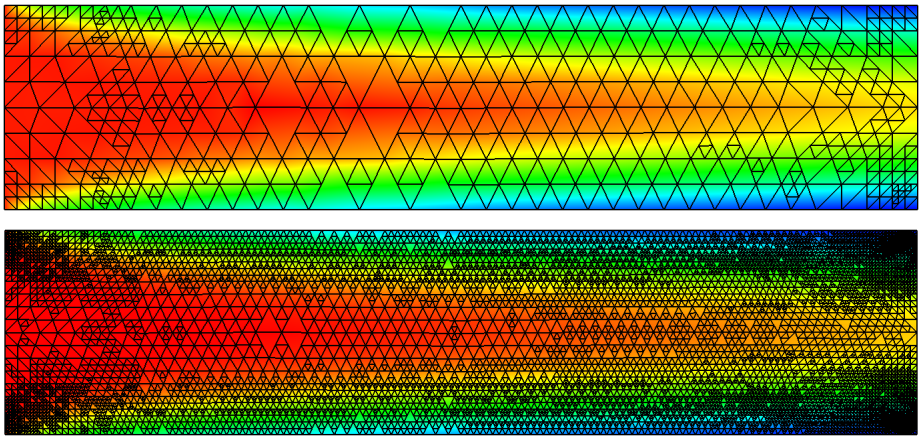


Figure 6.10. Adapted meshes using adjoint error indicator. Top solution is 4x refined, bottom solution is 8x refined.

$$\mathcal{L}(c, \lambda, k) = \mathcal{F}(c, k) + \lambda B(c, w) \tag{6.26}$$

First order optimality conditions are obtained by taking variations of the Lagrange functional with respect to c, λ, k. Taking variations with respect to the Lagrange multipliers and equating the results for all admissible variations, recovers the forward problem $B(c, w)$. Taking variations with respect to c and integrating by parts results in the adjoint equation in the strong form:

$$-u \cdot \nabla\lambda - D\Delta\lambda = 0 \quad \text{in} \quad \Omega \tag{6.27}$$

$$\lambda = \sum_{i=0}^{N} (c - c^*)\delta(\mathbf{x} - \mathbf{x}_i) \quad \text{on} \quad \Gamma_{\text{rxn}} \tag{6.28}$$

$$\lambda = 0 \quad \text{on} \quad \Gamma_{\text{in}} \tag{6.29}$$

$$\tag{6.30}$$

Although we have implemented the continuous adjoint for comparison purposes, our numerical experiments were conducted with the discrete adjoint, which is obtained by simply transposing the forward jacobian. It is the discrete adjoint that is exact and prefered for optimization ([50]) even though the continous and discrete adjoint are equivalent as the mesh parameter h goes to zero. Details on a general derivation of the discrete adjoint can be found in [58].

For completeness, we show the inversion equation which results from taking variations with respect to the inversion parameter k:

$$\int_{\Omega} \beta k - \int_{\Gamma_{\text{rxn}}} \lambda c = 0 \tag{6.31}$$

These optimality conditions are semilinear, which requires a nonlinear solver. A SQP method has been used from the NPSOL library which was accessed through the DAKOTA toolkit [33].

6.6 Numerical Results - contact tank

Wang et al. [62] developed a 2D numerical model to investigate tranport of a tracer in a contact tank and focused primarily on resolving the fluid flow with different turbulence models. The paper concludes that solute transport predictions depend on the accuracy of the hydrodynamics. A finite difference method was used to spatially discretize the Navier Stokes and the convection-diffusion-reaction. Although in Wang's study the flow is turbulent, we have reduced Reynolds number to the laminar case to allow for a more focused investigation of adaptivity and optimization for SUPG stabilized flows.

The contact tank consists of a flow domain with a single inlet and outlet. The domain has multiple turns at right angles to form a serpentine structure. We plot two computed flow fields for Re $= 100$ and 500 in Figure 6.11. The outlet channel was extended (not shown) in order to allow the fluid to return to a near fully developed flow at the outlet. The reaction zones were located where the flow would be in closest proximity to the walls, in order to increase mass transport. The chemical reactions which consume the reactant species were assumed to be first order linear.

Our goal was to investigate the ability to solve inverse problems for the reaction coefficients based on either

- a prescribed average concencration at the outlet, or
- a prescribed concentration profile on the reaction surfaces.

For the first case, we chose to fit only a single global constant reaction rate, in order that the problem be well-posed. In the second case, we allowed the reaction rate to take on three different values on segments of the reaction surface. More generally, we hope to use these results to drive work on large scale optimization where the reaction rate could be optimized as a discrete function that varies at all nodal locations on the reaction surface.

Our optimization procedure was multilevel, with optimization driven by DAKOTA at the outer loop. For each step of the SQP method, the value and gradients of the objective function were computed by the application code Aria. The gradients were computed using the adjoint approach.

In order to insure accuracy of the simulation, an adaptive mesh refinement loop was used by Aria. This could be performed for either uniformly or adaptively. In the latter case, the adjoint based error estimator was used to drive the adaptivity. Elements with high error contributions (in absolute value) were subdivided into child elements. The solution was recomputed and the procedure repeated for a fixed number of iterations.

While this approach to combining adaptive error control and optimization may not be the most efficient approach, our results below will demonstrate that is it preferrable to performing optimization on uniform fine grids. Essentially, the adaptive approach can produce accurate results for the optimization problem at a fraction of the cost of using uniform fine grids. In the near future, we hope to implement a more tightly coupled version of optimization with adaptive error control.

6.6.1 Optimization of a Single Reaction Parameter for an Outlet Response

Our first application will be to solve the inverse problem of choosing a single reaction parameter so that the average concentration at the outlet is a specified value. This problem is always mathematically solvable as long as the value to fit lies in the interval $(0, 1)$. We chose the value to fit to be 0.5.

Using a very fine uniform mesh with 366,592 elements, we calculate the optimal reaction rate to be 2.21861e-05. In Figure 6.12 we plot the forward solution (concentration) as well as the adjoint solution computed on this fine grid for $\mathrm{Re} = \mathrm{Pe} = 100$. The structures of the forward solution and the adjoint solution (near the outlet) are similar to those found in the 2D straight channel prototype problem shown previously in Figure **??**.

Next we compare the optimization with adaptive grids to compare the efficiency and accuracy tradeoffs. In Table 6.1 we plot the value of the optimal parameter value for both uniform and adaptive meshes. The uniform approach requires about 366K elements to get an error of about one percent. In contrast, the adaptive approach achieves this level of accuracy with only about 75K elements.

Two of the adaptive grids used for refinement levels three and six are shown in Figure 6.13.

	Uniform Refinement		Adaptive Refinement	
Refine Level	Elems	Optimal k	Elems	Optimal k
1	883	1.9993e-05	1431	1.8586e-05
2	3217	2.3077e-05	4000	1.9248e-05
3	12161	2.2455e-05	8008	2.0684e-05
4	47233	2.2237e-05	16717	2.1433e-05
5	366592	2.2186e-05	35374	2.1828e-05
6			74857	2.1992e-05

Table 6.1. Optimization results for the contact tank using a single parameter

6.6.2 Optimization of a Multiple Reaction Parameters to Fit a Prescribed Concentration on the Reaction Surfaces

Our second application will be to solve the inverse problem of choosing a multiple reaction parameters in order to fit a prescribed concentration on the reaction surfaces. The function that we fit is a linear function of x that decreases along the overall flow direction

$$c_{\text{rxn}}(x) \equiv 1 - x/4. \tag{6.32}$$

Since the length of the domain in the x-direction is two, this should result in a concentration profile from one to one half in the x-direction. The response function in this case is defined by

$$J(c) \equiv \frac{1}{2} \int_{\Gamma_{\text{rxn}}} |c - c_{\text{rxn}}|^2 \, dx. \tag{6.33}$$

The solutions to the forward and adjoint problem are shown in Figure 6.14 for $\text{Re} = \text{Pe} = 100$. The forward solution has a similar general profile as in Figure 6.12, but now takes on a minimum value near the outlet closer to one half. The adjoint is very different, exhibiting plumes that flow off the reaction surface in the upwind direction. Since the objective function is nonlinear (quadratic), the adjoint is now a function of the forward solution.

As before, we compare the optimal parameter values computed either by using uniform meshes or adaptive meshes obtained from the adjoint based error estimator. In Table 6.2 we present the optimal values computed using either uniform meshes or adaptive meshes. As in Table 6.1, the adaptive case is again more accurate, this time by about a factor of three in terms of number of elements.

Again we plot in Figure 6.15 the adapted mesh used for the optimal values in Table 6.2. Again, adaptivity is concentrated along the reaction surfaces. However, there is also increased adaptivity in

	Uniform Refinement			
Refine Level	Elems	Optimal k_1	Optimal k_2	Optimal k_3
1	883	4.9566e-06	7.9407e-06	1.0656e-05
2	3217	4.7849e-06	8.0191e-06	1.1157e-05
3	12161	4.6785e-06	7.8340e-06	1.0907e-05
4	47233	4.6506e-06	7.7810e-06	1.0829e-05
	Adaptive Refinement			
Refine Level	Elems	Optimal k_1	Optimal k_2	Optimal k_3
1	2173	4.9121e-06	7.7475e-06	1.0611e-05
2	4459	4.8848e-06	7.9435e-06	1.0813e-05
3	7879	4.6882e-06	7.7049e-06	1.0671e-05
4	17185	4.6630e-06	7.7061e-06	1.0687e-05

Table 6.2. Optimization results for the contact tank using multiple parameters

regions where the adjoint solution "plumes" are located. Also, because the adjoint is approximately zero near the outlet, no adaptivity occurs there.

6.6.3 Optimization for High Peclet Numbers

Finally, we present some optimization results for high Peclet numbers. Throughout this section, we fix Re $= 100$. We first verified that for the moderate value of Pe $= 100$, we obtain consistent results with and without SUPG stabilization. This was done by again fitting the average concentration at the outlet to the value of 0.5.

Next we increased the Peclet number to be closer to the actual value for a contact tank containing water as the fluid. The outlet value used in the optimization was also increased with Pe in order to make the problem more feasible. The results are summarized in Table 6.3.

6.7 Conclusions

In this chapter we presented a macroscale model and for the a contact tank water treatment application with microscale surface reactions represented using macroscale chemistry. We verified the forward and adjoint solutions to the transport equation, as well as the SUPG stabilization for moderate Peclet numbers. Efficient optimization of macroscopic reaction parameters with adaptive

Pe	Elems	Average Value to Fit	Optimal k
1e+2	10117	0.5	2.1655e-05
1e+3	14404	0.9	7.9533e-06
1e+4	39331	0.95	4.9624e-4

Table 6.3. Optimization results using SUPG stabilization and adaptivity

error control was demonstrated. Finally we provided initial data on extension of the approach to high Peclet numbers found in a realistic contact tank problem.

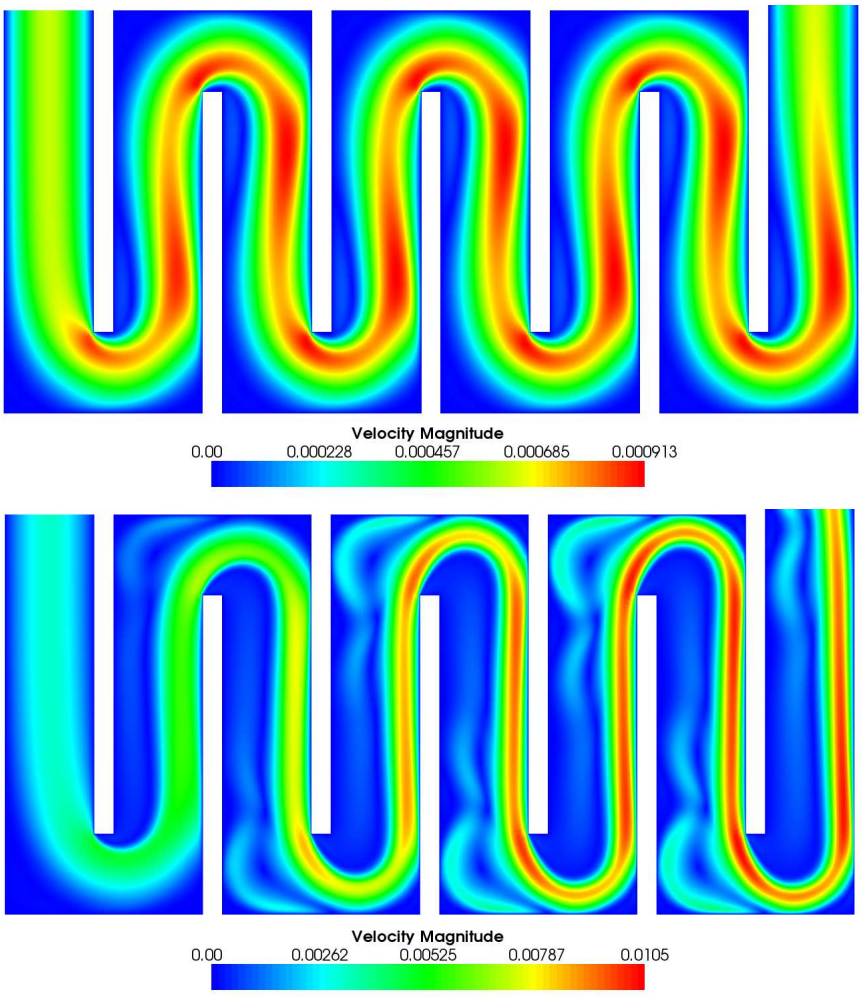


Figure 6.11. Contact tank flow fields for Re $= 100$ (top) and 500 (bottom).

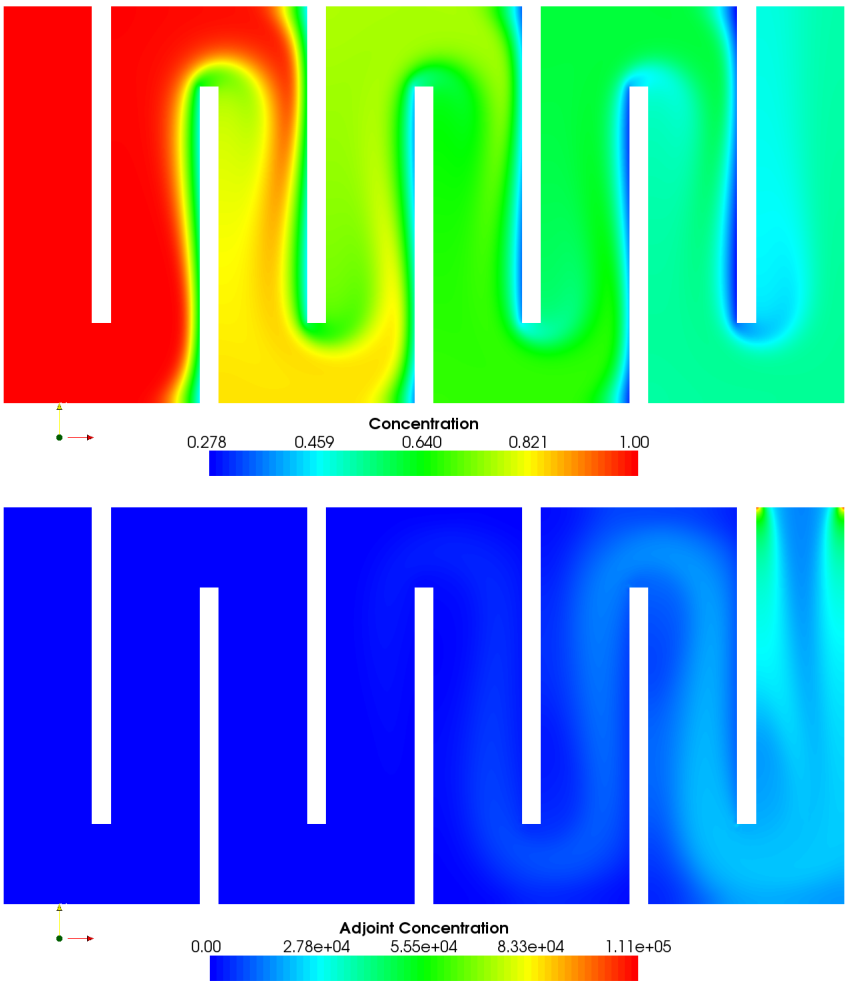


Figure 6.12. Concentration (top) and adjoint concentration (bottom) for the single parameter case.

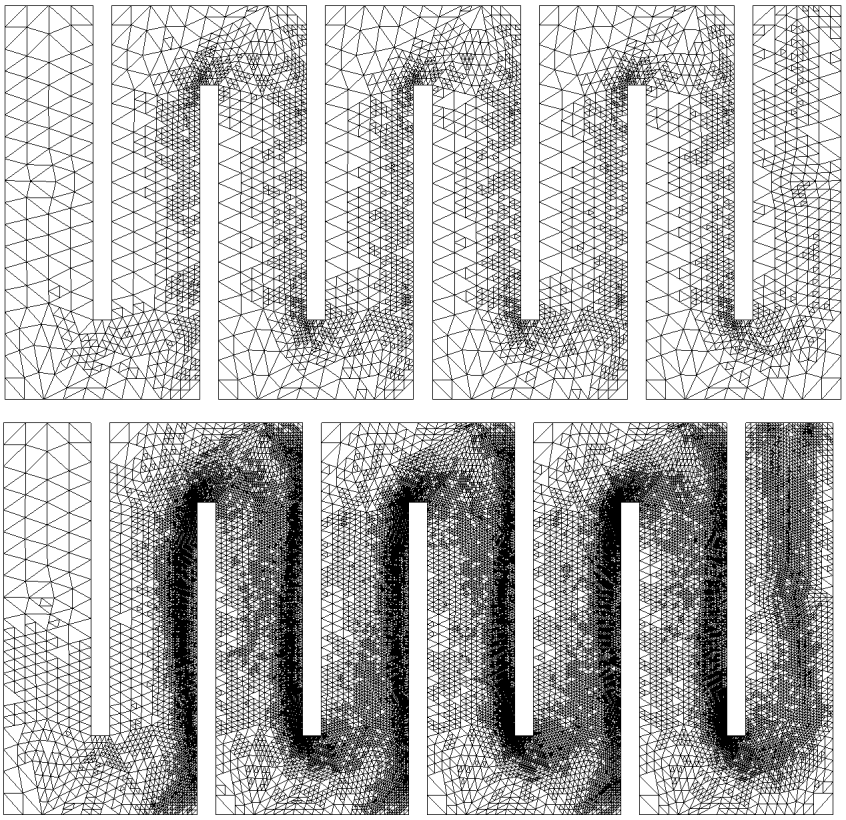


Figure 6.13. Adaptive meshes used at optimal value of reaction rate parameter for refinement level three (top) and six (bottom)

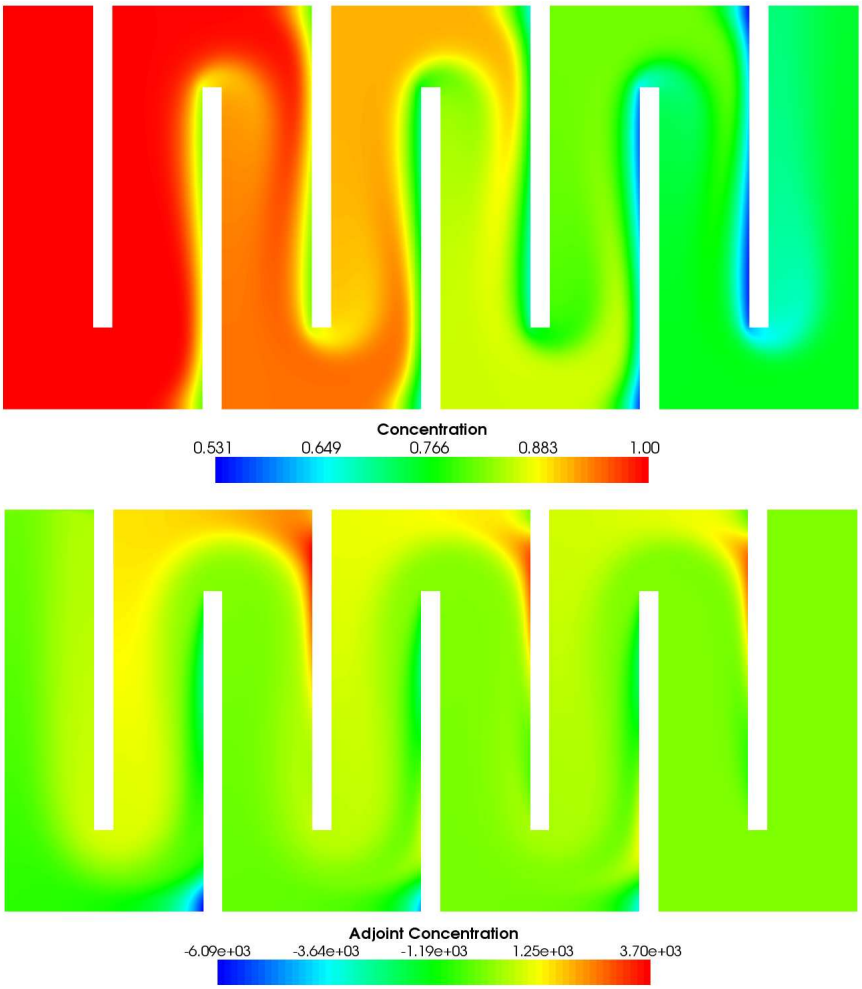


Figure 6.14. Concentration (top) and adjoint concentration (bottom)

for the multiple parameter case.

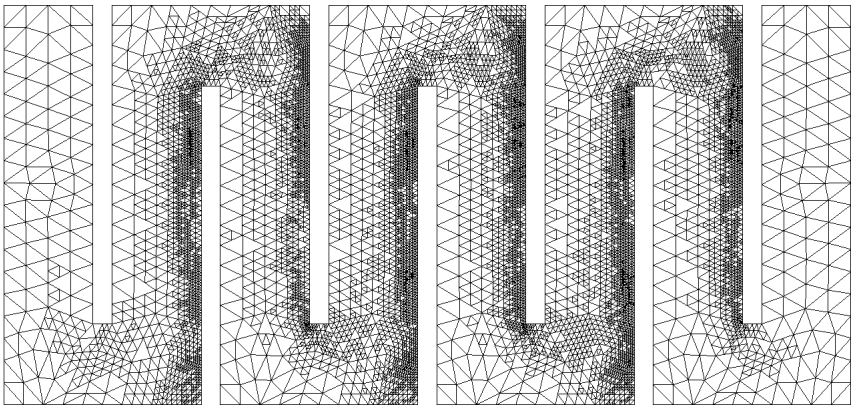


Figure 6.15. Adaptive meshes used at optimal value of reaction rate parameters for refinement level four.

Chapter 7

Integration and decontamination of *Bacillus cereus* in *Pseudomonas fluorescens* biofilms

7.1 Introduction

The Environmental Protection Agency stated that a better understanding is needed regarding which contaminants may attach to the interior of the water distribution system and how they can best be removed as a key research need . To gain insight on this topic more knowledge is needed on the integration of potential biological pathogens in biofilms. The interiors of pipes in water distribution systems are often covered with biofilms . The questions that arise include: 1) if a biological pathogen is introduced into a drinking water system will it become integrated and retained in the biofilms on the pipe-wall surfaces?, 2) if so, what fraction of an introduced pathogen is retained in the biofilm?, 3) once integrated into the biofilms, how long will the pathogens persist in the biofilms?, 4) what variables impact the pathogen integration? and 5) can the biofilms protect the pathogens from disinfection, or will traditional disinfection methods be able to remove the pathogens.

Szabo et al. found that Klebsiella pneumoniae persisted only temporarily (9 - 17 d) in both chlorinated and dechlorinated drinking-water biofilms grown on corroded iron surfaces in annular reactors. In contrast, these same authors found that Bacillus atrophaeus subsp. globigii, a surrogate for Bacillus anthracis, persisted under the same conditions for up to 70 days with target chlorine concentrations as high as 70 mg l-1 . Likewise, Langmark et al. found that 1-m hydrophilic and hydrophobic microspheres, Salmonella bacteriophages 28B and Legionella pneumophila persisted in drinking-water biofilms that were monitored for over 38 days. In this case, the drinking-water biofilms were grown on a 1-km long, 50-mm diameter pilot-scale distribution system and glass coupons were used for sampling. These varying results indicate that more studies are needed to better address the key EPA need.

This study examines the integration and retention of *Bacillus cereus* (ATCC14579) spores and polystyrene microspheres in *Pseudomonas fluorescens* (ATCC700830) biofilms grown on polyvinyl chloride (PVC) coupons in annular reactors in chlorine-free water. P. fluorescens was chosen as the biofilm organism because it has been found in drinking water biofilms , the ease of

growth and because it is Gram negative, in contrast to the Gram-positive *B. cereus* allowing for straightforward distinction between the two organisms. *B. cereus* was chosen as a surrogate for Bacillus anthracis, the causative agent for anthrax. The 2001 postal anthrax attacks brought attention to the ease of dissemination and potential lethality of B. anthracis in small doses. Carboxylate-modified microspheres were chosen because of the similarity of size (1 m diameter) and the negative surface charge at the pH of our system [Molecular Probes, MP 05001]. Use of *B. cereus* and microspheres allow for a comparison of living, reproducible particles to inert particles. Polyvinyl chloride coupons were chosen because water distribution systems contain PVC pipes, their ease of use, and demonstration that significant biofilms could grow on the pipe material .

Three variables were examined as to their effect on pathogen integration and retention: shear stress or Reynoldss number (Re) in the system during and after pathogen introduction, number of *B. cereus* spores and polystyrene spheres introduced to the system, and initial bacterial cell density in the biofilms. After examination of the above stated parameters, preliminary experiments investigating the effects of adding sodium hypochlorite with a goal of free chlorine concentrations in the reactor between 2 and 3 mg l-1 were conducted. Changes in P. fluorescens and pathogen surface density in the biofilms and reactor water were observed.

7.2 MATERIALS AND METHODS

7.2.1 Cultivation, preparation and enumeration of microorganisms

Bacillus cereus (ATCC14579) was initially grown on 2X SG sporulation agar for five days at 30C . After scraping the cells off the agar, the remaining vegetative cells were killed through repeated washing, centrifuging and suspension of centrifuged pellets in 50% ethanol. The pellets were suspended in 50% ethanol solution with overnight storage at 4C. Spores were washed three times in deionized (DI) water to remove the ethanol. To kill remaining vegetative cells, the pellets were suspended in 2 ml of 20% meglumine diatrizoate (Sigma). The 2 ml of suspension was then combined with 10-20 ml of 50% meglumine diatrizoate and centrifuged for 30 minutes at 14,500g. The presence of spores was confirmed using phase-contrast microscopy. The spores were washed four times to remove the excess chemical and stored in DI water at 4C until use. Prior to use, the spores were vortexed and enumerated on Trypticase Soy Agar (TSA) to determine the correct volume to add to the reactor. Spores were also vortexed prior to introduction to the reactor in order to minimize coagulation of the spores .

For each experiment, a new stock of *Pseudomonas fluorescens* (ATCC700830) was prepared using a CryobankTM (Copan Diagnostics Inc) bead with the P. fluorescens culture. The beads were stored in a -20 C freezer prior to use. The bead was placed in 9 ml of Trypticase Soy Broth (TSB) and incubated for 24 hours at 30 C. Two of these samples were then vortexed and centrifuged and the pellets combined and reconstituted in 9 ml of sterile DI water.

P. fluorescens surface density and the amount of *B. cereus* in the biofilms and reactor water were quantified using traditional pour plating methods. Pour plating was chosen because when spread or drop plating was used the P. fluorescens colonies spread, which interfered with enumeration. Quadruplicate plates were prepared for each dilution. Plates were incubated at 30C for approximately 24 hours for the *B. cereus* and 48 hours for the P. fluorescens prior to enumerating. Results are reported as the mean plus and minus the standard deviation of the counts from the four plates when the results for one sample are reported. Results for multiple samples are reported as the mean plus and minus one standard deviation of the counts from all the plates. Trypticase Soy Agar was used to culture both organisms. Triphenyl Tetrazolium Chloride (BD Biosciences) was added to the TSA at a final volume of 0.01% to facilitate P. fluorescens enumeration. TSA mixed with Polymyxin B (an antibiotic that inhibits Gramx negative growth) was used to plate *B. cereus*.

To enumerate cells in the biofilms the coupons were first scraped with a sterile polypropylene cell lifter (Corning 3008) into 9 ml of sterile DI water . The coupon was then rinsed with the diluent that was transferred into a sterile test tube. One ml of sterile DI water used to rinse the beaker and then water poured into the test tube. Cells were dispersed using established methods of sonication for 5 to 10 minutes and vortexing . From this suspension, two samples of 3 ml each were used for spectrophotometric analysis and 1 ml was used for plating. The detection limit for *B. cereus* in the biofilms was 1 CFU on the first dilution, converting to 5.3 CFUcm-2.

7.2.2 Experimental apparatus and setup

Experiments were conducted with annular reactors (BioSurface Technologies, Corporation, Model DFR 110) to simulate a water distribution system. Unused PVC coupons that had been disinfected in a 20% bleach solution were used for each experiment. Each reactor has 20 15 1.25 cm coupons in the inner rotating cylinder, which has a diameter of 13.9 cm. Coupons constituted approximately 25% of the surface area of the reactor. The volume of the annulus was approximately 1 L. Prior to running the experiment, the reactor was first sterilized in an autoclave at 121C and 15 pounds per square inch (psi) for 30 minutes. After sterilization, the reactor was filled with 10% bleach solution and the coupons placed in the inner cylinder. Finally, the reactor was rinsed by circulating sterile DI water through the system until free chlorine concentrations in the outflow solution were less than 0.01 mg l-1.

The correlation between Reynolds number (Re), shear stress and inner cylinder rotation speed as a function of pipe diameter were calculated based on the methods suggested by BioSurface Technologies, Corporation (B. Warwood, personal communication). Shear stress at the wall surface (w) was calculated assuming turbulent flow as follow:

$$\tau_w = \frac{f \rho v^2}{2} \tag{7.1}$$

Where f is the Fanning friction factor, ρ is the water density and v is the average water velocity through the pipe. The friction factor was calculated using the Blasius formula:

$$f = \frac{0.0791}{Re^{0.25}} \tag{7.2}$$

Finally, the RPM was calculated from the shear stress as follows:

$$RPM = (1023.37\tau_w - 3.05579)^{0.77628} \tag{7.3}$$

7.2.3 Operation of reactor

Pseudomonas fluorescens biofilms were grown under the same conditions to form reproducible biofilms. A nutrient solution was transported through the reactors for the duration of each experiment at concentrations of 5, 2.5, and 1.25 mg l-1 of glucose (as Difco Dextrose, BD Biosciences), peptone (Fisher Scientific), and yeast extract (Fisher Scientific), respectively. NaOH was also added to the reactor for the duration of the experiment from a separate container to maintain a concentration of 2.0 mgl-1 NaOH (Lab Chem, Inc.) to sustain a neutral pH. The reactor was first filled with the nutrient and NaOH solutions. Five ml of the P. fluorescens inoculum was then pipetted into the reactor along with 9 ml of sterile TSB. The inoculum was allowed to sit in the reactor for approximately four hours prior to initiating flow. The flow rate was then maintained at 1.2 mlmin-1 (residence time = 13.9 h) for 10 days. Then, to minimize the growth of planktonic bacteria in the reactor water, the flow rate was increased to 15.6 mlmin-1 (residence time = 1 h) for another seven days prior to adding the microspheres and spores. Previous experiments demonstrated that this was sufficient time to reach a steady-state bacterial cell density in biofilms on the order of 105 to 107 CFUcm-2 . The observed range of P. fluorescens colony counts was similar to those of biofilms grown with drinking water . The rotation speed of the inner cylinder was maintained at 60 revolutions per minute (RPM) during the P. fluorescens biofilm growth phase.

Just prior to surrogate pathogen introduction, two coupons were sampled from the annular reactor, the inner cylinder rotation speed was adjusted to that specified for the specific experiment, and the flow rate of the reactor was decreased to 2.6 mlmin-1 (residence time = 6 h). Micropsheres and *B. cereus* were then introduced separately and almost simultaneously into the reactor. The 6 h residence time was maintained for 24 hours after pathogen introduction to increase the initial residence time of the pathogens in the reactor. Twenty-four hours after the pathogens were introduced the flow rate was increased again so that the residence time was one hour.

A total of eight experiments were conducted with different initial amounts of introduced *B. cereus* spores and 1 -diameter TransFluoSpheres carboxylate-modified fluorescent microspheres (Molecular Probes, # T8883), and different shear stresses, as controlled by the inner cylinder rotation speeds (Table 7.1). Table 7.2presents the correlation between the inner cylinder rotation speed

Experiment Number	Average (Standard Deviation) *P. fluorescens* Surface Density (CFU cm^{-2})	Average (Standard Deviation) *B. cereus* Spores Introduced (CFU)	Microspheres Introduced (spheres)	RPM	Start of Chlorination (Days after pathogen intro)	T-Test Results	F-Test Results
50 RPM	7.5×10^4 (3.8×10^3), n=4	7.0×10^7 (2.7×10^6), n=4	9.1×10^8	50	36.0	1.8×10^{-4}	7.7×10^{-5}
100 RPM August	2.8×10^5 (2.5×10^5), n=8	9.9×10^5 (9.5×10^4), n=4	1.4×10^8	100	15.0	1.0×10^{-2}	5.0×10^{-23}
100 RPM March	1.6×10^5 (6.4×10^4), n=8	1.0×10^5 (2.0×10^4), n=4	4.2×10^9	100	14.0	N/A	N/A
150 RPM May	7.9×10^5 (4.5×10^5), n=8	4.4×10^7 (8.1×10^6), n=4	1.2×10^9	150	14.1	1.5×10^{-4}	2.2×10^{-8}
150 RPM July*	8.3×10^5 (9.9×10^4), n=8	2.6×10^8 (9.3×10^6), n=4	9.5×10^7	150	3.0	N/A	N/A
150 RPM August	2.5×10^5 (4.9×10^4), n=8	8.2×10^8 (8.6×10^7), n=4	5.0×10^8	150	14.0	0.19	0.14
300 RPM Jan	3.5×10^5 (2.5×10^5), n=4	7.6×10^7 (3.7×10^6), n=4	7.3×10^9	300	35.0	0.96	0.70
[illegible]	300	15.1	0.85	0.11	300 RPM May	3.5×10^5 (1.3×10^5), n=8	8.0×10^6 (6.6×10^5), n=4	1.2×10

[illegible] Experiment terminated early because of clogging at outflow line

Figure 7.1. Experiment variables and statistical test results

and shear stress, Re, average linear velocity and flow rate for different pipe diameters.

Samples of reactor water and biofilms were collected throughout the duration of the experiment. Ten ml of reactor water was collected at each sample event for analysis of P. fluorescens, *B.cereus* and microspheres as described below. Reactor water samples were collected periodically both before and after pathogen introduction. After pathogen introduction only one coupon was collected at each time interval. Each coupon was analyzed to quantify P. fluorescens, *B.cereus* and microspheres.

Approximately 14 days after the pathogens were introduced into the reactors, chlorine treatment began for four of the experiments. The goal of the chlorine treatment was to maintain free chlorine concentrations in the reactor between 2 and 3 mg l-1. This range was chosen because it is below the EPA allowable level for drinking water of 4 mg l-1 as Cl2.

Chlorine was added to samples of reactor water to determine the concentration of sodium hypochlorite that would be added to the system for the chlorine treatment. Between 5 and 10 times (depending on the experiment) prior to starting the chlorine treatment, reactor water samples were collected with a sterile pipette. Sodium hypochlorite was added to the sample until the free chlorine levels were within the desired range between 2 and 3 mg l-1. The amount of sodium hypochlorite that was added was recorded and used to estimate the desired sodium hypochlorite

RPM ↓	Shear Stress ($N\ m^{-2}$) ↓	Reynolds Number (Re)			Avg. Linear Vel. ($m\ s^{-1}$)			Flow (gpm)		
	Pipe → Diameter (in)	2	4	8	2	4	8	2	4	8
50	0.154	10,833	12,951	52,872	0.20	0.22	0.24	6.40	28.31	125.0
100	0.372	17,939	39,621	87,457	0.33	0.36	0.40	10.60	46.84	206.8
150	0.624	24,134	53,286	---	0.44	0.49	0.54	14.26	63.0	278.2
300	1.52	40,134	---	---	0.74	---	---	23.72	---	---

Figure 7.2. Correlation of inner cylinder rotation rate and Reynolds number, average linear velocity and flow rate.

concentration in the annular reactor.

At the start of the chlorine treatment, a third container was connected to the system containing the desired concentration of a sodium hypochlorite solution. This solution was continuously pumped through the system for 7 to 8 d. Sampling of reactor water continued and the concentration of sodium hypochlorite was adjusted to attain the desired free chlorine range in the reactor. After chlorine treatment was terminated, additional samples of reactor water and biofilms were collected and analyzed.

Unfortunately, it was difficult to maintain stable free chlorine concentrations. For this reason, our results related to the chlorine treatment should be considered preliminary. In some experiments it took approximately one day until the free chlorine concentrations reached the specified range. In one experiment, free chlorine concentrations exceeded 4 mg l-1 at one time.

One experiment was conducted to determine whether *B. cereus* alone could form a biofilm. The experiment was conducted as described in the first paragraph in this section, except the annular reactor was inoculated with *B. cereus* spores instead of P. fluorescens. Samples were collected after the increase in flow rate and for 17 more days. In this case, two coupons were collected at each sampling time.

7.2.4 Analytical methods

Fluorescence spectrophotometry (Varian Cary Eclipse Fluorescence Spectrophotometer, system ID Eclipse was used to measure the amount of microspheres in the reactor water and biofilms. Standards between 5103 and 106 spheres/ml were used to generate linear calibration curves between log concentration (spheres/ml) and log measured intensity (arbitrary units). The detection limit for the microspheres was assumed to be the concentration of the lowest standard, which is the equivalent to a surface density of 2.7 103 spherescm-2. Microsphere concentrations from the

biofilms samples collected prior to pathogen introduction were consistently measured at below the detection limit.

Periodic biofilm samples were collected for visualization using a Nikon 80i Epifluorescent microscope. Samples were examined with 10, 20, and 100 objectives. Molecular Probes LIVE BacLight Bacterial Gram Stain Kit (L-7005) was used to differentiate Gram-positive from Gram-negative organisms. SYTO9 labels both live Gram-negative and Gram-positive bacteria. In contrast, hexidium iodide preferentially labels Gram-positive bacteria. The hexidium iodide will displace the SYTO9 stain, thus Gram-negative bacteria should fluoresce at a wavelength of 500 nm (green) and the Gram-positive bacteria should fluoresce at 625 nm (red). Biofilms were stained with a mixture of 25 L of SYTO9 and 5 L of hexidium iodide mixed in 9970 L of sterile DI. 0.5 ml of the stain was pipetted on the coupon and incubated for 1 hour in the dark. The coupons were then rinsed three times with 500 L of sterile DI for each rinse and viewed immediately.

Periodic pH measurements were made of the reactor water using an Orion 520A+pH meter. At the same time the pH measurements were made, the room temperature was recorded. Free chlorine measurements were made using a HACH Pocket Chlorimeter II.

7.2.5 Data Analysis

The percent of introduced pathogens that were integrated and retained in the biofilms was calculated by multiplying the measured surface density of *B. cereus* or microspheres by the total surface area of the coupons and normalizing by the number of spores or microspheres introduced. This value represents the average percentage of the pathogen that may be captured and retained on the PVC. It does not account for the surface area of the outer cylinder of the reactor and the spaces between the coupons, which may also have biofilms on them. A T-test and an F-test was used to compare the mean and variance, respectively, of the percent of measured *B. cereus* and the microspheres in the biofilms (Microsoft Excel 2003). Results are presented as the probability that the samples from each experiment came from the same underlying population. Regression analyses were run using Kaleidagraph (version 4.03).

7.3 RESULTS

Both plating (Fig. 7.3) and microscopy (Fig. 7.4) give evidence of pathogens - microspheres and *B. cereus* becoming integrated and retained in biofilms. For the most part, the amount of the pathogens in the biofilms for each individual experiment remained relatively constant (Fig. 7.3).

For this reason, when comparisons were made between experiments, the average surface density of the pathogen in the biofilm for the samples collected after pathogen introduction and before chlorine treatment were used.

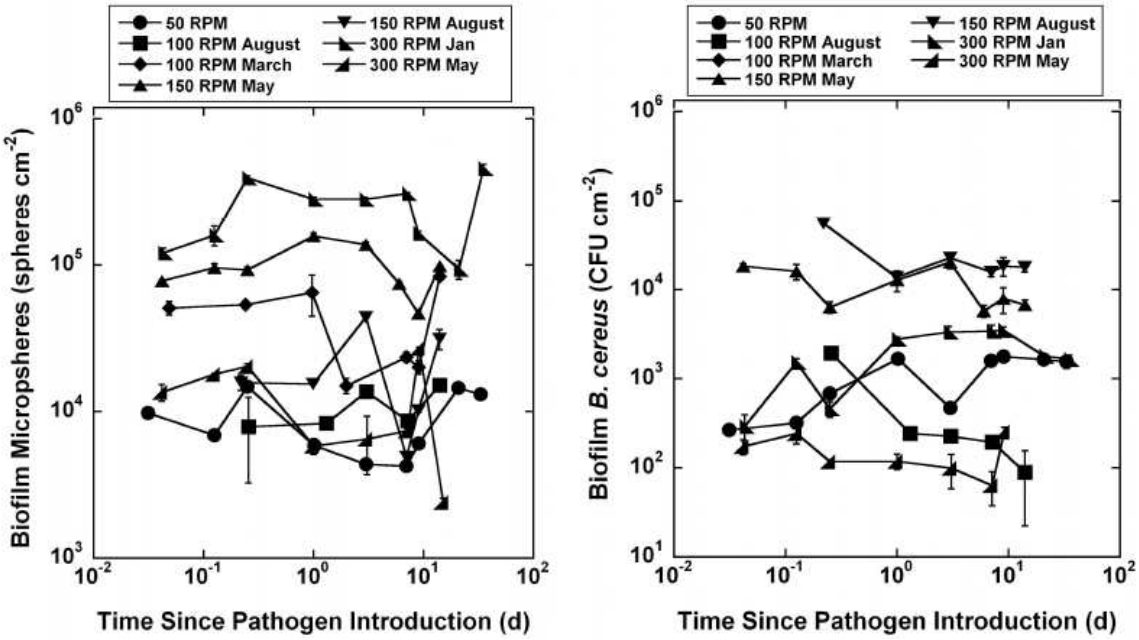


Figure 7.3. Surface densities of microsphere quantified by fluorenscence spectrophometry (left) and *B. cereus* quantified by pour plating (right) in biofilms over 14 days prior to chlorine treatment

A clear correlation between the amount of surrogate pathogen introduced to the system and the surface density of integrated pathogens in the biofilm is observed (Fig. 7.5A). As the number of spores or spheres introduced into the system increased over a range of four orders of magnitude, the surface density of the spores or spheres also increased over approximately three orders of magnitude. The R2 value of 0.73 for the regression through these data supports the positive correlation. As the regression gives a negative intercept, it is possible that when smaller numbers of pathogens are introduced into the system a linear relationship is not the best model for the correlation.

There is also an indication that, under the conditions of these experiments, a threshold amount of pathogens must be introduced to get detectable integration into the biofilm. Not included in Fig. 7.5 is the *B. cereus* surface density from the 100 RPM March experiment where 1.0 105 spores were introduced into the system. *B. cereus* was only occasionally detected in the biofilms at the detection limit with 1 CFU counted in the first dilution for a surface density of 5.3 CFUcm-2. These surface densities were only measured in one of four plates for two biofilm samples collected 0.05 and 2.0 days after pathogen introduction.

With the exception of two outliers, the percent of pathogens measured in the biofilms was 3% or less of the amount introduced (Fig. 7.5B). The absolute value of 3% should not be extrapolated

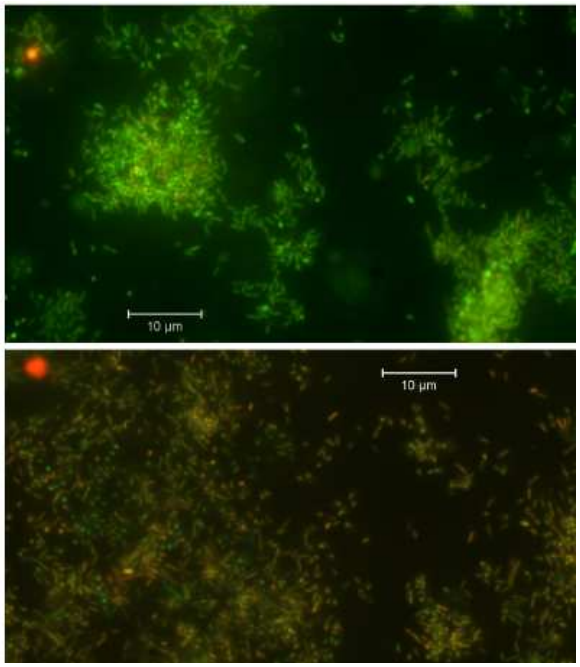


Figure 7.4. Epifluorescent images of P. fluorescens biofilms stained with LIVE BacLight Becterial Gram Stain Kit with microspheres (red) and *B. cereus* (orange) integrated within. Upper image shows clustering of biofilm organisms. Lower image shows area with concentrated *B. cereus*. Images taken with a 100objective.

to other systems. In the annular reactor, the PVC coupons accounted for approximately 25% of the available surface area on which biofilms can grow. Biofilm growth on the glass of the outer cylinder or the space between the coupons was not monitored.

The *B. cereus* colony counts from two experiments (100 RPM August and 150 RPM May) show that a larger percentage (10% and 21%) of integration is possible (Fig. 7.5B). As the amount of *B. cereus* in the biofilm does not increase significantly over the duration of the experiment (Fig. 7.3), we assume, for these two experiments, that there is insignificant net *B. cereus* growth in the biofilm. However, it is possible that there is growth and detachment at steady state with no net increase of *B. cereus*.

The growth of *B. cereus* biofilms in our annular reactors when inoculated alone gives evidence that the *B. cereus* spores can germinate to grow into biofilms. Counts ranged from 3.0105 to 5.7106 with an average of 1.9106 and a standard deviation of 1.3106 for the 16 coupons sampled and the 64 plates counted. The number of spores introduced into the reactor was approximately 4.1105.

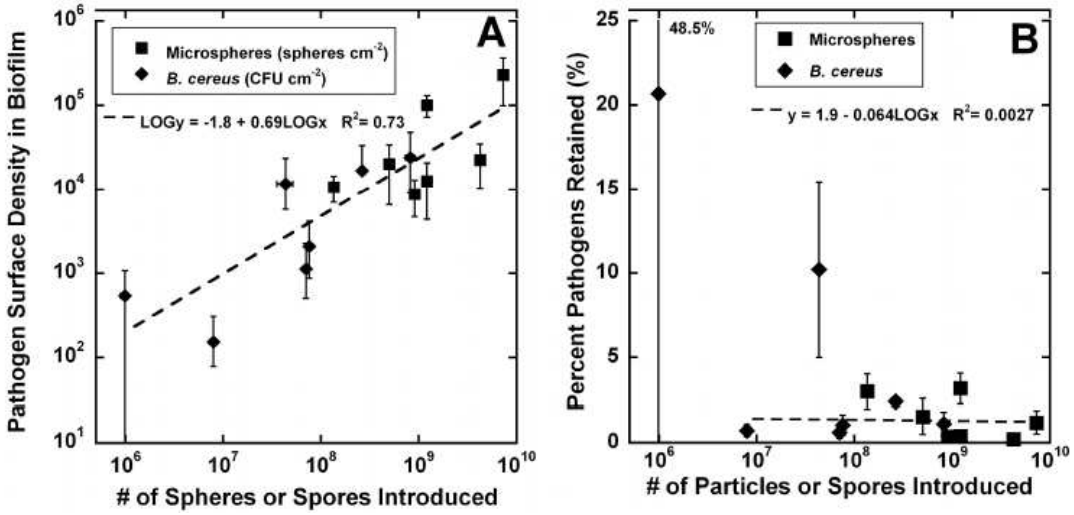


Figure 7.5. Relationship between the number of spheres or spores introduced to the system and the amount integrated and retained in the biofilms as surface density (A) and percent of the quantity introduced (B). Regression line and equation in B are for data with percent captured less than 10.

Strong trends in the amount of *B. cereus* measured in the biofilm over the 17 days that the biofilms were monitored were not observed.

There appears to be a slight correlation between the P. fluorescens colony counts just prior to pathogen introduction and the percent of pathogens measured in the biofilms (Fig. 7.6A), with greater capture associated with the higher P. fluorescens surface density. There is a positive slope for the regression through the non-outlier points and the R2 is 0.41. As our experiments were designed to grow repeatable biofilms, the density of bacterial cells in biofilms only span approximately one order of magnitude, thus whether our measured correlation can be extrapolated to a broader range of colony counts is uncertain. Also, since the shear stress was increased just prior to pathogen introduction and biofilm sloughing may have occurred, these initial P. fluorescens colony counts may not be indicative of the biofilm configuration encountered by the pathogens. While the cell numbers in our biofilms are in the same range of those found in drinking-water biofilms, we recommend running experiments in the future under higher nutrient conditions to determine whether the increasing trend we observed can be extrapolated.

A linear relationship between shear stress at the surface of the biofilm and the amount of pathogens captured and retained in the biofilms is not observed (Fig 7.6B). A regression through the non-outlier points yielded a very slow R2 value and a slope close to 0. However, a larger percentage of pathogens appear to be captured and retained for the mid-shear stress ranges of these

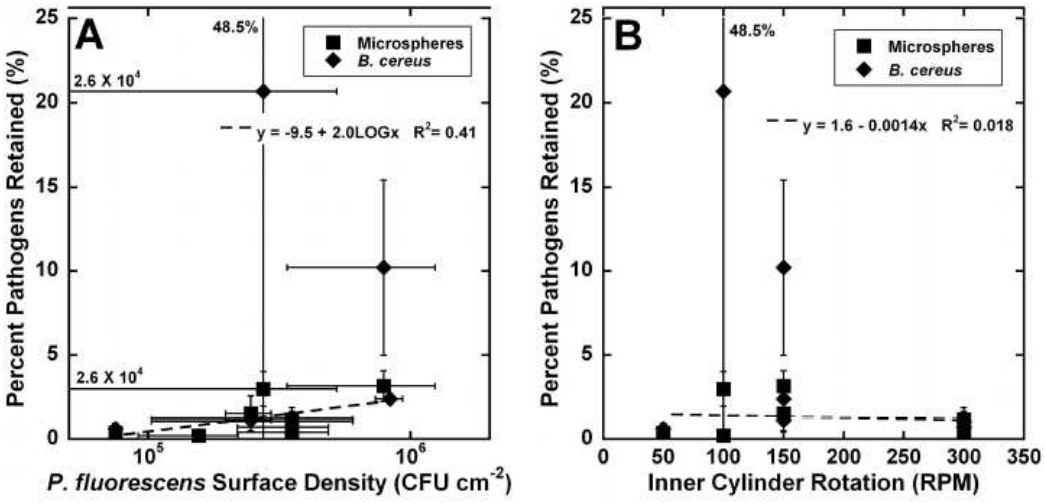


Figure 7.6. Percent of pathogens introduced to the system that is integrated and retained in the biofilms as a function of the initial P. fluorescens surface density (A) and rotation of the inner cylinder of the reactor, an indicator of Reynolds number Table 7.2) (B). Regression line and equations are for data with percent retained less than 10.

experiments.

Statistical analyses indicate that *B. cereus* may behave differently than the microspheres. In half of the six experiments that had enough data to do a statistical analysis, there is a less than 1% probability that the *B. cereus* and microsphere data came from the same underlying population (Table 7.1, T-Test Result). Thus, in these three experiments, it is likely that the *B. cereus* was either captured or behaved differently in the biofilm than the microspheres. For all three of these experiments, the percentage of *B. cereus* measured in the biofilms was greater than the percentage of microspheres. Two of these experiments are the ones with what we called outlier points for the percent of *B. cereus* capture. The third experiment, 50 RPM, had less than 1% of both *B. cereus* and microspheres captured. In this experiment, the amount of *B. cereus* measured in the biofilm increased by approximately one order of magnitude through the duration of the experiment. The probability that the variances of the *B. cereus* and microsphere populations came from the same underlying populations followed the same patterns as the T-test data (Table 7.1, F-Test Results).

As mentioned in the Materials and Methods Section, our attempt to maintain free chlorine levels between 2 and 3 mg l-1 for the duration of the chlorine treatment portion of the experiments was not entirely successful. For this reason, we present results from only two experiments, one of which had the most constant free chlorine concentrations and the other had the most data after the

termination of the chlorine treatment. The other three experiments are discussed where applicable. Because of the small number of experiments, interpretations should be considered preliminary.

For the experiments with a sufficient concentration of free chlorine (greater than 1 mg l-1), surface density of the microspheres and *B. cereus* in the biofilms decreased during the chlorine treatment portion of the experiment (e.g. Fig. 7.7A and B). The P. fluorescens colony counts also decreased. Concurrently, the concentration of microspheres and *B. cereus* in the reactor water increased immediately after the start of the chlorine treatment, presumably due to the release of the pathogens from the biofilms (Fig. 7.7C). Concentrations of the pathogens then decreased in the reactor water. After termination of the chlorine treatment, surface density of the pathogens in the biofilm appeared to level off or increase slightly (Fig. 7.7B). However, the counts of P. fluorescens increased significantly to counts higher than that before the pathogen introduction.

7.4 DISCUSSION

Our results demonstrated the integration of pathogens into existing biofilms, with approximately 3% or less of the pathogen introduced to the system becoming integrated. Results give evidence that *B. cereus* behaves differently than the microspheres and that shear stress may impact the amount of pathogens captured and retained by the biofilms. Finally, results of one experiment demonstrated that both the biofilms and pathogens within the biofilms can recover after a chlorine treatment.

7.4.1 Comparison of *B. cereus* to abiotic microspheres

An understanding is needed for why much larger amounts of *B. cereus* where observed in the biofilms for two experiments. One explanation is that the *B. cereus* spores are germinating in the reactor water and the planktonic cells are either captured differently than the spores or there are more cells to capture. There are five lines of evidence that the *B. cereus* spores germinate: 1) the observation that biofilms could be grown when only *B. cereus* spores were introduced into the reactor and 2) the fact that we could stain and visualize large clusters of *B. cereus* in the biofilms (Fig. 7.4), something we did not observe when we stained the spores alone (data not shown), 3) the observation that the stained *B. cereus* in the biofilms was more rod-shaped (Fig. 7.4) than the spherical-shaped spores observed with the phase-contrast microscopy, 4) the results of the statistical analyses showing the *B. cereus* behaves differently than the microspheres, and 4) the observation that we could not completely flush the *B. cereus* out of the reactor water, in contrast to the microspheres. We assume the germination is mostly happening in the reactor water as we do not see a significant increase in *B. cereus* colony counts in the biofilms over the time of the experiment. The exception is the 50 RPM, where the amount of *B. cereus* measured in the biofilm increased by approximately one order of magnitude through the duration of the experiment. This

observation suggests that either *B. cereus* multiplied in the biofilms or more *B. cereus* was captured allowing for an increase in the number of cells. Why the *B. cereus* surface density in the biofilms are so much higher for these two particular experiments is unclear. In comparing conditions of these two experiments versus the others, there are no obvious trends.

7.4.2 Impact of Shear Stress

If the explanation for the T-statistic result of *B. cereus* behaving differently than the microspheres is that the *B. cereus* spores are germinating and multiplying, then this study suggests that the capture of planktonic cells or germination of spores in the biofilm may be more likely with lower shear stress.

Studies have been conducted showing that the shear environment can control the amount of bacteria in a biofilm , biofilm structure , and biofilm detachment . With the exception of the findings of Peyton , it is generally thought that biofilms become thinner and denser with increasing shear force . In addition, the adhesive strength of a biofilm increases with increasing shear and also increases towards the substratum to which the biofilm is attached . However, we have not seen studies demonstrating the impact of shear environment on the integration of pathogens in a biofilm.

In this study, more pathogens are measured in the biofilm at the mid shear (100 and 150 RPM) ranges, however, not all the data are consistent. Fig. 7.3 shows that the greatest amount of *B. cereus* detected in the biofilms were both run at 150 RPM. The 150 RPM and 100 RPM August experiments had the greatest percent of captured microspheres. As the highest amounts of pathogens appears to be in the mid range of the shear environments, it is not surprising that a regression through the non-outlier points of RPM versus amount of pathogens in the biofilm yielded a very low R2 value and a slope close to 0 (Fig. 7.6B).

The change in shear stress immediately prior to adding the pathogens may have some impact on the existing biofilms. Ramasamy and Zhang postulate that when the shear stress of an annular reactor increases, sloughing may occur and then the polysaccharides in the EPS increases to re-establish the biofilms. At a later time, when the biofilm has recovered, the secretion of EPS diminished to its original level and shear stress no longer has an impact on the biofilm structure. The shear environments in our experiments bound those that Ramasamy and Zhang used 100 to 200 RPM.

We hypothesize that for shear stresses in the range of the 100 and 150 RPM experiments, there was some sloughing that led to an increase in EPS production that in turn led to greater pathogen capture. At 50 RPM and the March 100 RPM experiment the change in shear environment was not great enough to change the biofilm structure and capture is less. Finally, at the shear stress of the 300 RPM experiments, it is possible for the sloughing of the biofilm to be so large that the capture of pathogen is, for the most part, less than the 100 RPM and 150 RPM experiments. As noted

above, understanding capture of *B. cereus* is more complicated as the spores may be germinating in the reactor water or the biofilm. More studies examining the structure of the biofilms soon after the pathogen introduction are needed to confirm this hypothesis.

7.4.3 Impact of Chlorination

Even in our simple system, we show that biofilms and pathogens in biofilms can persist in a chlorinated system. The chlorine is effective in decreasing the amount of pathogens in the biofilms. However, the chlorination did not always completely eradicate the pathogens from the biofilms, or the biofilms themselves. In one case, the P. fluorescens was able to recover to colony counts higher than the original counts after termination of the chlorination treatment. At the same time, the pathogen surface density in the biofilms remained relatively constant or increased slightly, as in the case of *B. cereus* after termination of the chlorine treatment (Fig. 7.7C).

Our system had relatively simple geometry and biology compared to drinking water distribution systems. We grew single-species biofilms on PVC in an annular reactor. Yet, integration of pathogens in biofilms and recovery of the pathogens and biofilms after chlorine treatment were observed. A water-distribution system will have pipes of different materials, including concrete and iron, which can corrode and generate a physically and chemically much more complicated environment. Szabo et al. show that spores of C. atrophaeus subsp. globegii are able to persist in corroded iron coupons in annular reactors with chlorine concentrations as high as 70 mg l-1. In addition, pipe junctions and changes in flow rates can lead to stagnant regions and difference in chlorine concentration as a function of geometry. This information indicates that biofilms may act as a safe harbor for bio-pathogens in drinking water systems, making it difficult to decontaminate the systems.

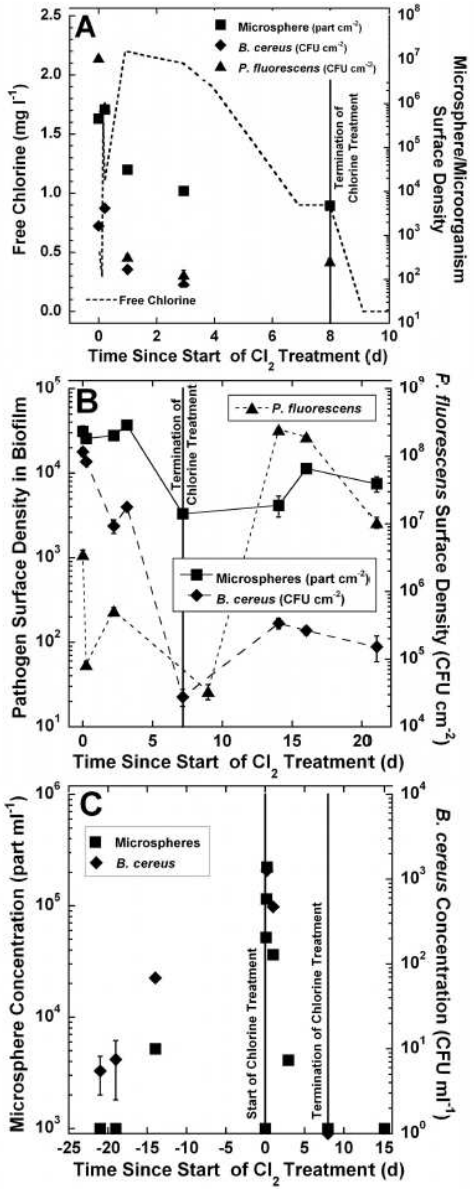


Figure 7.7. Impact of chlorine treatment on P. fluorescens and pathogen surface density in biofilms for 300 RPM January (A) and 150 RPM August (B) experiment. Impact of chlorine treatment on pathogen concentration in reactor water for 300 RPM January experiment (C). Note that concentrations or surface densities below the microsphere detection limit are plotted as 1 103 CFUml-1 or 10 spherescm-2, respectively. No detection for *B. cereus* is plotted as 1 CFUml-1.

Chapter 8

High Level Abstraction Software

8.1 Introduction

Besides forward discretization of a various models and problem formulation, Advanced simulation of realistic systems governed by Partial Differential Equations (PDEs) frequently require the discretization of operators appearing neither in the problem statement nor in a traditional solution of the system. As a simple example, a residual-based implementation of a nonlinear problem may not provide for the evaluation of a Jacobian matrix. Certain promising physics-based preconditioners for the Navier-Stokes equations require a discrete advection-diffusion equation in the pressure space. Optimization algorithms require adjoints of cost functionals. Such algorithms requiring such additional operators may be called *intrusive* in that they require the user to modify their code base before using them.

We may view Newton's method as a canonical example of an intrusive algorithm. Typically, legacy codes for nonlinear PDE provided a residual evaluation needed to solve some $F(x) = 0$ but did not provide a capability for evaluating the Jacobian matrix $F'(x)$ or its action on a vector. Obtain superior convergence offered by Newton-type solvers required significant additional code development. This is the original motivation for source-to-source automatic differentiation (AD), which allows users to gain extra mileage from legacy codes by generating new programs that evaluate gradients without modifying the original source code. These techniques are quite successful at obtaining Jacobians for Newton-based nonlinear solves, sensitivity analysis, and certain optimization algorithms. These tools, however, are inherently limited to finding operators which are derivatives of operators that are already implemented in existing code.

A complementary technique to AD receiving recent research attention is the generation of the numerical operators from a mathematical description, without directly writing traditional low-level code. Such techniques may involve actual code generation and compilation (Analysa, FFC) or interpretation of a formal grammar or embedded language (LifeV,FreeFEM). Such codes typically allow users to define variational forms at a high level of abstraction, greatly reducing development time. The capability to define basic linear problems seems to be the most mature aspect of these projects; users typically must provide their own linearization for a nonlinear problem or adjoint-based optimization, albeit at a high syntax level.

Our work in Sundance fuses the insights of these projects into a high-performance finite simulator with unified capabilities for intrusive techniques. With high-level PDE projects, we maintain that modern PDE codes should provide syntax comparable to the problem description. Like AD, we derive important operators from other operators via a process of differentiation. Going beyond this, the contribution of our work is the recognition that an extremely wide class of finite element operators required for everything from simple forward solution of linear problems to complex intrusive optimization techniques can be derived from a single abstract kind of Fréchet differentiation of variational forms. In this chapter, we present our mathematical framework for variational forms and their differentiation to obtain required operators, describe the software implementation of this technique, and present examples indicating some of the code's capabilities.

Computational science is continuously changing as exhibited by evolving hardware architectures, maturing simulation technologies and advancing analysis algorithms. One of the more notable changes is the emergence of efficient and high level tools capable of producing complete and fully functional simulation software. This new development paradigm leverages high level abstraction concepts and emphasizes a focus on complex analysis and design algorithms. Historically, simulation development consisted of implementing algorithms for individual supporting services (i.e. assembly, linear/nonlinear solver, pre/post processing, parallelism, etc) to collectively generate a complete simulation capability. This process was extremely time-consuming as each instantiation of physics required duplication of underlying modules. However, with the maturation of simulation technologies many of these supporting service have become available as robust libraries or software components. Consequently, the technical simulation community is transitioning to a process in which simulators are created in less time by leveraging existing and tested software modules. Not only is this a much more time efficient development process, but the much wider use has automatically verified and robustified these components. This process has further evolved into a top-down approach that continuously seeks to drive the user to higher levels of abstraction, thereby isolating the user from as much low level programming as possible.

The purpose of this chapter is to highlight our development of near-real time simulation capabilities which enables sophisticated and computationally efficient analysis capabilities. Our toolkit is called "Sundance". The key idea is to transform high level mathematical notation into fully functional simulators and thereby avoid the time-consuming implementation of underlying services associated with solving discretized sets of partial differential equations (PDEs). Our approach allows the replication of the finite element weak form notation in C++. Sets of PDEs are thereby easily specified and in combination with additional transitional code, transformed into fully functional simulators. This is not entirely a new concept as several activities in the technical community have attempted to achieve similar goals. However, the primary differentiating technology of our approach is an emphasis on massively parallel scalability and a focus on large scale analysis algorithms including optimization, uncertainty analysis, and reduced order modeling.

8.2 Efficient Development

One of the compelling features of Sundance is the ability to not only prototype dynamics efficiently but provide a convenient transition to production quality simulators. By virtue of eliminating significant software implementation efforts typical of PDE type development processes, the user can now focus on complex physics enhancements and on the analysis of solutions, such as optimization, uncertainty quantification, and error estimation. Instead of analysis serving a secondary role and often an afterthought, this can be a primary focus. An important distinction for the Sundance simulators is that the "products" are based on native C++ code and are not a result of a code generation process. The clear benefit is that developers have full functionality available to consider modifications and complex enhancements.

Modularity of software is an important consideration and continuously drives our toolkit development. Our goal is to eventually produce modular components at the lowest possible levels, even though the toolset in its current form already consists of three distinct packages: symbolic services, framework, and meshing (see Figure 8.1). The general concept of our design is to provide users and developers convenient interface access points through which modules can be extended, replaced or modified. The quality of these interface are directly related to the efficiency of the transition from prototyping to production. It is beyond this chapter however to describe the interface syntax but in forthcoming sections the descriptions of the toolkit design should provide insight into the current extent of software modularity and how further the ideal modularity can be accomplished within Sundance.

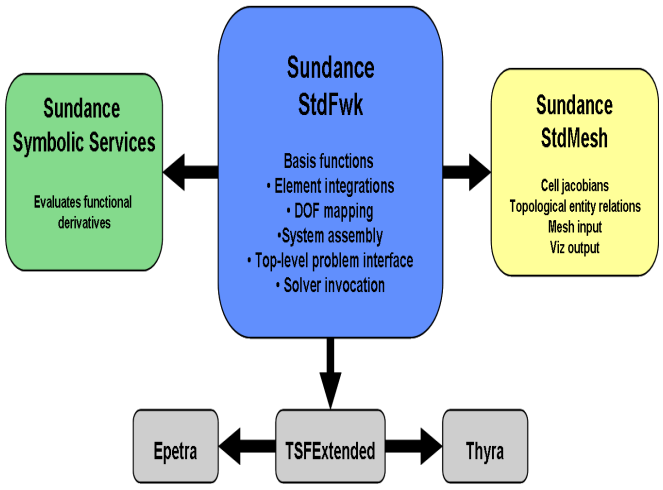


Figure 8.1. Package design

The two aspects of the Sundance architecture relevant to this project are

- **Symbolic representation and abstract evaluation.** A user specifies a PDE and boundary

conditions in weak form using a suite of high-level symbolic objects which build, internally, an expression graph representing the problem. Evaluation of this expression, or of auxiliary quantities derived from it, is done in-place (*i.e.* without symbolic transformations) using a hierarchy of `Evaluator` objects. Enhancements to this system for optimal performance with stochastic projection are described below.

- **The abstract assembly loop.** Several different types of operations must be carried out by an intrusion-enabled finite element code: Jacobian assembly, residual assembly, objective function evaluation, and sensitivity and gradient computation. Additionally, one might want to compute the action of a Jacobian without explicitly forming it. Sundance uses an abstract `AssemblyKernel` interface through which these operations can be managed by a unified assembly loop; operation-specific details are accessed via callbacks to the `AssemblyKernel` subtypes, and operation-specific data structures are maintained as data members of the `AssemblyKernel` subtype implementation.

8.3 A survey of high-level PDE codes

Implementing finite element methods is notoriously complicated, and Sundance is by no means unique in using advanced programming techniques to simplify the process.

Several projects have developed domain-specific languages for finite element methods. Analysa [11] is an early attempt at such a language. Scott and Bagheri developed a variant of Scheme that included grammatical support for variational forms and arbitrary-degree Lagrange elements. This Scheme dialect is then compiled to C++ code and executed. In similar spirit, FreeFEM [51] provides a domain-specific language by interpreting a Bison grammar. The GetDP [31] project also includes a domain-specific language The FEniCS project [32] includes FFC, a Python module that allows the definition of variational forms that are compiled to low-level code. A just-in-time compilation mode allows these forms to be run within Python. Sundance is more similar to the LifeV project, which embeds the language for variational forms in C++. While Sundance uses a C++ class hierarchy with a sophisticated evaluator to handle expressions, LifeV relies on expression templates. LifeV provides some support for automatic differentiation, but this has not been fully developed in the case of nonlinear problems or optimization. Short of domain-specific languages, many packages provide high-level support for possibly adapted finite element meshes, basis functions, and solver interfaces. We include the widely-used Deal.II library [13].

While many of these projects surpass Sundance in certain finite element functionality in the sense of more kinds of discretizations and adaptivity, Sundance tackles a larger overarching problem such as the automated support for automatic functional differentiation greatly simplifying nonlinear problems, PDE-constrained optimization, and uncertainty quantification. This capability allows Sundance to go beyond a linear forward solve to improve the capacity to efficiently solve scientific and engineering. Other projects do not prohibit this, but they do not provide the same

level of functionality.

8.4 Discretization as Functional Differentiation

Sundance is a toolkit for the development of high-performance PDE simulation codes based on a high-level description of a problem and its discretization. It was designed specifically to enable intrusive algorithms for optimization and uncertainty quantification with a minimum of developer effort beyond that needed to write a deterministic forward problem.

Sundance is based on the insight that a major step in building a finite element simulator, that of associating coefficient functions with element integration functions, can be represented as a mathematical problem. Computational solution of that problem then provides automation of that step. For example, in computing a residual, we can compute the derivative of a functional F involving v and $D_x v$ with respect to an expansion coefficient is

$$\frac{\partial F}{\partial v_i} = \int \frac{\partial \mathcal{F}}{\partial v}\, \phi_i(x)\, d\Omega + \int \frac{\partial \mathcal{F}}{\partial (D_x v)}\, D_x \phi_i(x)\, d\Omega. \tag{8.1}$$

Differentiation thus automatically associates coefficients with derivatives of basis functions. Equation 8.1 contains three distinct kinds of mathematical object, each of which plays a specific role in the structure of a simulation code.

1. $\frac{\partial F}{\partial v_i}$, which is a vector in $\mathbb{R}^N$. This discrete object is typical of the sort of information to be produced by a simulator's discretization engine for use in a solver or optimizer routine.

2. $\frac{\partial F}{\partial v}$ and $\frac{\partial F}{\partial (D_x v)}$, which are Frechet derivatives of the operator $\mathcal{F}$. The operator $\mathcal{F}$ defines the functional; it is a symbolic object, containing by itself no information about the finite-dimensional subspace on which the problem will be discretized. Numerical evaluation of $\mathcal{F}$ and its derivatives is equivalent to coefficient evaluation.

3. $D_x \phi_i$, which is a spatial derivative of a basis function.

Equation 8.1 is the bridge leading from a symbolic specification of a problem as a symbolic operator $\mathcal{F}$ to a discrete vector for use in a solver or optimizer algorithm. The central ideas behind Sundance are that (1) the discretization of many apparently disparate problem types can be represented in a unified way through functional differentiation as in equation 8.1, and (2), that this ubiquitous mathematical structure provides a path for connecting high-level symbolic problem representations to high-performance low-level discretization components.

By streamlining and partially automating the creation of a high-performance simulator, Sundance greatly simplifies the deployment of stochastic projection algorithms for UQ.

The weak form of a scalar PDE for $u \in V$ in d spatial dimensions will be, in a very abstract form, the requirement that

$$G(u, v) = \sum_r \int_{\Omega_r} F_r \left(\{D_\alpha v\}_\alpha, \{D_\beta u\}_\beta, x \right) \, d\mu_r = 0 \tag{8.2}$$

for all v in some subspace $\hat{V}$. The functions F_r are homogeneous linear functions of v and its derivatives, but can be arbitrary nonlinear functions of u and its derivatives as well as the independent variable $x \in \mathbb{R}^d$. We use the notation $D_\alpha f$ to indicate partial differentiation of f with respect to the combination of spatial variables indicated by the multi-index α. When we use a set $\{D_\alpha u\}_\alpha$ as the argument to F_r we mean that F_r may depend on any one or more members of the set of partial spatial derivatives of u. The summation is over geometric subregions Ω_r. The integrand F_r may take different functional forms on different subregions; for example it will usually have different functional forms on the boundary and on the interior. Finally, note that we may use different measures $d\mu_r$ on different subdomains; this allows, for instance, the common practice of enforcing Dirichlet boundary conditions by fixing values at nodes.

As usual we discretize u on a finite-dimensional subspace V^h and also consider only a finite-dimensional space $\hat{V}^h$ of test functions; we then expand u and v as a linear combination of basis vectors $\phi \in V^h$ and $\psi \in \hat{V}^h$,

$$u = \sum_j^N u_j \phi_j(x) \tag{8.3}$$

$$v = \sum_i^N v_i \psi_i(x). \tag{8.4}$$

The requirement that (8.2) holds for all $v \in V$ is met by ensuring that it holds for each of the basis vectors ψ_i. Because G has been defined as a homogeneous linear function in v, this condition is met if and only if

$$\frac{\partial G}{\partial v_i} = \sum_r \sum_\alpha \int_{\Omega_r} \frac{\partial F_r}{\partial (D_\alpha v)} D_\alpha \psi_i \, d\mu_r = 0, \tag{8.5}$$

where $\frac{\partial F_r}{\partial (D_\alpha v)}$ is the derivative of F with respect to the formal variable $D_\alpha v$. Differentiating with respect to a variable that is itself a derivative of a field variable is a notational device commonly used in Lagrangian mechanics and we will use it throughout this chapter. Repeating this process for $i = 1$ to N gives N (generally nonlinear) equations in the N unknowns u_j. We now linearize (8.5) with respect to u about some $u^{(0)}$ to obtain a system of linear equations for the full Newton step δu,

$$\frac{\partial G}{\partial v_i} + \left. \frac{\partial^2 G}{\partial v_i \partial u_j} \right|_{u_j^{(0)}} \delta u_j = 0. \tag{8.6}$$

In the case of a linear PDE (or one that has already been linearized with an alternative formulation, such as the Oseen approximation to the Navier-Stokes equations, the "linearization" would be done about $u^{(0)} = 0$, and δu is then the solution of the PDE.

Writing the above equation out in full, we have

$$\left[\sum_r \sum_\alpha \int_{\Omega_r} \frac{\partial F_r}{\partial(D_\alpha v)} D_\alpha \psi_i \, d\mu_r\right] + \\ + \sum_j \delta u_j \left[\sum_r \sum_\alpha \sum_\beta \int_{\Omega_r} \frac{\partial^2 F_r}{\partial(D_\alpha v)\partial(D_\beta v)} D_\alpha \psi_i D_\beta \phi_j \, d\mu_r\right] = 0. \quad (8.7)$$

The two bracketed quantities are the load vector f_i and stiffness matrix K_{ij}, respectively.

With this approach, we can compute a stiffness matrix and load vector by quadrature provided that we have computed the first and second order functional derivatives of F_r. Were we free to expand F_r algebraically, it would be simple to compute these functional derivatives symbolically, and we could then evaluate the resulting symbolic expressions on quadrature points. We have devised an algorithm and associated data structure that will let us compute these functional derivatives using a variant of AD at the symbolic level, saving us the combinatorial explosion of expanding F_r.

Finally, we note that the method above generalizes immediately to problems with multiple unknown fields. In addition sensitivity analysis seeks the derivatives of a field u with respect to a parameter p. When u is determined by (8.2), we do implicit differentiation to find

$$\sum_r \sum_\beta \int_{\Omega_r} \left[\frac{\partial F_r}{\partial D_\beta u} D_\beta (\frac{\partial u}{\partial p}) + \frac{\partial F_r}{\partial p}\right] d\mu_r = 0 \quad \forall v \in \hat{V}. \quad (8.8)$$

Differentiating by v_i to obtain discrete equations gives

$$\sum_r \sum_\alpha \left[\int_{\Omega_r} \frac{\partial^2 F_r}{\partial D_\alpha v \partial p} D_\alpha \psi_i \, d\mu_r\right] + \\ + \sum_j \frac{\partial u_j}{\partial p} \left[\sum_r \sum_\alpha \sum_\beta \int_{\Omega_r} \frac{\partial^2 F_r}{\partial D_\alpha v \partial D_\beta u} D_\alpha \psi_i D_\beta \phi_j \, d\mu_r\right] \quad (8.9)$$

Sundance has basic support for stochastic projection methods for UQ. This prototype capability is implemented through transformation of the DAG for the original deterministic problem. A more efficient implementation would be to do stochastic projection in-place without expansion of the DAG, just as is already done for differentiation in Sundance's linearization and optimization capabilities. The abstract `Evaluator` system provides a natural way to incorporate this enhancement into the existing system.

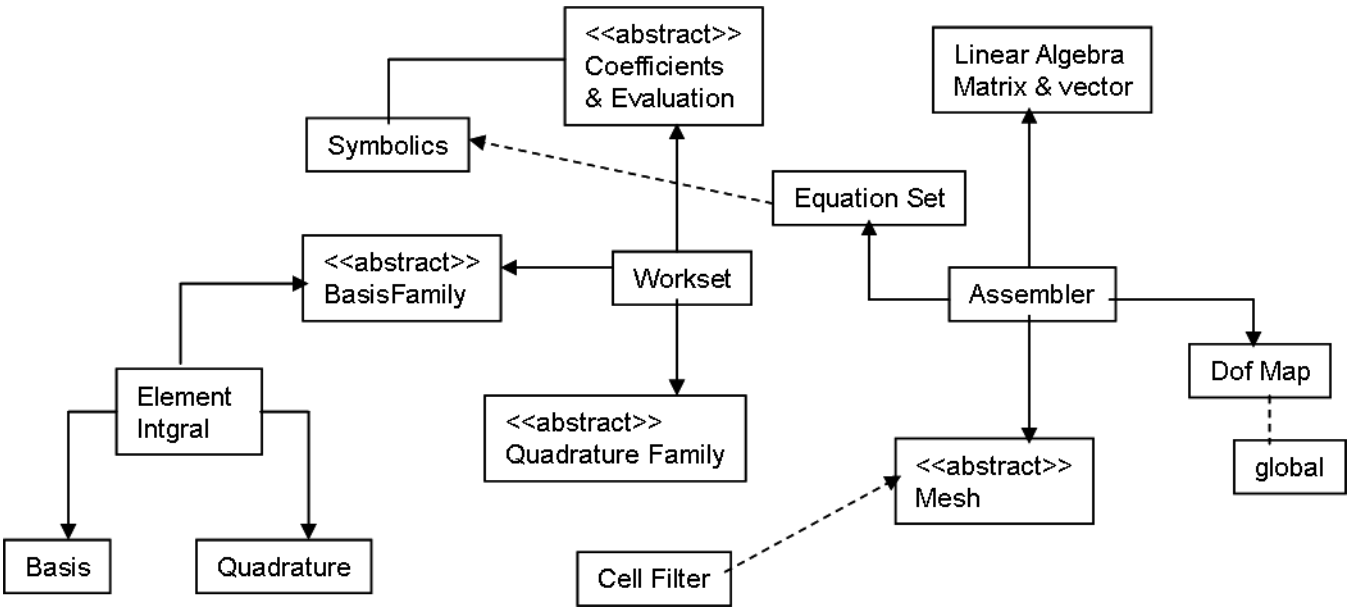


Figure 8.2. Sundance Package Overview

8.5 Modular Design

Sundance provides high level abstraction methods to transform a finite element based weak form of a set of PDE based equation into C++ code. Sundance comprises of three major packages, 1) the symbolic system, 2) the mesh systems, and the finite element module. The job of the symbolic system is to evaluate functional derivatives and provide the finite element package the appropriate integral coefficients. The finite element package is also referred to as the Standard framework (StdFwk) consist of a facility to calculate basis families, perform element integration, manage the degree of freedom mapping, perform the assembly routine, manage the top level interfaces, and provide the functionality to interact with solver routines. The meshing package (stdMesh) consists of facilities to calculate cell Jacobians, manage the mesh topology, store mesh input, and provide visualization output. Each of these packages are relatively independent from each other. Within each package, multiple services and supporting functions are implemented in a modular design fashion. The general architecture is presented in figure 8.2 and highlights the major and most important package components. This figure presents inter-relationship only in a general way and forthcoming descriptions will further elaborate on the inner workings of the design.

8.5.1 Symbolics

The purpose of the symbolic package is to represent and evaluate all functional and spatial derivatives and mixtures thereof such as the partial derivatives $\frac{\partial F_\alpha}{\partial D_\alpha v}$ and $\frac{\partial^2 F_\alpha}{\partial D_\alpha v \partial D_\beta u}$ in the following simple example:

$$\frac{\partial G}{\partial v_i} = 0 = \left[\sum_\alpha \int_{\Omega_\alpha} \frac{\partial F_\alpha}{\partial D_\alpha v} D_{\alpha \Psi_i} \right] + \sum_j \Delta u_j \left[\sum_\alpha \sum_\beta \int_{\Omega_\alpha} \frac{\partial^2 F_\alpha}{\partial D_\alpha v \partial D_\beta u} D_\alpha \Psi_i D_\beta \phi_j \right] \tag{8.10}$$

Several classes within the symbolic package provide the primary functionality. The Expr class is a user-level representation of the symbolic expressions and is a handle to a reference-counted pointer to the ExprBase subtype. A range of subtypes are supported including unknown functions, test functions, derivatives, coordinate expressions, and cell diameters. It supports overloaded operators and has the necessary hooks to apply transformation rules. It should be noted that expression copied and assignments are shallow.

Mathematical objects such as weak forms, boundary conditions, or postprocessing operations are assembled from expression (`Expr`) objects, subtypes of which include test functions, unknown functions, discrete functions, products, sums, spatial derivatives, nonlinear operators, integrals, and others; the generic expression is a directed acyclic graph composed of expressions. These are referred to as "symbolic" expressions, however this is something of a misnomer, as in the context of discretization many expression types must often be annotated with non-symbolic information; a better description is "annotated symbolics expressions" or "quasi-symbolic expressions."

The EvaluatableExpr class is a ExprBase subtype that defines the interface for evaluation and preprocessing. Figure 8.3 shows an interaction diagram of the assembly routine in which the EvaluatableExpr::evaluate() function is called for each workset. Constant and vector coefficients are returned and then used in the integral evaluation.

The EquationSet class is the interface for clients of the symbolic system.

A schematic of the asbtract assembly loop is show in the UML sequence diagram in figure 8.4.

8.6 Numerical Examples

In this section a range of numerical examples are presented to demonstrate the core feature set. We start with a relatively simple and familiar example using advection-diffusion physics. The primary purpose of this example is to expose the more fundamental components common to all simulator

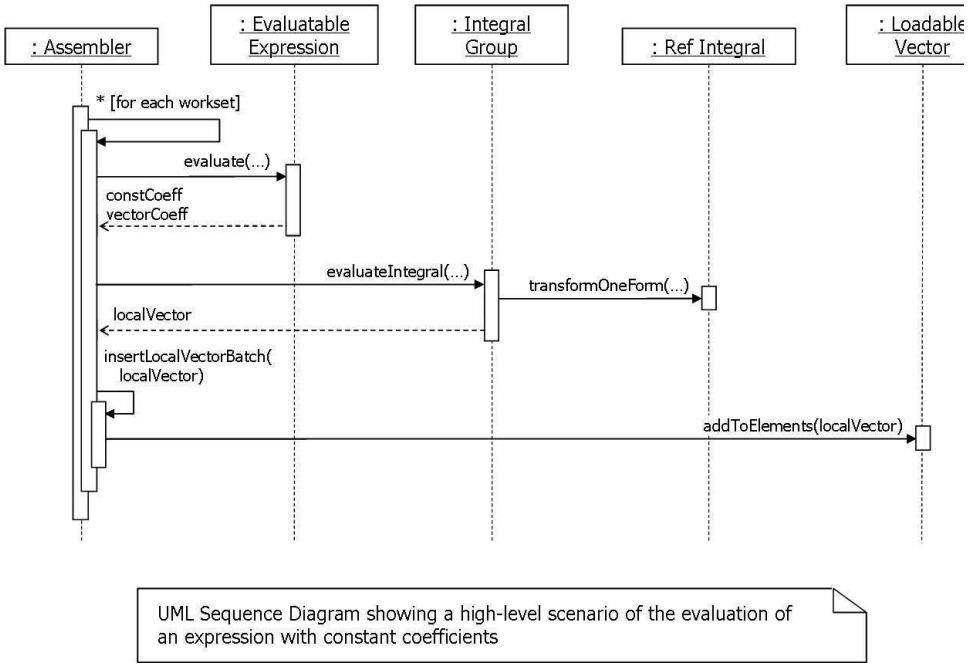


Figure 8.3. UML Sequence Diagram showing a high level scenario of an expression with constant coefficients

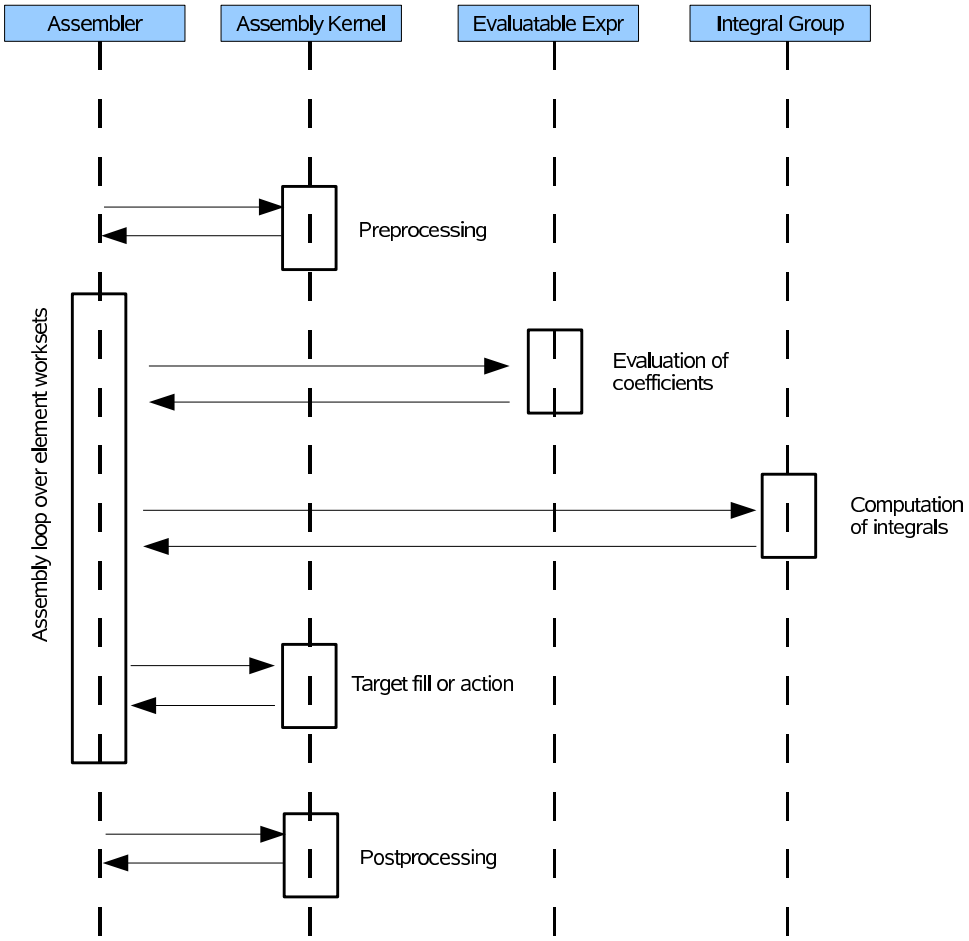


Figure 8.4. UML sequence diagram showing the structure of the Sundance assembly loop.

instantiations. Following this relatively example, more complicated features are discussed, including nonlinearities, time integration, and optimization.

We first introduce a simple example to cover the fundamental functionality of Sundance.

$$V \cdot \nabla r - k \cdot \Delta r = 0 \quad \in \Omega \tag{8.11}$$
$$r = 0 \quad on\ \Gamma_1 \tag{8.12}$$
$$r = x \quad on\ \Gamma_2 \tag{8.13}$$
$$r = y \quad on\ \Gamma_3 \tag{8.14}$$

where r represents concentration, k is the diffusivity, and V is the velocity field, which in this case is set to potential flow:

$$\Delta u = 0 \quad \in \Omega \tag{8.15}$$
$$u = \frac{1}{2}(x^2 - 1.0) \quad on\ \Gamma_1 \tag{8.16}$$
$$u = -\frac{1}{2}y^2 \quad on\ \Gamma_2 \tag{8.17}$$
$$u = \frac{1}{2}(1.0 - y^2) \quad on\ \Gamma_3 \tag{8.18}$$

In weak form the advection-diffusion is written as:

$$\int_\Omega \nabla s \cdot \nabla r + \int_\Omega s \cdot V \cdot \nabla r = 0 \quad \in \Omega \tag{8.19}$$

where $V = \nabla u$ and s is the Lagrange polynomial test function. The dynamics is defined in one line this is represented verbatim as:

```
Expr adEqn = Integral(Omega, (grad*s)*(grad*r), quad2)
  + Integral(Omega, s*V*(grad*r), quad4);
```

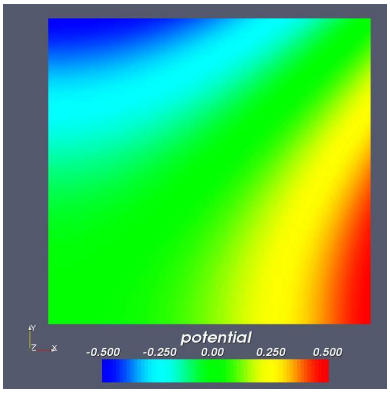


Figure 8.5. Advection Diffusion Solution

The internal mesher is used to create a finite element domain of $50 * 50$ simplicial elements in 2D. Figure 8.5 shows the final concentration solution. The complete Sundance code is included in the Appendix which includes basic boiler plate code to enable boundary conditions, meshing, test and trial function definitions, quadrature rules, interface for linear solver, and post processing.

8.6.1 Thermal-fluid coupling

As an example of a nonlinear coupling between processes, we consider the problem of Benard convection []. In this problem, a Newtonian fluid is initially stationary, but heated from the bottom. Because of thermal effects, the density of the fluid decreases with increasing temperature. At a critical value of a certain parameter, the fluid starts to overturn. Fluid flow transports heat, which in turn changes the distribution of buoyant forces.

In nondimensional form, the steady state of this system is governed by a coupling of the Navier-Stokes equations and heat transport. Let $u = (ux, uy)$ denote the velocity vector, p the fluid pressure, and T the temperate of the fluid. The parameter Ra is called the Rayleigh number and measures the ratio of energy from buoyant forces to viscous dissipation and heat condition. The parameter Pr is called the Prandtl number and measures the ratio of viscosity to heat conduction. The model uses the Boussinesq approximation, in which density differences are assumed to only alter the momentum balance through buoyant forces. The model is

$$\begin{aligned} -\Delta u + u \cdot \nabla u - \nabla p - \frac{Ra}{Pr} T \hat{\mathbf{j}} &= 0 \\ \nabla \cdot u &= 0 \\ -\frac{1}{Pr} \Delta T + u \cdot \nabla T &= 0. \end{aligned} \tag{8.20}$$

No-flow boundary conditions are assumed on the boundary of a box. The temperature is set to 1

on the bottom and 0 on the top of the box, and no-flux boundary conditions are imposed on the temperature on the sides.

We implemented this problem in Sundance by discretizing the fluid equations with standard Taylor-Hood elements (quadratic velocity and linear pressure) and the temperature equation with piecewise linear elements. The tight coupling between the fluid and thermal unknowns means that Newton's method requires a rather accurate initial guess to converge. This requires us to deploy two different solution strategies. First, we implemented a kind of nonlinear Gauss-Seidel iteration in which the fluid equations are solved with a fixed temperature field, and then the temperature equation is solved with the updated fluid velocity. Moreover, to avoid a nonlinear solve at each iteration, we lagged the advective velocity in the fluid equations, so that our iteration is to start from some initial u^0, p^0, T^0 and then compute

$$\begin{aligned}(\nabla u^{i+1}, \nabla v) + (u^i \cdot \nabla u^{i+1}, v) - (p^{i+1}, \nabla \cdot v) - \frac{Ra}{Pr}(T^i \hat{j}, v) = 0 \\ (\nabla \cdot u^{i+1}, q) = 0\end{aligned} \tag{8.21}$$

The new temperature T^{i+1} is computed using the newly found value of u^{i+1} by

$$\frac{1}{Pr}(\nabla T^{i+1}, \nabla w) + (u^{i+1} \cdot \nabla T^{i+1}) = 0 \tag{8.22}$$

After some N such steps, the values (u^N, p^N, T^N) are used as a starting guess for a full Newton method.

In order to minimize the code required, we made use of the top-level polymorphism of Sundance's `Expr` class. We wrote functions that form the weak form of the fluid and temperature equations

```
Expr flowEquation( Expr flow , Expr lagFlow
                   Expr varFlow , Expr temp ,
      Expr rayleigh, Expr inv_prandtl ,
   QuadratureFamily quad )
{
  CellFilter interior = new MaximalCellFilter();
  /* Create differential operators  */
  Expr dx = new Derivative(0);
  Expr dy = new Derivative(1);
  Expr grad = List(dx, dy);

  Expr ux = flow[0]; Expr uy = flow[1]; Expr u = List( ux , uy );
  Expr lagU = List( lagFlow[0] , lagFlow[1] );
  Expr vx = varFlow[0]; Expr vy = varFlow[1];
  Expr p = flow[2]; Expr q = varFlow[2];
```

```
  Expr temp0 = temp;
  return Integral(interior,
     (grad*vx)*(grad*ux) + (grad*vy)*(grad*uy)
  + vx*(lagU*grad)*ux + vy*(lagU*grad)*uy
  - p*(dx*vx+dy*vy) - q*(dx*ux+dy*uy)
  - temp0*rayleigh*inv_prandtl*vy,quad);
}
Expr tempEquation( Expr temp , Expr varTemp , Expr flow ,
      Expr inv_prandtl ,
   QuadratureFamily quad )
{
  CellFilter interior = new MaximalCellFilter();
  Expr dx = new Derivative(0); Expr dy = new Derivative(1);
  Expr grad = List(dx, dy);

  return Integral( interior ,
      inv_prandtl * (grad*temp)*(grad*varTemp)
   + (flow[0]*(dx*temp)+flow[1]*(dy*temp))*varTemp ,
   quad );
}
```

Then, to form the Gauss-Seidel strategy, we formed two separate equations. The first calls `flowEquation` the actual `UnknownFunction` flow variables for `flow` and the previous iterate stored in a `DiscreteFunction` for `lagFlow` and for the temperature. The second equation does the analogous thing in `tempEquation`. This allows us to form two linear problems and alternately solve them. After enough iterations, we used these same functions to form the fully coupled system by calling

```
Expr fullEqn = flowEquation( List( ux , uy , p ) ,
                             List( ux , uy , p ) ,
                             List( vx , vy , q ) ,
                             T , rayleigh, inv_prandtl , quad )
  + tempEquation( T , w , List( ux , uy , p ) , inv_prandtl , quad );
```

where `ux,uy,p,T` are the `UnknownFunction` objects.

Benard convection creates many interesting numerical problems. We have already alluded to the difficulty in finding an initial guess for a full Newton method. Moreover, early in the iterations, the solutions change very little, which can fool solvers into thinking they have converged when they haven't actually . A more robust solution strategy (which could also be implemented in Sundance) would be solving a series of time-dependent problems until a steady state has been reached. Besides difficulties in the algebraic solvers, large Rayleigh numbers can lead to large

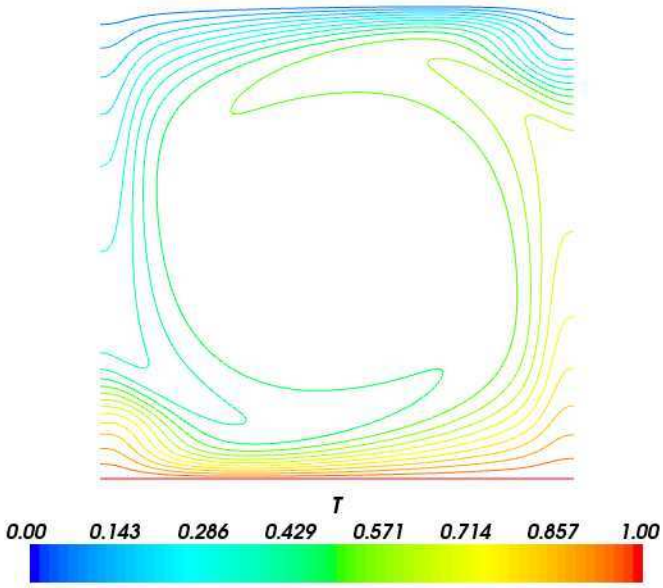


Figure 8.6. Solution of Benard convection on a 128x128 mesh subdivided into triangles with $Ra = 5 \times 10^5$ and $Pr = 1$.

fluid velocities, which imply a high effective Peclet number and need for stabilized methods in the temperature equation.

Here, we show the temperature computed for $Ra = 5 \times 10^5$ and $Pr = 1$ on a 128x128 mesh subdivided into right triangles. We performed several nonlinear Gauss-Seidel iterations before starting a full Newton solve. Figure 8.6 shows the temperature profile after the full Newton method converged.

8.6.2 Level Set Biofilm with Adjoint Based Optimization

We refer the reader to Chapters 2 and 3 for a description of the algorithms, level set strategies, and optimization approaches. In this section, we describe the general implementation details of the level set based biofilm simulator.

The internal mesh generation method is used and a xml input file to set the values of commonly used parameters, such as grid spacing, time steps, reaction coefficients, etc. Cellfilters, basis functions, quadrature rule follow the same syntax as shown in previous examples.

A special sensor location methods was implemented and the following code shows the general strategy. Either the MaximalCellFilter method executes a cellfilter method to include all internal discretization points as sensor information or a special xml file is read which contains sensor locations.

```
if (sensor_flag)
{
  cerr << "Sensors Everywhere" << endl ;
  sensors = new MaximalCellFilter() ;
  boolExprdf_ = new DiscreteFunction(discreteSpace, 1.0) ;
}
else
{
  cerr << "Sensors From Sensor File" << endl ;
  // Identify where the sensor locations are (in an xml file)
  FileInputSource fs("./sensors.xml");
  XMLObject sensorxml = fs.getObject();

  PointData pointData(sensorxml, mesh_);
  sensors = pointData.sensorLocations();

  boolExprdf_ = new DiscreteFunction(discreteSpace, 0.0) ;
  CellSet sensorCells = sensors.getCells(mesh_) ;

  Vector<double> a =
                        DiscreteFunction::discFunc(boolExprdf_)->getVector() ;
  RefCountPtr<DOFMapBase> map = DiscreteFunction::discFunc(boolExprdf_)->map();

  CellIterator iter = sensorCells.begin() ;
  Array<int> dofs(1) ;
  while (iter!=sensorCells.end())
  {
    map->getDOFsForCell(0, *iter, 0, dofs) ;
    a.setElement(dofs[0],1.0) ;

    iter++ ;
  }
  boolExprdf_->setVector(a) ;
}
```

Initial conditions are set with special methods that are essentially stack several gaussian distributions on top of each other with varying magnitude and location.

```
int ic_flag = xml.getRequiredInt("initial_condition");
```

```
  switch (ic_flag)
  {
    case(0) :
      polynomialInitialCondition() ;
      break ;
    case(1) :
      gluedPolynomialInitialCondition(2) ;
      break ;
    case(2) :
      gluedPolynomialInitialCondition(8) ;
      break ;
    case(3) :
       cerr << "GAUSSIAN IC_FLAG NOT IMPLEMENTED IN test.xml" << endl ;
      //gaussianInitialCondition() ;
      break ;
    default :
       cerr << "UNKNOWN IC_FLAG IN test.xml" << endl ;
  }
```

An example methods for one of the initial condition case is shown as follows:

```
void
DistribControlPoissonObj::polynomialInitialCondition()
{
  Expr x = new CoordExpr(0);
  Expr y = new CoordExpr(1);

  VectorType<double> vecType = new EpetraVectorType();
  DiscreteSpace discreteSpace(mesh_, new Lagrange(1), vecType);

  // Set up the Target Concentrations (i.e. the sensor measurements)
  double max_height = maxHeight_ * 1.0;
  Expr interface_fn = 0.944133+20.0787*x-388.802*pow(x,2.)-444.205*pow(x,3.)
                     +30580.6*pow(x,4.)-181036.0*pow(x,5.)+445133.*pow(x,6.)
                     -348158.*pow(x,7.)-647104.0*pow(x,8.)+1.87644e6*pow(x,9.)
                     -1.964e6*pow(x,10.)+987296.0*pow(x,11.)-198341.*pow(x,12.);

  Expr interface = -10.0*(y - 1.0*max_height*interface_fn - 1.0*max_height) ;

  L2Projector projection(discreteSpace, interface) ;
  Expr target = projection.project() ;

  Vector<double> va = DiscreteFunction::discFunc(target)->getVector() ;
  Vector<double> vb = DiscreteFunction::discFunc(rhoTarget_)->getVector() ;
```

```
  vb.acceptCopyOf(va) ;
  DiscreteFunction::discFunc(rhoTarget_)->setVector(vb) ;

  return ;
}
```

Our implementation includes a forward simulation that is used to calculated the sensor information. The equations are implemented as follows:

```
 // SUPG parameter -- Brooks and Hughes, '84
  Expr tau = sqrt(1.0/(4.0/(deltaT*deltaT) + 4.*((grad*t2)*(grad*t2))/(h_*h_))) ;

  // Crank-Nicholson time discretization
  Expr targetEqn = Integral(interior_, (trho - rhoTarget_)*trhoHat/deltaT
           +0.5*(D_/rhoB_)*((grad*trho)*(grad*tc))*trhoHat
           +0.5*(D_/rhoB_)*((grad*rhoTarget_)*(grad*cTarget_))*trhoHat

           +(trho - rhoTarget_)/deltaT
             *tau*(D_/rhoB_)*((grad*t2)*(grad*trhoHat))
           +0.5*(D_/rhoB_)*((grad*trho)*(grad*tc))
             *tau*(D_/rhoB_)*((grad*t2)*(grad*trhoHat))
           +0.5*(D_/rhoB_)*((grad*rhoTarget_)*(grad*cTarget_))
             *tau*(D_/rhoB_)*((grad*t2)*(grad*trhoHat))
           -D_*(grad*tc)*(grad*tcHat)
              -Htrho*k1_*tc/(k2_+tc) * tcHat
         , new GaussianQuadrature(2) );
```

To solve this problem we call upon a Newton based nonlinear solver and use BiCGStab linear solver. All this is wrapped in Crank-Nicholson time stepping scheme.

For the inverse problem, the continuous adjoint problem is formed:

```
  Expr atau = sqrt(1.0/(4.0/(deltaT*deltaT) + 4.*((grad*at2)*(grad*at2))/(h_*h_)))

  Expr adjoint1 = Integral(interior_, (lrho - lrho1_ )*lrhoHat/deltaT *deltaT

           +0.5*(D_/rhoB_)*((grad*lrhoHat)*(grad*c0_))*lrho*deltaT
           -0.5*k1_*dHlrho*c0_/(k2_+c0_)*lc*lrhoHat

           +0.5*(D_/rhoB_)*((grad*lrhoHat)*(grad*c1_))*lrho1_ *deltaT
           -0.5*k1_*dHlrho1*c1_/(k2_+c1_)*lc*lrhoHat
```

```
          +(lrho - lrho1_ )/deltaT
             *atau*(D_/rhoB_)*((grad*at2)*(grad*lrhoHat))
          +0.5*(D_/rhoB_)*((grad*lrho)*(grad*c0_))*lrho
             *atau*(D_/rhoB_)*((grad*at2)*(grad*lrhoHat))
          -0.5*k1*dHlrho*c0_/(k2_+c0_)*lc
             *atau*(D_/rhoB_)*((grad*at2)*(grad*lrhoHat))
          +0.5*(D_/rhoB_)*((grad*lrho1_)*(grad*c1_))
             *atau*(D_/rhoB_)*((grad*at2)*(grad*lrhoHat))
          -0.5*k1*dHlrho1_*c1_/(k2_+c1_)*lc
             *atau*(D_/rhoB_)*((grad*at2)*(grad*lrhoHat))

          -D_*(grad*lc)*(grad*lcHat)
          -Hlrho*k1_*k2_/((k2_+c0_)*(k2_+c0_))*lc*lcHat
          +(D_/rhoB_)*lrho*((grad*lcHat)*(grad*rho0_))*deltaT
                          ,q2)  ;
Expr adjoint2 = adjoint1
      + Integral(interior_, 0.5*lrhoHat*(rho1_ - rhoTarget1_ )
                *boolExpr_*deltaT, q2)
      + Integral(interior_, 0.5*lrhoHat*(rho0_ - rhoTarget0_ )
                *boolExpr_*deltaT, q2) ;
```

This equation is solved by time integrating backward and supplying the state equation at each time step. Finally, the gradient of the objective function is obtained by forming the inversion equation, setting up a “linearProblem” object, multiplying by -1 and taking the right hand side. Snippets of the code is show below:

```
/* Form inversion eqn */
Expr u      = new    UnknownFunction(new Lagrange(1));

Expr sens = Integral(interior_, -Reg_ * rho0_*beta + lrho1_*beta + beta*u, q2);
Expr sensBC;
sensProb_ = rcp(new LinearProblem(mesh_, sens, sensBC, beta, u, vecType));
.
.
.
gradF = -1.0*sensProb_->getRHS();
```

References

[1] *Markov Chain Monte Carlo In Practice*. Chapman and Hall/CRC, 2000.

[2] V. Akcelik, G. Biros, O. Ghattas, K. R. Long, and B. van Bloemen Waanders. A variational finite element method for source inversion for convective-diffusive transport. *Finite Elements in Analysis and Design*, 39(8):683–705, 2003.

[3] Volkan Akçelik, George Biros, Andrei Drăgănescu, Omar Ghattas, Judith C. Hill, and Bart G. van Bloemen Waanders. Dynamic data driven inversion for terascale simulations: real-time identification of airborne contaminants. November 2005. to appear.

[4] E. Alpkvist and I. Klapper. A multidimensional multispecies continuum model for heterogeneous biofilm development. *Bulletin of Mathematical Biology*, 69(7), 2006.

[5] E. Alpkvist, N Chr Overgaard, S. Gustafsosn, and A. Heyden. A new mathematical model for chemotactic bacterial colony growth. *Water Sci. Technol*, 49(11-12):187–92, 2004.

[6] Erik Alpkvist, Niels Chr. Overgaard, Anders Heydon, and Stefan Gustafsson. A new mathematical model for biofilm growth simulated by the level set method. In *Proceedings of the 5th International Conference on Biofilm Systems*, In press.

[7] Richard Aster, Brian Borchers, and Clifford Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2004.

[8] P. Attard. *Thermodynamics and Statistical Mechanics - Equilibrium by Entropy Maximization*. Academic Press, 2002.

[9] I. Babuska and A. Miller. The post-processing approach in the finite element method part 1: Calculation of displacements, stresses and other higher derivatives of the displacements. *Int. J. Numer. Methods Eng.*, 20:1085–1109, 1984.

[10] I. Babuska and A. Miller. The post-processing approach in the finite element method part 2: The calculation of stress intensity factors. *Int. J. Numer. Methods Eng.*, 20:1111–1129, 1984.

[11] B. Bagherri and R. Scott. About analysa. Technical report, The University of Chicago, 2003.

[12] W. Bangerth. A framework for the adaptive finite element solution of large-scale inverse problems. *SIAM J. Sci Comput.*, 30(6):2665–2989, 2008.

[13] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II — a general-purpose object-oriented finite element library. *ACM Trans. Math. Softw.*, 33(4).

[14] J.W. Barrett and C.M. Elliott. Total flux estimates for a finite-element approximation of elliptic equations. *IMA J. Numer. Anal.*, 7:129–148, 1987.

[15] Lorenz T. Biegler Carl D. Laird Bart G. van Bloemen Waanders, Roscoe A. Bartlett. Nonlinear programming strategies for source detection of municipal water networks. Philadelphia, Pennsylvania, June. EWRI.

[16] T. Barth and H. Deconinck, editors. *Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics. Lecture Notes in Computational Science and Engineering*.

[17] O. Bashir, K. Willcox, O. Ghattas, B. van Bloemen Waanders, and J. Hill. Hessian-based model reduction for large-scale systems with initial-condition inputs. *International Journal for Numerical Methods in Engineering*, 73(6):844–868, May 2007.

[18] R. Becker and R. Rannacher, editors. *Weighted a posteriori error control in finite element methods. In Block, H.G. et al. (eds.) Proceedings of ENUMATH-97*. World Scientific Publishing, 1998.

[19] R. Becker and R. Rannacher, editors. *An optimal control approach to error control and mesh adaptation.* Acta Numerica Cambridge: Cambridge University Press, 2001.

[20] J. Berry, L. Fleisher, W.E. Hart, and C. E. Phillips. Sensor placement in municipal water networks. *J. Water Resources Planning and Management*, 131(3):237–243, 2005.

[21] J. Berry, W.E. Hart, C. E. Phillips, J.G. Uber, and Watson J. Sensor placement in municipal water networks with temporal integer programming methods. *J. Water Resources Planning and Management*, 132(4):218–224, 2006.

[22] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[23] Alexander N. Brooks and Thomas J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.

[24] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2), June 2007.

[25] H-J. Bungartz, M. Kuehn, M. Mehl, M. Hauser, and S. Wuertz. Fluid flow and transport in defined biofilms: experiments and numerical simualtions on a microscale. *Water Sicend and Technology*, 41(4-5):331–338, 2000.

[26] R.D. Carr, Greenberg H.J, W.E. Hart, G. Konejevod, E Lauer, T. Morrison, and C.E. Phillips. Robust optimization of contaminant sensor placment for community water systems. *Mathematical Programming Series B.*, 107:337–356, 2005.

[27] Prodip K. Das and Subir Bhattacharjee. Electrostatic double layer force between a sphere and a planar substrate in the presence of previously deposited spherical particles. *Langmuir*, 21:4755–4764, 2005.

[28] J. Dockery and I. Klapper. Finer formation in biofilm layers. *SIAM Journal of Applied Mathematics*, 62(3):853–869, 2001.

[29] J. Dockery and I. Klapper. Finger formation in biofilm layers. *SIAM J. Appl. Math.*, 62(3):853–869, 2001.

[30] R. Duddu, S. Bordas, D. Chopp, and B. Moran. A combined extended finite element and level set method for biofilm growth. *International Journal for Numerical Methods in Engineering*, 2007. To appear.

[31] Patrick Dular and Christophe Geuzaine. Getdp: a general environment for the treatment of discrete problems. Technical report, University of Lige, 2006.

[32] T. Dupont, J. Hoffman, C. Johnson, R. C. Kirby, M. G. Larson, A. Logg, and L. R. Scott. The fenics project. Technical report, Chalmers Finite Element Center Preprint Series, 2003.

[33] M. S. Eldred, S.L. Brown, B.M. Adams, D.M. Dunlavy D. M. Gay, L.P. Swiler, A.A. Giunta, W.E. Hart, J.P. Eddy J.P. Watson, J.D. Griffin, P.D. Hough, T.G. Kolda, M.L. Martinez-Canales, and P.J. Williams. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification and sensitivity analysis: Version 4.1 users manual. Technical Report SAND2006-6337, Sandia National Laboratories, 2007.

[34] R. Evans. *Fundamentals of Inhomogeneous Fluids*. Dekker, 1992.

[35] M.B. Giles and E. Sli. Adjoints methods for pdes: A posteriori error analysis and postprocessing by duality. *In Iserles, A. (ed.) Acta Numerica Cambridge: Cambridge University*, 2002.

[36] Wences P. Gouveia and John A. Scales. Resolution of seismic waveform inversion: Bayes versus Occam. *Inverse Problems*, 13:323–349, 1997.

[37] D. Henderson, editor. *Fundamentals of Inhomogeneous Fluids*. Dekker, 1992.

[38] Charles Jackson, Mrinal K. Sen, and Paul L. Stoffa. An efficient stochastic Bayesian approach to optimal parameter and uncertainty estimation for climate model predictions. *Journal of Climate*, 17:2828–2841, 2004.

[39] C. Johnson, R. Rannacher, and M. Boman. Numerics and hydrodynamic stability toward error control in computational fluid dynamics. *SIAM J. Numer. Anal*, 32(4):1058–1079, 1995.

[40] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2005.

[41] I. Klapper. Effect of heterogenous structure in mechanically unstressed biofilm on overall growth. *Bulletin of Mathematical Biology*, 66:809–824, 2004.

[42] Carl D. Laird, Lorenz T. Biegler, Bart G. van Bloemen Waanders, and Roscoe A. Bartlett. Contamination source determination for water networks. *J. Water Resour. Plng and Mgmt*, 131(2):125–134, 2005.

[43] K. Long. Sundance. Technical report, Texas Tech University, 2008. http://www.math.ttu.edu/ klong/Sundance/html/.

[44] Alberto Malinverno. Parsimonious Bayesian Markov chain Monte Carlo inversion in a non-linear geophysical problem. *Geophysical Journal International*, 151:675–688, 2002.

[45] S.A. McKenna and D.B. Hart. On-line identification of adverse water quality event from monitoring of surrogate data: Canary software. In *Singapore International Water Week*, 2008.

[46] S.A. McKenna, M. Wilson, and K.A. Klise. Detecting changes in water quality data. *American Water Works Association Journal*, 100(1):74–85, 2008.

[47] Sean A. McKenna, Bart G. van Bloemen Waanders, Carl D. Laird, Steven G. Buchberger, Zhiwei Li, and Rob Janke. Source location inversion and the effect of stochastically varying demand. In *World Water and Environmental Resources Congress*, Anchorage, AL, May 2005. ASCE.

[48] Ali Mohammad-Djafari. Bayesian inference for inverse problems. In *Bayesian inference and Maximum Entropy Methods in Science and Engineering*, volume 21, pages 477–496, 2002.

[49] P. Monk and E. Sli. The adaptive computation of far field patterns by a posteriori error estimates of linear functionals. *SIAM J. Numer. Anal*, 36:251–274, 1998.

[50] Siva K. Nadarajah and Antony Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. volume 0667. AIAA, 2000.

[51] Antoine Le Hyaric Olivier Pironneau, Frdric Hecht. freefem++. Technical report, Universit Pierre et Marie Curie Laboratoire Jacques-Louis Lions, 2008.

[52] M. Paraschivoiu, J. Peraire, and A. Patera. A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations. *Comput. Methods Appl. Mechanics Eng.*, 150(1-4):289–312, 1997.

[53] Cristian Picioreanu, Mark C. M. van Loosdrecht, and Joseph J. Heijnen. A theoretical study on the effect of surface roughness on mass transport and transformation in biofilms. *Biotechnology and Bioengineering*, 68(4):355–369, 2000.

[54] E. Ruckenstein and D.C. Prieve. Rate of deposition of brownian particles under the influence of london and double-layer forces. *J.C.S. Faraday II*, 69:1522, 1973.

[55] J.A. Sethian. *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, Cambridge Monograph on Applied and Computational Mathematics, 1999.

[56] Ian W. Sutherland. Biofilm exopolysaccharides: a strong and sticky framework. *Microbiology*, 147(1):3–9, 2001.

[57] Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[58] B. G. van Bloemen Waanders, R. A. Bartlett, S.S. Collis, E.R. Keiter, C.C. Ober, T.M. Smith, V. Akcelik, O. Ghattas, J.C. Hill, M. Berggren, M. Heinkenschloss, and L.C. Wilcox. Sensitivity technologies for large scale simulation. Technical Report SAN2004-6574, Sandia National laboratories, 2005.

[59] D. Venditti and D. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *J. Comput. Phys.*, 164:204–227, 2000.

[60] D. Venditti and D. Darmofal. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *J. Comput. Phys.*, 176:40–69, 2002.

[61] E.J.W Verwey and J.TH. G. Overbeek. *Theory of The Stability of Lyophobic Colloids*. Dover, 1999.

[62] Hong Wang and Roger A. Falconer. Simulating disinfection processes in chlorine contact tanks using various turbulence models and high-order accurate difference schemes. *Water Research*, 32(5):1529–1543, May 1998.

[63] Jingbo Wang and Nicholas Zabaras. Hierarchical Bayesian models for inverse problems in heat conduction. *Inverse Problems*, 21:183–206, 2005.

[64] Jingbo Wang and Nicholas Zabaras. Using Bayesian statistics in the estimation of heat source in radiation. *International Journal of Heat and Mass Transfer*, 48:15–29, 2005.

Appendix

Appendix A

A.1 Capillary Tube Experiment Protocol

A.1.1 Day 1

- Assemble the capillary tube apparatus. The capillary tube apparatus contains tubing, bubble trap, and a flow break. The inlet line contains the bubble trap, and a Luer Lock fitting for the syringe. The outlet line contains a flow break. Remove the vent port from the bubble trap. The vent port is not autoclavable.
- Place size 13 MasterFlex tubing on the capillary tubes and connect to the inlet and outlet tubing Place aluminum foil on the open ends of the tubing.
- Autoclave the apparatus and capillary tubes at 121C for 15 minutes.
- Attach the vent port to the bubble trap after the apparatus is done autoclaving.
- Reconstitute GFP Pseudomonas fluorescens mut3 by placing one GFP Pseudomonas fluorescens mut3 cyrobead in 9mL TSB with $2\mu g/mL$ kanamycin. Do this with two separate vials of TSB with $25\mu g/mL$ kanamycin. Incubate overnight at 30^0C

A.1.2 Day 2

- Combine the two vials of overnight growth bacteria into one sterile conical vial. Centrifuge at 10,000 g pour off solution and add sterile DI water, repeat 2X. Then add 5 mL sterile DI water to the solution. (GFP mut 3 Pseudomonas fluorescens grows at a slower rate then normal Pseudomonas fluorescens, so only 5 mL is add to make the inoculum)
- Place the sterilized capillary tube in the capillary tube holder. Secure it in place by dripping hot wax over the ends.
- Add 60 mL of 1:100 TSB to the 60 mL sterile syringe.
- Take 0.5 mL of inoculant and add it to the syringe.
- Attach the syringe to the inlet line of the capillary tube via the Luer Lock fitting.

- Place the capillary tube holder on the confocal microscope.
- Secure the bubble trap by taping it upright to the microscope.
- Secure the flow break by taping it upright to the microscope.
- Open up the vent port on the bubble trap.
- Fill the bubble trap with 4 mL of liquid by depressing the syringe plunger manually.
- Close the vent port on the bubble trap.
- Connect syringe to the syringe pump.
- Check to make sure that the infuse rate is 0.05 mL/min and the syringe diameter is set to 26.7 mm.
- Start flow though the system. (It generally takes about 15 minutes for the liquid to fill the capillary tube.)
- Begin collecting image stacks.

  Stacks are collect by focusing on the bottom of bacteria and moving the stage until the bacteria are out of focus.

  Stacks are collected every 15 for the first day.

A.2 Biofilm Images

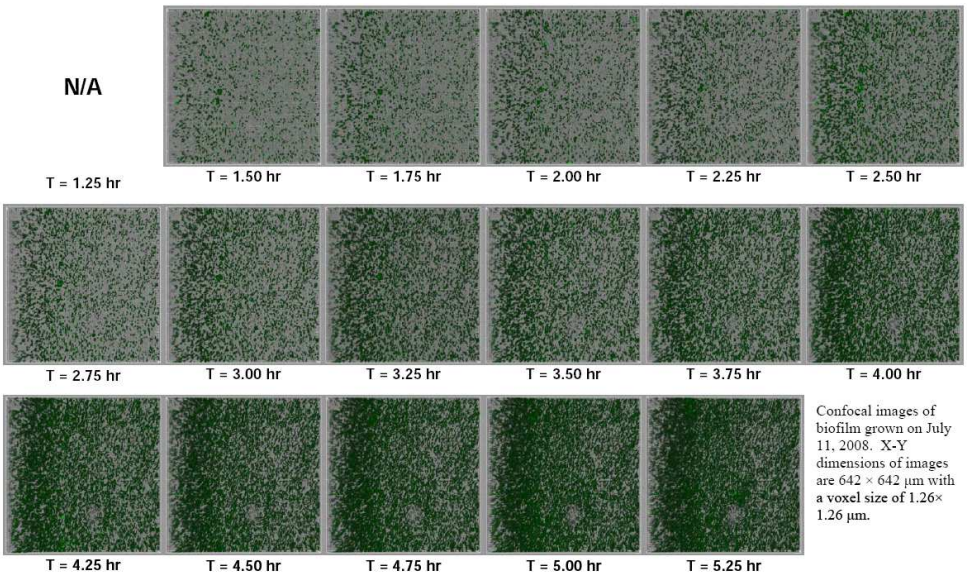


Figure A.1.

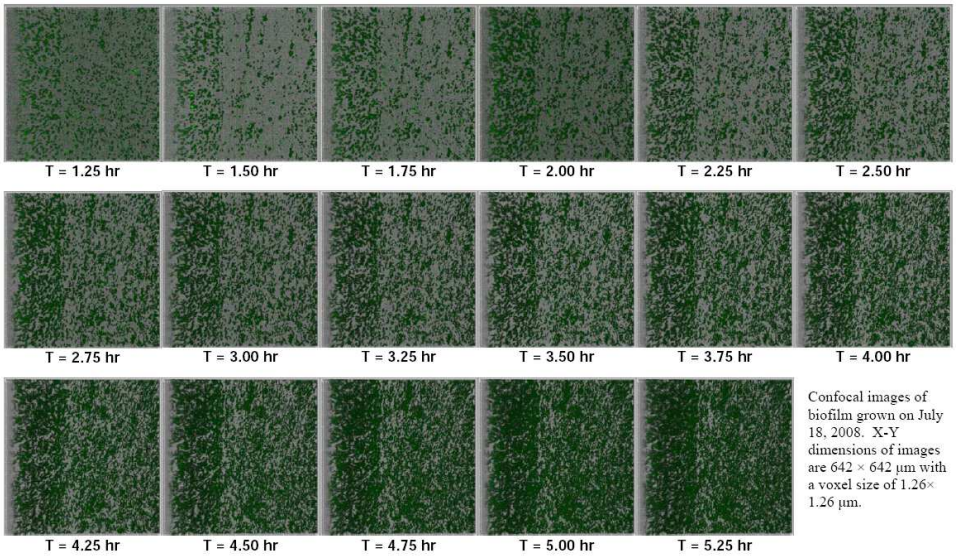
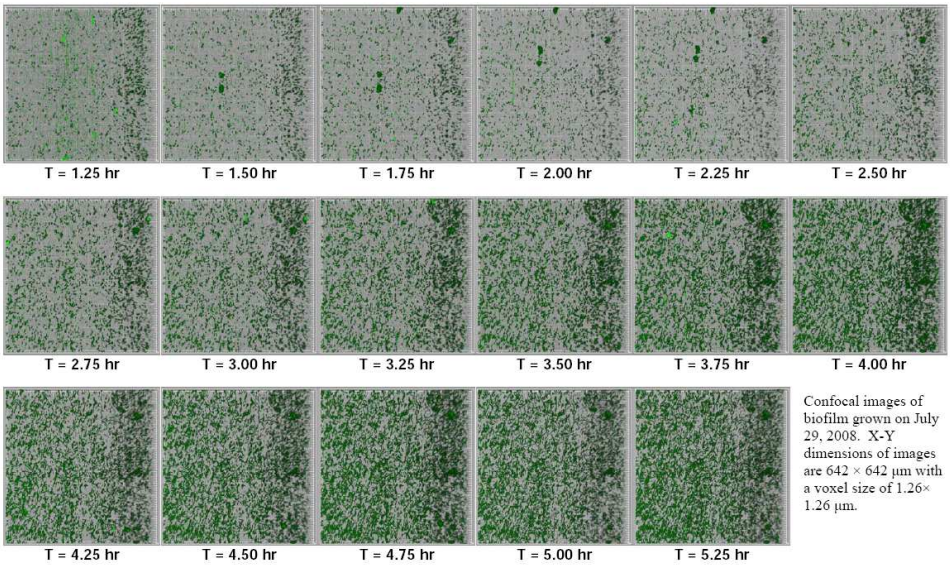
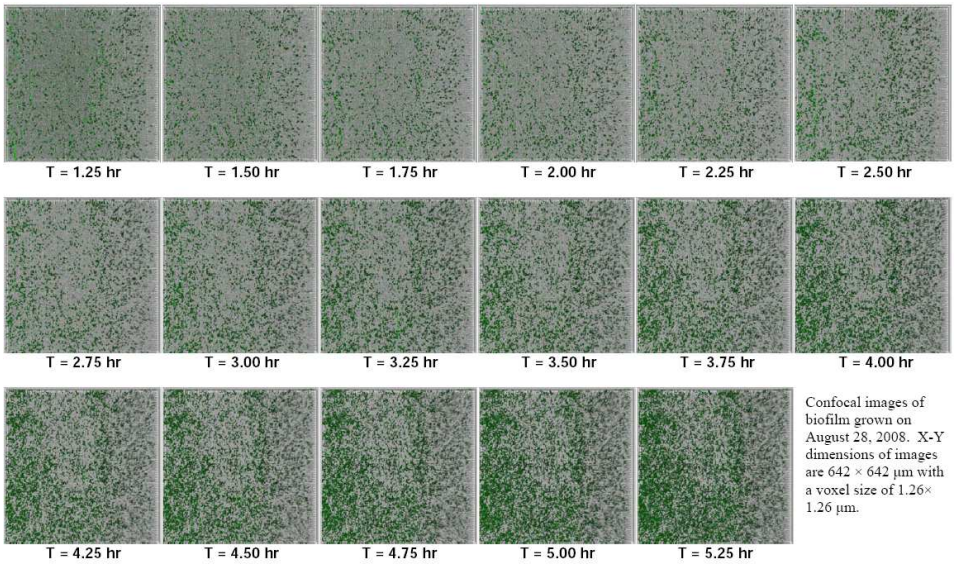


Figure A.2.



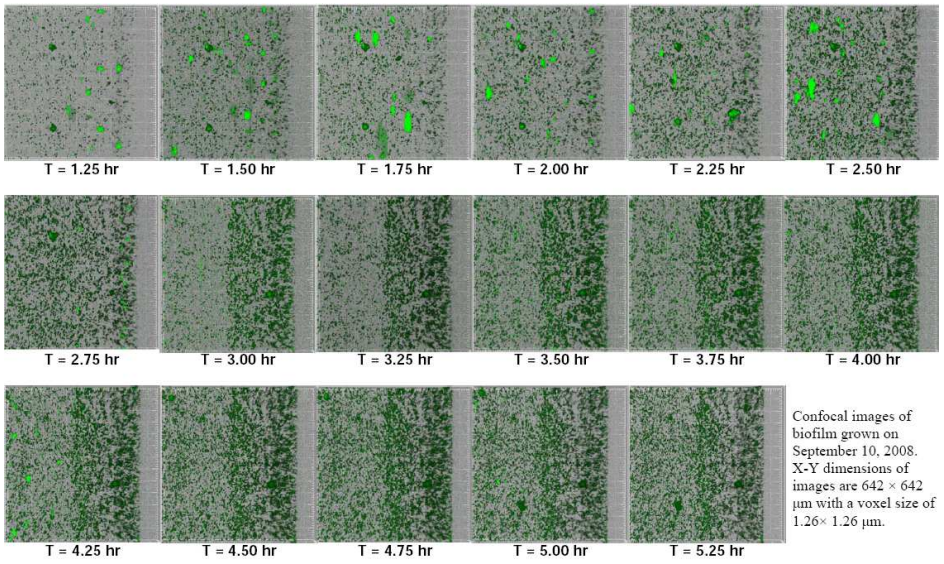
Confocal images of biofilm grown on July 29, 2008. X-Y dimensions of images are 642 × 642 μm with a voxel size of 1.26× 1.26 μm.

Figure A.3.



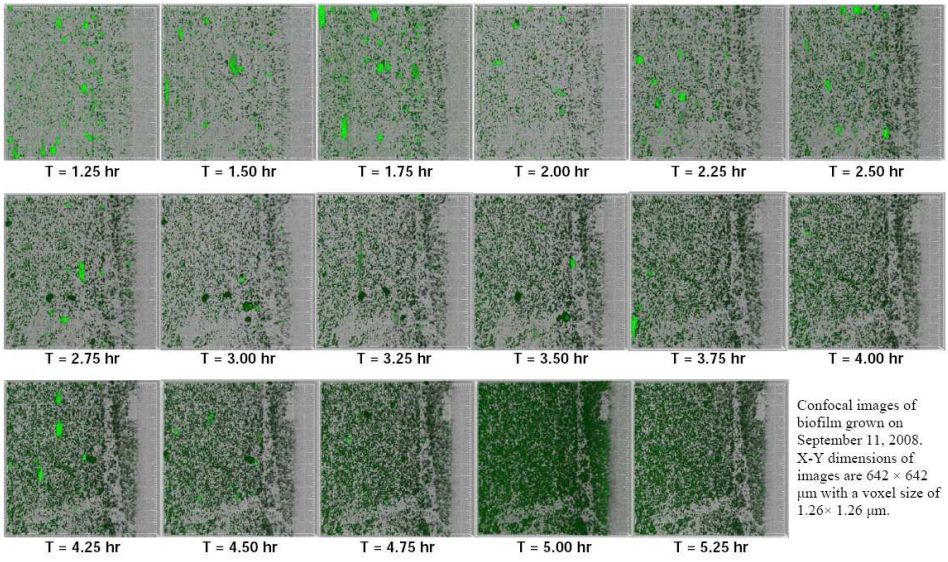
Confocal images of biofilm grown on August 28, 2008. X-Y dimensions of images are 642 × 642 μm with a voxel size of 1.26× 1.26 μm.

Figure A.4.



Confocal images of biofilm grown on September 10, 2008. X-Y dimensions of images are 642 × 642 µm with a voxel size of 1.26× 1.26 µm.

Figure A.5.



Confocal images of biofilm grown on September 11, 2008. X-Y dimensions of images are 642 × 642 µm with a voxel size of 1.26× 1.26 µm.

Figure A.6.

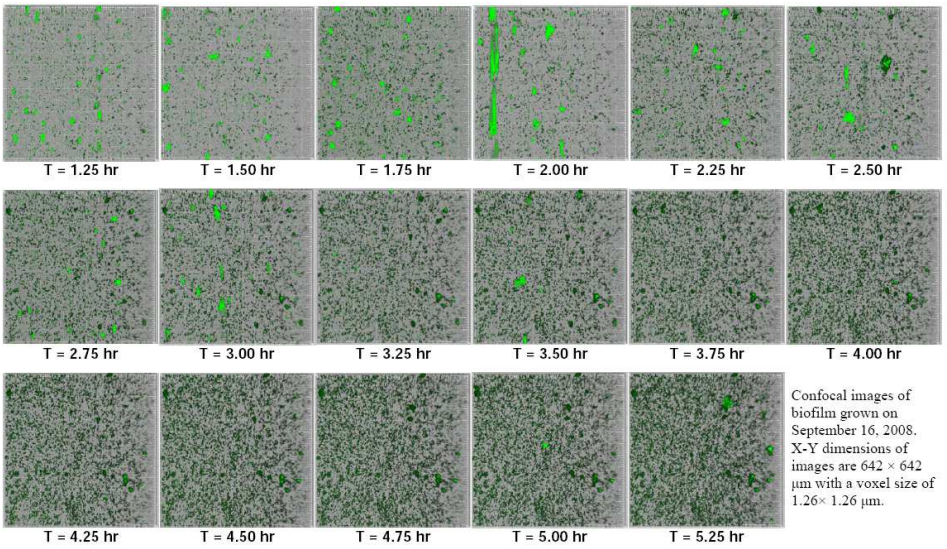


Figure A.7.

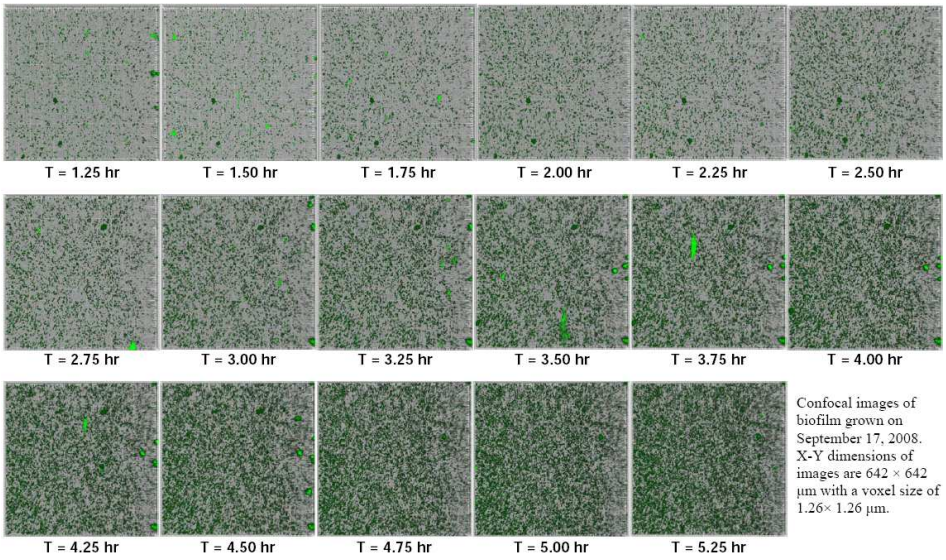


Figure A.8.

Appendix B

```
// Sundance AD.cpp for Advection-Diffusion with Potential flow

#include ‘‘Sundance.hpp’’

CELL_PREDICATE(LeftPointTest, {return fabs(x[0]) < 1.0e-10;})
CELL_PREDICATE(BottomPointTest, {return fabs(x[1]) < 1.0e-10;})
CELL_PREDICATE(RightPointTest, {return fabs(x[0]-1.0) < 1.0e-10;})
CELL_PREDICATE(TopPointTest, {return fabs(x[1]-1.0) < 1.0e-10;})

int main(int argc, char** argv)
{

  try
{
      Sundance::init(&argc, &argv);
      int np = MPIComm::world().getNProc();

      /* linear algebra using Epetra */
      VectorType<double> vecType = new EpetraVectorType();

      /* Create a mesh */
      int n = 50;
      MeshType meshType = new BasicSimplicialMeshType();
      MeshSource mesher = new PartitionedRectangleMesher(0.0, 1.0, n, np,0.0, 1.0,
      Mesh mesh = mesher.getMesh();

      /* Create a cell filter to identify maximal cells in the interior (Omega) of
      CellFilter Omega = new MaximalCellFilter();
      CellFilter edges = new DimensionalCellFilter(1);
      CellFilter left = edges.subset(new LeftPointTest());
      CellFilter right = edges.subset(new RightPointTest());
      CellFilter top = edges.subset(new TopPointTest());
      CellFilter bottom = edges.subset(new BottomPointTest());

      /* Create unknown & test functions, discretized with first-order Lagrange int
      int order = 2;
```

```
Expr u = new UnknownFunction(new Lagrange(order), "u");
Expr v = new TestFunction(new Lagrange(order), "v");

/* Create differential operator and coordinate functions */
Expr dx = new Derivative(0);
Expr dy = new Derivative(1);
Expr grad = List(dx, dy);
Expr x = new CoordExpr(0);
Expr y = new CoordExpr(1);

/* Quadrature rule for doing the integrations */
QuadratureFamily quad2 = new GaussianQuadrature(2);
QuadratureFamily quad4 = new GaussianQuadrature(4);

/* Define the weak form for the potential flow equation */
Expr flowEqn = Integral(Omega, (grad*v)*(grad*u), quad2);

/* Define the Dirichlet BC */
Expr flowBC = EssentialBC(bottom, v*(u-0.5*x*x), quad4)
  + EssentialBC(top, v*(u - 0.5*(x*x - 1.0)), quad4)
  + EssentialBC(left, v*(u + 0.5*y*y), quad4)
  + EssentialBC(right, v*(u - 0.5*(1.0-y*y)), quad4);

/* Set up the linear problem! */
LinearProblem flowProb(mesh, flowEqn, flowBC, v, u, vecType);

ParameterXMLFileReader reader(searchForFile("bicgstab.xml"));
ParameterList solverParams = reader.getParameters();
cerr << "params = " << solverParams << endl;
LinearSolver<double> solver = LinearSolverBuilder::createSolver(solverParams)

/* Solve the problem */
Expr u0 = flowProb.solve(solver);

/* Set up and solve the advection-diffusion equation for r */
Expr r = new UnknownFunction(new Lagrange(order), "u");
Expr s = new TestFunction(new Lagrange(order), "v");

Expr V = grad*u0;
Expr adEqn = Integral(Omega, (grad*s)*(grad*r), quad2)
  + Integral(Omega, s*V*(grad*r), quad4);

Expr adBC = EssentialBC(bottom, s*r, quad4)
  + EssentialBC(top, s*(r-x), quad4)
  + EssentialBC(left, s*r, quad4)
```

```
        + EssentialBC(right, s*(r-y), quad4);

      LinearProblem adProb(mesh, adEqn, adBC, s, r, vecType);
      Expr r0 = adProb.solve(solver);

      FieldWriter w = new VTKWriter("AD-2D");
      w.addMesh(mesh);
      w.addField("potential", new ExprFieldWrapper(u0[0]));
      w.addField("potential2", new ExprFieldWrapper(u0[1]));
      w.addField("concentration", new ExprFieldWrapper(r0[0]));
      w.write();

}
catch(exception& e)
Sundance::handleException(e);
Sundance::finalize();
}
```

DISTRIBUTION:

2	MS 0123	LDRD Office, 1011
1	MS 0370	Bart van Bloemen Waanders, 1414
1	MS 0899	Technical Library, 9536 (electronic)

Sandia National Laboratories