2012-04-16

Using an Experimental Mixture Design to Identify Experimental Regions with High Probability of Creating a Homogeneous Monolithic Column Capable of Flow

Charles C. Willden
Brigham Young University - Provo

Using an Experimental Mixture Design to Identify Experimental Regions with High Probability of Creating a Homogeneous Monolithic Column Capable of Flow

Charles C. Willden

A selected project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

John S. Lawson, Chair
Scott D. Grimshaw
Del T. Scott

Department of Statistics

Brigham Young University

June 2012

ABSTRACT

Using an Experimental Mixture Design to Identify Experimental Regions with High Probability of Creating a Homogeneous Monolithic Column Capable of Flow

Charles C. Willden
Department of Statistics, BYU
Master of Science

Graduate students in the Brigham Young University Chemistry Department are working to develop a filtering device that can be used to separate substances into their constituent parts. The device consists of a monomer and water mixture that is polymerized into a monolith inside of a capillary. The ideal monolith is completely solid with interconnected pores that are small enough to cause the constituent parts to pass through the capillary at different rates, effectively separating the substance. Although the end objective is to minimize pore sizes, it is necessary to first identify an experimental region where any combination of input variables will consistently yield homogeneous monoliths capable of flow. To accomplish this task, an experimental mixture design is used to model the relationship between the variables related to the creation of the monolith and the probability of creating an acceptable polymer.

The results of the mixture design suggest that, inside of the constrained experimental region, mixtures with higher proportions of monomer and surfactant, low amounts of initiator and salt, and DEGDA as the monomer have the highest probability of producing a workable monolith. Confirmatory experiments are needed before future experimentation to minimize pore sizes is performed using the refined constrained experimental region determined by the results of this analysis.

Keywords: optimal design, unreplicated split-plot, generalized linear models, column chromatography

ACKNOWLEDGMENTS

This document is a testament to the love and support of so many people around me during the last two years. I would not have been able to pursue a graduate degree without the assistance of my parents. I want to thank Dr. John Lawson and Dr. Dennis Tolley for getting me involved in this research and for providing invaluable direction and support. I would be remiss if I did not acknowledge Dr. Lawson for taking me under his wing and sparking my interest in industrial statistical methods. Also, I want to acknowledge the work of Pankaj Aggarwal from the Chemistry Department in performing all of the experiments and providing valuable insight throughout this process. Finally, I want to thank my wife, Summer, who has been long-suffering during my graduate education. I owe you an awesome vacation.

CONTENTS

CHAPTER 1

INTRODUCTION

The identification of unknown chemical compounds is an important task carried out everyday by a wide range of professionals, and can even be a matter of life or death. For example, a hazardous material crew might be able to save many lives if it is able to quickly identify a highly-toxic substance so that the victims can be given the correct treatment before they are beyond help. Many methods have been developed to identify unknown compounds, such as spectroscopy methods, X-ray crystallography, and mass spectrometry; however, these methods work best when the substance being studied consists of a single compound. For this reason, it is often helpful to separate a mixture of compounds into its constituent parts, and then perform the aforementioned identification methods on the separated compounds.

The process of separating mixtures of compounds is called chromatography, and there exists a plethora of methods to perform this task with a wide range of sophistication and required skill. Flash column chromatography, for example, uses a glass tube, called a column or capillary, filled with silica gel particles or a porous monolith. The column is attached to a pressure source and the mixture is pushed through the column. As the mixture passes through, the individual compounds travel at different rates, separating the mixture on the other end of the column (Harwood et al. 1999). The efficiency of separation can be increased by using smaller silica beads or monoliths with smaller pores, but these often require greater amounts of pressure to push the substance through the column. Within the last twenty years, ultra high performance liquid chromatography (UHPLC) has become the gold standard for commercially available chromatography techniques, but with very expensive equipment that provides the ultra high pressure. Columns used in UHPLC are often packed with silica beads less than two microns in size. A group from the Brigham Young University (BYU)

Department of Chemistry has been working to achieve the same level of performance using monolithic columns without the requirement of ultra high pressure (P. Aggarwal, personal communication, February, 28, 2012).

1.1 Proposed Column Chromatograph

The group of chemists at BYU led by Dr. Milton Lee have been working towards creating monolithic column that will ideally have a pore size and skeleton thickness of a single micron. Theoretically, such a device would be capable of achieving the same efficiency as other UHPLC techniques but with lower pressure.

The expectation is that future experimentation can optimize the polymer such that it has the smallest pore size that this process is capable of creating. This, however, is complicated by two problems: First, the process does not always create a polymer that is completely solid, which we will call a "homogeneous" monolith. Second, the pores are not always interconnected, meaning the compound to be separated will not flow through the column. In order to design an experiment to find the conditions to create a monolith with minimal pore size, we would first need to know what process settings will consistently produce homogenous polymers capable of flow.

1.2 Mixture Design

An experimental design is created to understand how the variables related to the creation of the polymer, such as mixture ingredient proportions and polymerization temperature, effect the probability of creating a homogenous polymer capable of flow. Designing experiments to study mixtures requires special considerations because of the dependent nature of proportions. Increasing the proportion of one ingredient necessarily decreases the proportion of one or more of the other ingredients. Mixture designs are a special class of designs that properly account for the dependencies of the proportions and model how a response variable is effected by the joint blending properties of the mixture components (Cornell 2002).

The design in this study was created to model how the blending properties of three mixture variables and the levels of several other process variables effect the probability of creating an acceptable monolith, meaning homogenous and capable of flow. The mixture variables consisted of water, a monomer, and a surfactant. In this study, two different monomers are included, but only one is used in a single mixture. The first monomer is diethylene glycol diacrylate (DEGDA), which is the preferred monomer of the researchers because it is compatible with biological samples, or biocompatible (P. Aggarwal, personal communication, February, 28, 2012). Ethylene glycol dimethacrylate (EGDmA) is the other monomer used in the study, and the surfactant is polyethylene oxide co-polypropylene oxide co-polyethylene oxide (EPE 4400).

As previously mentioned, the effects of several process variables are of interest. Some ingredients in the mixture are not classified as mixture components because their addition to the mixture does not change the total volume. These ingredients include calcium chloride, which is an organic salt, and potassium persulfate, which is an initiator. The non-ingredient process variables are mixing time measured in minutes, heat in degrees celsius used in the hot water bath used to polymerize the mixture, and a factor variable indicating whether the DEGDA or EGDmA is used in the mixture.

The results of this analysis are analyzed using logistic regression. This statistical technique makes it possible to model the relationship between probabilities and explanatory variables using a link function. This technique is discussed in greater detail in the Chapter 3.

Once a model for the probability of producing a homogenous monolith capable of flow is obtained using logistic regression, the model is used to draw contour plots over a triangular region called a simplex, whose coordinates denote the proportions of each mixture ingredient. These contour lines indicate a level of predicted probability of creating an acceptable polymer given the proportions of mixture components and process variable settings. Using these contour plots, regions of high probability in the simplex can be identified. These regions, if confirmed to be of high probability by a confirmatory experiment, can then be used as constraints on the experimental region for future experimentation designed to minimize pore size.

The remainder of this document contains a literature review and a thorough explanation the methods of experimental design and analysis. The literature reviewed in the following chapter is from academic journals and textbooks related to recent work in column chromatography and the use of mixture designs in a variety of circumstances. Chapter 3 explains the type of monolith chromatography columns being developed, and then discusses the mixture design, model-fitting techniques, and visualization methods used to obtain results.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews several items of literature related to recent work in developments in column chromatography and landmark statistical papers and textbook material that give the theoretical basis for the methods used in this study.

2.1 Current State of Column Chromatography

Chromatography is a family of techniques used to separate mixtures of chemical compounds. The work in this paper deals with a specific class of chromatography techniques called 'column chromatography,' in which the mixture of compounds called the mobile phase is moved through a tube occupied by a filtering structure called the stationary phase. The different compounds in the mobile phase have varying affinity for the stationary phase, which causes the mobile phase to separate into its constituent compounds as it moves through the column (Tunç et al. 2010; Aggarwal et al. 2011).

As column chromatography has evolved, two types of stationary phases have become dominant. Columns packed with tiny beads, most often silica gel particles, are called particle packed columns, and the mobile phase separates as it maneuvers through the tiny spaces between the particulates. Figure 2.1 contains an SEM micrograph of a particle packed column. The other common type of stationary phase is constructed of a single skeletal structure called a monolith. Monolithic columns are most often silica-based, or created using an organic polymer (Aggarwal et al. 2011).

Several parameters common to both types of stationary phase govern the performance of the column. Smaller pore sizes achieve higher separation efficiency, but have a negative effect on the permeability of the column. Permeability controls how quickly the mobile phase passes through the column. An ideal column would offer both high permeability and high

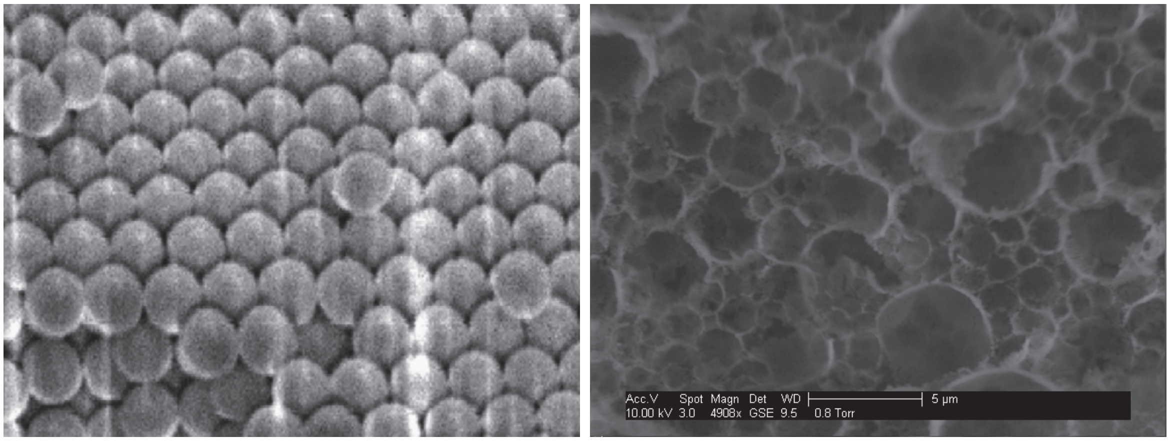
Acc.V Spot Magn Det WD 5 µm
10.00 kV 3.0 4908x GSE 9.5 0.8 Torr

Figure 2.1: SEM micrographs of a particle packed column on the left and a monolithic column on the right. The particles in the packed column are 1 μm in size, which is commonly used in UHPLC. The micrograph on the right shows a column produced in one of the experimental runs of this study. This particular emulsion has a 'foam' morphology, which is preferred for this application, but the pore sizes are too large for UHPLC.

efficiency; however, these two qualities are inversely related and often require a compromise. Morphology is a parameter that describes the form of the stationary phase, and different morphologies are ideal for different applications (Aggarwal et al. 2011).

In the 1990s, particle packed columns and monolithic columns seemed to be on even footing with each other. However, with the advent of UHPLC using particles smaller than 2 μm and ultra high pressure pumps have placed monolithic columns slightly out of favor. Since then, many have labored to advance monolithic column technology to the same performance level as UHPLC devices (P. Aggarwal, personal communication, February, 28, 2012).

These efforts to put monolithic columns back on par with particle packed columns are done with good reason. According to Tunç et al. (2010), monolithic columns offer several advantages over its counterpart, the primary advantage being a diminished trade-off between efficiency and permeability with the more open pore structures monoliths are capable of creating. Theoretically, UHPLC efficiency can be achieved using lower pressures available with much more affordable equipment. Also, particle packed columns require retaining frits to hold the particles in place. This creates inhomogeneity in the separation media, which is not a problem associated with monolithic columns. Finally, monolithic columns offer control over the morphology of the stationary phase, which can be optimized to increase flow velocity and efficiency.

However, the variables that govern morphology and pore structure in organic polymer monoliths are not very well understood. Porras et al. (2008) state that "there is no general theory for predicting the microemulsion structure in a particular system of oil, water, and surfactant. However, there are rules and phenomenological parameters that enable a microemulsion with a particular desired structure to be prepared." For clarification, microemulsions refer to pore sizes measured on the micron scale. The work here is preliminary to further experimentation designed to accurately model the relationship between those parameters and the pore structure of the monoliths.

For now, the objective is to understand what set of operating conditions and proportions of water, oil, and surfactant will consistently yield homogeneous monoliths with open pore structures. Hopefully, it is an early step towards the creation of a biocompatible UHPLC monolithic column that will be an inexpensive yet powerful tool in a wide variety of applications.

2.2 Statistical Literature Related to Design & Analysis Methods

The completion of this project requires an eclectic combination of statistical tools. This section discusses the theoretical foundation for the techniques used to plan the experiments and analyze the results. It begins with an explanation of experimental designs for mixtures; followed by a discussion of optimal designs, generalized linear models, model selection using stepwise algorithms, and diagnostic methods.

Mixture Designs

Experimentation with mixtures, such as the water-in-oil mixture used to create organic polymer monoliths in column chromatography, cannot be performed using traditional design methods. To do so would make statistical inference from the design impossible because increasing the amount of one ingredient necessarily decreases the proportion of the other ingredients to the total mixture. This means that any observed effect from increasing one ingredient is confounded with the effect of diluting the others.

Also, as Lawson (2010, p. 443) explains, "The characteristics of a product that is composed of a mixture of components is usually a function of the proportion of each component in the mixture and not the total amount present." Therefore, it makes more sense to use proportions as the explanatory variables in place of total amounts of each mixture component.

Designs for mixtures accommodate the interdependencies of the proportions by restricting the experimental region. In the case of a three factor experiment, which is the case for this study, the experimental region is generally represented graphically as a cube. If we take the same region to represent three mixture components, the three axes of the cube span from zero to one and represent the proportion of each factor in the total mixture. The vertex of the cube with coordinate (1,1,1) would hypothetically denote a mixture of 100% component 1, 100% component 2, and 100% component 3. Of course, that is impossible. To restrict the space to only include coordinates that make sense, a constraint is imposed that the coordinates must sum to one. In the three factor example, this results in an equilateral triangular region, which is illustrated graphically in Figure 2.2.

SCHEFFÉ MODEL The constraint that the proportions sum to one not only modifies the experimental region, but also the linear model that represents the relationship between the proportions of the mixture components and the response. A three-factor factorial experiment can be represented with the model $\mathbf{y} = \beta_0 + \beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 + \beta_3\mathbf{x}_3 + \boldsymbol{\epsilon}$, where there are three

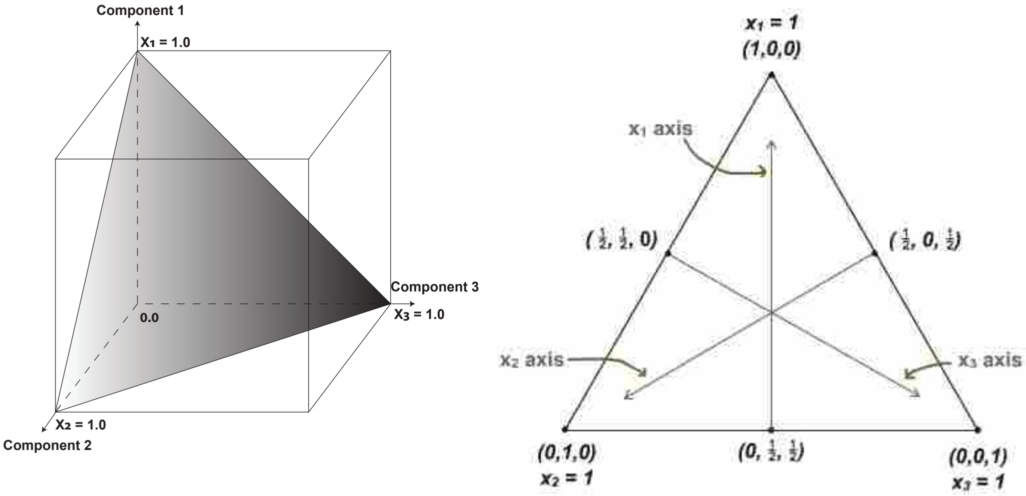

Figure 2.2: The figure on the left shows how the triangular region is derived from the experimental region of a three factor factorial design. The constrained triangular region, called a 'simplex', is used to plot points using a triangular coordinate system. Any coordinate inside the simplex represents a valid mixture, meaning the elements of the coordinate sum to one. Each vertex represents a mixture consisting entirely of the corresponding mixture component.

main effect terms and an intercept. However, the constraint that $\sum x_i = 1$ makes one of the x_is redundant since it can be derived from the proportions of the other two. This problem will cause the **X** matrix to be rank deficient and the least squares solution to be nonunique.

Scheffé (1958) introduced an alternative polynomial regression model to accommodate this problem by removing the intercept term. For a three-component mixture design, the first order Scheffé model is

$$\eta = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3, \tag{2.1}$$

where β_i is interpreted as the response for mixture consisting entirely of component i. Response levels at points off of the vertices are then weighted averages of the response levels on the vertices of the simplex. This is often referred to as the *linear blending* of the mixture.

Extending the model to include interaction effects allows for curvature in the response surface and (2.1) becomes

$$\eta = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3. \tag{2.2}$$

The interpretation of these interactions is a natural extension of the interpretation of the main effect terms. According to Scheffé (1958, p. 347), excess response η accounted for by the interactions over the linear blending is often called the 'synergism' of the mixture. If the response levels decrease on account of the interaction effects, it is called the 'antogonism'.

Although it is possible to include cubic and higher order terms, the total number of terms in the model increases rapidly, and these models would require many experiments to estimate. It is common practice to limit the degree of polynomial terms in the model due to constraints on time and resources for experimentation.

ADDING PROCESS VARIABLES Often times, characteristics of products created from mixtures are also affected by variables related to the process, or *process variables*, such as mixing time, temperature, and pressure. The effects of process variables can be estimated simultaneously with the mixture effects and incorporated into the Scheffé model. This is done by first defining a model for mixture components, $\eta(x) = f(x)$, and a model for the process variables, $\eta_{pv}(z) = g(z)$. A combined model is obtained by the cross-product of the two models, $\eta(x, z) = (f(x)) \times (g(z))$ (Kowalski et al. 2000).

With Kowalski's model, the problem associated with number of terms with higher-order polynomials is compounded when crossing the mixture variable model with the process variable model. If the number of terms in $\eta(x)$ is a and the number of terms in $\eta_{pv}(z)$ is b, then the number of terms in the combined model will be $a \times b$. It is easy to see how the number of terms can quickly get out of hand if an experimenter attempts to model too many factors in the same design.

MIXTURE DESIGNS WITH CONSTRAINTS Another common technique with mixture designs, and designs in general, is to incorporate constraints when only a subregion of the experimental region is of interest. In mixture designs, the experimental region has already been constrained to a simplex by $\sum x_i = 1$, but it is sometimes clear that certain areas of the simplex would create unusable mixtures that should be avoided in the design. Using constraints allows more resources to be directed to the region of interest for a more precise model.

When consistent lower bounds are placed on the proportions of mixture components, the constrained region reduces to a smaller simplex inside of the original simplex. For convenience, component proportions can be transformed to the scale of the smaller simplex, which are called *pseudo-components*. If l_i is the lower bound for component i, then the ith pseudo-component is

$$x'_i = \frac{x_i - l_i}{1 - \sum_{i=1}^{k} l_i}, \tag{2.3}$$

where k is the number of mixture components (Lawson 2010, pp. 461–462).

Pseudo-components are especially useful for plotting the response surface on the constrained region. When contour lines on the full simplex are hard to distinguish or interpret, pseudo-components create a "zoomed-in" view of the surface over the constrained region. This capability will be incorporated into the 'mixplot' function in the next version of the *mixexp* package in R (the current version at the time of this writing is 0.5-1).

The addition of upper constraints makes the selection of design points more difficult. The intersections of the constraints for each component can create a wide variety of irregular shapes. McLean and Anderson (1966) introduced the Extreme Vertices Design (EVD) for this situation, which generates design points from the extreme vertices, edge centroids, and facet centroids of the constrained region. In designs with many components, the number of extreme vertices can greatly outnumber the terms in the Scheffé models. For this situation, Snee and Marquardt (1974) introduced the XVERT algorithm, which can select a subset of design points from a candidate list consisting of extreme vertices and centroids that minimizes the trace of $\mathbf{(X'X)}^{-1}$, which is called an A-optimal design. Optimal designs are discussed in greater detail in a later section of this chapter.

MIXTURE DESIGNS IN SPLIT-PLOT ARRANGEMENTS One of the greatest challenges of experimental design is working around time and budgetary constraints. Split-plot designs relax the requirement to run the experiments in a completely randomized order in order to reduce the total amount of time required to perform the entire list of runs, or to make the execution less tedious. For example, in a design where oven temperature is a factor, it can take a long time for an oven to reach the correct temperature for the next run in the design. If runs are performed in blocks with a constant oven temperature, it would be possible to finish the list of runs much faster.

However, sacrificing complete randomization comes at a cost. Easy-to-vary factors are randomized within blocks called whole-plots in which the hard-to-vary factors are held constant. The levels of the hard-to-vary factors are randomized across the different whole-plots. In this two-stage randomization scheme, the hard-to-vary factors have a different experimental unit than the easy-to-vary factors. This requires an additional error term associated with the hard-to-vary factors. The result is a more complex model and less statistical power to detect the effects for hard-to-vary factors compared to the easy-to-vary factors (Lawson 2010, pp. 301-302). Many times, the potential savings in time and resources for this type of design outweigh these costs.

In the context of mixture designs, it is often convenient to create a large batch of a single mixture to be used for several runs of the design. Mixture designs in split-plot arrangements are called split-plot mixture process variable (SPMPV) designs (Lawson 2010, p. 484). In SPMPV designs, usually any variable that is part of the process to create the mixture is considered a hard-to-vary factor.

An example of this type of study can be found in Kowalski et al. (2002). The authors compare and contrast three methods for fitting models to this type of data using a simulation study followed by an example with real data: The first was ordinary least squares, which was included to benchmark the performance of the other two models. The second model was mixed model approach with a compound symmetric covariance structure to model the correlation of observations within whole-plots. The final model was called the "pure error method," which performed relatively equal to the second model with no apparent advantage. Both methods estimate the subplot error variance, but require at least partial replication within whole-plots, which is not always possible under time and budgetary constraints.

D-OPTIMAL DESIGNS The final step in planning the design is to actually generate the design points. Referring back to the discussion on mixture designs with upper and lower bounds on the mixture variables, a candidate-free algorithm already exists in JMP to generate D-optimal designs for irregularly shaped constrained experimental regions. This is the algorithm used to generate the design for this study. Both D-optimality in general and the algorithm proposed by Jones and Goos are discussed here.

D-optimal designs choose the set of design points that maximizes the determinant of the information matrix $\mathbf{X'X}$, or more generally $\mathbf{X'V}^{-1}\mathbf{X}$, where $\mathbf{V}$ is the variance of the response vector, $\mathbf{y}$. This is equivalent to minimizing the determinant of $(\mathbf{X'V}^{-1}\mathbf{X})^{-1}$, which simultaneous minimizes the generalized variance of $\hat{\boldsymbol{\beta}}$. The generalized variance is an overall measure of variance computed from the determinant of a covariance matrix (Atkinson and Donev 1992, pp. 42, 116-117). In summary, this an approach to choosing the design points that will provide the most precise estimates of the regression coefficients.

The algorithm proposed by Jones and Goos (2007) was designed to be a swiss-army-knife approach to generating D-optimal designs. With the growing popularity of the D-optimal criterion, experimenters have tried to find ways to generate D-optimal designs in the presence of complicating factors common to experimental design such as split-plot arrangements, mixture components, and high-dimensional constrained regions that yield an enormous list of candidate design points. Finding the D-optimal subset of such a large candidate set can be computationally very expensive. Jones and Goos' algorithm employs a candidate-free approach that can accommodate any of the previously mentioned complicating factors and is integrated into the Custom Design feature in JMP.

In summary, methods exist for mixture designs to overcome any of the challenges common to traditional designs; however, the presence of mixtures components does add a layer of complexity to any design. The literature summarized in this section provides a sturdy foundation for the design of this experiment, and the following sections explain the tools used in the analysis of the results.

Generalized Linear Models

Generalized linear models are a class of models that employ the use of link functions to fit linear models to response variables with nonnormal likelihoods in the exponential family of distributions. Nelder and Wedderburn (1972) introduced this class of models and specified a procedure based on iterative weighted least squares to obtain maximum likelihood estimates of the regression coefficients.

A generalized linear model consists of three parts:

1. *A random component* which consists of a response variable $\mathbf{y}$, which is a vector of independent observations from a distribution in the natural exponential family. All distributions in this family can be factored into the form

$$f(y_i, \theta_i) = a(\theta_i)b(y_i)\text{exp}[w(\theta_i)t(y_i)], \tag{2.4}$$

   where $w(\theta_i)$ is the canonical parameter, and $t(y_i)$ is the sufficient statistic.

2. *A systematic component* that is the linear combination of explanatory variables used to predict the response variable and has the form

$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij}. \tag{2.5}$$

3. *A link function* which connects the linear component of the model to the random component by $\eta_i = g(\mu_i)$. Link functions are functions of the parameters related to the mean of y_i in the random component.

The canonical parameter $w(\theta_i)$ can always be used as a link function and is called the canonical link. In the case of bernoulli or binomial data, $w(p_i) = \ln(\frac{p_i}{1-p_i})$, which is the *logit* link in logistic regression (Agresti 2002, p. 117). Using the logit link, the linear component models the log-odds of the event recorded in the binomial or bernoulli response variable. The log-odds can then be converted back to probability of a "success" given the explanatory variable by using the inverse of the logit transform, which is

$$p_i = \frac{\exp(\sum_{j=1}^{p} \beta_j x_{ij})}{1 + \exp(\sum_{j=1}^{p} \beta_j x_{ij})}. \tag{2.6}$$

However, there are alternative link functions for logistic regression that are considered in this study. The probit link relates the linear component to the random component using the inverse of the cumulative distribution function of the standard normal distribution $\Phi^{-1}(p_i)$. Often times, the resulting model using the probit link is nearly indistinguishable from the same model fit using the logit link. Finally, there is the complementary-log-log link, abbreviated cloglog, which is $g(p_i) = log[-log(1 - p_i)]$. The advantage of the cloglog link is that it does not have to be symmetric about the point where $\hat{p}_i = 0.50$, but it is rarely used in practice compared to the probit and logit links (Agresti 2002, pp. 245–250). Choosing the appropriate link function is usually a matter of personal judgment or comparing the models using some information criterion such as AIC or BIC.

Model Selection

In many situations, it is not known which explanatory variables are actually related to the response variable. Unless it is known beforehand which terms in the model are significant, it is almost always necessary to eliminate the terms that do not significantly contribute to the model. Statisticians consider themselves to be scientists, and when competing hypotheses of varying complexity explain the same phenomenon equally well, Occam's razor demands that we assume the simplest hypothesis. There are also many consequences to over-fitting a model, such as poor, overconfident prediction performance and possibly economic costs of tracking more explanatory variables than necessary. Simplifying a model in experimental design can greatly reduce the number of experimental runs needed in future experimentation. At the same time, under-fitting yields biased estimates of the regression coefficients. Finding a good balance between model parsimony and model fit is a difficult but pervasive problem in statistical model building. In the last half-century, there has been a lot of work done in the field of model selection.

Hocking (1976) published a landmark paper on the subject and discussed a wide-variety of techniques for model selection. In this paper, Hocking introduced stepwise methods, which either add or delete variables one at a time based on some criterion chosen by the analyst. Stepwise methods encompass two methods which are both variations on the same idea: forward selection (FS) and backward elimination (BE).

The FS method begins with no variables in the model and adds one variable at a time until some stopping criteria is met. In the paper, Hocking suggests adding the variable with largest single degree of freedom F-ratio among the remaining eligible variables. This is done iteratively until no remaining eligible variable has a higher F-ratio than some predetermined stopping value F_{in}.

BE does something similar, except it begins with all terms in the model and iteratively deletes the term in the model that has the lowest F-ratio until all of the remaining terms have an F-ratio larger than some predetermined stopping value F_{out} (Hocking 1976, p. 8).

Since stepwise methods were first introduced, they have been adapted to use many different selection criteria, such as R^2, R^2_{adj}, AIC, BIC, SBIC, Mallow's Cp, and PRESS. PROC LOGISTIC in SAS software has an option to use a FS or BE algorithm for model selection by using the SELECTION= option in the MODEL statement. By default, this procedure uses the Wald χ^2 statistic as the criteria for including and eliminating terms from the model. The user can specify a significance level such as 0.01 or 0.10 using the SLENTRY= option, which is used as the stopping criterion for the selection algorithm (SAS Institute 2010, pp. 3943–3944).

Model Fit Diagnostics

Once a model has been selected, it should be examined for goodness-of-fit, violations of assumptions, and influential points. Goodness-of-fit for logistic regression models can be examined using a generalization of ANOVA, which is the analysis of deviance introduced by Nelder and Wedderburn (1972, pp. 375–376). Goodness-of-fit is tested using the deviance statistic D, which is

$$D = -2[\ell(\mathbf{y};\mathbf{x},\boldsymbol{\beta}) - \ell(\mathbf{y};\mathbf{y})], \tag{2.7}$$

where $\ell(\mathbf{y};\mathbf{x},\boldsymbol{\beta})$ is the log-likelihood for the fitted model, and $\ell(\mathbf{y};\mathbf{y})$ is the log-likelihood for the saturated model. D is a likelihood ratio test statistic, which asymptotically follows a χ^2_{n-p} distribution under regularity conditions, where n is the number of observations in the data and p is the number of terms in the fitted model.

Another tool to evaluate the model fit using deviance is the residual deviance statistic ΔD. Residual deviance is simply the difference in the deviance of the fitted model with the deviance of the intercept-only, or null, model. ΔD also has a χ^2 asymptotic distribution with $p-1$ degrees of freedom.

When these tests indicate a poor fit, the cause can often be identified graphically through residual plots. However, traditional residuals of the from generalized linear models of the form $\hat{y} - y$ are not always helpful. An alternative residual is the Pearson residuals, e_i, where

$$e_i = \frac{y_i - \hat{\mu}_i}{\left[\hat{V}(Y_i)\right]^{1/2}}. \tag{2.8}$$

Pearson residuals originate from Pearson's χ^2 statistic for contingency tables, which is equal to the sum of squared Pearson residuals (Agresti 2002, p. 142).

Finally, the model should be checked for data points that have disproportional influence on the estimated regression coefficients. DFBETAS measures the absolute difference in coefficient estimates when each observation is temporarily removed from the model. Both

Pearson residual plots, and DFBETA plots are available as an option with ODS graphics output from PROC LOGISTIC in SAS software (SAS Institute 2010, p. 3996).

When fitting a model, being thoughtful about every aspect in the process is critical. Careful model selection and inspection can save the analyst a lot of pain and embarrassment, and it is worth taking the time and effort to be thorough. The next chapter discusses how each of these considerations are implemented into this study.

CHAPTER 3

METHODS

This chapter provides the details for each step in the process of planning and creating the design, executing the experiments, and analyzing the results using the methods described in the literature review. The first section explores the variables used in the process, and the following sections discuss the design created in JMP, fitting the logistic regression model in SAS, and visualizing the response surfaces using R.

3.1 Explanatory and Response Variables

The chemists in this work already spent years studying monolithic columns for chromatography in literature and in practice, so their expert knowledge makes the identification of potential explanatory factors very easy. They also provided constraints on the experimental region and specified any time and budgetary constraints. All of these considerations are integrated into the design. This section clarifies what are the response and explanatory variables in this study, the constraints, and why a split-plot design is used.

Response Variables

The ultimate goal of this research is to discover a way to make UHPLC-capable devices from monolithic columns. However, it is not very well understood at this time what settings of the proposed process will create a homogeneous monolith with an open pore structure. Analytically, it would be difficult to make inference from a design to study pore sizes if a large number of experimental runs yielded unusable columns, whether they be inhomogeneous or homogeneous with closed pores. For this reason, the researchers first need to know what blends of the mixtures and what process-variable settings will consistently produce a workable monolithic column.

Initially, the two response variables specified were two binary responses: whether or not the monolith was homogeneous, and whether or not the monolith had an open pore structure. For simplification of the analysis, the two response variables are combined so that the response is still binary, but a "success" is determined to be a monolith that is both homogeneous and capable of flow. As previously discussed, binary responses can be modeled using logistic regression.

Explanatory Variables

The covariates in the logistic regression model are used to model the probability of creating a homogeneous monolith capable of flow. The researchers determined that this probability was most likely a function of three mixture variables and five process variables, which are detailed in the following paragraphs.

MIXTURE VARIABLES The emulsion used to create the monolith is a mixture of a porogen, a monomer, and a surfactant. The monomer and porogen are two immiscible fluids, and the surfactant is a chemical agent that reduces the surface tension of the two fluids. The monomer is what eventually becomes the solid structure of the monolith, and the water droplets in the mixture are replaced during the polymerization process by pores. The idea is to create a mixture where the porogen is well dispersed throughout the monomer in tiny droplets.

Four different mixture component chemicals are actually used in this study. The researchers want to study two different monomers, DEGDA and EDGmA. Only one of these chemicals is used at a time for an individual mixture, but the identity of the monomer is indicated using factor variable (this variable is classified as a process variable and is explained later). The porogen component used throughout this study is water, and the surfactant used is EPE 4400.

PROCESS VARIABLES Five process variables are included in this study. These are initiator, organic salt, mixing time, heating temperature, and an indicator variable specifying which monomer is used in the mixture. Initiator is a substance that initiates a chemical reaction, such as polymerization in this case. The initiator in this study is potassium persulfate. The organic salt, calcium chloride, works in conjunction with the surfactant to stabilize the emulsion. Even though the initiator and organic salt are ingredients in the mixture, they do not change the volume of the mixture, and therefore are not considered mixture components.

A magnetic mixer mixes the ingredients between 45 and 75 minutes. The capillary columns filled with the mixture were then polymerized using a hot-water bath set to a temperature between $55\,^{\circ}$ and $65\,^{\circ}C$.

Constraints

In order to prevent the design from exploring areas of the experimental region that are not of interest, constraints were imposed on the proportions of the mixture components and on the levels of all of the process variables except the indicator variable for which monomer is used. The constraints on the mixture variables are specified below, followed by the constraints on the process variables.

MIXTURE VARIABLES The researchers determined the constraints on the mixture components based on literature and experience, but are still somewhat arbitrary. Table 3.1 contains the constraints on the proportions of the mixture components. Water is constrained to be a very large proportion of the mixture. A graphical depiction of the constrained region on a simplex is shown in Figure 3.1.

PROCESS VARIABLES The constraints on the process variables are displayed in Table 3.2. It should be noted here that the salt and initiator are both in terms of proportions of the amount of a mixture components.

Table 3.1: Constraints on the mixture components.

	Bounds on Proportions	
Variable	Lower	Upper
Porogen	0.70	0.86
Monomer	0.10	0.25
Surfactant	0.02	0.05

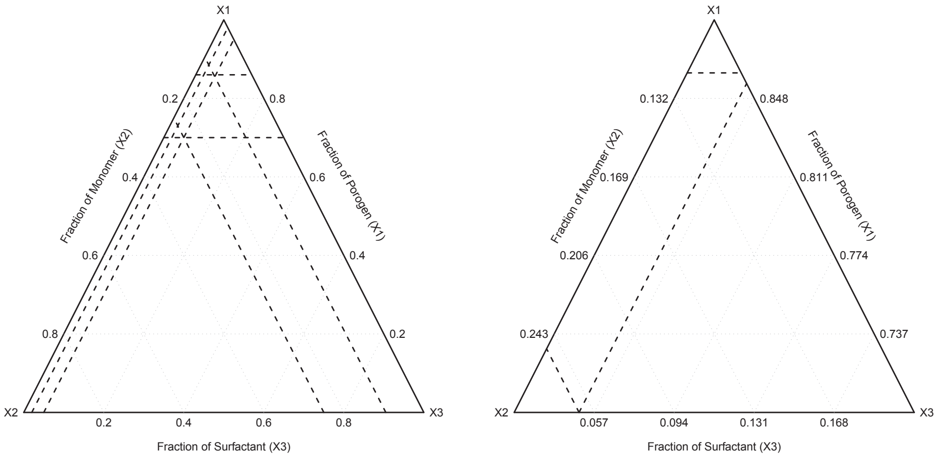

Figure 3.1: This figure is a graphical representation of the constrained experimental region. The dashed lines on the simplex indicate a constraint. The constrained region appears to be a trapezoid shape on a very small region of the simplex. The plot on the left shows what region looks like zoomed-in using pseudo-components.

Split-plot Design

The equipment used for this study belongs to a laboratory shared by multiple students and faculty, therefore the availability of the lab equipment is limited. It was decided that a maximum of two mixture formulations could be created and four total runs could be performed in a single day. This constraint required the use of a split-plot arrangement where two whole-plot blocks could be performed on a daily basis. Each whole-plot uses a single mixture formulation, and two columns filled by that mixture are polymerized at two different temperatures in the hot-water bath. Since all of the factors except temperature are part of the process to create the mixture, these are all classified as hard-to-vary factors. For

Table 3.2: Constraints on the mixture components.

		Bounds	
Variable	Unit	Lower	Upper
Initiator	Proportion of monomer	0.002	0.10
Salt	Proportion of porogen	0.001	0.02
Mix Time	Minutes	45	75
Temperature	Degrees Celsius	55	65

example, we cannot change the amount of salt or change which monomer is used without having to create an entire new batch. Temperature is the only easy-to-vary factor.

One complication with this design is that there is no replication inside of the whole-plots. The result is that the subplot variance σ_ϵ^2 is not estimable. Section 3.4 explains how this problem is overcome.

3.2 Model Specification

Using the model formulation specified by Kowalski et al. (2000), separate models for the mixture components and the process variables are fit.

The model for the mixture variables $\eta(\mathbf{x})$ is the quadratic Scheffé model of the form

$$\eta(\mathbf{X}) = \beta_p \mathbf{x}_p + \beta_m \mathbf{x}_m + \beta_s \mathbf{x}_s + \beta_{p\times m}\mathbf{x}_p\mathbf{x}_m + \beta_{p\times s}\mathbf{x}_p\mathbf{x}_s + \beta_{m\times s}\mathbf{x}_m\mathbf{x}_s, \tag{3.1}$$

where subscripts p denotes porogen, m denotes monomer, and s denotes surfactant.

For the process variable model, $\eta_{pv}(\mathbf{z})$, is just a simple main effects model with six terms of the form

$$\eta_{pv}(\mathbf{Z}) = \alpha_0 + \alpha_{init} z_{init} + \alpha_{salt} z_{salt} + \alpha_{mix} z_{mix} + \alpha_{temp} z_{temp} + \alpha_{mon2} z_{mon2}, \tag{3.2}$$

where subscripts *init* denotes initiatior, *salt* denotes salt, *mix* denotes mix time, *temp* denotes temperature, and *mon*2 denotes the indicator variable that EGDmA is used in the mixture instead of DEGDA.

With six terms in $\eta(\mathbf{X})$ and six terms in $\eta(\mathbf{Z})$, the combined Kowalski model would have 36 terms. To reduce the number of experiments required, the model is simplified by removing the interaction terms between process variable main effects and mixture variable interactions from the model. The result is that it is only possible to estimate the effects of the process variables on the linear blending of the mixture components, and not on the quadratic blending. The resulting model is

$$\begin{aligned}
\eta(\mathbf{X},\mathbf{Z}) = {} & \gamma_1^0\mathbf{x}_p + \gamma_2^0\mathbf{x}_m + \gamma_3^0\mathbf{x}_s + \gamma_4^0\mathbf{x}_p\mathbf{x}_m + \gamma_5^0\mathbf{x}_p\mathbf{x}_s + \gamma_6^0\mathbf{x}_m\mathbf{x}_s \qquad (3.3)\\
& + \gamma_{11}^1\mathbf{z}_{init}\mathbf{x}_p + \gamma_{12}^1\mathbf{z}_{init}\mathbf{x}_m + \gamma_{13}^1\mathbf{z}_{init}\mathbf{x}_s + \gamma_{21}^2\mathbf{z}_{salt}\mathbf{x}_p + \gamma_{22}^2\mathbf{z}_{salt}\mathbf{x}_m + \gamma_{23}^2\mathbf{z}_{salt}\mathbf{x}_s\\
& + \gamma_{31}^3\mathbf{z}_{mix}\mathbf{x}_p + \gamma_{32}^3\mathbf{z}_{mix}\mathbf{x}_m + \gamma_{33}^3\mathbf{z}_{mix}\mathbf{x}_s + \gamma_{41}^4\mathbf{z}_{mon2}\mathbf{x}_p + \gamma_{42}^4\mathbf{z}_{mon2}\mathbf{x}_m + \gamma_{43}^4\mathbf{z}_{mon2}\mathbf{x}_s\\
& + \gamma_{51}^4\mathbf{z}_{temp}\mathbf{x}_p + \gamma_{52}^4\mathbf{z}_{temp}\mathbf{x}_m + \gamma_{53}^4\mathbf{z}_{temp}\mathbf{x}_s, \qquad (3.4)
\end{aligned}$$

which has a total of 21 terms.

3.3 JMP® Custom Design Tool

The Custom Design feature in JMP available from the DOE menu is used to generate a D-optimal set of design points based on the model. Generating the design required specifying a response, factors, constraints, and model terms.

Two responses are specified so that the researcher carrying out the experiments can record separately whether or not the monolith is homogeneous and if it has open pores. Nothing is entered into the fields for lower and upper limits or importances.

In entering the factors, porogen, monomer, and surfactant are specified as hard-to-vary mixture variables; salt, initiator and mix time are specified as continuous hard-to-change variables; temperature is specified as continuous and easy-to-change; and finally the factor variable "Mon1or2" is specified as a categorical hard-to-change variable. The factor constraints previously discussed are entered in as lower and upper bounds for each of the variables. Another constraint was placed on surfactant to ensure that it was never outside the range of 20% to 50% of monomer.

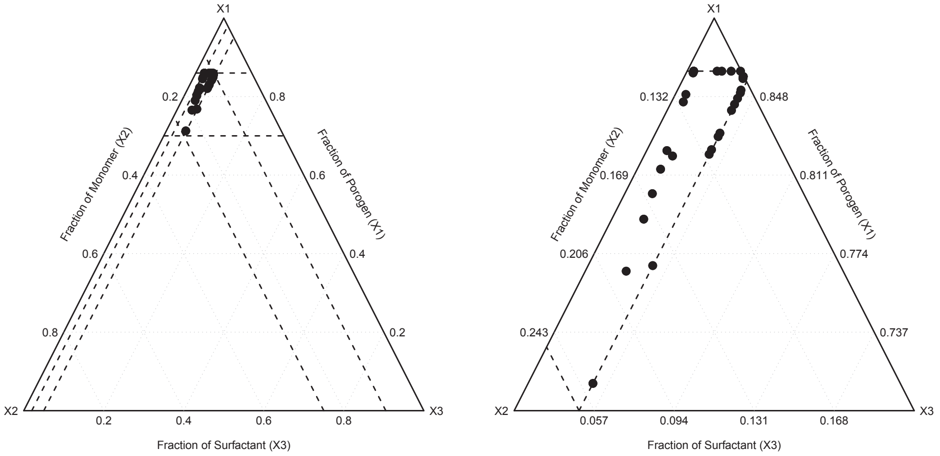

Figure 3.2: The simplex on the left shows the location of the design points on the full simplex. The simplex on the right shows the design points on the simplex using pseudo-components. Several runs were performed on the same point on the simplex, but with different process variable settings; therefore, several points are super-imposed on other points.

The terms from the model previously specified are then input into the 'Model' section. It was determined that 33 whole-plot blocks could be completed in a little over two weeks for a total of 66 runs. Figure 3.2 shows the location of the design points on the simplex. A JMP script that recreates the design is also available in Appendix A. After the design points are generated in randomized order, they are output in a spread-sheet. The list of runs is then provided one of the researchers with instructions to perform the experiments in the given order.

3.4 Analysis Methods

After the experiments are completed, fitting and visualizing the estimated response surface are the next steps. A discussion of how the model is fit and visualized using SAS software and R concludes the chapter.

Logistic Regression Models

The lack of replication within whole-plots makes it impossible to estimate the subplot error variance term σ_ϵ^2, which is necessary to fit a mixed model. In more typical situations with straightforward models, this problem can be overcome by fitting a model with whole-plot effects only, and a model with subplot effects only with a factor variable indicating which whole-plot the observation belongs to. After independently determining which effects are significant in each model, the terms corresponding to the significant effects, except for the whole-plot factor variable, are pooled into a combined model. Mimicking this approach, the effect for temperature modeled by the subplot model will not be consistent with the Kowalski model, where process variables are only present in the model through interactions with mixture variables.

In an effort to maintain the framework of the Kowalski model, the modeling procedure illustrated in Figure 3.3 was created for this study. The idea is to heuristically assess whether or not the whole-plot error variance is negligible and all the terms can be estimated together as if the data are independent.

Model 1 is the subplot model that only includes only a temperature effect and factor variable for whole-plot. Model 2 is the model with all terms in the model, including temperature-by-mixture-component interactions, estimated as if the data are independent. If both models agree that temperature is significant, there is evidence that the whole-plot error term is negligible and Model 2 is used. The model for this scenario is the original model specified in (3.3). If neither model has significant temperature effects, temperature is ignored, and modeling becomes straightforward with only whole-plot effects still present and the model for this scenario is (3.5).

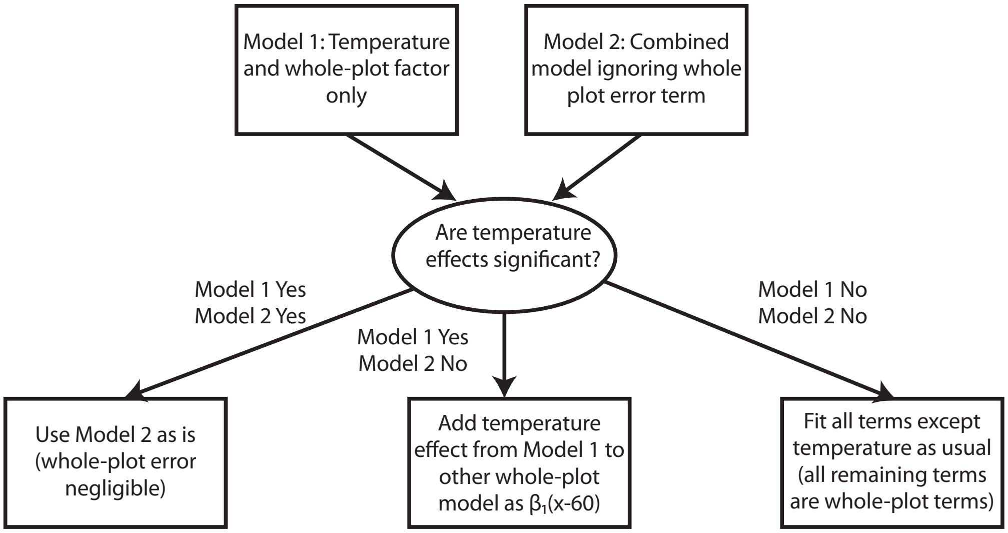
Model 1: Temperature and whole-plot factor only
Model 2: Combined model ignoring whole plot error term
Are temperature effects significant?
Model 1 Yes Model 2 Yes
Model 1 Yes Model 2 No
Model 1 No Model 2 No
Use Model 2 as is (whole-plot error negligible)
Add temperature effect from Model 1 to other whole-plot model as β1(x-60)
Fit all terms except temperature as usual (all remaining terms are whole-plot terms)

Figure 3.3: This decision tree shows how any temperature effect will be handled in the final stage of model fitting. Ultimately, it is the role of the subplot model to decide if temperature will appear in the model in any form. A fourth scenario, "Model 1 No Model 2 Yes", is excluded from the chart. This scenario is very unlikely since Model 1 should be far more powerful to detect any temperature effect than Model 2.

$$\begin{aligned}\eta(\mathbf{x},\mathbf{z}) = {} & \gamma_1^0\mathbf{x}_p + \gamma_2^0\mathbf{x}_m + \gamma_3^0\mathbf{x}_s + \gamma_4^0\mathbf{x}_p\mathbf{x}_m + \gamma_5^0\mathbf{x}_p\mathbf{x}_s + \gamma_6^0\mathbf{x}_m\mathbf{x}_s \\ & + \gamma_{11}^1\mathbf{z}_{init}\mathbf{x}_p + \gamma_{12}^1\mathbf{z}_{init}\mathbf{x}_m + \gamma_{13}^1\mathbf{z}_{init}\mathbf{x}_s \\ & + \gamma_{21}^2\mathbf{z}_{salt}\mathbf{x}_p + \gamma_{22}^2\mathbf{z}_{salt}\mathbf{x}_m + \gamma_{23}^2\mathbf{z}_{salt}\mathbf{x}_s \\ & + \gamma_{31}^3\mathbf{z}_{mix}\mathbf{x}_p + \gamma_{32}^3\mathbf{z}_{mix}\mathbf{x}_m + \gamma_{33}^3\mathbf{z}_{mix}\mathbf{x}_s \\ & + \gamma_{41}^4\mathbf{z}_{mon2}\mathbf{x}_p + \gamma_{42}^4\mathbf{z}_{mon2}\mathbf{x}_m + \gamma_{43}^4\mathbf{z}_{mon2}\mathbf{x}_s\end{aligned} \tag{3.5}$$

If Model 1 has a significant temperature effect but Model 2 does not, then there is evidence that the whole-plot error term is not negligible. In this case, the temperature effect is combined with a model of whole-plot effects by adding the coefficient for temperature estimated by Model 1 multiplied by a centered value for temperature. The reason for the centered temperature value is that the prediction from the whole plot model represents a prediction averaged across the two levels of temperature, $55\,^\circ$ and $65\,^\circ C$. The temperature effect in this scenario is unique from the other process variable effects because changing the level temperature results in a simple raising or lowering of the entire response surface instead of altering the blending properties of the mixture components. The model for this scenario is (3.6)

$$
\begin{aligned}
\eta(\mathbf{x},\mathbf{z}) = {} & \gamma_1^0\mathbf{x}_p + \gamma_2^0\mathbf{x}_m + \gamma_3^0\mathbf{x}_s + \gamma_4^0\mathbf{x}_p\mathbf{x}_m + \gamma_5^0\mathbf{x}_p\mathbf{x}_s + \gamma_6^0\mathbf{x}_m\mathbf{x}_s \\
& + \gamma_{11}^1\mathbf{z}_{init}\mathbf{x}_p + \gamma_{12}^1\mathbf{z}_{init}\mathbf{x}_m + \gamma_{13}^1\mathbf{z}_{init}\mathbf{x}_s \\
& + \gamma_{21}^2\mathbf{z}_{salt}\mathbf{x}_p + \gamma_{22}^2\mathbf{z}_{salt}\mathbf{x}_m + \gamma_{23}^2\mathbf{z}_{salt}\mathbf{x}_s \\
& + \gamma_{31}^3\mathbf{z}_{mix}\mathbf{x}_p + \gamma_{32}^3\mathbf{z}_{mix}\mathbf{x}_m + \gamma_{33}^3\mathbf{z}_{mix}\mathbf{x}_s \\
& + \gamma_{41}^4\mathbf{z}_{mon2}\mathbf{x}_p + \gamma_{42}^4\mathbf{z}_{mon2}\mathbf{x}_m + \gamma_{43}^4\mathbf{z}_{mon2}\mathbf{x}_s \\
& + \beta_{temp}(\mathbf{z}_{temp} - 60)
\end{aligned} \tag{3.6}
$$

The model is fit using PROC LOGISTIC in SAS. All of the terms specified in (3.3) are included in the MODEL statement and the NOINT option is used to prevent fitting an intercept. Model selection is performed using forward selection with a stopping criterion $\alpha = 0.20$ for the Wald statistics. This stopping criterion was chosen to have smaller type II error rate than using $\alpha = 0.05$ because the cost of falsely determining an effect to be non-significant now could be very costly in future experimentation when that factor is no longer considered. All of the effects involving only mixture components are forced in the model using the START= option.

This same procedure is performed three times using each of the previously discussed link functions and comparing the fits using AIC. The logit link function is preferred by default since it is the canonical link and fairly easy to interpret. If there is a large disparity in AIC favoring one of the other two link functions over the logit link, then that link function is used instead.

After the final model is selected, assumptions are checked using diagnostic plots. Pearson's residual plots are examined for dependence, non-linearity, heteroscedasticity, and non-normality. The DFBETAS plots reveal any influential points.

A common problem with logistic models is complete separation or quasi-complete separation. This simply means that one or more of the predictor variables is able to perfectly predict the response, creating a situation where the maximum likelihood solution is not unique. Fortunately, this problem is easily overcome using the FIRTH option in the model statement, a new feature appearing in SAS 9.2 (SAS Institute 2012). This option makes it possible to get converged estimates of the coefficients and Wald χ^2 tests in the presence of separation problems.

Mixture Plot

Once all assumptions seem to be satisfied, the fitted surface can be visualized using contour plots over the simplex. The 'mixplot' function in the *mixexp* package in R (Lawson 2011) is modified to handle user-supplied prediction equations, plot mixture component constraints, and have an option to plot using pseudo-components. The code for this version of the function, mixplot2, is provided in Appendix C. Conclusions about which areas of the experimental region can be made by viewing these plots conditional on the levels of process variables.

CHAPTER 4

RESULTS

The design points and results appear in the tables contained in Appendix D. This chapter examines the fitted model, model diagnostics, and the estimated surface plots.

4.1 Model Fit

The first step of fitting the model is to decide whether or not temperature, the single subplot effect, should be included in either form discussed in the previous chapter. Following this step, several link functions are compared using AIC. Finally, the coefficients of the model fit with the chosen link function are examined.

Examination of Temperature Effects

Temperature is only included in the model in any form if the subplot model determines that it is significant. In this case, the subplot model fit using PROC LOGISTIC produces an error due to a quasi-complete separation. The culprit in this case is several levels of the factor variable denoting which whole-plot. Using the FIRTH option in the MODEL statement, the estimate for the temperature effect converges to 0.0646 with standard error 0.064. The Wald χ^2 statistic is 1.0169 with one degree of freedom and a corresponding p-value 0.3133. Based on this result, temperature effects are excluded from the model, and all remaining terms are whole-plot effects as described by (3.5).

Comparison of Link Functions

In order to determine an appropriate link function for the final model, the logit, probit, and cloglog link functions are compared using AIC and interpretability of parameter estimates.

Forward-selection is used for all three models starting with the model specified in (3.5). This shows if the link function specification alters the subset of terms chosen by stepwise selection algorithm. In this case, the same subset of terms chosen for the model is consistent across all link functions, so that is not a problem. Table 4.1 shows the AIC corresponding to each link function. AIC prefers the cloglog link function, but the disparity between the logit and cloglog link appears to be small. The logit link is chosen since it has superior interpretation and an AIC only slightly worse than the cloglog link.

Table 4.1: AIC values for each link function.

Link Function	AIC
Logit	77.983
Probit	78.436
Cloglog	76.578

Parameter Estimates

Using the logit link, the model selection procedure reduces the full model down to the linear blending terms, the quadratic blending terms, and interactions for initiator, salt, and 'mon2' with the linear blending terms. Table 4.2 displays the estimated model coefficients.

The negative coefficients on the quadratic blending terms suggest that the surface dips toward the middle as a result of antagonism between all of the mixture components. Initiator, salt, and mon2 all interact with two of the mixture components. The fact that each of those process variables interacts positively with one mixture component and negatively with another makes it difficult to assess which levels of those process variables is optimal for producing workable monoliths with high probability.

Table 4.2: Table of regression coefficient estimates.

Parameter	DF	Estimate	Standard Error	Wald χ^2	Pr > χ^2
Porogen	1	1.0996	12.8468	0.0073	0.9318
Monomer	1	221.7	372.8	0.3537	0.5520
Surfactant	1	5040.0	5758.3	0.7661	0.3814
Porogen*Monomer	1	-245.7	438.6	0.3138	0.5754
Porogen*Surfactant	1	-5267.6	6076.4	0.7515	0.3860
Monomer*Surfactant	1	-6148.0	7535.3	0.6657	0.4146
Monomer*Initiator	1	-2628.7	1438.9	3.3375	0.0677
Surfactant*Initiator	1	11192.7	5272.9	4.5059	0.0338
Porogen*Salt	1	800.3	304.0	6.9317	0.0085
Surfactant*Salt	1	-17525.3	6161.1	8.0913	0.0044
Monomer*Mon2	1	-39.6059	13.6094	8.4692	0.0036
Surfactant*Mon2	1	105.1	37.3340	7.9233	0.0049

4.2 Model Diagnostics

The appropriateness of the model is examined using a variety of diagnostic tools. This section discusses the use of the Pearson residual plot, DFBETA plots, and an analysis of deviance.

Pearson Residuals

The plot of the Pearson residuals diagnose a poor fit by the presence of outliers and patterns due to dependence over time or nonlinearity. The plot in Figure 4.1 shows that observation 28 may be an outlier. If that point is also influential, it should also stand out in the DFBETAS plots If that is the case, observation 28 would be deleted. However, at this point, no corrective action seems warranted.

Influence

The next diagnostic tool, DFBETAS plots, is used to assess influence. With a moderately small data set with many parameters, it is expected that some points will be moderately in-

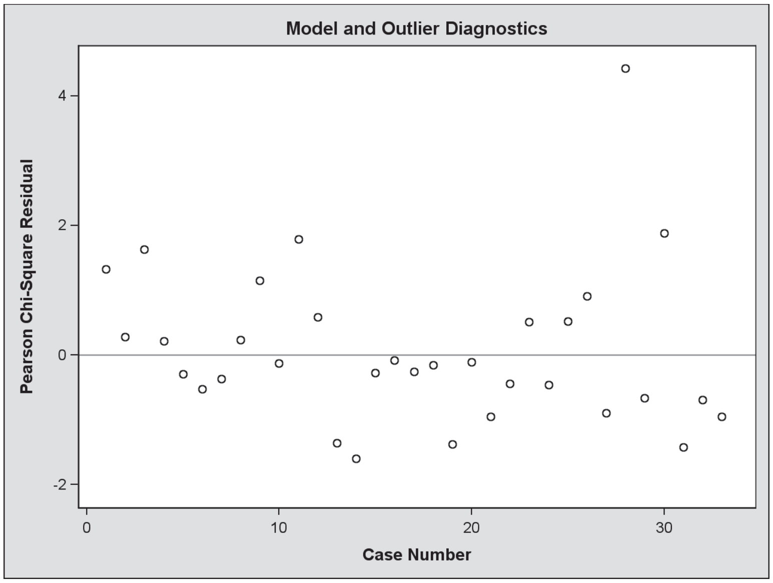

Figure 4.1: The Pearson residual plot does not seem to suggest that there are any problems with nonlinearity or dependence. Observation 28 seems to stand out and may be an outlier.

fluential. However, observation 1 seems to have high influence on the $Porogen \times Surfactant$ and $Monomer \times Surfactant$ interactions, as seen in Figure 4.2. This observation comes from the upper left-hand corner area of the constrained region, which is an area populated by several design points. Therefore, it is not immediately clear what makes this point so influential. The result for this observation is that two out of two trials had homogeneous monoliths with open pores, which means that this corner area may have lower probability in reality than what is predicted by this model. This region may be checked for accuracy using confirmatory experiments.

Also, Observation 28, a potential outlier, was of particular concern that it may also be influential, but did not stand out as being highly influential for any of the terms in the model.

Analysis of Deviance

An analysis of deviance provides a goodness-of-fit diagnostic. The deviance statistic for this model is 40.1198 with 21 degrees of freedom. The test has a p-value of 0.0072, which leads

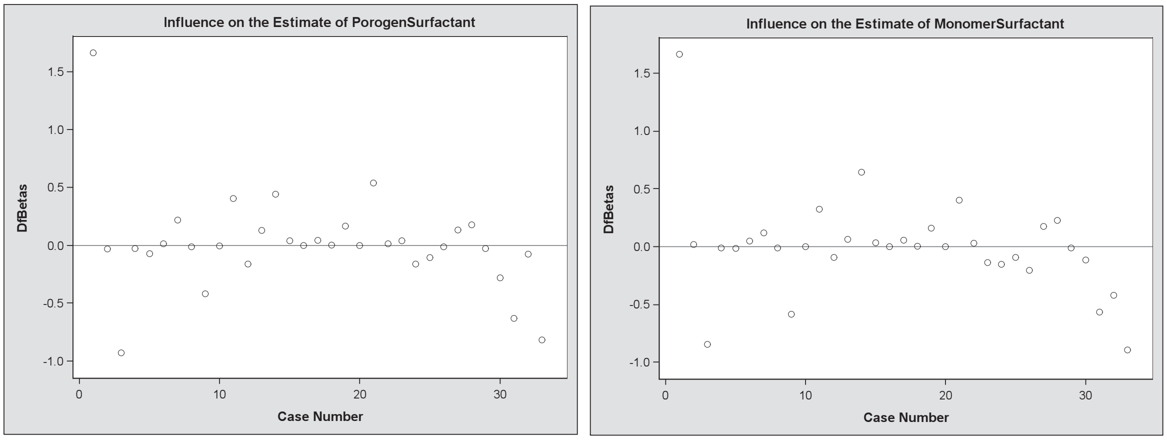

Figure 4.2: DFBETAS plots for selected terms with concerning amounts of influence present. DFBETAS plots help assess if any individual points are extraordinarily influential on the coefficient estimates of the model. In these plots, observation 1 seems to be disproportionately influential on the coefficient estimates corresponding to the $Porogen \times Surfactant$ and $Monomer \times Surfactant$ interaction effects.

to the conclusion that the model performs significantly worse than a saturated model. Also, the data are overdispersed for this model. The overdispersion parameter using the deviance method is 1.9105.

Comparing to the intercept only model, $\Delta D = 19.653$ with 12 degrees of freedom. This model has a marginally significant p-value of 0.0743. Therefore, the model seems to perform better than the intercept-only model, but not as well as the saturated model.

In conclusion, some of the diagnostic tools do suggest that the model can be potentially misleading. There is a possible outlier, influential points, and the model does not compare very well to a saturated model. However, these are not severe enough to warrant any action until confirmatory experiments suggest that model does not perform well.

4.3 Surface Plots

Finally, the surface plots are created using the ‘mixplot’ function in R. Figure 4.3 contains a selected surface plots intended to illustrate the effects of each process variable. Based on these plots, it can be concluded that increasing levels of salt or initiator tend to increase the area of low probability regions inside the constrained region. Using EGDmA instead of DEGDA in the mixture has a similar effect.

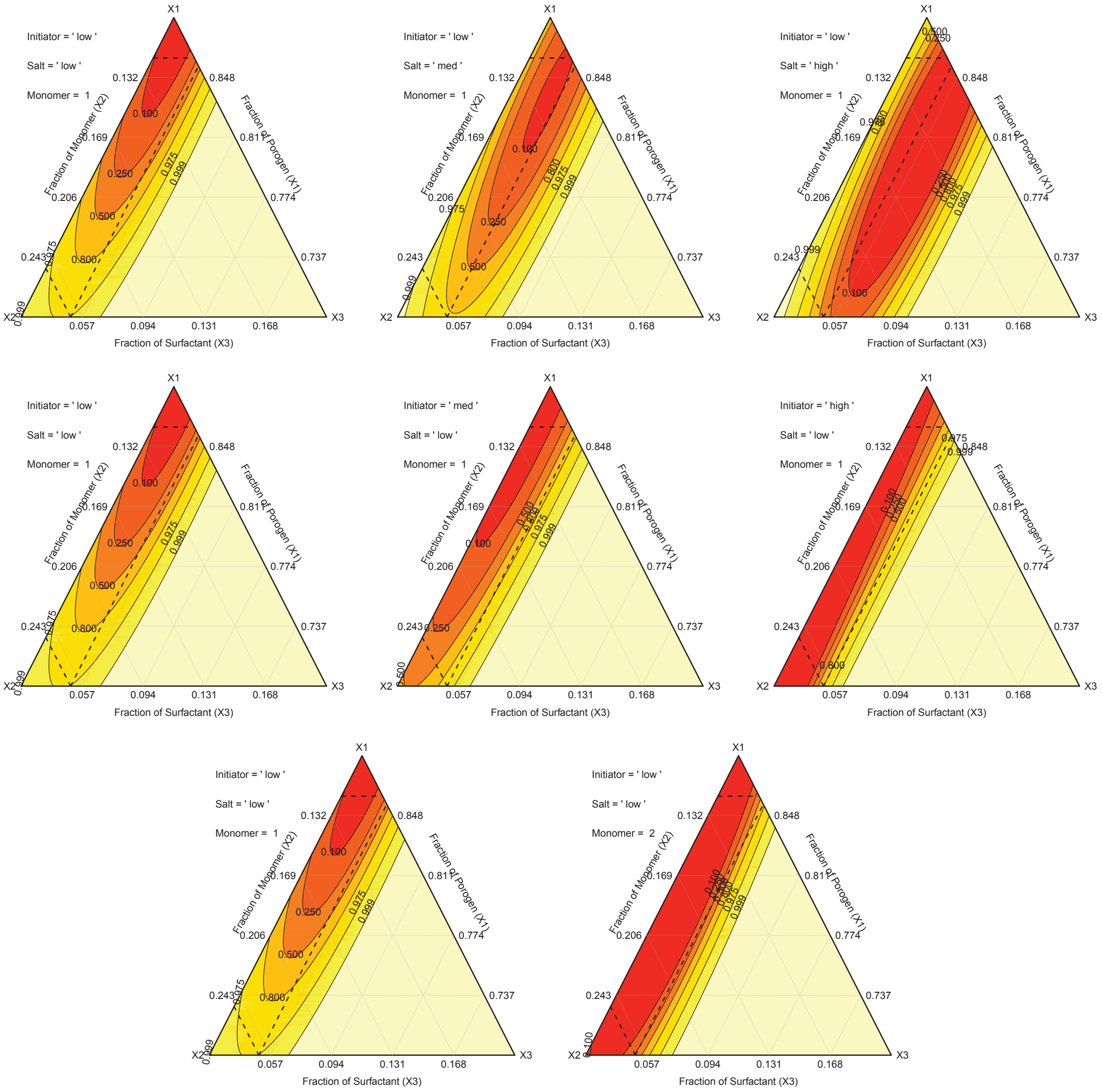

Figure 4.3: Selected surface plots for various combinations of process variable settings. The color gradient from red to light yellow corresponds to low to high probability. Each row is designed to illustrate the effect of increasing the level of one process variable while holding all others constant. The top row illustrates the effect of increasing salt, the middle row illustrates increasing initiator, and the bottom row illustrates the effect of changing monomers from DEGDA to EGDmA.

Another troubling feature of this plot is that, in general, the areas of highest probability for the surface are either outside the constrained region (which is an extrapolation), or in the area of the constrained region where the fewest experiments were performed. Due to both of these observations, it is highly recommended that confirmatory experiments be carried out to validate the models in these potentially promising areas.

At this point, it is difficult to recommend new constraints for further experimentation to minimize pore sizes. The areas of the highest probability are also the areas that inspires the least amount of confidence based on the location of design points and the tight constraints. Conclusions based on these results are discussed in more detail in the following chapter.

CHAPTER 5

CONCLUSIONS

Even though there is some indication that the fitted model may not be very trustworthy, there are many encouraging conclusions that can be made from the model. Temperature and mix time are likely important factors in determining the pore size of monoliths created with this process. Since there is little to no evidence suggesting that either variable has a significant effect on the probability of producing a workable monolith, these variables will be less constrained in future experimentation to minimize pore sizes. On the other hand, the model suggests that initiator should be fixed at its lowest setting.

There is strong evidence that DEGDA is superior to EGDmA in producing homogeneous monoliths capable of flow. Fortunately, DEGDA is the preferred monomer in this study because it is biocompatible. EGDmA can be excluded entirely from future experimentation.

The surface plots that have the largest areas of high probability are the plots that hold initiator constant at its minimum level and use DEGDA. Figure 5.1 demonstrates that increasing the level of salt seems to increase the area of high probability along the back of the region.

Also, if the model is accurate beyond the upper bound on surfactant used in this study, then the points with higher proportions of surfactant provide a large area of high probability to study in future experimentation. In addition, most of that same area would have high probability regardless of process variable settings. The accuracy of the model in these regions will be assessed in confirmatory experiments.

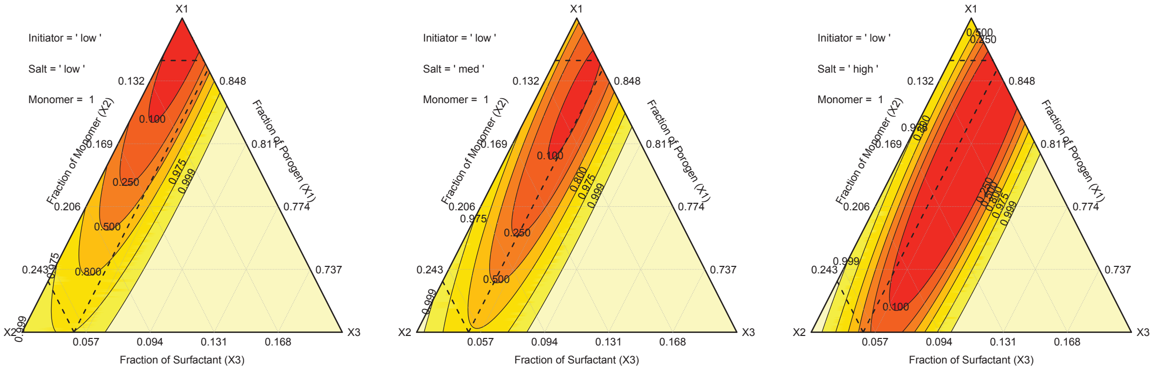

Figure 5.1: Surface plots holding initiator and mon2 constant. The amount of salt increases from left to right.

These future studies are discussed in the following sections, as well as a critical review of the methods used in this study. These sections are included to provide a framework for future studies that build upon the work done here.

5.1 Review of Methods

The purpose of this section is discuss any perceived shortcomings of this study. These are identified for the benefit of those who carry on this work in the future. Recommended changes to constraints, design generation, and experimental procedures are suggested.

Constraints

One of the interesting results of the fitted model is that it predicts an area of very high probability in regions with higher surfactant than what is permitted in this study. Porras et al. (2008) suggest higher amounts of surfactant are necessary for desirable emulsions. In hindsight, the proportion of surfactant is likely over-constrained. It is recommended that the bounds on surfactant be extended to 0.12 for future studies.

In the mean time, confirmatory experimental runs can explore this area to see if the extrapolated predictions in this area are accurate. If predictions are accurate, this region probably represents the largest region of high probability that future experimentation to optimize pore sizes can explore.

Design Generation

The mixture plots showing the design points in Figure 3.2 show that the algorithm in JMP for generating D-optimal design point seems to have largely ignored the lower area of the constrained design region. No reason for this phenomenon is offered, but in an effort to find an answer, a new design was created using the same inputs except for removing the split-plot arrangement. As Figure 5.2 illustrates, the new design placed all 66 runs in mostly the same areas as the design used in this study, so the split-plot arrangement is not causing this issue.

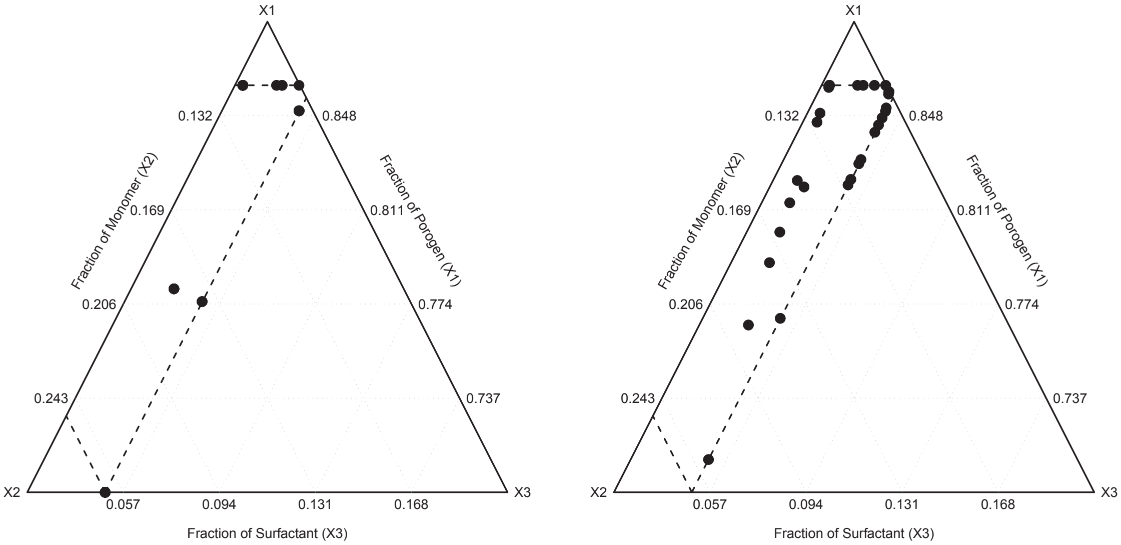

Figure 5.2: Design points generated without the split-plot structure are plotted on the left, and the design points used in this study are plotted on the right.

Unfortunately, the design points were not plotted until after the experiments had been completed. Otherwise, points in that lower region may have been added manually for better coverage. For future experimentation, it is recommended to plot the design points before moving forward with running the experiments. Also, D-optimal designs can be generating using the XVERT algorithm in SAS ADX or with the ‘Xvert’ function in the *mixexp* package in R. XVERT designs for the mixture components can be crossed with a separate design for the process variables, and the D-optimal subset of those points can be identified using the ‘optFederov’ function in the *AlgDesign* package in R.

Experimental Procedures

The experimental design in this study specifies filling only two columns with the same mixture in a whole plot, and to test one column at the low and high levels of temperature. This was specified under the impression that any additional runs inside a whole plot would take too much time. However, it was later learned that it would have been relatively easy to perform two or more runs at each temperature in a whole-plot, eliminating the problem of not being able to estimate the subplot error and increasing the total amount of data gathered from the study.

5.2 Future Work

This work is only an initial step toward the potential creation of the UHPLC-capable monolith column. The next steps are to validate the model estimated in this study through confirmatory experiments, model monolith morphology, and finally to design a study to minimize pore sizes.

Confirmatory Experiments

Confirmatory experiments for model validation are already underway. Ten design points have been selected from areas of low predicted probability and high predicted probability, including points with greater amounts of surfactant than allowed in this study.

The performance of the model will be evaluated by comparing the predicted and observed response, and by calculating the observed misclassification rate.

Studying Morphology

Studying the morphology of the monoliths had not been discussed initially. However, the chemist who carried out the experiments also recorded the morphology of every homogeneous monolith. A foam morphology is preferred, and so a similar analysis can be performed using a new binary response variable where homogenous monoliths with foam morphology and open pores is considered a success. The same analysis methods used in this study is recommended.

Minimizing Pore Sizes

Designing an experiment to minimize the pore sizes is the final and most important stage of this work. The purpose of these preliminary steps is to provide constraints for this design so that most, if not all, of the experimental runs will produce workable polymers so that a pore size measurement can be taken at each experimental run.

At the very least, the study to minimize pore sizes should provide an idea of the minimal pore size this process is capable of producing, and what process settings of the polymerization process will achieve that minimum.

In summary, the reality of time and budget constraints often means that the ideal experimental design is not an option. However, as this study shows, designs that accommodate compromises can still provide helpful information. Confirmatory experiments will dictate what recommendations are finally given for constraints used in future experiments; however, in the mean time, the results that have already been obtained provide a good idea about which process variables have a significant effect on the probability of producing homogenous monoliths capable of flow. The fitted model also provides rough understanding of the blending properties of the mixture components.

BIBLIOGRAPHY

Aggarwal, P., Tolley, H. D., and Lee, M. L. (2011), "Monolithic Bed Structures for Capillary Liquid Chromatography," *Journal of Chromatography A, to appear*, 1–14.

Agresti, A. (2002), *Categorical Data Analysis*, John Wiley & Sons, Inc.

Atkinson, A. C., and Donev, A. N. (1992), *Optimal Experimental Designs*, Oxford University Press.

Cornell, J. A. (2002), *Experiments with Mixtures* (3rd ed.), John Wiley & Sons, Inc.

Harwood, L. M., Moody, C. J., and Percy, J. (1999), *Experimental Organic Chemistry* (2nd ed.), Blackwell Science Ltd.

Hocking, R. R. (1976), "The Analysis and Selection of Variables in Linear Regression," *Biometrics*, 32, 1–49.

Jones, B., and Goos, P. (2007), "A Candidate-Set-Free Algorithm for Generating D-Optimal Split-Plot Designs," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 56, 347–364.

Kowalski, S., Cornell, J. A., and Vining, G. G. (2000), "A New Model and Class of Designs for Mixture Experiments with Process Variables," *Communications in Statistics – Theory and Methods*, 29, 2255–2280.

—— (2002), "Split-Plot Designs and Estimation Methods for Mixture Experiments with Process Variables," *Technometrics*, 44, 72–79.

Lawson, J. S. (2010), *Design and Analysis of Experiments with SAS*, Chapman & Hall.

—— (2011), *mixexp: Design and analysis of mixture experiments*, r package version 0.5-1.

McLean, R. A., and Anderson, V. L. (1966), "Extreme Vertices of Mixture Experiments," *Technometrics*, 8, 447–454.

Nelder, J. A., and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society. Series A (General)*, 135, 370–384.

Porras, M., Solans, C., González, C., and Guti'errez, J. M. (2008), "Properties of Water-in-Oil (W/O) Nano-Emulsions Prepared by a Low-Energy Emulsification Method," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 324, 181–188.

SAS Institute (2010), *SAS/STAT 9.22 User's Guide*, SAS Institute.

—— (2012), "Usage Note 22599: Understanding and Correcting Complete or Quasi-complete Separation Problems," `http://support.sas.com/kb/22/599.html`.

Scheffé, H. (1958), "Experiments with Mixtures," *Journal of the Royal Statistical Society. Series B (Methodological)*, 20, 344–360.

Snee, R. D., and Marquardt, D. W. (1974), "Extreme Vertices Designs for Linear Mixture Models," *Technometrics*, 16, 399–408.

Tunç, Y., et al. (2010), "Acrylic-Based High Internal Phase Emulsion Polymeric Monolith for Capillary Electrochromatography," *Journal of Chromatography A*, 1217, 1654–1659.

APPENDICES

APPENDIX A

JMP® SCRIPT

```
DOE(
Custom Design,
{Add Response( Maximize, "Solid", ., ., . ),
Add Response( Maximize, "Flow", ., ., . ),
Add Factor( Mixture, 0.7, 0.86, "Porogen", 1 ),
Add Factor( Mixture, 0.095, 0.25, "Monomer", 1 ),
Add Factor( Mixture, 0.02, 0.05, "Surfactant", 1 ),
Add Factor( Continuous, 0.2, 10, "Initiator", 1 ),
Add Factor( Continuous, 0.1, 2, "Salt", 1 ),
Add Factor( Continuous, 45, 75, "Mix Time", 1 ),
Add Factor( Continuous, 55, 65, "Heat", 0 ),
Add Factor( Categorical, {"L1", "L2"}, "Mon1or2", 1 ), Set Random Seed( 309862 ),
Number of Starts( 1 ), Add Constraint(
[0 -0.5 1 0 0 0 0 0, 0 0.2 -1 0 0 0 0 0]
), Add Term( {1, 1} ), Add Term( {2, 1} ), Add Term( {3, 1} ),
Add Term( {1, 1}, {2, 1} ), Add Term( {1, 1}, {3, 1} ),
Add Term( {2, 1}, {3, 1} ), Add Term( {1, 1}, {2, 1}, {3, 1} ),
Add Term( {1, 1}, {2, 1} ), Add Term( {1, 1}, {3, 1} ),
Add Term( {2, 1}, {3, 1} ), Add Term( {1, 1}, {4, 1} ),
Add Term( {1, 1}, {5, 1} ), Add Term( {1, 1}, {6, 1} ),
Add Term( {1, 1}, {7, 1} ), Add Term( {1, 1}, {8, 1} ),
Add Term( {2, 1}, {4, 1} ), Add Term( {2, 1}, {5, 1} ),
Add Term( {2, 1}, {6, 1} ), Add Term( {2, 1}, {7, 1} ),
Add Term( {2, 1}, {8, 1} ), Add Term( {3, 1}, {4, 1} ),
Add Term( {3, 1}, {5, 1} ), Add Term( {3, 1}, {6, 1} ),
Add Term( {3, 1}, {7, 1} ), Add Term( {3, 1}, {8, 1} ),
Add Term( {4, 1}, {5, 1} ), Add Term( {4, 1}, {6, 1} ),
Add Term( {4, 1}, {7, 1} ), Add Term( {4, 1}, {8, 1} ),
Add Term( {5, 1}, {6, 1} ), Add Term( {5, 1}, {7, 1} ),
Add Term( {5, 1}, {8, 1} ), Add Term( {6, 1}, {7, 1} ),
Add Term( {6, 1}, {8, 1} ), Add Term( {7, 1}, {8, 1} ), Add Term( {4, 2} ),
Add Term( {5, 2} ), Add Term( {6, 2} ), Add Alias Term( {1, 1}, {2, 1} ),
Add Alias Term( {1, 1}, {3, 1} ), Add Alias Term( {1, 1}, {4, 1} ),
Add Alias Term( {1, 1}, {5, 1} ), Add Alias Term( {1, 1}, {6, 1} ),
Add Alias Term( {1, 1}, {7, 1} ), Add Alias Term( {2, 1}, {3, 1} ),
Add Alias Term( {2, 1}, {4, 1} ), Add Alias Term( {2, 1}, {5, 1} ),
Add Alias Term( {2, 1}, {6, 1} ), Add Alias Term( {2, 1}, {7, 1} ),
Add Alias Term( {3, 1}, {4, 1} ), Add Alias Term( {3, 1}, {5, 1} ),
Add Alias Term( {3, 1}, {6, 1} ), Add Alias Term( {3, 1}, {7, 1} ),
Add Alias Term( {4, 1}, {5, 1} ), Add Alias Term( {4, 1}, {6, 1} ),
Add Alias Term( {4, 1}, {7, 1} ), Add Alias Term( {5, 1}, {6, 1} ),
Add Alias Term( {5, 1}, {7, 1} ), Add Alias Term( {6, 1}, {7, 1} ),
Add Alias Term( {1, 1}, {8, 1} ), Add Alias Term( {2, 1}, {8, 1} ),
Add Alias Term( {3, 1}, {8, 1} ), Add Alias Term( {4, 1}, {8, 1} ),
Add Alias Term( {5, 1}, {8, 1} ), Add Alias Term( {6, 1}, {8, 1} ),
Add Alias Term( {7, 1}, {8, 1} ), Set Sample Size( 66 ), Set N Whole Plots( 33 ),
Make Design, Make Table}
);
```

APPENDIX B

SAS$^{®}$ CODE

```
PROC IMPORT OUT= SASUSER.Phase1
            DATAFILE= "C:\Users\Cameron\Dropbox\MastersProject\Data\Results_Phase_1_clean2.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
PROC LOGISTIC DATA=SASUSER.PHASE1;
CLASS WHOLEPLOTS;
MODEL COMBINED(DESC)= WHOLEPLOTS HEAT/firth;
RUN;
PROC GLIMMIX DATA=SASUSER.PHASE1;
CLASS WHOLEPLOTS;
MODEL COMBINED(DESC)=HEAT / S DIST=BIN;
RANDOM WHOLEPLOTS;
RUN;
PROC LOGISTIC DATA=SASUSER.PHASE1;
CLASS IS_MON_1;
MODEL COMBINED(DESC)=POROGEN MONOMER SURFACTANT
POROGEN*MONOMER POROGEN*SURFACTANT MONOMER*SURFACTANT
INITIATOR*POROGEN INITIATOR*MONOMER INITIATOR*SURFACTANT
SALT*POROGEN SALT*MONOMER SALT*SURFACTANT
MIX_TIME*POROGEN MIX_TIME*MONOMER MIX_TIME*SURFACTANT
IS_MON_1*POROGEN IS_MON_1*MONOMER IS_MON_1*SURFACTANT
HEAT*POROGEN HEAT*MONOMER HEAT*SURFACTANT/NOINT SCALE=P
SELECTION=FORWARD START=6 SLENTRY=0.2;
RUN;
/*USES FORWARD STEPWISE STARTING AFTER PURE MIXTURE EFFECTS WITH ALPHA=0.20*/


/*Compare Logit link to probit and cloglog*/
PROC LOGISTIC DATA=SASUSER.PHASE1_WP DESC;
CLASS IS_MON_1;
MODEL R/N=POROGEN MONOMER SURFACTANT POROGEN*MONOMER POROGEN*SURFACTANT MONOMER*SURFACTANT
INITIATOR*POROGEN INITIATOR*MONOMER INITIATOR*SURFACTANT
SALT*POROGEN SALT*MONOMER SALT*SURFACTANT MIX_TIME*POROGEN MIX_TIME*MONOMER MIX_TIME*SURFACTANT
IS_MON_1*POROGEN IS_MON_1*MONOMER IS_MON_1*SURFACTANT /NOINT LINK=PROBIT
SELECTION=FORWARD START=6 SLENTRY=0.2;
RUN;

PROC LOGISTIC DATA=SASUSER.PHASE1_WP DESC;
CLASS IS_MON_1;
MODEL R/N=POROGEN MONOMER SURFACTANT POROGEN*MONOMER POROGEN*SURFACTANT MONOMER*SURFACTANT
INITIATOR*POROGEN INITIATOR*MONOMER INITIATOR*SURFACTANT
```

```
SALT*POROGEN SALT*MONOMER SALT*SURFACTANT MIX_TIME*POROGEN MIX_TIME*MONOMER MIX_TIME*SURFACTANT
IS_MON_1*POROGEN IS_MON_1*MONOMER IS_MON_1*SURFACTANT /NOINT LINK=CLOGLOG
SELECTION=FORWARD START=6 SLENTRY=0.2;
RUN;

ODS LISTING CLOSE;
ODS LATEX GPATH='C:\Users\Cameron\Dropbox\MastersProject\Plots\DiagPlots';
ods graphics on;
PROC LOGISTIC DATA=SASUSER.PHASE1_WP DESC PLOTS=INFLUENCE(UNPACK);
CLASS IS_MON_1;
MODEL R/N=POROGEN MONOMER SURFACTANT POROGEN*MONOMER POROGEN*SURFACTANT MONOMER*SURFACTANT
INITIATOR*MONOMER INITIATOR*SURFACTANT SALT*POROGEN SALT*SURFACTANT
IS_MON_1*MONOMER IS_MON_1*SURFACTANT /NOINT SCALE=D;
ODS SELECT WHERE = (lowcase(_label_) ? 'pearson') GlobalTests GoodnessOfFit;
RUN;
ODS LATEX CLOSE;
ODS LISTING;

ODS LISTING CLOSE;
ODS PDF FILE='C:\Users\Cameron\Dropbox\MastersProject\Plots\Deviance.pdf';
ods graphics on;
PROC LOGISTIC DATA=SASUSER.PHASE1_WP DESC;
CLASS IS_MON_1;
MODEL R/N=POROGEN MONOMER SURFACTANT POROGEN*MONOMER POROGEN*SURFACTANT MONOMER*SURFACTANT
INITIATOR*MONOMER INITIATOR*SURFACTANT SALT*POROGEN SALT*SURFACTANT
IS_MON_1*MONOMER IS_MON_1*SURFACTANT /NOINT SCALE=D;
RUN;
ODS PDF CLOSE;
ODS LISTING;

ODS LISTING CLOSE;
ODS LATEX GPATH='C:\Users\Cameron\Dropbox\MastersProject\Plots\DiagPlots';
ods graphics on;
PROC LOGISTIC DATA=SASUSER.PHASE1_WP DESC PLOTS=DFBETAS(UNPACK);
CLASS IS_MON_1;
MODEL R/N=POROGEN MONOMER SURFACTANT POROGEN*MONOMER POROGEN*SURFACTANT MONOMER*SURFACTANT
INITIATOR*MONOMER INITIATOR*SURFACTANT SALT*POROGEN SALT*SURFACTANT
IS_MON_1*MONOMER IS_MON_1*SURFACTANT /NOINT SCALE=D;
ODS SELECT WHERE = (lowcase(_label_) ? 'porogensurfactant')
WHERE = (lowcase(_label_) ? 'monomersurfactant');
RUN;
ODS LATEX CLOSE;
ODS LISTING;
```

APPENDIX C

R CODE

C.1 Mixture Plot Function

```
MixturePlot2 = function(x=NULL,y=NULL,z=NULL,w=NULL,des=NULL,
                        res=400,lims=c(rep(0,6)),color.palette = heat.colors,
                        constrts=FALSE,contrs=TRUE,n.breaks=10,levels=NULL,
                        cols=FALSE, despts=TRUE, mod=NA,x3lab="Fraction X3",
                        x2lab="Fraction X2", x1lab="Fraction X1",
                        corner.labs = c("X3", "X2", "X1"),
                        colorkey=list(dx=0.04,x0=0.95,y0=0.45,y1=0.90,add=TRUE,mode="all"),
                        pseudo=FALSE,user.func=NULL,...)
{
##############################################
#Argument list
# des design matrix with known points (data frame)
# x, y, z locations for known points
# w values at x,y,z locations
# res number of color blocks between 0 and 1 of x
# lims vector of lower and upper constraints for x1,x2,x3
# constrts if TRUE constraints found in lims will be added to the graph
# contrs if TRUE contour lines will be added to the graph
# n.breaks number of breaks between levels (used for contours if 'levels' not specified)
# levels takes a list of contour levels (e.g. levels=c(1,3,5,10) will draw contours at those heights)
# cols if TRUE regions between contour lines will be colored
# despts if TRUE plots the design points in data frame des or vectors x, y, z
# color.palette is the color palette to use.
# mod is an indicator for the model 1=linear, 2=quadratic, 3=special cubic, NA=user specified function
#  in user.func
# x3lab label for the x3 axis
# x2lab label for the x2 axis
# x1lab label for the x1 axis
# corner.labs labels for x3, x2 and x1 vertices
# colorkey  list with locations of the color key
# psuedo if TRUE uses psuedo components to zoom in on constrained region.  Will create the smallest
# equilateral triangle that still contains the whole constrained region.
# user.func is a function supplied by the user that takes as arguments a dataframe called 'grid'
#  containing columns 'x', 'y', and 'z' and returns a predicted 'w' for each row in 'grid'.
# ... additional arguments for user.func besides 'grid'


if (is.null(des)){
  if (is.null(x))
      stop("There must be a data frame containing the design, or vectors of known points")
```

```
                 }  else {
x<-des$x3
y<-des$x2
z<-des$x1
w<-des$y
                         }


#Depends on the following libraries
library(lattice)
library(grid)
######################
## Creation of Grid ##
######################
trian <- expand.grid(base=seq(0,1,l=res), high=seq(0,sin(pi/3), l=res))#87
trian <- subset(trian, (base*sin(pi/3)*2)>high)
trian <- subset(trian, ((1-base)*sin(pi/3)*2)>high)
new2 <- data.frame(z=trian$high*2/sqrt(3))
new2$x <- trian$base-trian$high/sqrt(3)
new2$y <- 1 - new2$x-new2$z


if(pseudo){
l.bnds <- lims[seq(1,5,by=2)]
sum.bnds <- sum(l.bnds)
new2$x <- l.bnds[3]+(1-sum.bnds)*new2$x
new2$y <- l.bnds[2]+(1-sum.bnds)*new2$y
new2$z <- l.bnds[1]+(1-sum.bnds)*new2$z
x.pseudo <- (x-l.bnds[3])/(1-sum.bnds)
y.pseudo <- (y-l.bnds[2])/(1-sum.bnds)
z.pseudo <- (z-l.bnds[1])/(1-sum.bnds)
}

if(is.na(mod)==FALSE){
## Fit the data to a model
if (mod==1) {
## This is the Scheffe Linear model
fm1 = lm(w~x+y+z-1)
}
if (mod==2) {
## This is the Scheffe Quadratic model
fm1 = lm(w~x+y+z+x*y+x*z+y*z-1)
}
if (mod==3) {
## This is the Scheffe Special Cubic Model
fm1 = lm(w~x+y+z+x*y+x*z+y*z+x*y*z-1)
}
}
## Create a new dataset using the model
if (is.na(mod)==TRUE){
if(is.null(user.func)==TRUE){
stop("There must be a model specified or a user supplied function for predictions on the simplex")
}else{
trian$w=user.func(grid=new2,...) #add ... to argument when creating the function
}
}else{
trian$w = predict(fm1, newdata=data.frame(new2))
}

## Function for laying out barycentric coordinates
grade.trellis <- function(from=0.2, to=0.8, step=0.2, col=1, lty=3, lwd=.5){
if (constrts) {
#Constraints on x1
  f1<-lims[1]
  t1<-lims[2]
  s1=t1-f1
  x1 <- seq(f1, t1, s1)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x2, y2, 1-x2, y2, col=col, lty=2, lwd=2.0)
#Constraints on x2 (note backwards f2-1-upper, t2=1-lower
  f2<-1-lims[4]
  t2<-1-lims[3]
  s2<-t2-f2
  x1 <- seq(f2, t2, s2)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x1, 0, x2, y2, col=col, lty=2, lwd=2.0)
#Constraints on x3
  f3<-lims[5]
  t3<-lims[6]
  s3<-t3-f3
  x1 <- seq(f3, t3, s3)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x1, 0, x3, y3, col=col, lty=2, lwd=2.0)
}
#Grid lines
  x1 <- seq(from, to, step)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x1, 0, x2, y2, col="darkgrey", lty=lty, lwd=lwd)
  panel.text(x1, 0, label=x1, pos=1)
  panel.segments(x1, 0, x3, y3, col="darkgrey", lty=lty, lwd=lwd)
  panel.text(x2, y2, label=rev(x1), pos=2)
  panel.segments(x2, y2, 1-x2, y2, col="darkgrey", lty=lty, lwd=lwd)
  panel.text(x3, y3, label=rev(x1), pos=4)

}
```

```
grade.trellis.pseudo <- function(from=0.2, to=0.8, step=0.2, col=1, lty=3, lwd=.5){
#Constraints on x1
  x1 <- (lims[2]-l.bnds[1])/(1-sum.bnds)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x2, y2, 1-x2, y2, col=col, lty=2, lwd=2.0)
#Constraints on x2 (note backwards f2-1-upper, t2=1-lower
  x1 <- 1-(lims[4]-l.bnds[2])/(1-sum.bnds)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x1, 0, x2, y2, col=col, lty=2, lwd=2.0)
#Constraints on x3
  f3<-lims[5]
  t3<-lims[6]
  s3<-t3-f3
  x1 <- (lims[6]-l.bnds[3])/(1-sum.bnds)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  panel.segments(x1, 0, x3, y3, col=col, lty=2, lwd=2.0)
#Grid lines
  x1 <- seq(from, to, step)
  x2 <- x1/2
  y2 <- x1*sqrt(3)/2
  x3 <- (1-x1)*0.5+x1
  y3 <- sqrt(3)/2-x1*sqrt(3)/2
  labx1 <- l.bnds[3]+(1-sum.bnds)*x1
  labx2 <- l.bnds[2]+(1-sum.bnds)*x1
  labx3 <- l.bnds[1]+(1-sum.bnds)*x1
  panel.segments(x1, 0, x2, y2, col="darkgrey", lty=lty, lwd=lwd)
  panel.text(x1, 0, label=labx1, pos=1)
  panel.segments(x1, 0, x3, y3, col="darkgrey", lty=lty, lwd=lwd)
  panel.text(x2, y2, label=rev(labx2), pos=2)
  panel.segments(x2, y2, 1-x2, y2, col="darkgrey", lty=lty, lwd=lwd)
  panel.text(x3, y3, label=rev(labx3), pos=4)

}

## Perform the actual plotting
if(is.null(levels)){
p <- levelplot(w~base*high, trian, aspect="iso", xlim=c(-0.1,1.1), ylim=c(-0.1,0.96),
          x3lab=NULL, x2lab=NULL, contour=contrs, cuts=n.breaks, labels=TRUE, pretty=TRUE, region=cols,
          col.regions = color.palette(n=n.breaks+1), cex.lab=1.3,
          par.settings=list(axis.line=list(col=NA), axis.text=list(col=NA)),
          panel=function(..., at=pretty(trian$w,n=11), contour=TRUE, labels=pretty(trian$w,n=11)){
                          panel.levelplot(..., at=pretty(trian$w,n=11), contour=TRUE,
                          labels= pretty(trian$w,n=11),
                                          lty=2, lwd=0.5, col=1)}
          )
}else{
p <- levelplot(w~base*high, trian, aspect="iso", at=levels, xlim=c(-0.1,1.1), ylim=c(-0.1,0.96),
          x3lab=NULL, x2lab=NULL, contour=contrs, labels=TRUE, pretty=TRUE, region=cols,
          col.regions = color.palette(n=n.breaks+1), cex.lab=1.3,
          par.settings=list(axis.line=list(col=NA), axis.text=list(col=NA)),
          panel=function(..., at=pretty(trian$w,n=11), contour=TRUE, labels=pretty(trian$w,n=11)){
                          panel.levelplot(..., at=pretty(trian$w,n=11), contour=TRUE,
                          labels=pretty(trian$w,n=11),
                                          lty=2, lwd=0.5, col=1)}
          )
}
##labels and legend
grid.newpage()
pushViewport(viewport(xscale = p$x.limits, yscale = p$y.limits))
do.call(panel.levelplot, trellis.panelArgs(p, 1))
## update the trellis panel
#trellis.focus("panel", 1, 1, highlight=TRUE)
panel.segments(c(0,0,0.5), c(0,0,sqrt(3)/2), c(1,1/2,1), c(0,sqrt(3)/2,0),lwd=2)
if(pseudo){
grade.trellis.pseudo()
}else{
grade.trellis()
}
panel.text(0, 0, label=corner.labs[2], pos=2)
panel.text(1/2, sqrt(3)/2, label=corner.labs[3], pos=3)
# This point is x1 vertex
#panel.points(1/2,sqrt(3)/2,col="black",cex=1.4,pch=19)
panel.text(1, 0, label=corner.labs[1], pos=4)
panel.text(.5,-.075,x3lab)
panel.text(.18,.5,x2lab,srt=60)
panel.text(.82,.5,x1lab, srt=-60)

if (despts) {
if(pseudo){
# Plot the design points
# using the transformation x=z*sqrt(3)/4+x+.065*z, y=z*sqrt(3)/2
xpts<-(z.pseudo*sqrt(3)/4)+.065*z.pseudo+x.pseudo
ypts<-z.pseudo*sqrt(3)/2
panel.points(xpts,ypts,pch=19,cex=1.4,col="black")
}else{
xpts<-(z*sqrt(3)/4)+.065*z+x
ypts<-z*sqrt(3)/2
panel.points(xpts,ypts,pch=19,cex=1.4,col="black")
}
}

ck.x=colorkey$x0
ck.y.b=colorkey$y0 #.45
ck.y.t=colorkey$y1 #.90
ck.y = seq(ck.y.b,ck.y.t,len=n.breaks+2)
d.x = colorkey$dx
d.y = diff(ck.y[1:2])
}
#Function over
```

C.2 User-defined Function with Final Model

```
mymod=function(grid,init='low',salt='low',mon=1,logodds=FALSE){
x=grid$x #surf
y=grid$y #mon
z=grid$z #por
if(init=='low'){
intr=0.002*y
}
if(init=='med'){
intr=0.05*y
}
if(init=='high'){
intr=0.1*y
}
if(salt=='low'){
slt=0.001*z
}
if(salt=='med'){
slt=0.0105*z
}
if(salt=='high'){
slt=0.02*z
}
w2=1.0996*z+221.7*y+5040*x-245.7*z*y-5267.6*z*x-
6148*x*y-2628.7*intr*y + 11192.7*intr*x + 39.6059*(1-mon)*y -
105.1*(1-mon)*x + 800.3*slt*z - 17525.3*slt*x
w=(1/(1+exp(-w2)))
if(logodds==TRUE){
return(w2)
}else{
return(w)
}
}
```

C.3 Code to Create All Mixture Plots

```
setwd("/Users/cameronwillden/Dropbox/MastersProject")
data=read.csv("Data/Results_Phase_1_clean2.csv")
z=data$Porogen
y=data$Monomer
x=data$Surfactant
w=data$Combined

levels=c('low','med','high')
for(i in 1:2){
for(j in 1:3){
for(k in 1:3){
```

```
pdf(paste(’Plots/MixPlots/Mon’,i,’I_’,j,’S_’,k,’.pdf’,sep=""),6,6)
MixturePlot2(x,y,z,user.func=mymod,lims=c(.7,.86,0.095,0.25,0.02,0.05),
constrts=TRUE,pseudo=TRUE,cols=T, despts=FALSE,x1lab="Fraction of Porogen (X1)",
x2lab="Fraction of Monomer (X2)",x3lab="Fraction of Surfactant (X3)",
levels=c(-1,0,0.1,0.25,0.5,0.8,0.975,0.999,1.001),
init=levels[j],salt=levels[k],mon=i)

grid::grid.text(paste("Initiator = ’",levels[j],"’"), x=unit(0.1, "npc"),
y=unit(0.86, "npc"),just=’left’)
grid::grid.text(paste("Salt = ’",levels[k],"’"), x=unit(0.1, "npc"),y=unit(0.78, "npc"),just=’left’)
grid::grid.text(paste("Monomer = ",i), x=unit(0.1, "npc"), y=unit(0.70, "npc"),just=’left’)
dev.off()
}
}
}

###Show Constraints Points
pdf("ConstraintsPseudo.pdf",7,7)
MixturePlot2(x,y,z,n.breaks=0,user.func=mymod,lims=c(.7,.86,0.095,0.25,0.02,0.05),
constrts=TRUE,pseudo=TRUE,cols=F,levels=c(-1,2),despts=F,x1lab="Fraction of Porogen (X1)",
x2lab="Fraction of Monomer (X2)",x3lab="Fraction of Surfactant (X3)",
init=levels[1],mon=1)
dev.off()

pdf("Constraints.pdf",7,7)
MixturePlot2(x,y,z,n.breaks=0,user.func=mymod,lims=c(.7,.86,0.095,0.25,0.02,0.05),
constrts=TRUE,pseudo=FALSE,cols=F,levels=c(-1,2),despts=F,x1lab="Fraction of Porogen (X1)",
x2lab="Fraction of Monomer (X2)",x3lab="Fraction of Surfactant (X3)",
init=levels[1],mon=1)
dev.off()


###Show Design Points
pdf("DesPointsPseudo.pdf",7,7)
MixturePlot2(x,y,z,n.breaks=0,user.func=mymod,lims=c(.7,.86,0.095,0.25,0.02,0.05),
constrts=TRUE,pseudo=TRUE,cols=F,levels=c(-1,2),despts=T,x1lab="Fraction of Porogen (X1)",
x2lab="Fraction of Monomer (X2)",x3lab="Fraction of Surfactant (X3)",
init=levels[1],mon=1)
dev.off()

pdf("DesPoints.pdf",7,7)
MixturePlot2(x,y,z,n.breaks=0,user.func=mymod,lims=c(.7,.86,0.095,0.25,0.02,0.05),
constrts=TRUE,pseudo=FALSE,cols=F,levels=c(-1,2),despts=T,x1lab="Fraction of Porogen (X1)",
x2lab="Fraction of Monomer (X2)",x3lab="Fraction of Surfactant (X3)",
init=levels[1],mon=1)
dev.off()
```

APPENDIX D

DESIGN POINTS AND RESULTS

Run	Whole Plot	Porogen	Monomer	Surfactant	Initiator	Salt	Mix Time	Heat	Is.Mon.1	Combined
1	1	0.86	0.10	0.04	0.01	0.00	45	65	0	1
2	1	0.86	0.10	0.04	0.01	0.00	45	55	0	1
3	2	0.82	0.13	0.05	0.00	0.00	75	65	0	1
4	2	0.82	0.13	0.05	0.00	0.00	75	55	0	0
5	3	0.86	0.12	0.02	0.00	0.00	75	65	1	1
6	3	0.86	0.12	0.02	0.00	0.00	75	55	1	1
7	4	0.86	0.12	0.02	0.01	0.02	75	65	1	1
8	4	0.86	0.12	0.02	0.01	0.02	75	55	1	1
9	5	0.81	0.16	0.03	0.00	0.02	75	55	0	0
10	5	0.81	0.16	0.03	0.00	0.02	75	65	0	0
11	6	0.86	0.10	0.05	0.01	0.02	45	55	1	0
12	6	0.86	0.10	0.05	0.01	0.02	45	65	1	0
13	7	0.71	0.24	0.05	0.02	0.00	75	55	0	0
14	7	0.71	0.24	0.05	0.02	0.00	75	65	0	0
15	8	0.85	0.13	0.03	0.00	0.02	45	55	1	1
16	8	0.85	0.13	0.03	0.00	0.02	45	65	1	1
17	9	0.86	0.10	0.05	0.01	0.00	75	55	1	1
18	9	0.86	0.10	0.05	0.01	0.00	75	65	1	1
19	10	0.82	0.15	0.03	0.00	0.00	45	55	0	0
20	10	0.82	0.15	0.03	0.00	0.00	45	65	0	0
21	11	0.86	0.10	0.04	0.00	0.02	75	65	0	1
22	11	0.86	0.10	0.04	0.00	0.02	75	55	0	0

Run	Whole Plot	Porogen	Monomer	Surfactant	Initiator	Salt	Mix Time	Heat	Is.Mon.1	Combined
23	12	0.83	0.12	0.05	0.01	0.00	45	55	1	1
24	12	0.83	0.12	0.05	0.01	0.00	45	65	1	1
25	13	0.84	0.11	0.05	0.01	0.01	75	55	0	0
26	13	0.84	0.11	0.05	0.01	0.01	75	65	0	1
27	14	0.86	0.10	0.05	0.00	0.00	60	55	0	0
28	14	0.86	0.10	0.05	0.00	0.00	60	65	0	0
29	15	0.85	0.10	0.05	0.01	0.02	60	55	0	1
30	15	0.85	0.10	0.05	0.01	0.02	60	65	0	0
31	16	0.80	0.16	0.03	0.02	0.00	45	55	0	0
32	16	0.80	0.16	0.03	0.02	0.00	45	65	0	0
33	17	0.85	0.10	0.05	0.00	0.01	45	55	1	0
34	17	0.85	0.10	0.05	0.00	0.01	45	65	1	0
35	18	0.83	0.12	0.05	0.00	0.02	75	65	1	0
36	18	0.83	0.12	0.05	0.00	0.02	75	55	1	0
37	19	0.86	0.12	0.02	0.00	0.02	45	65	0	0
38	19	0.86	0.12	0.02	0.00	0.02	45	55	0	0
39	20	0.86	0.12	0.02	0.01	0.00	75	55	0	0
40	20	0.86	0.12	0.02	0.01	0.00	75	65	0	0
41	21	0.86	0.12	0.02	0.01	0.00	45	65	1	0
42	21	0.86	0.12	0.02	0.01	0.00	45	55	1	0
43	22	0.86	0.12	0.02	0.00	0.01	75	65	0	0
44	22	0.86	0.12	0.02	0.00	0.01	75	55	0	0

Run	Whole Plot	Porogen	Monomer	Surfactant	Initiator	Salt	Mix Time	Heat	Is.Mon.1	Combined
45	23	0.79	0.17	0.03	0.00	0.00	75	55	1	1
46	23	0.79	0.17	0.03	0.00	0.00	75	65	1	1
47	24	0.86	0.10	0.04	0.00	0.00	45	55	1	0
48	24	0.86	0.10	0.04	0.00	0.00	45	65	1	0
49	25	0.84	0.11	0.05	0.01	0.00	45	65	0	1
50	25	0.84	0.11	0.05	0.01	0.00	45	55	0	1
51	26	0.77	0.20	0.04	0.02	0.02	75	65	1	1
52	26	0.77	0.20	0.04	0.02	0.02	75	55	1	1
53	27	0.85	0.10	0.05	0.01	0.02	75	65	0	0
54	27	0.85	0.10	0.05	0.01	0.02	75	55	0	0
55	28	0.82	0.13	0.05	0.00	0.02	45	55	0	0
56	28	0.82	0.13	0.05	0.00	0.02	45	65	0	1
57	29	0.85	0.13	0.03	0.01	0.02	45	55	0	0
58	29	0.85	0.13	0.03	0.01	0.02	45	65	0	0
59	30	0.86	0.12	0.02	0.01	0.02	75	65	0	1
60	30	0.86	0.12	0.02	0.01	0.02	75	55	0	1
61	31	0.82	0.15	0.03	0.01	0.00	75	65	1	0
62	31	0.82	0.15	0.03	0.01	0.00	75	55	1	0
63	32	0.77	0.18	0.05	0.00	0.02	45	55	1	0
64	32	0.77	0.18	0.05	0.00	0.02	45	65	1	0
65	33	0.86	0.11	0.03	0.01	0.01	60	65	1	0
66	33	0.86	0.11	0.03	0.01	0.01	60	55	1	0