2007-02-16

Analysis Using Smoothing Via Penalized Splines as Implemented in LME() in R

John R. Howell
Brigham Young University - Provo

ANALYSIS USING SMOOTHING SPLINES AS IMPLEMENTED IN LME() IN R

by

John R. Howell

A project submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Statistics

Brigham Young University

April 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a project submitted by

John R. Howell

This project has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Gilbert W. Fellingham, Chair
Date	G. Bruce Schaalje
Date	Del T. Scott

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the project of John R. Howell in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date	Gilbert W. Fellingham Chair, Graduate Committee

Accepted for the Department

Date	Scott D. Grimshaw Graduate Coordinator

Accepted for the College

Date	Thomas W. Sederberg Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

ANALYSIS USING SMOOTHING SPLINES AS IMPLEMENTED IN LME() IN R

John R. Howell

Department of Statistics

Master of Science

Spline smoothers as implemented in common mixed model software provide a familiar framework for estimating semi-parametric and non-parametric models. Following a review of literature on splines and mixed models, details for implementing mixed model splines are presented. The examples use an experiment in the health sciences to demonstrate how to use mixed models to generate the smoothers. The first example takes a simple one-group case, while the second example fits an expanded model using three groups simultaneously. The second example also demonstrates how to fit confidence bands to the three-group model. The examples use mixed model software as implemented in lme() in R. Following the examples a discussion of the method is presented.

ACKNOWLEDGEMENTS

Many thanks to my always supportive wife Jennifer. I also wish to extend my heartfelt appreciation to my advisor Dr. Fellingham, and the members of my graduate committee, Dr. Scott, and Dr. Schaalje, for the many suggestions they have made that made this possible.

CONTENTS

TABLES

Table

FIGURES

Figure

1. INTRODUCTION

Smoothing splines offer more flexibility than traditional polynomial regression for fitting non-linear and non-polynomial relationships. In this project, smoothing splines are fit to experimental data from the health sciences. The data consist of tendon temperatures taken every 30 seconds over a 55-minute treatment period from subjects in three separate treatment groups. The temperature was measured by a thermometer surgically embedded in the Achilles tendon of each subject. The experimental groups consist of a control (no intervention) and two treatments. Both treatments include a 5-minute pre-treatment measurement period followed by: (1) applying an icepack to the tendon for 20 minutes followed by 30 minutes of post-treatment measurement, and (2) immersing the tendon in a whirlpool ice bath for 20 minutes followed by 30 minutes of post-treatment measurement. Following a review of literature, a description on how to analyze the experiment using smoothing splines implemented in popular mixed model software will be given.

Eilers and Marx (1996) proposed the technique of P-Splines, or penalized splines, a method of fitting a smoothing spline using knots and simple penalties. Prior to Eilers and Marx, penalty methods used an integral of the second derivatives as a smoothing penalty. Eilers and Marx proposed using a simple difference as the penalty. They claimed that this technique retained all the benefits of the integral methods, but had the added benefit of being less computationally expensive. Ngo and Wand (2004) demonstrated a method using existing mixed model software in `S-Plus`® and `SAS`® that retained the use of knots, but controlled the smoothing using the shrinkage property of mixed models instead of an externally defined penalty.

Smoothing spline techniques have been well developed, but not widely implemented by researchers in disciplines outside of probability and statistics. This project

is an introduction to smoothing splines for practicioners in the health sciences. It assumes no prior knowledge of penalized splines, but a basic knowledge of R (2004) and mixed models is beneficial. The project will help researchers deal with longitudinal data collected by health science researchers that is not amenable to analysis using regular parametric methods.

This paper shows how to apply splines to the tendon temperature data. Chapter 2 is a review of literature containing a brief history of literature concerning smoothing and an examination of P-splines. Chapter 3 outlines the proposed scope of the project. In Chapter 4, results of the analysis are presented along with the key portions of the computer code necessary to obtain the results. Chapter 5 concludes by reviewing the strengths and weaknesses of the method and identifying pitfalls that were encountered when implementing the model. An appendix is also included which contains the complete computer programs in `R`.

2. REVIEW OF LITERATURE

2.1 Early Literature

Smoothing splines are non-parametric or semi-parametric techniques for fitting smooth curves to data. Named after the tool used by ship builders to create the curve of the hull, a mathematical spline is a series of piecewise polynomials with smooth joints. Splines first appeared in actuarial literature near the turn of the twentieth century. Spencer's paper "On the graduation of rates of sickness and mortality" (1904) is the earliest reference to smoothing as a mathematical technique. Whittacker proposed a new method of smoothing data in 1923. Whittacker's paper is the first that used probability tools for fitting a smooth curve. He also used a roughness penalty to control the amount of smoothing. The use of this penalty was later refined and used in many forms of splines, including P-Splines.

The term "spline" was coined by Schoenberg to describe the process of fitting a smooth function to data points (Schoenberg 1946; Epperson 1998). While working for the U.S. Army at the Ballistics Research Laboratory during World War II, he was assigned to determine the trajectory of munitions shells. The spatial nature of the data prompted him to develop basis functions for piecewise polynomial fits. The application was computationally intensive and not practical for widespread use at the time. Schoenberg spent much of his later career fine-tuning and expanding his initial work. He is considered to be the father of B-splines or basis splines (covered in section 2.2) and the many applications which stem from them (Karlin 1973; Schoenberg 1988; Davis 1996).

Due to insufficient computing power and unstable boundary estimates, B-splines were not widely used until the 1970's. The boundary problem was solved in 1972 by de Boor, who published a paper detailing a mathematically stable formula for calculating

B-splines. De Boor used the concept of divided differences and recursively computed higher-order splines. This algorithm became the basis for much of the spline literature that followed in fields as diverse as statistics and computer-aided design.

2.2 B-splines

While many other smoothing methods exist such as LOWESS, kernel density estimators, and running-mean and running-line (Cleveland 1979; Ramlau-Hansen 1983; Buja et al. 1989), B-Splines are especially popular among statisticians because they are easy to implement and allow greater flexiblity than other spline methods.

2.2.1 Constructing Basis Functions

B-splines are smoothing splines that are based on B-spline basis functions. The term B-spline has been used for both the basis functions and the resulting smoothed function. This paper will use the term "B-spline basis function" to refer to the basis function and "smoothed function" to refer to the function fit using the B-spline basis function. Basis functions are functions that are mathematically independent from one another. The calculation of a basis function of a given order depends only on the number and location of the knots within the basis function interval. It does not depend on any other basis functions. When basis functions are taken in linear combinations, they can be used to approximate any target function. The larger the number of basis functions used in the linear combination, the better the smoothed function is approximated (Ramsay and Silverman 2005).

Knots divide the interval over which basis functions are calculated. The number and location of the knots along with the order of the basis function determine the shape of the basis function. When the knots are equally spaced, they only determine the scale of the basis function. When they are not equally spaced, the B-spline basis functions can be skewed or truncated. Figure 2.1 graphically represents a set of basis

functions calculated using 4 equally spaced interior knots and 4 boundary knots on each end. The knots were $(0, 0, 0, 0, 2, 4, 6, 8, 10, 10, 10, 10)$. Notice that center basis function is symmetrical. As the basis functions move closer to the edge of the graph, they become more skewed. This is due to the stacking of the knots at the endpoints.

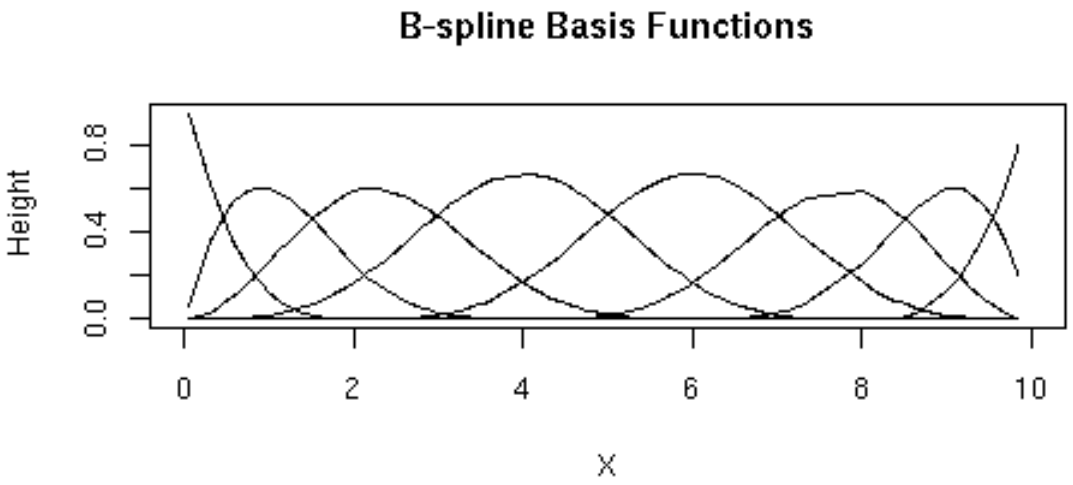

Figure 2.1: B-Spline Basis Functions

Generally knots are located at unique values. There are two purposes for stacking knots. The first is when the knots are located on a boundary. A basis function requires order δ knots on each side of all independent variables, t, evaluated where δ is the order of the B-spline. Because data is not generally found outside the interval δ, equal knots are often specified at each boundary so that the B-spline basis functions do not extend beyond the interval that contains data. Another reason to stack knots is to create a discontinuity in the smoothed curve. Only the data contained in the interval that the B-spline basis function covers can influence the smoothed function. When knots are stacked, the B-spline basis functions are truncated. If enough knots are stacked, the B-spline basis functions can be truncated such that two adjacent data points may never be contained in the same B-spline basis function. This would lead to a discontinuity in the smoothed function.

The order of the basis function and the number of knots is predetermined. The smoothness of the curve is determined by the order of the basis function as well as the number of knots. At each knot, all of the $\delta - 2$ derivatives of the adjoining functions are equal due to the construction of the B-spline. For splines of order four or higher, this leads to a smooth fit provided each knot is located at a unique value. Order four basis functions are the most common type of splines since they provide for equal first and second derivatives at each knot.

The basis functions in Figure 2.1 were created using the recursive formula developed by de Boor (1972). This formula is:

$$N_{i,1}(t) = \begin{cases} 1, & k_i \leq t < k_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,\delta}(t) = \frac{t - k_i}{k_{i+\delta-1} - k_i} N_{i,\delta-1}(t) + \frac{k_{i+\delta} - t}{t_{i+\delta} - t_{i+1}} N_{i+1,\delta-1}(t)$$

where $N_{i,\delta}(t)$ is the basis function evaluated at t_i, $k_1, ..., k_n$ are the knots; and δ is the order of the basis function being calculated. The basis functions are recursively calculated by first calculating the lower degree splines. $N_{i,\delta}(t)$ is greater than 0 only at δ knots; it is 0 everywhere else. The recursive nature of the formula can be shown by creating a triangle for each value of t:

	Order 1	**Order 2**	$\cdots$	**Order δ**
$\boldsymbol{k_{i-1}}$	0	0	$\cdots$	0
$\boldsymbol{k_i}$	$N_{i,1}(t) = 1$	$N_{i,2}(t)$	$\cdots$	$N_{i,\delta}(t)$
$\boldsymbol{k_{i+1}}$	0	$N_{i+1,2}(t)$	$\cdots$	$N_{i+1,\delta}(t)$
$\vdots$		0	$\ddots$	$\vdots$
$\boldsymbol{k_{i+\delta}}$			0	$N_{i+\delta,\delta}(t)$
$\boldsymbol{k_{i+\delta+1}}$				0.

Although it is not shown, all $N_{i,\delta}(t)$ outside the triangle are 0. The column corre-

sponding to the desired order can be transposed to create the X matrix when the smoothed function is fit.

2.2.2 Fitting the Smoothed Function

Ordinary Least Squares is used to determine a set of coefficients for the basis functions that minimize the sums of squares between the observed data and the linear combination of B-spline basis functions (Dierckx 1993). Using these coefficients, the smoothed function is calculated and can be plotted.

Figure 2.2 is an example of a B-Spline fit that was calculated from randomly generated data. The **X** values were randomly generated from a Uniform(0, 10) distribution. The **Y** values were then calculated as $sin(X) + .4 \times$ Random Normal$(0, 1)$. The knots are the same ones that were used to create Figure 2.1, (0, 0, 0, 0, 2, 4, 6, 8, 10, 10, 10, 10). Using the the recursive formula described in Section 2.2.1, the B-spline basis functions were calculated for an order four B-spline, This results in the same basis functions plotted in Figure 2.1. The basis functions then form the rows of the **X** matrix. The **X** matrix is 100×8 so it is not possible to show the entire table in this text; however, the first three rows are shown in Table 2.1. After calculating the B-Spline basis functions, a simple linear model, $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$, was fit using the generated **y** values and the **X** matrix created by the basis functions. The spline can then be plotted using predicted values from the simple linear model. Figure 2.2 is the fitted spline from this exercise. The dark, heavy line is the fitted curve and the dashed line is the actual function. The points represent the generated data values.

0.943	0.057	0.001	0.000	0.000	0.000	0.000	0.000
0.618	0.352	0.030	0.001	0.000	0.000	0.000	0.000
0.369	0.528	0.099	0.004	0.000	0.000	0.000	0.000

Table 2.1: First three rows of the constructed X matrix

B-splines have a tendancy to over-fit the data as more knots are used. Green

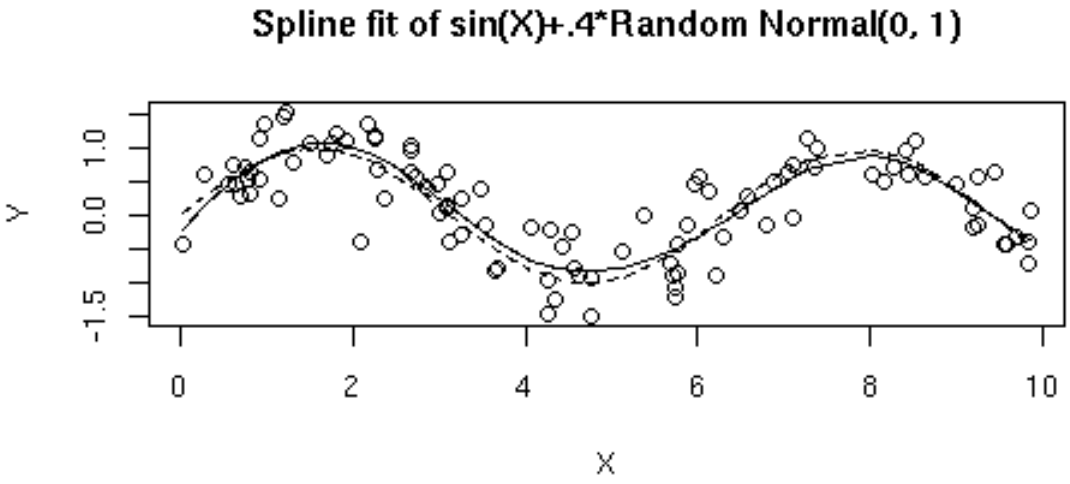

Figure 2.2: B-Spline fit example

and Silverman (1995) overcame one of the problems with B-splines by using a penalty to control the smoothness of the splines instead of using rules and algorithms to determine the optimal number of knots.

2.3 P-splines

In 1996, Eilers and Marx coined the term P-splines. P-splines, also called penalized splines (Ruppert and Carroll 2000) or pseudosplines (Hastie 1996), are an extension of B-splines and share many of the same desirable properties. Both P-spline and B-spline approximations do not degrade at the boundaries of the interval containing them. The mean of B-splines or P-splines evaluated at the data points is equal to the mean of the data points and the variance is less than the variance of the data points due to the smoothing. B-splines and P-splines also extend common generalized linear models. However, B-splines suffer from a practical problem: the number and location of the knots must be decided prior to fitting the model. The choice of the number of knots used is problematic because too few knots can lead to

under-fitting the data and too many knots can lead to over-fitting the data (Eilers and Marx 1996).

Many approaches have been proposed to help determine how many knots should be used. One method involves automatically optimizing the number of knots based on a predetermined algorithm. Friedman and Silverman (1989) used a stepwise procedure to find the optimal position of the knots for a smoothing spline. The fit of the knots is evaluated using a procedure called Generalized Cross Validation (GCV). Knots are added one by one to give the best piecewise fit while ensuring that the knots are spaced by an amount determined by the researcher. This stepwise procedure is repeated until a predetermined maximum number of knots is placed. The GCV is then computed for each step in the procedure (from 1 knot to K_{max} knots) and the solution with the lowest GCV is chosen. The procedure then uses a backwards stepwise procedure to determine if deleting any one of the knots will improve the GCV. Kooperberg and Stone (1991; 1992) use an automatic procedure based on AIC or BIC and the Wald statistic to determine the appropriate number of knots. This stepwise procedure removes the knot that is the least statistically significant, as measured by its Wald statistic, until only one knot remains. The iteration that minimizes either AIC or BIC is chosen as the final model. These solutions lead to complex computation and decision rules which are fairly arbitrary.

Good and Gaskins (1971) proposed that a roughness penalty be subtracted from the log-likelihood in order to control over-fitting. The penalty they proposed is $\lambda \int f''(x)^2 dx$, where $f(x)$ is the function to be fit and λ is the smoothing parameter. Silverman (1982) refined this suggestion by penalizing the second derivative of the logarithm of $f(x)$ instead of the second derivative of $f(x)$. Instead of being forced to directly select knots, a large number of knots are used and over-fitting is controlled by the value of the smoothing parameter λ.

O'Sullivan (1986; 1988) expanded on this idea by proposing an algorithm to au-

tomatically optimize the value of the smoothing parameter λ using a cross-validation procedure based on a Newton-Raphson approximation. The algorithm is complex and only an approximation of the optimal smoothing parameter but it can be computed quickly. Ruppert (2002) showed that with penalized splines, the number of knots has little effect on the resulting smoothed curve as long as a sufficient number of knots are used to fit the main features of the data. In rare cases the MSE increases as the number of knots increases. The computation time also increases as the number of knots increases, so there is an incentive to limit the number of knots used.

P-Splines expand the method of using a penalty to control the smoothing. Instead of basing the penalty on the second-order derivative, the penalty is based on the difference of the coefficients of adjacent B-splines. The non-penalized approach determines weights a_j such that

$$\sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{k} a_j B_j \left(x_i \right) \right\}^2$$

is minimized where n is the number of observations and B_j is the jth B-spline basis function. This is an Ordinary Least Squares fit to the B-splines. Traditional penalized procedures add a penalty function to this equation and minimize

$$\sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{k} a_j B_j \left(x_i \right) \right\}^2 + \lambda \int_{x_{min}}^{x_{max}} \left\{ \sum_{j=1}^{k} a_j B_j'' \left(x \right) \right\}^2 dx.$$

The traditional penalized procedure leads to complex mathematical equations and difficulties in fitting the model. Eilers and Marx (1996) simplified the previous equation by minimizing

$$\sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{k} a_j B_j \left(x_i \right) \right\}^2 + \lambda \sum_{j=k+1}^{n} \left(\Delta^l a_j \right)^2,$$

where a_j is the least squares weight, B_j is the jth B-spline, n is the number of observations, λ is the constant controlling the smoothing, and $\Delta^l a_j$ is the l^{th} difference between adjacent B-spline weights ($\Delta^2 a_j = \Delta\Delta a_j = (a_j - a_{j-1})(a_{j-1} - a_{j-2})$).

This simplified criterion is a good approximation of higher-order penalties. The equation reduces the mathematical complexity of the problem and replaces a computationally expensive operation with a cheap one. The attractive properties of penalized B-splines are preserved using this method and the computation is much easier than using the second derivative as a penalty (Eilers and Marx 1996).

2.4 Linear Mixed Models

Linear mixed models have been used for many years to analyze data from randomized block designs, longitudinal data, spatial data, and many other types of data in which the observations are correlated (Laird and Ware 1982; Littell et al. 1996; Diggle et al. 2002). The popularity of linear mixed models has led to the development of sophisticated procedures for their analysis in software packages such as SAS®, S-PLUS®, and R.

Linear mixed models are an extension of simple linear models. Recall the simple linear model

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{y}$ represents a vector of dependent variables, $\mathbf{X}$ is the full rank design matrix of known values, $\boldsymbol{\beta}$ is a vector of regression parameters, and $\boldsymbol{\epsilon}$ is a vector of error terms. It is common to assume that $\boldsymbol{\epsilon}$ is distributed $MVN(\mathbf{0}, \mathbf{V})$. In the simplest form, $\mathbf{V}$ is $\sigma^2\mathbf{I}$. $\boldsymbol{\beta}$ is estimated by minimizing the sum of squares $\boldsymbol{\epsilon'\epsilon} = (\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta})$. This leads to the normal equations $\boldsymbol{X'X\beta} = \boldsymbol{X'y}$ and the solution $\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$. If $\boldsymbol{\epsilon}$ is distributed normally with mean $\mathbf{0}$ and variance $\sigma^2\mathbf{I}$, then $\hat{\boldsymbol{\beta}}$ is the best linear unbiased estimator of $\boldsymbol{\beta}$, meaning that there is not a $\hat{\boldsymbol{\beta}}$ that has a smaller variance, is a linear function of $\mathbf{y}$, and is unbiased for $\boldsymbol{\beta}$.

The linear mixed model is an extension of the simple linear model. The equation

for the linear mixed model is

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Zu} + \boldsymbol{\epsilon},$$

where $\mathbf{y}$, $\mathbf{X}$, and $\boldsymbol{\beta}$ have the same meaning as the simple linear model. $\mathbf{Z}$ is the design matrix for the random effects, $\boldsymbol{u}$ is distributed as $MVN(\mathbf{0}, \mathbf{G})$, and $\boldsymbol{\epsilon}$ is distributed as $MVN(\mathbf{0}, \mathbf{R})$. Assuming that $\boldsymbol{u}$ is multivariate normal relaxes the iid restriction of the simple linear model. It allows for observations to be correlated and to have heterogeneous variances. This type of data is common in longitudinal experiments where there are repeated measures on an experimental unit, as is the case in the examples in this text.

In linear mixed models, both $\boldsymbol{u}$ and $\boldsymbol{\beta}$ are estimated. $\hat{\boldsymbol{\beta}}$ estimates the effect due to the fixed treatments and $\hat{\boldsymbol{u}}$ estimates the difference between the subgroups and the population mean as defined by the $\mathbf{Z}$ matrix. Most of the time $\boldsymbol{\beta}$ is of most interest since it involves the effects of a treatment; however, it is often useful to examine the $\boldsymbol{u}$ parameters as well.

If $\mathbf{G}$ and $\mathbf{R}$ are known the estimation problem is straightforward and easily solved by separately solving the mixed model equations for $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{u}}$. However, $\mathbf{G}$ and $\mathbf{R}$ are generally unknown. The most commonly used approach for estimating $\mathbf{R}$ and $\mathbf{G}$ is Restricted Maximum Likelihood (REML)(Swallow and Monahan 1984).

Once we find reasonable estimates of $\mathbf{G}$ and $\mathbf{R}$ we can solve the mixed model equations in the standard way. The mixed model equations are:

$$\begin{bmatrix} \boldsymbol{X'}\hat{\boldsymbol{R}}^{-1}\boldsymbol{X} & \boldsymbol{X'}\hat{\boldsymbol{R}}^{-1}\boldsymbol{Z} \\ \boldsymbol{Z'}\hat{\boldsymbol{R}}^{-1}\boldsymbol{X} & \boldsymbol{Z'}\hat{\boldsymbol{R}}^{-1}\boldsymbol{Z} + \hat{\boldsymbol{G}}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X'}\hat{\boldsymbol{R}}^{-1}\boldsymbol{y} \\ \boldsymbol{Z'}\hat{\boldsymbol{R}}^{-1}\boldsymbol{y} \end{bmatrix}.$$

The solution to these equations are:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X'}\hat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{X'}\hat{\mathbf{V}}^{-1}\boldsymbol{y}$$
$$\hat{\boldsymbol{u}} = \hat{\mathbf{G}}\mathbf{Z'}\hat{\mathbf{V}}^{-1}(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{\beta}}),$$

where $\mathbf{V} = \mathbf{V}(\mathbf{y}) = \mathbf{ZGZ}' + \mathbf{R}$ (Littell et al. 1996). If $\mathbf{G}$ and $\mathbf{R}$ were known then $\hat{\boldsymbol{\beta}}$ would be a best linear unbiased estimator (BLUE) of $\boldsymbol{\beta}$ and $\hat{\boldsymbol{u}}$ would be the best linear unbiased predictor (BLUP) of $\boldsymbol{u}$. However, since $\mathbf{G}$ and $\mathbf{R}$ are only estimates the term "empirical" is often used to indicate that $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{u}}$ are only approximations, referred to as EBLUE and EBLUP.

One result that is a byproduct of specifying $\boldsymbol{u} \sim MVN(\mathbf{0}, \mathbf{G})$ is that the estimates of $\boldsymbol{u}$ shrink toward 0. For prediction, this is generally a good thing because the variability inherent in the different experimental units is generally greater than the variability due to the different treatments (Robinson 1991). The shrinkage minimizes the effect that the observed experimental units have on predicting the outcome of a trial and give more weight to the effect of treatment itself. Mixed model splines as discussed in section 2.5 are possible because of the shrinkage properties of the $\hat{u}$ estimators. Because of the shrinkage, the fitted curve is smooth instead of reflecting every random variation in the data.

2.5 P-Splines and Linear Mixed Models

Many researchers have combined smoothing splines and mixed models. These include Lin and Zhang (1999) who proposed using General Additive Mixed Models to jointly estimate variance components and smoothing parameters. They applied this procedure to longitudinal data on respiratory infections as well as simulation studies. Because of the difficult numerical integration required, they also developed an extension of REML called double penalized quasi-likelihood. Wang (1998a) fit a random effects analysis of variance model with smoothing splines as the main effects and estimated the model using generalized maximum likelihood. He suggested that existing software could be used to fit the nonparametric mixed effects models and found a way to fit his model using a two-step process with SAS®. The first step is to calculate the B-splines and the second is to fit the model using SAS® PROC

MIXED. Verbyla et al.(1999) used REML implemented in a software package ASREML (Gilmour et al. 2002) to fit mixed models with smoothing spline components. Verbyla et al.(1999) used three examples from a variety of research areas. Brumbank and Rice (1998) also used REML and traditional B-splines to fit a non-parametric mixed model to grouped repeated measures data.

Researchers have also shown that splines themselves can be fit as mixed models. Commenting on the Robinson (1991) paper on BLUPs, Speed (1991) showed that it was possible to fit penalized splines as mixed models and therefore splines could also be considered BLUPs. Eilers (1999) showed that the P-spline formulation he helped develop could also be represented as mixed models. Both of these references are in comments on other spline and mixed model papers and do not contain a complete proof and examples. Wand (2003) took a different approach and used the shrinkage property of mixed models to fit the splines. This approach makes splines much more accessible since it relies on well understood mixed model theory and the notation can be simplified. His approach also relies on only specifying the fixed and random effects so that the models can be completely fit with widely available software such as SAS®, S-Plus®, or R.

The model Wand uses can be simply described with few mathematical formulas and a simple graph. Wand simplified the mathematics by using a truncated line basis function instead of a B-spline basis function. B-spline basis functions would work as well, but the mathematics are more complex. The truncated line basis function is represented by

$$(x-\kappa_k)_+ = \begin{cases} 0, & x \le \kappa_k \\ (x-\kappa_k), & x > \kappa_k \end{cases}, \tag{2.1}$$

where κ_k is the k-th knot. Consider the model

$$y_i = \beta_0 + \beta_1 x_i + \sum_{k=1}^{K} u_k (x_i - \kappa_k)_+ + \epsilon_i. \tag{2.2}$$

If this model is fit as a standard linear model with only fixed effects, then the plot

pictured on the left of Figure 2.3 would be the result. The fitted line is very rough due to the number of truncated line basis functions fit. The roughness of the fit can be remedied by imposing the restriction:

$$u_k \sim N(0, \sigma_u^2).$$

Thus the model is now a mixed model. For $\sigma_u^2 < \infty$, the u_k's are shrunk toward the mean and the fit is smooth. This is demonstrated on the right side of Figure 2.3.

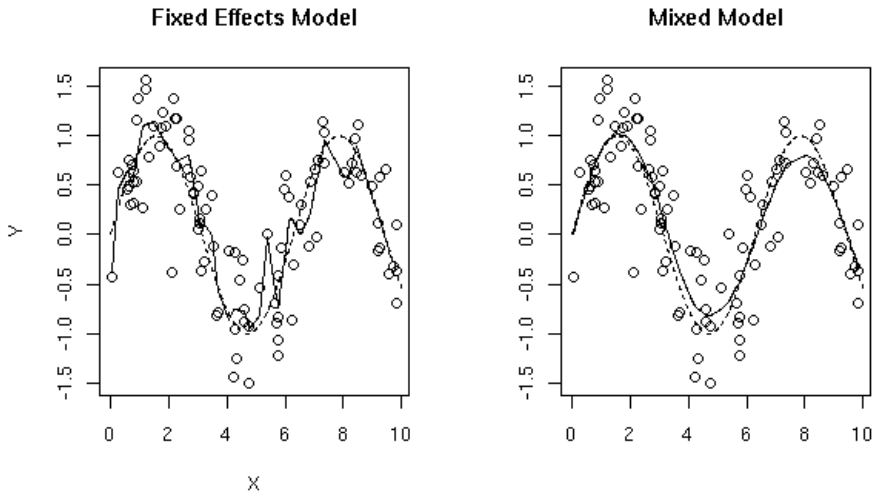

Figure 2.3: Truncated line basis splines fit with fixed effects model on the left and mixed effects model on the right. The data is the same as used in Figure 2.2. However, twenty-seven equally spaced interior knots are used instead of the twelve knots for the B-spline example.

If design matrices are defined as:

$$\boldsymbol{X} = [1 \; x_i]_{1 \le i \le n}, \quad \boldsymbol{Z} = \left[(x_i - \kappa_k)_+\right]_{1 \le i \le n, 1 \le k \le K},$$

with the β's set to $\boldsymbol{\beta} = [\beta_0, \beta_1]'$ and the u_ks set to $\boldsymbol{u} = [u_1, \ldots, u_K]'$, then you can rewrite equations 2.1 and 2.2 as a linear mixed model,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}, \quad \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\epsilon} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \sigma_u^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_\epsilon^2 \boldsymbol{I} \end{bmatrix} \right),$$

and perform estimation using REML. This formulation allows for great flexibility in the mixed model fit since both random and fixed effects can be easily added. Wand and Ngo (2004) published a follow-up paper that walks through specific examples of smoothing that include computer code for both S-Plus® and SAS®.

Fitting the models with R is a trivial extension of the S-Plus® examples because both S-plus® and R are based on the programming language "S". R is appealing as a statistical programming langauge because it is open-source, free, and widely available. R lacks the graphic interface that S-Plus® has for inexperienced users, but implements nearly all of the programming functions of S-Plus®. It also facilitates the creation of high-quality graphs which are indispensable for smoothing problems.

Formal inference has not been well-developed for traditional spline models. Complicated, numerically intensive methods are required for an exact fit. This has led researchers to simplify the challenge of fitting confidence intervals and performing hypothesis tests by using lower-dimensional approximations (Gray 1994). P-splines estimated using mixed model software avoid this problem because standard mixed-effects diagnostics, inference, and model selection can be used (Wand 2003). The smoothing coefficients do not have any direct meaning, but degrees of freedom and standard errors can be calculated. The degrees of freedom and standard errors can then be used to create a variety of inference procedures including variability bars. The amount of smoothing implemented is determined by the number of knots or the basis function used in the Z matrix. Researchers who are familiar with mixed models but have little experience with smoothing can use the methods in this paper to take advantage of the flexibility that splines offer (Ruppert et al. 2003).

3. PROJECT SCOPE

3.1 Research Objectives

The purpose of this project is to create a "how-to" guide for fitting splines using mixed models. The project is written for statisticians and health science researchers who have knowledge of mixed models, but do not have significant knowledge of splines or programming. The project walks the reader through an example of creating a set of splines from a blocked design. The splines for each of the three groups are created both individually and simultaneously, and confidence bands are fit to the data.

The data come from a health science experiment. It is a simple three-group design. Forty-five subjects were divided into three groups. Each subject had an electronic thermometer surgically embedded into their Achilles tendon and then went through an exercise routine designed to increase the temperature of the tendon. Immediately following the exercise, the subjects received one of the following treatments. The control group was allowed to recover without intervention. The second group had an ice pack placed on the Achilles tendon five minutes after the completion of the exercise. The ice pack was administered for 20 minutes and then the group was allowed to recover for 30 minutes. The third group was allowed the same five-minute resting period as the second group, but had their leg immersed in a whirlpool ice bath for 20 minutes, followed by a 30-minute recovery. The temperature of all three groups was measured every 30 seconds for the entire 55-minute period. The primary questions of interest are how fast the tendon cools and how long the coolness persists. The faster the temperature is dropped and the longer the temperature stays below normal, the greater the therapeutic effect. The raw data are graphed in Figure 3.1. The dataset is very rich, so the graphs already show a strong respondent trend.

The graphs for the two treatment groups show two change-points. These points

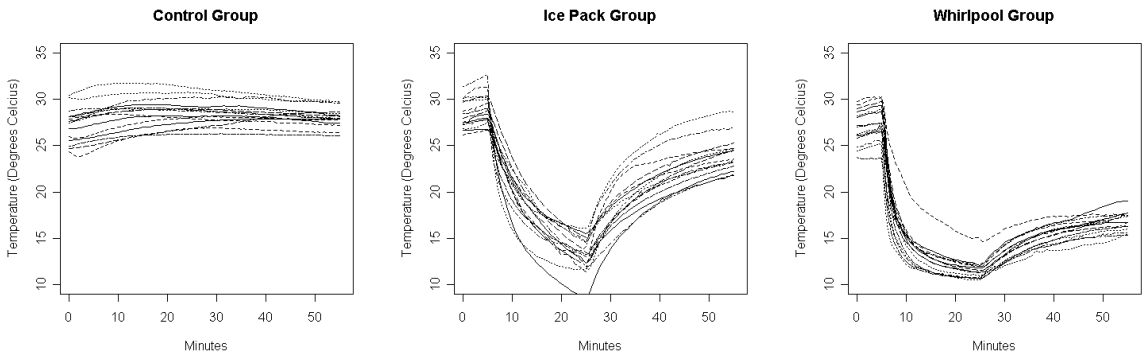

Figure 3.1: Control, Ice pack, and Whirlpool Data

occur at the five-minute mark and the 25-minute mark where the treatment began and ended. Because the change points in the data are so sharp, it is not appropriate to fit a simple parametric model. Splines offer a simple alternative. Using splines, a flexible model can be fit which does not follow a strict functional form. The data set contains repeated measures of the same observational unit. In the interest of simplicity, this paper ignores the serial correlation within the observational units, and models within subject correlation only via the random effects. In Chapter 4, which addresses challenges of the model, there is a discussion of various ways to fit a model that incorporate the correlation due to repeated measures on the same respondent.

3.2 One-Group Solution

The whirlpool icebath group will be used as an example for fitting a one-group solution. A simplified version of the actual R code is presented to reduce the complexity of the problem. The full program is presented in Appendix B. The basic procedure for fitting the model involves six basic steps:

(1) Set up the dependent and independent variables

(2) Specify knots

(3) Create **Z** matrix

(4) Prepare model

(5) Estimate parameters

(6) Plot results

3.2.1 Set up variables

In the whirlpool dataset there are two variables. The temperature of the tendon is the dependent variable and the time that the measurement was taken is the independent variable. There are 110 observations per respondent.

```
y <- whirlpool.temperature    #The dependent variable
x <- whirlpool.times          #The independent variable
```

3.2.2 Specify knots

The second step is to set up the knots which are needed in step 5, the estimation step. Wand (2004) proposed a default knot specification that functions well for most problems. The simple rule used in the default.knots function is

$$\kappa_k = \left(\frac{k+1}{K+2}\right) \text{th sample quantile of unique } x_i\text{'s}, \quad 1 \leq k \leq K,$$

with

$$K = \begin{cases} 5, & \frac{1}{4}N < 5 \\ \lfloor \frac{1}{4}N \rfloor, & 5 \leq \frac{1}{4}N \leq 35 \\ 35, & \frac{1}{4}N > 35 \end{cases},$$

where N is the number of unique X values. These knots seem to work well for many problems and are a reasonable initial assumption for knot placement in this example.

```
default.knots <- function(x,num.knots)
{
   if (missing(num.knots))
      num.knots <- max(5,min(floor(length(unique(x))/4),35))

   return(quantile(unique(x),seq(0,1,length=
                  (num.knots+2))[-c(1,(num.knots+2))]))
}

knots <- default.knots(x)
```

The function `default.knots` returns 27 unique knots based on the algorithm above.

3.2.3 Create Z matrix

The third step in fitting the model is to create the $\boldsymbol{Z}$ matrix. The $\boldsymbol{Z}$ matrix represents the difference between each x value and each knot. Its dimensions are the length of the $\boldsymbol{X}$ matrix by the number of knots. The values of the $\boldsymbol{Z}$ matrix are constrained to be greater than or equal to 0, so any differences that result in a negative number are set to 0. A simple example of a $\boldsymbol{Z}$ matrix with knots $\kappa_k = (1.5, 3, 4.5)$ and X values $(1, 2, 3, 4, 5)$ is

$$\begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0 \\ 1.5 & 0 & 0 \\ 2.5 & 1 & 0 \\ 3.5 & 2 & .5 \end{bmatrix}.$$

```
z <- outer(x, knots, "-")     #Number of obs. X Number of knots
z <- z * (z > 0)              #All negative elements of z set to 0
```

3.2.4 Prepare Model

The linear mixed model function in R, lme(), requires a grouped data structure in order to fit the model using a specialized $\boldsymbol{Z}$ matrix. Normally the grouped data object would contain information on which group each observation belongs to so that the $\boldsymbol{Z}$ matrix could be properly constructed. Since the $\boldsymbol{Z}$ matrix is explicitly defined in this problem, a single dummy group will be created so that the software does not attempt to alter the $\boldsymbol{Z}$ matrix that has been provided.

```
group <- rep(1, length(x))    #Create the dummy grouping variable
temp.dataframe <- data.frame(x, y)
                              #Combine the raw data into a data
                              #frame
model.data <- groupedData(y~x|group, data=temp.dataframe)
                              #Create the grouped data object
                              #for the fixed effects
```

3.2.5 Estimate Parameters

The actual model can be fit once the grouped data object has been created. The model should be fairly clear. The model is $temperature = \beta_0 + time * \beta_1 + \mathbf{Z}u$. The β_0 parameter in the model is an intercept and the β_1 is a pseudo-slope related to the time that each temperature was taken. The random effects are specified in the $\boldsymbol{Z}$ matrix that was previously created.

```
library(nlme)
fit <- lme(y~x, random=pdIdent(~-1+z), data=model.data)
```

The function `lme()` is a part of the `nlme` library. This library needs to be loaded before the function can be called. The fixed effects section of the model is a

standard intercept and slope model. The intercept retains its standard meaning, but since the actual curve is determined by a combination of the slope and the random effects the slope has no meaning. In the random statement the `pdIdent()` function is used to construct the $\boldsymbol{Z}$ matrix exactly as it appears in the argument and specifies that the $\boldsymbol{G}$ matrix should be a multiple of the identity matrix. The `~-1+z` specifies that the $\boldsymbol{Z}$ matrix created should be used exactly as specified without the addition of an intercept.

3.2.6 Plot Results

The coefficients of the mixed model are not meaningful by themselves. They are only useful when combined to create the fitted curve. The easiest way to display the results for this problem is using a two-dimensional line plot.

```
fit.beta.hat <- fit$coef$fixed
                              #extract the beta-hats
fit.u.hat <- unlist(fit$coef$random)
                              #extract the u-hats
#Construct the plot dimensions
x.grid <- seq(min(x), max(x), length=length(x))
X.grid <- cbind(1, x.grid)
Z.grid <- outer(x.grid, knots, "-")
Z.grid <- Z.grid * (Z.grid > 0)
yhat.grid <- X.grid%*%fit.beta.hat + Z.grid%*%fit.u.hat
plot(x, y, type='n')
for( i in 1:15) {
     lines(x, each.respondents.y, lty=i)
}
lines(x.grid, yhat.grid, col="green", lwd=3)
```

Figure 3.2 is the resulting plot. The plot shows that the fit is quite good except in two locations. At the beginning of the treatment near the five-minute mark, the curve anticipates the beginning of the treatment and at the beginning of the recovery

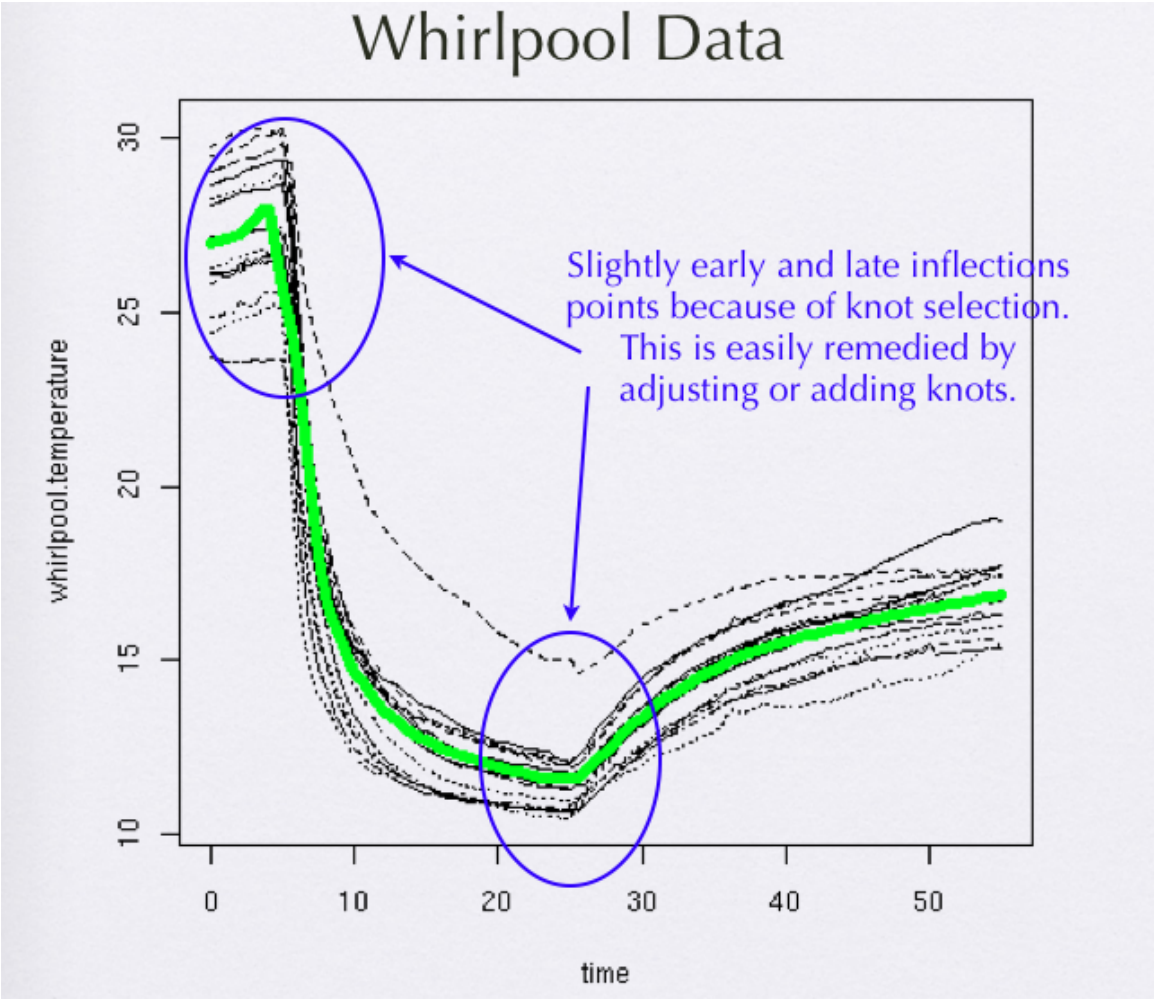

Figure 3.2: Fit of whirlpool group. The dashed lines are the data points. The thick line is the fitted spline. The circle highlight the change points in the graph. Because of knot position, the fit at the change points is less than ideal.

period near the 25-minute mark the curve slightly lags the data. This can be corrected to some degree by adjusting the knots to explicitly take into account the treatment change points. By adding two knots at the 5-minute and the 25-minute marks the model fit is greatly improved.

```
knots <- default.knots(x)
knots <- append(knots, 5, after=2)
knots <- append(knots, 25, after = 12)
```

The 5-minute mark is the third knot and the 25-minute mark is the thirteenth knot. The parameters can be re-estimated and the model re-plotted.

As seen in Figure 3.3, the fit is now better at the change points. The reversal of the slope of the spline is closer to the 5-minute mark and the second slope reversal comes at exactly the 25-minute mark. With traditional P-splines the change point could be emphasized even further by placing additional knots at the 5-minute mark to create a discontinuity in the fitted curve. The mixed model approach used in this paper does not allow for these additions since it creates redundant columns in the $\boldsymbol{Z}$ matrix and `R` removes those columns from the model.

3.3 Three-Group Solution

The next step in fitting the tendon temperature model is to fit the three groups simultaneously. Since mixed models allow for different groups to have different variance and covariance structures, each treatment can be fitted as a group of random effects. The same steps used in Section 3.2 can be followed for the three-group solution.

(1) Set up the dependent and independent variables

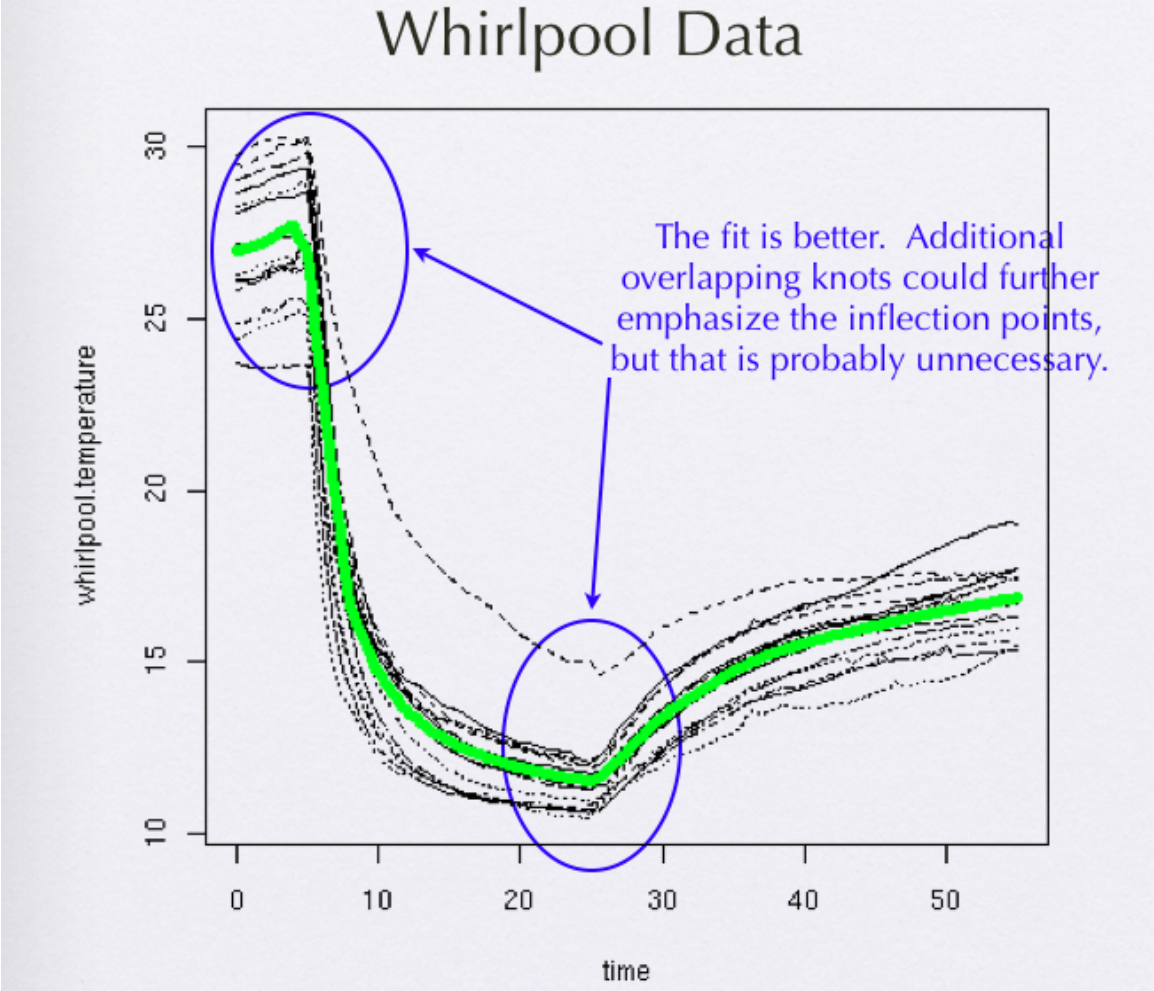

Figure 3.3: Fit of whirlpool group with adjusted knots. The thick line is the fitted spline. The circles highlight the 5-minute mark and the 25-minute mark where the knots were adjusted.

(2) Specify knots

(3) Create $\boldsymbol{Z}$ matrix

(4) Prepare model

(5) Fit model

(6) Plot results

The notation used in this section will be more complex than used in the previous section as the problem is now more complicated.

3.3.1 Set up the dependent and independent variables

The data were originally stored in three separate files, one for each treatment group. The data need to be stacked so that there is one dependent variable per row.

```
control <- read.csv('control.csv', header=T)
icepack <- read.csv('icepack.csv', header=T)
whirlpool <- read.csv('whirlpool.csv', header=T)
temperature <- rbind(control,icepack,whirlpool)
```

Temperature is the dependent variable and it is sorted by respondent and then by order within each respondent. The next step is to construct the independent variable. In this situation it is easier to create the independent variable using `R` than it is to read it in from a file. For this model a reasonable specification is a separate slope and intercept for each group. Other specifications would be possible and appropriate, but the separate slope and intercept model maintains the group structure and is easy to interpret. This process creates six columns in the $\boldsymbol{X}$ matrix. The first three columns represent the intercept for each group; a 1 if the subject is a member of the group and a 0 if it is not. The fourth through sixth columns are the measurement times for each group. The value generated is the actual time of the

measurement if the subject was a member of the group associated with the column or 0 if the subject was not a member of the group.

```
n <- length(temperature)
time <- rep(seq(0, 55, .5), 45)

index.control <- (1:(n/3))
index.icepack <- n/3 + (1:(n/3))
index.whirlpool <- 2*(n/3) + (1:(n/3))

int.control <- rep(0, n)
int.icepack <- rep(0, n)
int.whirlpool <- rep(0, n)

int.control[index.control] <- 1
int.icepack[index.icepack] <- 1
int.whirlpool[index.whirlpool] <- 1

X <- cbind(int.control, int.icepack, int.whirlpool, int.control*time,
int.icepack*time, int.whirlpool*time)
```

Since there are no missing data, time is simply a sequence from 0 minutes to 55 minutes in half-minute increments. This is repeated for all 45 respondents. The next step is to create a stretched identity matrix for the intercept. Because the respondents are sorted according to group, the first third of the respondents belong to the control group, the second third to the ice-pack group, and the final third to the whirlpool group. By creating an index variable for the group, the three intercept columns can quickly be fill. The slope is then the intercept column multiplied by the column of times. The final $\boldsymbol{X}$ matrix is just the combination of those columns. There are 4,995 total observations, so the $\boldsymbol{X}$ matrix is $4,995 \times 6$.

3.3.2 Specify Knots

The number of knots that are chosen affects the size of the model. In this example there will be 3 times as many random effects as there are knots in the model. This will greatly increase the number of parameters in the model and can substantially increase the computation time. A balance needs to be maintained since more knots mean a smoother curve, while with too few knots the model will not be sufficiently smooth.

In this model there are 111 unique X values. If one knot for every four X values were used, as in the previous example, then there would be 27 knots. This number can be reduced by incorporating some information from the experiment. The graphs of the data show a flat portion for the first five minutes of the experiment prior to the application of the treatments. There can be few knots in this period of time since there is little curvature in the model. As demonstrated in Section 3.2.6, a knot is needed at the first change point at 5 minutes. One other knot between zero and the the change point should be sufficient. The next section of the data is the region of the steepest change due to the treatment being applied. Ten knots should be enough to model this with a knot at the 25 minute mark, but not one at the five minute mark since the previous piecewise section contains one. The third section also has significant curvature, but not quite as much as the second section. Eight knots should be sufficient in this section for a total of 20 knots. The end points are not included in the knot specification since there are implicit knots at the beginning and end of the interval.

```
knots.1 <- c(2.5, 5)
knots.2 <- quantile(c(5, 25), seq(0, 1, length=11))[-c(1)]
knots.3 <- quantile(c(25, 55), seq(0,1, length=10))[-c(1, 10)]
knots <- c(knots.1, knots.2, knots.3)
```

3.3.3 Create Z Matrix

The $\boldsymbol{Z}$ matrix needs to be a block diagonal matrix, with one block for each of the three treatment groups. Since the knots and the $\boldsymbol{X}$ matrix are the same for each of the treatment groups, each of the blocks will be the same. Using the same procedure as that used in Section 3.2.3, first take the difference between each time and each knot and then set the negative values to 0. Then, using the previously created grouping variable, create the blocked diagonal matrix.

```
Z.overall <- outer(time, knots, "-")
Z.overall <- Z.overall*(Z.overall > 0)

Z <- cbind(Z.overall*int.control, Z.overall*int.icepack,
Z.overall*int.whirlpool)
```

The creation of the `Z.overall` is fairly straight forward. To expand the overall $\boldsymbol{Z}$ matrix element-wise multiply the overall $\boldsymbol{Z}$ matrix with the intercept for each treatment group. This is a shorter process than creating the entire $\boldsymbol{Z}$ matrix by hand.

3.3.4 Prepare Model

The formula used in the random statement of the mixed model can be created prior to use to simplify the model specification. There needs to be one formula for each treatment group that identifies which columns of the $\boldsymbol{Z}$ matrix are used in each solution.

```
block.ind <- list(1:K, (K+1):(2*K), (2*K+1):(3*K))
Z.block  <-  list()
```

```
for (i in 1:length(block.ind))
    Z.block[[i]] <- as.formula(paste("~Z[,c(",paste(block.ind
         [[i]],collapse=","),")]-1"))
```

The code is best described by showing the results.

```
[[1]]
~Z[, c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
    17, 18, 19, 20)] - 1

[[2]]
~Z[, c(21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
    35, 36, 37, 38, 39, 40)] - 1

[[3]]
~Z[, c(41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
    55, 56, 57, 58, 59, 60)] - 1
```

`Z.block` is a list that contains three elements. Each of the elements is the one-sided formula for one of the treatment groups. It explicitly calls out the columns in the $\boldsymbol{Z}$ matrix that are used for each of the groups fit.

The grouped data structure used in the mixed model also needs to be created. Just as in Section 3.2.4, a dummy variable of ones needs to be created to prevent the software from expanding the specified $\boldsymbol{Z}$ matrix. The `groupedData`object is required by the software when a $\boldsymbol{Z}$ matrix is explicitly specified, but provides no additional information.

```
dummy <- rep(1, n)
tendon <- groupedData(temperature~X|dummy,
data=data.frame(X, temperature))
```

3.3.5 Estimate Parameters

The estimate parameters step for this model is fairly straightforward. The formula for the random effects and data frame object were created in the previous step.

```
fit <- lme(temperature~-1+X, data=tendon, random=pdBlocked(Z.block,
pdClass="pdIdent"))
```

The only thing that is unusual in the `lme`call is the `pdBlocked` function in the random statement. It specifies that the $\boldsymbol{G}$ matrix will have a blocked diagonal structure and takes two arguments. The first argument is a list of formulas that define which random effects the blocks will consist of. Each formula represents one block and in this example each block of the $\boldsymbol{G}$ matrix will represent a different group. This list of formulas was created in Section 3.3.4. The second argument, `pdClass=`, specifies the structure of an individual block of the $\boldsymbol{G}$ matrix. For this example we will use a multiple of an identity matrix for each group which corresponds to the `"pdIdent"` class.

3.3.6 Plot Results

Plotting the results of the model is also fairly straightforward. It is simply a matter of extracting the $\hat{\boldsymbol{\beta}}$'s and $\hat{\boldsymbol{u}}$'s, and calculating the predicted value for each group. Any values can be used, but since there is an observation every 30 seconds, it is convenient to use the actual time values for predictions.

```
beta.hat <- fit$coef$fixed
u.hat <- unlist(fit$coef$random)
```

```
grid.control <- 1:111
grid.icepack <- (15*111+1):(16*111)
grid.whirlpool <- (30*111+1):(31*111)

X.grid <- X[c(grid.control, grid.icepack, grid.whirlpool),]
Z.grid <- Z[c(grid.control, grid.icepack, grid.whirlpool),]

fhat <- X.grid%*%beta.hat + Z.grid%*%u.hat
```

The $\hat{\boldsymbol{\beta}}$'s and the $\hat{\boldsymbol{u}}$'s are first extracted from the model. The X grid used for the prediction is made up of the rows of the $\boldsymbol{X}$ matrix for the first respondent in each group. The same is done for the $\boldsymbol{Z}$ matrix. Note that the $\boldsymbol{Z}$ matrix is dependent on the $\boldsymbol{X}$ matrix. It does not need to be recalculated in this example because the same $\boldsymbol{X}$ matrix values are used in the prediction as were used to fit the model. If different $\boldsymbol{X}$ values were desired for prediction, the $\boldsymbol{Z}$ matrix would need to be recalculated using those $\boldsymbol{X}$ values and the knots used to fit the model.

The fitted values can then be used to overlay the splines on a plot of the raw data. This is shown in Figure 3.4. The complete code for the plots can be found in Appendix B, but the code to fit the splines is:

```
grid.control <- 1:111
grid.icepack <- (15*111+1):(16*111)
grid.whirlpool <- (30*111+1):(31*111)


X.grid <- X[c(grid.control, grid.icepack, grid.whirlpool),]
Z.grid <- Z[c(grid.control, grid.icepack, grid.whirlpool),]

fhat <- X.grid%*%beta.hat + Z.grid%*%u.hat

lines(time[1:111], fhat[1:111], col='red', lwd=3)
lines(time[(111+1):(2*111)], fhat[(111+1):(2*111)],
        col='blue', lwd=3)
```

```
lines(time[(2*111+1):(3*111)], fhat[(2*111+1):(3*111)],
        col='green', lwd=3)
```

3.4 Confidence Bands

The splines themselves are useful, but they only tell a part of the story. Since the model was fit with mixed models we can add confidence bands to the plot of the splines. These confidence bands are useful for detecting significant differences. The first step in computing the confidence bands is to calculate the complete variance covariance matrix for the mixed model. Using estimates of the $\boldsymbol{G}$ matrix and the $\boldsymbol{R}$ matrix from the REML procedure, the joint variance-covariance matrix, $\boldsymbol{C}$, of $(\hat{\beta} - \beta, \hat{u} - u)$ can be estimated. As seen in Littel et al. (1996) and Robinson (1991),

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{X'R^{-1}X} & \boldsymbol{X'R^{-1}Z} \\ \boldsymbol{Z'R^{-1}X} & \boldsymbol{Z'R^{-1}Z + G^{-1}} \end{bmatrix}^{-1}$$

for full rank $\boldsymbol{X}$. Since $\boldsymbol{G}$ and $\boldsymbol{R}$ are not known, the estimates $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{R}}$ are used instead. $\boldsymbol{C}$ is notationally $\hat{\boldsymbol{C}}$ to signify that it is only an estimate.

Using this formula it is not difficult to create the $\hat{\boldsymbol{C}}$ matrix once $\hat{\boldsymbol{R}}$ and $\hat{\boldsymbol{G}}$ are extracted from the model.

```
sig.sq.eps <- fit$sigma^2
R <- diag(sig.sq.eps, 111*45)
G <- sapply(pdMatrix(fit$modelStruct$reStruct), "*", fit$sigma^2)
G <- matrix(G, 60, 60)
```

In the estimation step in Section 3.3.5 $\hat{\boldsymbol{G}}$ was specified as a matrix:

$$\begin{bmatrix} \hat{\sigma}^2_{G_1}\boldsymbol{I} & 0 & 0 \\ 0 & \hat{\sigma}^2_{G_2}\boldsymbol{I} & 0 \\ 0 & 0 & \hat{\sigma}^2_{G_3}\boldsymbol{I} \end{bmatrix}.$$

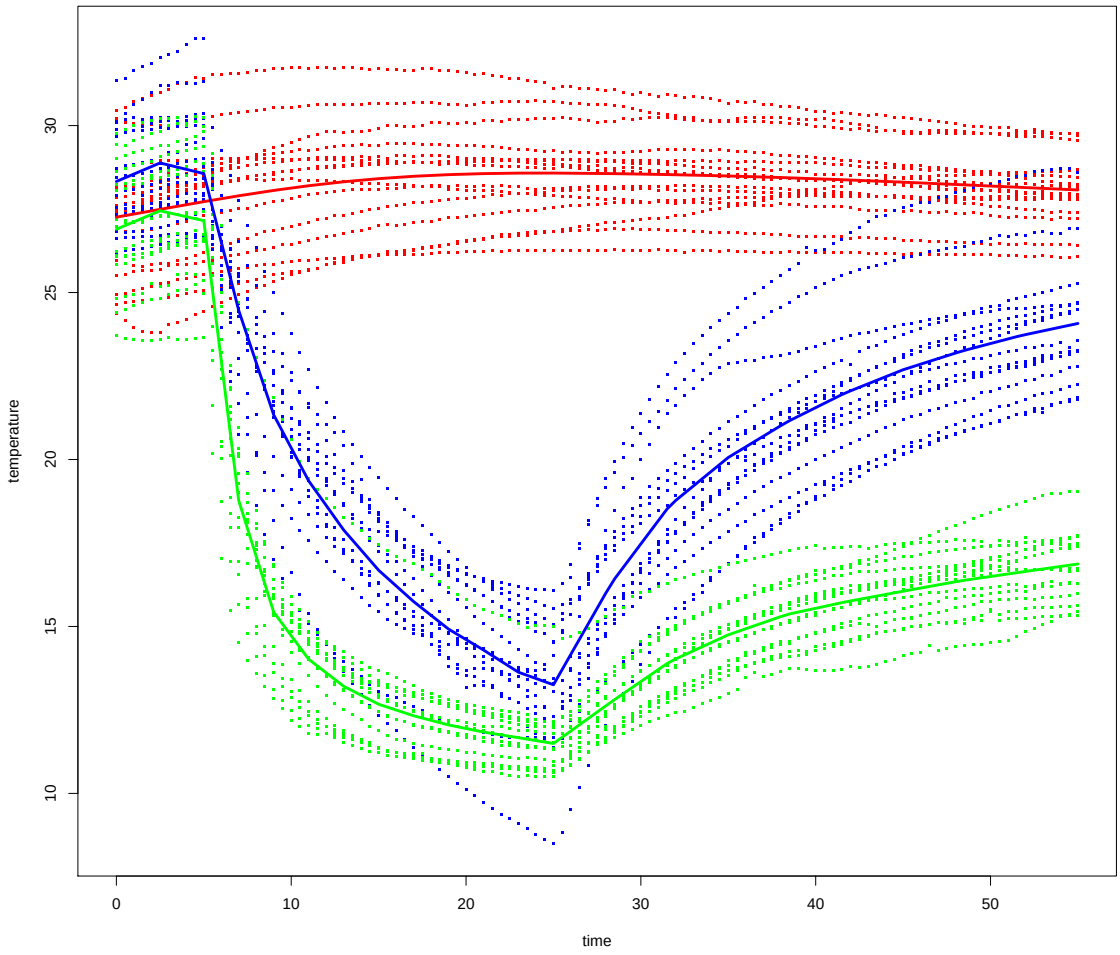

Figure 3.4: All three groups fitted simultaneously. The three thicker lines are the spline fits for each group. The top line (red) is the control group, the second line (blue) is the ice-pack group, and the bottom (green) line is the whirlpool group. The dots are the raw data.

and $\hat{\boldsymbol{R}}$ was specified as $\hat{\sigma}^2_\epsilon \boldsymbol{I}$. The R code extracts the parameters $\hat{\sigma}^2_{G_1}$, $\hat{\sigma}^2_{G_2}$, $\hat{\sigma}^2_{G_3}$, and $\hat{\sigma}^2_\epsilon$. It then constructs the $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{R}}$ matrix. The value for $\hat{\sigma}^2_\epsilon$ is stored as part of the lme object and is easily extracted. $\hat{\boldsymbol{G}}$ is decomposed into a positive definite matrix object with each element divided by $\hat{\sigma}^2_\epsilon$. Thus $\hat{\boldsymbol{G}}$ can be found by multiplying each element of the positive-definite matrix by $\hat{\sigma}^2_\epsilon$. The call to `pdMatrix` returns the positive-definite matrix used in the calculation as a list. The `sapply` statement multiplies each element in the list by $\hat{\sigma}^2_\epsilon$. The $\hat{\boldsymbol{G}}$ matrix can then be reconstructed.

The result is that

$$\hat{\boldsymbol{R}} = \hat{\sigma}^2_\epsilon \boldsymbol{I} = 2.56 \cdot \boldsymbol{I},$$

where $\boldsymbol{I}$ is a 4995 $\times$ 4995 identity matrix and

$$\hat{\boldsymbol{G}} = \begin{bmatrix} \hat{\sigma}^2_{G_1}\boldsymbol{I} & 0 & 0 \\ 0 & \hat{\sigma}^2_{G_2}\boldsymbol{I} & 0 \\ 0 & 0 & \hat{\sigma}^2_{G_3}\boldsymbol{I} \end{bmatrix} = \begin{bmatrix} 0.0003 \cdot \boldsymbol{I} & 0 & 0 \\ 0 & 0.3485 \cdot \boldsymbol{I} & 0 \\ 0 & 0 & 1.2947 \cdot \boldsymbol{I} \end{bmatrix},$$

where each $\boldsymbol{I}$ is a 20 $\times$ 20 identity matrix.

Once the $\boldsymbol{G}$ and $\boldsymbol{R}$ matrix are extracted the $\hat{\boldsymbol{C}}$ matrix can be created.

```
G.inv <- solve(G)
R.inv <- diag(1/sig.sq.eps, 4995)

C11 <- t(X)%*%R.inv%*%X
C21 <- t(Z)%*%R.inv%*%X
C22 <- t(Z)%*%R.inv%*%Z+G.inv
Chat <- cbind(rbind(C11, C21), rbind(t(C21), C22))
Chat <- solve(Chat)
```

Because $\boldsymbol{R}$ is a diagonal matrix the expensive operation of taking the inverse of a 4995 $\times$ 4995 matrix can be avoided by simply taking the reciprocal of each of the diagonal elements. Using this formulation the largest matrix that needs to be inverted is the $66X66$ matrix, one row and column for each parameter.

Note that $\hat{\boldsymbol{C}}$ underestimates the overall variability of the model since it does not take into account the variance of $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{R}}$. One way to correct this to use variance inflation factors; a second way is to use a conservative number of degrees of freedom in any t- or F-statistic (Littell et al. 1996). The second method is the procedure used in this example.

If $\boldsymbol{L}$ is a contrast vector, then confidence bands for each spline are

$$\boldsymbol{L}\begin{bmatrix}\hat{\beta}\\ \hat{u}\end{bmatrix} \pm t_{\hat{\nu},\alpha/2}\sqrt{\boldsymbol{L}\hat{\boldsymbol{C}}\boldsymbol{L}'},$$

where $\hat{\nu}$ is the approximate degrees of freedom and $t_{\hat{\nu},\alpha/2}$ is the $(1-\alpha/2)100^{th}$ percentile of the $t_{\hat{\nu}}$-distribution (Littell et al. 1996). Since multiple tests are being done, a Scheff correction is used for the critical value. Eight degrees of freedom are used as a conservative estimate for the denominator degrees of freedom. Eight was chosen because $N_{group} - p_{fixed\ effects} - 1$ $(15-6-1)$. Two numerator degrees of freedom are used for the three treatments minus 1. This is a fairly arbitrary choice, but should be sufficiently conservative.

```
se.control <- sqrt(diag(cbind(X[grid.control,], Z[grid.control,])
     %*%Chat%*%t(cbind(X[grid.control,], Z[grid.control,]))))
ul.control <- fhat[1:111]
     + (sqrt(2)*sqrt(qf(.95, 2, 8))*se.control)
ll.control <- fhat[1:111]
     - (sqrt(2)*sqrt(qf(.95, 2, 8))*se.control)
lines(seq(0, 55, .5), ll.control, col='red', lwd=3, lty=2)
lines(seq(0, 55, .5), ul.control, col='red', lwd=3, lty=2)

se.icepack <- sqrt(diag(cbind(X[grid.icepack,], Z[grid.icepack,])
     %*%Chat%*%t(cbind(X[grid.icepack,], Z[grid.icepack,]))))
ul.icepack <- fhat[(111+1):(2*111)]
     + (sqrt(2)*sqrt(qf(.95, 2, 8))*se.icepack)
ll.icepack <- fhat[(111+1):(2*111)]
     - (sqrt(2)*sqrt(qf(.95, 2, 8))*se.icepack)
lines(seq(0, 55, .5), ll.icepack, col='blue', lwd=3, lty=2)
```

```
lines(seq(0, 55, .5), ul.icepack, col='blue', lwd=3, lty=2)

se.whirlpool <- sqrt(diag(cbind(X[grid.whirlpool,], Z[grid.whirlpool,])
     %*%Chat%*%t(cbind(X[grid.whirlpool,], Z[grid.whirlpool,]))))
ul.whirlpool <- fhat[(2*111+1):(3*111)]
     + (sqrt(2)*sqrt(qf(.95, 2, 8))*se.whirlpool)
ll.whirlpool <- fhat[(2*111+1):(3*111)]
     - (sqrt(2)*sqrt(qf(.95, 2, 8))*se.whirlpool)
lines(seq(0, 55, .5), ll.whirlpool, col='green', lwd=3, lty=2)
lines(seq(0, 55, .5), ul.whirlpool, col='green', lwd=3, lty=2)
```

This uses a simple contrast for each X value and plots the results in Figure 3.5, which shows that even with the conservative degrees of freedom the confidence bands do not overlap once the treatment has begun, so one can conclude that treatments are significantly different at $\alpha = .05$

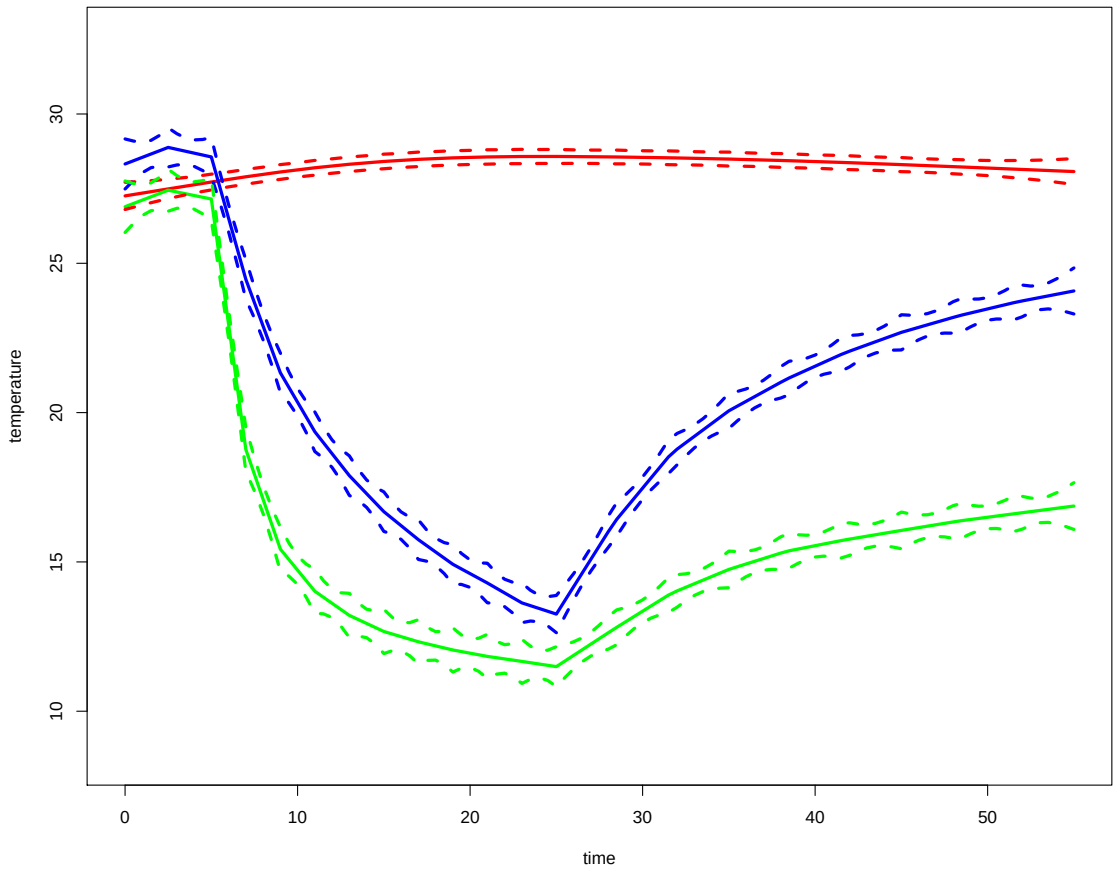

Figure 3.5: All three groups fitted simultaneously. The three thicker solid lines are the spline fits for each group. The top line (red) is the control group, the second line (blue) is the ice-pack group, and the bottom (green) line is the whirlpool group. The dashed lines are the confidence bands for each group.

4. CHALLENGES

Fitting splines with mixed models provides a fast and relatively simply method for fitting splines to data. There are some limitations to model fitted in this example. Some correlation due the repeated measures on the subjects was not accounted for. This correlation was ignored in order to simplify the problems and make it easy to understand. `R` makes this problem especially difficult to solve because the structure of the $\boldsymbol{R}$ matrix is difficult to specify independent of the $\boldsymbol{G}$ matrix. In theory, the correlation within each subject could be captured by specifying the correlation structure in the $\boldsymbol{R}$ matrix only; however, `R` applies the correlation structure in the $\boldsymbol{G}$ matrix. Using the `correlation=` argument in the `lme()` function with a grouped dataset creates a set of random effects for each respondent. Multiplying the already large number of random effects by the number of respondents would make this problem unwieldy. The constraint on the correlation structure is unique to the way `R` fits the model and is not a weakness of the method. It is possible that this could be overcome with a significant amount of programming or by using a different program such as `SAS`®.

Another limitation with mixed model splines is that the amount of smoothing can only be controlled by changing the number of knots or changing the distributional assumptions of $\boldsymbol{u}$. There is not a penalty parameter that can be tuned as there is in P-splines, but in the examples from this paper and the Wand (2003) paper that does not seem to create a problem. This may be because the mixed model shrinkage factor is calculated automatically from the model and this leads to an optimal fit. More research into this issue needs to be done before a conclusion can be reached as to what types of problems need additional control over the penalty.

In this paper `R` was used for fitting the models. `R` was chosen because it is widely

available and allows easy data manipulation. It has a strong mixed model estimation engine and is widely used among statisticians. The ease with which a researcher can manipulate matrices in `R` is one of its big advantages. Its weakness, however, is in the model specification. The `lme()` routine was not designed to allow direct specification of the $\boldsymbol{Z}$ matrix or to separately control the structure of the $\boldsymbol{R}$ and $\boldsymbol{G}$ matrices. This complicates the use of the program slightly. An alternative to `lme()` is currently being developed, but the replacement, `lmer()`, is even more restrictive and currently cannot fit mixed model splines. As development continues this may provide a better method. `SAS`® is another option for fitting mixed models, but data manipulation and the plotting require more programming than `R`. If specifying the desired model is much easier in `SAS`®, then it may be a better choice for some problems.

5. CONCLUSION

One of the biggest advantages of mixed model splines is that they are simply mixed models. There are no tricks or special considerations that need to be used to calculate the splines. Mixed model splines provide an easy method to incorporate the advantages of splines into a framework familiar to many researchers. The researchers can use all the techniques that they are familiar with to plot and analyze the splines. It is also possible to incorporate additional fixed and random effects in mixed model splines since the method does not rely on any special handling of the $\boldsymbol{X}$ or $\boldsymbol{Z}$ matrix. The only difficulty researchers face is in translating the model into the software. However, this is a difficulty that people familiar with the software are used to.

When fitting splines the model specification in R is straightforward and a direct extension of mixed models. Many different models can be used that combine the elements of both random effects and fixed effects. This paper demonstrated two models. The first fit a spline to a single treatment, the second simultaneously fit splines to three groups. Because mixed model splines are simply mixed models, the spline models can use the same inference procedures that standard mixed models use. Thus, fitting confidence bands is straightforward using standard mixed model inference procedures.

BIBLIOGRAPHY

Berry, S. M., Carroll, R. J., and Ruppert, D. (2002), "Bayesian smoothing and regression splines for measurement error problems," *Journal of the American Statistical Association*, 97, 160–169.

Brumback, B. A. and Rice, J. A. (1998), "Smoothing Spline Models for the Analysis of Nested and Crossed Samples of Curves," *Journal of the American Statistical Association*, 93, 961–976.

Buja, A., Hastie, T., and Tibshirani, R. (1989), "Linear Smoothers and Additive Models," *The Annals of Statistics*, 17, 453–510.

Cleveland, W. S. (1979), "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of the American Statistical Association*, 74, 829–836.

Davis, P. (1996), "B-Splines and Geometric Design," *SIAM News*, 29, accessed online Sept. 8, 2005.

de Boor, C. (1972), "On calculating with $\boldsymbol{B}$-splines," *J. Approximation Theory*, 6, 50–62, collection of articles dedicated to J. L. Walsh on his 75th birthday, V (Proc. Internat. Conf. Approximation Theory, Related Topics and their Applications, Univ. Maryland, College Park, Md., 1970).

— (2001), *A Practical Guide to Splines*, New York, NY: Springer-Verlag, revised edition ed.

Dierckx, P. (1993), *Curve and Surface Fitting with Splines*, New York, NY: Oxford Science Publications.

Diggle, P., Liang, K.-L., and Zeger, S. (2002), *Analysis of Longitudinal Data*, Oxford: Oxford University Press, second edition ed.

Eilers, P. H. C. (1999), "Comments on: The Analysis of Designed Experiments and Longitudinal Data by Using Smoothing Splines," *Applied Statistics*, 48, 307–308.

Eilers, P. H. C. and Marx, B. D. (1996), "Flexible Smoothing with B-Splines and Penalties," *Statistical Science*, 11, 89–121.

Epperson, J. (1998), "Numerical Analysis Digest 1998 post — History of Splines," [Online at `http://www.csc.fi/math_topics/Mail/NANET98-2/msg00228.html`; accessed October 25, 2005].

Eubank, R. L. (1999), *Nonparmetric Regression and Spline Smoothing*, Marcel Dekker, Inc., second edition ed.

Friedman, J. H. and Silverman, B. W. (1989), "Flexible Parsimonious Smoothing and Additive Modeling," *Technometrics*, 31, 3–21.

Gilmour, A., Gogel, B., Cullis, B., Welham, S., and Thompson, R. (2002), *ASReml User Guide Release 1.0*, Hemel Hempstead, HP1 1ES, UK: VSN International Ltd.

Good, I. J. and Gaskins, R. A. (1971), "Nonparametric Roughness Penalties for Probability Densities," *Biometrika*, 58, 255–277.

Gray, R. J. (1994), "Spline-Based Tests in Survival Analysis," *Biometrics*, 50, 640–652.

Green, P. and Silverman, P. (1995), *Nonparametric Regression and Generalized Linear Models: A roughness penalty approach*, Chapman and Hall.

Hastie, T. (1996), "Pseudosplines," *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 379–396.

Karlin, S. (1973), "To I. J. Schoenberg and his mathematics," *J. Approximation Theory*, 8, vi–ix, collection of articles dedicated to Isaac Jacob Schoenberg on his 70th birthday, I.

Kooperberg, C. and Stone, C. J. (1991), "A study of logspline density estimation," *Computational Statistics & Data Analysis*, 12, 327–347.

— (1992), "Logspline Density Estimation for Censored Data," *Journal of Computational and Graphical Statistics*, 1, 301–328.

Laird, N. and Ware, J. (1982), "Random-effects models for longitudinal data," *Biometrics*, 38, 963–974.

Lin, X. and Zhang, D. (1999), "Inference in Generalized Additive Mixed Models by Using Smoothing Splines," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61, 381–400.

Littell, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), *SAS®System for Mixed Models*, Cary, NC: SAS Institute Inc.

MathSoft Inc., . (2005).

Ngo, L. and Wand, M. (2004), "Smoothing with Mixed Model Software," *Journal of Statistical Software*, Volume 9, 1–54.

O'Sullivan, F. (1986), "A Statistical Perspective on Ill-Posed Inverse Problems," *Statistical Science*, 1, 502–518.

— (1988), "Fast computation of fully automated log-density and log-hazard estimators," *SIAM J. Sci. Statist. Comput.*, 9, 363–379.

R Development Core Team (2004), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, iSBN 3-900051-00-3.

Ramlau-Hansen, H. (1983), "Smoothing Counting Process Intensities by Means of Kernel Functions," *The Annals of Statistics*, 11, 453–466.

Ramsay, J. O. and Silverman, B. W. (2005), *Functional Data Analysis*, New York: Springer Science+Business Media, Inc., 2nd ed.

Robinson, G. K. (1991), "That BLUP is a Good Thing: The Estimation of Random Effects," *Statistical Science*, 6, 15–32.

Rupert, D. (2002), "Selecting the Number of Knots for Penalized Splines," *Journal of Computational and Graphical Statistics*, 11, 735–757.

Ruppert, D. and Carroll, R. (2000), "Spatially-adaptive penalties for spline fitting," *Australian and New Zealand Journal of Statistics*, 42, 205–224.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, New York, NY: Cambridge University Press.

SAS Institute Inc., . (2005).

Schoenberg, I. J. (1946), "Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae," *Quart. Appl. Math.*, 4, 45–99.

— (1964a), "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci. U.S.A.*, 52, 947–950.

— (1964b), "Spline interpolation and best quadrature formulae," *Bull. Amer. Math. Soc.*, 70, 143–148.

— (1964c), "Spline interpolation and the higher derivatives," *Proc. Nat. Acad. Sci. U.S.A.*, 51, 24–28.

— (1988), "A brief account of my life and work," *I.J. Schoenberg, Selected Papers*, 1, 1–10, accessed online Sept. 7, 2005, http://www.math.technion.ac.il/hat/fpapers/isobio.pdf.

Silverman, B. W. (1982), "On the Estimation of a Probability Density Function by the Maximum Penalized Likelihood Method," *The Annals of Statistics*, 10, 795–810.

Speed, T. (1991), "[That BLUP is a Good Thing: The Estimation of Random Effects]: Comment," *Statistical Science*, 6, 42–44.

Spencer, J. (1904), "On the graduation of rates of sickness and mortality," *Journal of the Institute of Actuaries*, 38, 334.

Swallow, W. H. and Monahan, J. F. (1984), "Monte Carlo Comparison of ANOVA, MIVQUE, REML, and ML Estimators of Variance Components," *Technometrics*, 26, 47–57.

Verbyla, A. P., Cullis, B. R., Kenward, M. G., and Welham, S. J. (1999), "The Analysis of Designed Experiments and Longitudinal Data by Using Smoothing Splines," *Applied Statistics*, 48, 269–311.

Wahba, G. (1990), *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics.

Wand, M. (2003), "Smoothing and mixed models," *Computational Statistics*, 18, 223–250.

Wang, Y. (1998a), "Mixed Effects Smoothing Spline Analysis of Variance," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60, 159–174.

— (1998b), "Smoothing Spline Models with Correlated Random Errors," *Journal of the American Statistical Association*, 93, 341–348.

Whittacker, E. (1923), "On a New Method of Graduation," *Proceedings of the Edinburgh Mathematical Society*, 41, 63–75.

Wikipedia (2005), "Spline (mathematics) — Wikipedia, the free encyclopedia," [Online; accessed August 2, 2005].

Table A.1: Control Data

Time	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Sub11	Sub12	Sub13	Sub14	Sub15
0	25.51	24.37	27.43	28.72	24.65	28.1	26.82	27.85	30.46	27.45	27.61	24.94	28.14	25.97	30.22
0.5	25.54	24.17	27.5	28.8	24.7	28.1	26.82	27.87	30.55	27.45	27.66	24.98	28.19	25.87	30.1
1	25.63	23.96	27.57	28.81	24.74	28.15	26.84	27.97	30.73	27.57	27.7	25.05	28.22	25.81	30.07
1.5	25.67	23.83	27.67	28.88	24.77	28.2	26.87	28.01	30.81	27.69	27.78	25.11	28.28	25.79	30.06
2	25.67	23.81	27.72	28.88	24.79	28.22	26.89	28.05	30.9	27.79	27.88	25.18	28.35	25.84	30.03
2.5	25.68	23.8	27.82	28.94	24.83	28.3	26.95	28.14	30.99	27.94	27.93	25.26	28.42	25.87	30.01
3	25.81	24.04	27.89	28.96	24.86	28.36	27.01	28.14	31.08	28.06	28.02	25.31	28.57	25.88	30.01
3.5	25.83	24.11	27.97	28.96	24.91	28.44	27.08	28.2	31.25	28.17	28.06	25.38	28.54	25.95	30.06
4	25.84	24.19	28.06	28.96	24.98	28.5	27.14	28.22	31.3	28.27	28.09	25.43	28.58	26.03	30.1
4.5	25.88	24.33	28.13	28.98	25.03	28.53	27.22	28.23	31.43	28.39	28.14	25.48	28.67	26.12	30.18
5	25.92	24.42	28.17	29	25.06	28.58	27.31	28.27	31.41	28.47	28.19	25.54	28.74	26.21	30.23
5.5	25.96	24.57	28.23	29.01	25.11	28.63	27.35	28.32	31.52	28.55	28.23	25.58	28.79	26.32	30.27
6	26.05	24.71	28.3	29	25.18	28.66	27.39	28.32	31.54	28.7	28.28	25.63	28.85	26.38	30.32
6.5	26.1	24.81	28.35	29	25.2	28.72	27.46	28.33	31.56	28.8	28.32	25.67	28.93	26.47	30.37
7	26.13	24.94	28.41	29	25.26	28.75	27.53	28.35	31.57	28.9	28.39	25.71	28.97	26.58	30.38
7.5	26.19	25.03	28.46	29.01	25.32	28.79	27.53	28.41	31.59	29.03	28.46	25.75	29.01	26.63	30.41
8	26.25	25.14	28.52	28.98	25.39	28.84	27.62	28.39	31.65	29.14	28.5	25.8	29.05	26.73	30.45
8.5	26.3	25.22	28.55	29	25.47	28.9	27.67	28.356	31.66	29.24	28.53	25.83	29.13	26.82	30.49
9	26.36	25.34	28.59	29	25.55	28.9	27.72	28.4	31.7	29.33	28.55	25.84	29.16	26.87	30.53
9.5	26.43	25.42	28.62	28.98	25.56	28.93	27.76	28.4	31.72	29.41	28.61	25.89	29.19	26.93	30.54
10	26.43	25.55	28.66	28.98	25.62	28.94	27.82	28.41	31.74	29.49	28.62	25.92	29.22	26.97	30.55
10.5	26.49	25.59	28.68	28.98	25.64	28.98	27.88	28.36	31.74	29.59	28.67	25.95	29.28	27.05	30.58
11	26.56	25.65	28.71	28.97	25.69	29.02	27.93	28.343	31.72	29.66	28.7	25.96	29.24	27.11	30.59
11.5	26.61	25.72	28.76	28.97	25.73	29.03	27.98	28.35	31.75	29.72	28.71	26	29.29	27.19	30.62
12	26.67	25.83	28.76	28.98	25.84	29.03	28.01	28.363	31.73	29.81	28.74	26.03	29.33	27.23	30.63
12.5	26.71	25.87	28.79	28.98	25.89	29.05	28.04	28.32	31.72	29.81	28.74	26.03	29.36	27.3	30.63
13	26.76	25.89	28.81	28.94	25.95	29.03	28.07	28.33	31.74	29.8	28.76	26.06	29.41	27.32	30.64

Time	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Sub11	Sub12	Sub13	Sub14	Sub15
13.5	26.82	25.99	28.84	28.97	26.01	29.05	28.1	28.3	31.75	29.83	28.8	26.08	29.4	27.37	30.63
14	26.87	26.05	28.85	28.96	26.08	29.06	28.1	28.27	31.72	29.81	28.81	26.1	29.42	27.43	30.64
14.5	26.93	26.09	28.87	28.97	26.12	29.03	28.1	28.26	31.69	29.86	28.84	26.1	29.4	27.48	30.66
15	27	26.14	28.87	28.94	26.14	29.06	28.1	28.26	31.7	29.92	28.84	26.12	29.41	27.53	30.66
15.5	27.04	26.18	28.87	28.94	26.22	29	28.13	28.23	31.67	29.99	28.88	26.13	29.45	27.57	30.67
16	27.02	26.22	28.87	28.92	26.23	29.1	28.11	28.23	31.69	29.99	28.9	26.13	29.45	27.61	30.67
16.5	27.05	26.23	28.88	28.9	26.25	29.11	28.15	28.23	31.66	29.97	28.9	26.14	29.45	27.65	30.67
17	27.09	26.27	28.9	28.92	26.28	29.06	28.18	28.2	31.64	29.98	28.92	26.15	29.45	27.71	30.68
17.5	27.14	26.34	28.9	28.9	26.31	29.09	28.19	28.19	31.71	29.99	28.9	26.15	29.42	27.71	30.68
18	27.14	26.39	28.9	28.88	26.36	29.09	28.2	28.18	31.7	30.09	28.9	26.18	29.44	27.74	30.68
18.5	27.19	26.41	28.93	28.84	26.39	29.09	28.2	28.17	31.65	30.09	28.9	26.19	29.42	27.76	30.66
19	27.26	26.45	28.92	28.85	26.44	29.06	28.19	28.1	31.64	30.05	28.93	26.21	29.41	27.78	30.63
19.5	27.26	26.52	28.94	28.87	26.48	29.03	28.22	28.1	31.61	30.06	28.94	26.19	29.41	27.83	30.62
20	27.28	26.56	28.93	28.84	26.54	29.01	28.19	28.05	31.59	30.1	28.94	26.22	29.4	27.83	30.6
20.5	27.3	26.6	28.93	28.81	26.57	29.02	28.18	28.07	31.54	30.07	28.94	26.23	29.4	27.84	30.63
21	27.34	26.63	28.93	28.81	26.58	29	28.2	28.07	31.56	30.14	28.98	26.23	29.37	27.87	30.68
21.5	27.37	26.67	28.93	28.79	26.65	29	28.19	28.05	31.5	30.14	28.98	26.26	29.36	27.93	30.68
22	27.4	26.74	28.96	28.78	26.67	28.98	28.2	28.02	31.48	30.11	28.98	26.23	29.35	27.91	30.71
22.5	27.43	26.75	28.96	28.76	26.67	28.94	28.18	28	31.41	30.19	28.97	26.25	29.31	27.96	30.73
23	27.46	26.78	28.98	28.76	26.7	28.93	28.17	27.96	31.42	30.19	28.97	26.26	29.28	27.97	30.73
23.5	27.46	26.8	28.96	28.75	26.71	28.89	28.13	27.96	31.35	30.18	28.97	26.26	29.27	27.98	30.73
24	27.46	26.83	28.93	28.74	26.76	28.85	28.1	27.93	31.33	30.2	28.97	26.25	29.26	28.01	30.76
24.5	27.53	26.82	28.94	28.72	26.8	28.83	28.13	27.92	31.28	30.2	28.97	26.25	29.22	28.01	30.72
25	27.56	26.83	28.98	28.71	26.84	28.83	28.13	27.91	31.12	30.22	28.93	26.26	29.24	28.05	30.71
25.5	27.58	26.84	28.97	28.76	26.88	28.83	28.13	27.89	31.18	30.23	28.97	26.27	29.2	28.07	30.68
26	27.6	26.86	28.97	28.74	26.93	28.81	28.14	27.89	31.18	30.2	28.97	26.27	29.22	28.07	30.67
26.5	27.62	26.87	28.98	28.7	26.97	28.79	28.15	27.84	31.12	30.18	28.93	26.26	29.19	28.09	30.66
27	27.65	26.88	28.97	28.68	27.02	28.78	28.17	27.85	31.12	30.15	28.93	26.25	29.19	28.07	30.63
27.5	27.65	26.87	28.96	28.66	27.05	28.79	28.17	27.83	31.07	30.1	28.93	26.26	29.16	28.09	30.64

Time	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Sub11	Sub12	Sub13	Sub14	Sub15
28	27.67	26.89	28.97	28.65	27.09	28.7	28.17	27.78	31.04	30.11	28.92	26.27	29.18	28.09	30.62
28.5	27.69	26.91	28.96	28.63	27.11	28.71	28.15	27.83	31.08	30.11	28.92	26.28	29.18	28.11	30.6
29	27.69	26.89	28.94	28.62	27.14	28.75	28.17	27.78	30.99	30.22	28.89	26.26	29.2	28.11	30.58
29.5	27.69	26.91	28.94	28.62	27.17	28.75	28.18	27.76	30.93	30.19	28.87	26.27	29.22	28.1	30.51
30	27.71	26.89	28.94	28.59	27.18	28.71	28.18	27.74	30.98	30.2	28.87	26.26	29.23	28.11	30.44
30.5	27.71	26.89	28.94	28.58	27.21	28.65	28.17	27.69	30.94	30.23	28.87	26.26	29.26	28.13	30.38
31	27.74	26.88	28.94	28.61	27.22	28.65	28.18	27.75	30.86	30.23	28.87	26.28	29.27	28.06	30.37
31.5	27.75	26.87	28.93	28.59	27.24	28.62	28.19	27.7	30.92	30.23	28.84	26.27	29.29	28.09	30.33
32	27.75	26.87	28.96	28.57	27.3	28.61	28.2	27.672	30.89	30.2	28.83	26.25	29.28	28.06	30.29
32.5	27.76	26.86	28.94	28.55	27.32	28.58	28.2	27.71	30.85	30.19	28.81	26.21	29.28	28.07	30.25
33	27.78	26.83	28.93	28.53	27.37	28.62	28.2	27.69	30.86	30.16	28.78	26.26	29.28	28.05	30.23
33.5	27.83	26.82	28.94	28.53	27.44	28.58	28.18	27.706	30.84	30.15	28.75	26.25	29.26	28.07	30.16
34	27.8	26.8	28.93	28.5	27.49	28.55	28.18	27.67	30.8	30.11	28.74	26.26	29.26	28.06	30.14
34.5	27.83	26.8	28.94	28.48	27.52	28.54	28.17	27.69	30.75	30.14	28.72	26.22	29.29	28.05	30.11
35	27.84	26.79	28.92	28.48	27.58	28.54	28.17	27.67	30.66	30.14	28.71	26.25	29.26	28.04	30.11
35.5	27.85	26.8	28.9	28.44	27.61	28.55	28.14	27.67	30.71	30.2	28.71	26.23	29.23	28.04	30.12
36	27.84	26.76	28.92	28.45	27.65	28.54	28.17	27.66	30.7	30.24	28.68	26.22	29.22	28.02	30.11
36.5	27.85	26.78	28.93	28.42	27.71	28.54	28.14	27.62	30.73	30.24	28.74	26.23	29.19	28.01	30.11
37	27.85	26.76	28.92	28.39	27.75	28.52	28.14	27.61	30.62	30.24	28.67	26.21	29.18	28	30.12
37.5	27.88	26.74	28.9	28.4	27.82	28.48	28.13	27.62	30.64	30.24	28.66	26.21	29.14	28	30.1
38	27.88	26.74	28.9	28.41	27.89	28.45	28.11	27.672	30.58	30.24	28.67	26.21	29.11	27.98	30.1
38.5	27.89	26.71	28.9	28.41	27.91	28.41	28.1	27.65	30.57	30.22	28.63	26.22	29.1	27.98	30.06
39	27.88	26.69	28.89	28.37	27.98	28.41	28.07	27.63	30.53	30.18	28.63	26.21	29.07	27.97	30.06
39.5	27.91	26.69	28.88	28.35	28.04	28.41	28.1	27.59	30.49	30.19	28.62	26.21	29.06	27.96	30.03
40	27.91	26.67	28.88	28.36	28.06	28.35	28.09	27.59	30.42	30.18	28.55	26.19	29.03	27.96	29.98
40.5	27.89	26.67	28.87	28.35	28.13	28.35	27.97	27.58	30.41	30.2	28.54	26.19	29.02	27.96	29.97
41	27.89	26.67	28.84	28.35	28.17	28.36	28.04	27.59	30.36	30.16	28.55	26.21	29	27.93	29.93
41.5	27.88	26.67	28.83	28.33	28.19	28.37	28.01	27.57	30.38	30.11	28.54	26.19	28.96	27.93	29.9
42	27.88	26.63	28.8	28.32	28.26	28.33	28	27.54	30.36	30.07	28.52	26.19	28.94	27.93	29.89

Time	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Sub11	Sub12	Sub13	Sub14	Sub15
42.5	27.89	26.66	28.78	28.32	28.32	28.33	27.96	27.48	30.33	30.03	28.53	26.19	28.93	27.91	29.9
43	27.87	26.65	28.75	28.3	28.35	28.31	27.93	27.45	30.31	29.99	28.52	26.18	28.89	27.91	29.86
43.5	27.87	26.62	28.74	28.3	28.37	28.28	27.88	27.46	30.32	29.94	28.49	26.18	28.88	27.89	29.83
44	27.87	26.61	28.74	28.24	28.41	28.27	27.89	27.46	30.31	29.85	28.49	26.18	28.87	27.89	29.85
44.5	27.92	26.61	28.74	28.24	28.44	28.26	27.85	27.46	30.24	29.84	28.44	26.17	28.83	27.88	29.84
45	27.89	26.57	28.72	28.23	28.45	28.2	27.84	27.44	30.23	29.77	28.42	26.15	28.8	27.89	29.83
45.5	27.88	26.57	28.7	28.2	28.46	28.18	27.82	27.46	30.2	29.77	28.42	26.18	28.8	27.85	29.83
46	27.84	26.57	28.68	28.18	28.46	28.17	27.8	27.44	30.19	29.77	28.42	26.17	28.79	27.83	29.79
46.5	27.85	26.54	28.66	28.17	28.48	28.13	27.79	27.41	30.16	29.84	28.4	26.15	28.75	27.83	29.77
47	27.87	26.53	28.67	28.17	28.49	28.1	27.76	27.41	30.12	29.86	28.37	26.14	28.74	27.82	29.79
47.5	27.84	26.53	28.65	28.2	28.49	28.08	27.71	27.43	30.1	29.85	28.33	26.15	28.72	27.8	29.77
48	27.87	26.49	28.63	28.11	28.52	28.06	27.69	27.37	30.05	29.84	28.33	26.15	28.68	27.8	29.76
48.5	27.85	26.51	28.61	28.11	28.52	28.04	27.67	27.37	30.01	29.83	28.31	26.14	28.66	27.8	29.77
49	27.85	26.48	28.59	28.11	28.54	28.05	27.66	27.39	29.96	29.8	28.3	26.13	28.63	27.82	29.77
49.5	27.85	26.47	28.58	28.09	28.53	28.06	27.63	27.36	29.96	29.76	28.28	26.14	28.61	27.79	29.77
50	27.88	26.43	28.57	28.07	28.54	28.06	27.58	27.3	29.94	29.75	28.27	26.13	28.62	27.79	29.77
50.5	27.88	26.47	28.55	28.09	28.57	28.02	27.58	27.27	29.93	29.74	28.26	26.12	28.54	27.8	29.77
51	27.88	26.44	28.55	28.05	28.59	28.02	27.54	27.24	29.93	29.7	28.27	26.12	28.49	27.82	29.77
51.5	27.88	26.44	28.54	28.02	28.59	27.98	27.52	27.272	29.85	29.67	28.24	26.12	28.45	27.79	29.76
52	27.85	26.48	28.5	28.01	28.59	27.96	27.48	27.259	29.83	29.66	28.26	26.09	28.42	27.8	29.77
52.5	27.85	26.44	28.5	28.01	28.59	27.96	27.48	27.245	29.89	29.68	28.23	26.08	28.24	27.79	29.79
53	27.84	26.44	28.49	28.02	28.61	27.94	27.45	27.252	29.81	29.68	28.18	26.1	28.33	27.82	29.75
53.5	27.83	26.44	28.49	27.98	28.62	27.93	27.41	27.19	29.79	29.64	28.2	26.09	28.32	27.84	29.76
54	27.82	26.44	28.48	27.97	28.62	27.89	27.43	27.24	29.76	29.62	28.18	26.06	28.28	27.8	29.75
54.5	27.84	26.43	28.46	27.96	28.65	27.91	27.4	27.22	29.75	29.59	28.17	26.06	28.26	27.79	29.76
55	27.85	26.41	28.45	27.94	28.67	27.85	27.41	27.22	29.71	29.55	28.15	26.08	28.23	27.78	29.75

Table A.2: Icepack Data

Time	Sub16	Sub17	Sub18	Sub19	Sub20	Sub21	Sub22	Sub23	Sub24	Sub25	Sub26	Sub27	Sub28	Sub29	Sub30
0	27.54	26.19	26.82	27.35	27.92	28.45	26.63	30.09	30.12	31.35	29.67	27.3	27.54	28.7	29.88
0.5	27.53	26.25	26.79	27.37	28.07	28.49	26.65	30.37	30.19	31.4	29.8	27.3	27.63	28.8	29.86
1	27.54	26.28	26.82	27.43	28.1	28.49	26.65	30.62	30.23	31.65	29.86	27.41	27.44	28.83	29.84
1.5	27.59	26.32	26.83	27.48	28.19	28.58	26.67	30.77	30.19	31.77	29.96	27.43	27.91	28.88	29.85
2	27.67	26.38	26.91	27.52	28.24	28.63	26.69	30.99	30.25	31.86	30.05	27.31	28.02	28.96	29.89
2.5	27.71	26.44	26.96	27.61	28.32	28.7	26.71	31.2	30.23	32.03	30.12	27.58	28.13	28.85	29.93
3	27.83	26.49	27.02	27.63	28.4	28.75	26.74	31.2	30.23	32.13	30.15	27.62	28.23	29.2	29.92
3.5	27.89	26.52	27.14	27.66	28.46	28.85	26.74	31.3	30.11	32.23	30.24	27.69	28.3	29.2	29.96
4	27.96	26.58	27.23	27.74	28.54	28.92	26.76	31.27	30.16	32.43	30.28	27.67	28.3	29.44	29.97
4.5	27.96	26.65	27.35	27.75	28.62	28.98	26.75	31.27	30.22	32.61	30.32	27.79	28.39	29.5	30.01
5	27.98	26.69	27.49	27.78	28.78	29.02	26.73	31.33	30.2	32.61	30.36	27.91	28.41	29.61	29.99
5.5	27.5	26.71	27.13	26.3	28.42	28.23	26.44	28.89	28.18	26.61	29.93	27.01	28.1	28.68	28.88
6	26.74	26.34	26.52	25.1	27.91	26.97	25.8	26.34	26.49	23.96	29.26	25.52	26.25	27.45	25.2
6.5	26.31	25.46	25.92	24.36	27.15	25.75	25.14	25.28	25.34	22.63	28.49	24.09	24.26	26.3	22.35
7	25.58	24.42	25.32	23.79	26.52	24.68	24.53	24.42	24.28	21.67	27.75	23.02	22.56	25.47	20.75
7.5	24.79	23.46	24.82	23.25	25.81	23.72	23.89	23.6	23.48	20.74	27.02	21.97	21.27	24.56	19.6
8	24.03	22.63	24.24	22.75	25.22	22.84	23.31	22.8	22.78	20.22	26.34	21.36	20.12	23.77	18.61
8.5	23.34	22.23	23.67	22.21	24.44	22.1	22.89	22.13	22.15	19.6	25.64	20.7	19.07	23.17	17.74
9	22.66	21.25	23.22	21.74	23.77	21.39	22.23	21.45	21.62	19.06	24.99	20.09	18.16	22.58	17.01
9.5	22.06	20.69	22.78	21.3	23.18	20.78	21.71	20.73	21.14	18.63	24.37	19.52	17.36	21.97	16.44
10	21.54	20.06	22.37	20.9	22.61	20.22	21.23	20.08	20.7	18.23	23.76	19.1	16.61	21.56	15.9
10.5	21.05	19.68	22	20.45	22.04	19.59	20.75	19.48	20.3	17.86	23.21	18.7	15.95	21.15	15.48
11	20.64	19.36	21.67	20.05	21.56	19.06	20.3	18.89	19.97	17.58	22.69	18.34	15.4	20.8	15.09
11.5	20.25	18.77	21.36	19.7	21.06	18.55	19.87	18.39	19.51	17.2	22.17	17.98	14.9	20.45	14.75
12	19.87	18.34	21.06	19.34	20.58	18.06	19.42	17.98	19.21	16.93	21.7	17.66	14.45	20.08	14.46
12.5	19.58	17.96	20.8	19.01	20.16	17.63	19.05	17.59	18.94	16.65	21.25	17.38	14.03	19.76	14.19

Time	Sub16	Sub17	Sub18	Sub19	Sub20	Sub21	Sub22	Sub23	Sub24	Sub25	Sub26	Sub27	Sub28	Sub29	Sub30
13	19.29	17.59	20.42	18.69	19.68	17.29	18.65	17.25	18.61	16.4	20.83	17.17	13.64	19.43	13.95
13.5	19.01	17.26	19.87	18.22	19.35	16.93	18.35	16.96	18.34	16.13	20.45	16.94	13.3	19.14	13.71
14	18.74	16.93	19.39	17.9	18.91	16.56	17.92	16.58	18.12	15.88	20.09	16.73	12.96	18.82	13.5
14.5	18.54	16.64	18.81	17.67	18.62	16.23	17.57	16.3	17.9	15.64	19.77	16.52	12.64	18.42	13.27
15	18.23	16.33	18.42	17.42	18.35	15.95	17.25	15.93	17.84	15.45	19.47	16.29	12.34	18.14	13.05
15.5	18	16.05	18.1	17.16	18	15.66	16.93	15.66	17.71	15.2	19.13	16.06	12.08	17.83	12.94
16	17.79	15.78	17.83	16.94	17.69	15.41	16.57	15.4	17.58	15.07	18.82	15.88	11.88	17.61	12.86
16.5	17.59	15.54	17.61	16.68	17.37	15.17	16.26	15.21	17.43	14.94	18.51	15.65	11.62	17.37	12.71
17	17.39	15.31	17.39	16.44	17.09	14.92	15.97	15.04	17.28	14.78	18.28	15.45	11.37	17.21	12.6
17.5	17.26	15.03	17.16	16.09	16.8	14.72	15.66	14.75	17.05	14.88	18.02	15.28	11.15	17.01	12.48
18	17.1	14.79	17.01	15.77	16.52	14.47	15.33	14.47	17	14.71	17.74	15.09	10.95	16.86	12.34
18.5	16.97	14.39	16.89	15.46	16.26	14.27	15.07	14.15	16.77	14.54	17.5	14.94	10.71	16.7	12.23
19	16.84	14.03	16.77	15.2	16.02	14.07	14.83	13.79	16.61	14.47	17.25	14.8	10.51	16.54	12.1
19.5	16.72	13.75	16.68	14.88	15.81	13.87	14.55	13.59	16.42	14.47	17	14.64	10.3	16.38	12
20	16.6	13.53	16.57	14.63	15.57	13.69	14.35	13.18	16.26	14.35	16.77	14.5	10.11	16.23	11.99
20.5	16.49	13.74	16.5	14.35	15.29	13.5	14.11	13.24	16.11	14.45	16.49	14.43	9.94	16.09	11.92
21	16.38	13.74	16.44	14.11	15.08	13.35	13.91	13.23	15.93	14.35	16.36	14.3	9.74	15.93	11.83
21.5	16.25	13.68	16.36	14.01	14.9	13.16	13.74	13.24	15.74	14.21	16.17	14.17	9.54	16.1	11.77
22	16.15	13.65	16.33	13.6	14.7	12.98	13.56	12.9	15.58	14.01	15.99	14.02	9.36	15.61	11.71
22.5	16.03	13.56	16.25	13.41	14.52	12.83	13.39	12.46	15.48	13.57	15.81	13.91	9.24	15.28	11.67
23	15.94	13.5	16.19	13.19	14.3	12.71	13.22	12.19	15.16	13.69	15.7	13.78	9.09	15.37	11.66
23.5	15.85	13.41	16.13	12.9	14.14	12.53	13.08	12.01	15.11	13.71	15.5	13.64	8.94	15.24	11.66
24	15.74	13.3	16.11	12.59	13.93	12.38	12.92	11.79	14.84	13.69	15.37	13.52	8.78	15.12	11.68
24.5	15.62	13.18	16.1	12.23	13.71	12.57	12.6	11.59	14.79	13.87	15.23	13.38	8.63	14.8	11.67
25	15.53	13.05	16.07	11.99	13.43	12.3	12.31	11.37	14.54	14.13	15.07	13.28	8.49	14.51	11.68
25.5	15.65	12.98	16.29	11.99	13.39	12.59	12.53	12.74	14.88	15.39	15.2	13.37	8.83	14.82	11.63
26	15.81	13.18	16.86	12.19	13.69	12.79	12.92	14.1	15.36	16.48	15.7	13.74	9.51	15.35	11.96
26.5	16.14	13.59	17.49	12.45	14.07	13.26	13.41	15.29	15.81	17.33	16.18	14.19	10.17	15.91	12.67
27	16.5	14.13	18.18	12.75	14.5	13.78	13.9	16.3	16.25	18.06	16.6	14.63	10.85	16.41	13.45

Time	Sub16	Sub17	Sub18	Sub19	Sub20	Sub21	Sub22	Sub23	Sub24	Sub25	Sub26	Sub27	Sub28	Sub29	Sub30
27.5	16.81	14.62	18.83	13.07	14.9	14.23	14.35	17.1	16.65	18.63	17.05	15.05	11.49	16.85	14.15
28	17.16	15.08	19.43	13.37	15.28	14.71	14.7	17.86	17	19.17	17.43	15.52	12.04	17.32	14.78
28.5	17.43	15.52	19.95	13.65	15.62	15.2	15.08	18.46	17.28	19.6	17.84	15.86	12.55	17.66	15.35
29	17.78	15.89	20.48	13.94	15.98	15.6	15.36	19.17	17.59	20.06	18.22	16.22	12.98	17.99	15.84
29.5	17.99	16.28	20.97	14.21	16.32	16.01	15.61	19.42	17.82	20.42	18.54	16.57	13.35	18.31	16.32
30	18.3	16.61	21.39	14.47	16.66	16.3	15.86	20.01	18.07	20.77	18.87	16.89	13.87	18.62	16.74
30.5	18.69	16.92	21.8	14.72	17.05	16.6	16.1	20.45	18.31	21.14	19.19	17.17	14.22	18.87	17.17
31	18.86	17.22	22.19	14.96	17.26	16.82	16.3	20.96	18.47	21.45	19.42	17.45	14.58	19.14	17.55
31.5	19.07	17.5	22.56	15.24	17.55	17.05	16.56	21.38	18.62	21.78	19.68	17.75	14.88	19.38	17.91
32	19.27	17.79	22.89	15.53	17.83	17.28	16.76	21.63	18.91	22.13	19.88	17.79	15.27	19.59	18.2
32.5	19.47	18.04	23.17	15.73	18.11	17.53	16.93	21.91	19.1	22.41	20.12	18.04	15.53	19.8	18.57
33	19.74	18.27	23.47	15.97	18.37	17.75	17.13	22.13	19.32	22.66	20.34	18.41	15.82	19.93	18.86
33.5	19.91	18.5	23.71	16.22	18.61	17.96	17.32	22.41	19.34	22.87	20.54	18.71	16.01	20.16	19.11
34	20.1	18.75	23.95	16.48	18.82	18.16	17.5	22.57	19.44	23.14	20.77	18.89	16.38	20.34	19.38
34.5	20.26	18.94	24.16	16.66	19.02	18.35	17.67	22.75	19.75	23.36	20.93	19.07	16.66	20.45	19.66
35	20.39	19.14	24.36	16.9	19.25	18.5	17.83	22.87	19.88	23.56	21.14	19.29	16.9	20.61	19.88
35.5	20.6	19.32	24.54	17.12	19.43	18.67	18.02	22.91	20.06	23.74	21.3	19.46	17.1	20.74	20.1
36	20.79	19.51	24.81	17.32	19.66	18.85	18.15	22.97	20.08	23.95	21.45	19.96	17.38	20.87	20.29
36.5	20.93	19.68	24.98	17.51	19.83	19.03	18.34	22.97	20.14	24.11	21.62	19.84	17.65	21	20.49
37	21.13	19.87	25.16	17.69	20.01	19.19	18.46	23.05	20.39	24.26	21.8	19.95	17.78	21.12	20.67
37.5	21.26	19.96	25.32	17.9	20.17	19.35	18.58	23.09	20.49	24.42	21.95	20.1	18.02	21.23	20.87
38	21.41	20.19	25.52	18.08	20.4	19.47	18.74	23.11	20.65	24.56	22.06	20.22	18.2	21.35	21.08
38.5	21.53	20.31	25.69	18.27	20.52	19.59	18.86	23.18	20.73	24.7	22.22	20.36	18.38	21.44	21.25
39	21.7	20.44	25.88	18.45	20.71	19.74	19.01	23.23	20.86	24.87	22.34	20.48	18.63	21.54	21.4
39.5	21.82	20.58	26.35	18.69	21.02	19.85	19.11	23.32	20.92	24.97	22.49	20.62	18.71	21.66	21.54
40	21.92	20.74	26.25	18.81	21.09	20	19.25	23.39	21.04	25.14	22.65	20.86	18.9	21.75	21.74
40.5	22.08	20.83	26.3	18.93	21.25	20.13	19.34	23.48	21.15	25.26	22.75	21.02	19.25	21.82	21.92
41	22.19	20.95	26.47	19.09	21.36	20.23	19.46	23.56	21.23	25.4	22.88	21.13	19.23	21.93	22.05
41.5	22.3	21.04	26.58	19.23	21.52	20.36	19.58	23.62	21.31	25.51	22.97	21.31	19.39	21.98	22.22

Time	Sub16	Sub17	Sub18	Sub19	Sub20	Sub21	Sub22	Sub23	Sub24	Sub25	Sub26	Sub27	Sub28	Sub29	Sub30
42	22.43	21.12	26.8	19.35	21.67	20.48	19.7	23.71	21.41	25.6	22.84	21.4	19.58	22.08	22.37
42.5	22.53	21.25	26.96	19.5	21.8	20.6	19.8	23.76	21.5	25.69	23.22	21.49	19.72	22.15	22.54
43	22.66	21.38	27.14	19.66	21.95	20.71	19.91	23.8	21.58	25.76	23.35	21.58	19.87	22.22	22.67
43.5	22.78	21.48	27.28	19.8	22.08	20.84	20	23.85	21.63	25.8	23.42	21.71	19.99	22.31	22.75
44	22.87	21.61	27.3	19.85	22.21	20.95	20.09	23.93	21.73	25.87	23.54	21.73	20.12	22.39	22.88
44.5	22.94	21.73	27.46	20	22.32	21.06	20.21	23.97	21.79	25.95	23.64	21.97	20.26	22.54	23.03
45	23.06	21.82	27.54	20.14	22.49	21.19	20.17	24.04	21.87	26.04	23.74	22.01	20.39	22.5	23.15
45.5	23.14	21.88	27.61	20.23	22.65	21.28	20.38	24.08	21.93	26.09	23.83	22.09	20.52	22.59	23.27
46	23.22	21.98	27.7	20.35	22.75	21.34	20.38	24.13	22.05	26.15	23.91	22.18	20.56	22.63	23.38
46.5	23.34	22.08	27.71	20.45	22.88	21.48	20.51	24.19	22.1	26.23	24.03	22.35	20.73	22.7	23.47
47	23.42	22.14	27.84	20.56	23.03	21.53	20.62	24.22	22.13	26.26	24.11	22.23	20.82	22.76	23.52
47.5	23.47	22.26	27.88	20.65	23.1	21.62	20.7	24.28	22.22	26.32	24.17	22.41	20.97	22.83	23.63
48	23.58	22.34	28	20.74	23.23	21.71	20.77	24.3	22.34	26.36	24.25	22.53	21.06	22.88	23.72
48.5	23.64	22.37	28.06	20.84	23.34	21.78	20.86	24.36	22.37	26.38	24.34	22.56	21.18	22.93	23.83
49	23.71	22.46	28.15	20.93	23.42	21.84	20.92	24.37	22.52	26.51	24.44	22.61	21.3	23.03	23.91
49.5	23.8	22.53	28.22	21.02	23.52	22	21	24.41	22.48	26.54	24.53	22.62	21.35	23.03	23.97
50	23.85	22.61	28.24	21.09	23.62	22.04	21.09	24.44	22.75	26.61	24.57	22.75	21.47	23.11	24.04
50.5	23.91	22.67	28.33	21.14	23.7	22.13	21.17	24.48	22.65	26.71	24.6	22.79	21.58	23.17	24.15
51	23.97	22.74	28.4	21.26	23.81	22.19	21.26	24.52	22.71	26.67	24.77	22.85	21.62	23.23	24.21
51.5	24.05	22.78	28.46	21.34	23.91	22.26	21.34	24.53	22.76	26.74	24.81	22.89	21.71	23.27	24.28
52	24.13	22.87	28.5	21.36	24.03	22.36	21.4	24.53	22.8	26.7	24.87	22.94	21.79	23.36	24.3
52.5	24.17	22.93	28.63	21.52	24.12	22.41	21.43	24.56	22.83	26.79	24.95	23.01	21.91	23.35	24.36
53	24.24	23.01	28.65	21.54	24.17	22.53	21.41	24.58	22.93	26.82	25.01	23.1	21.97	23.42	24.42
53.5	24.32	23.05	28.67	21.65	24.25	22.53	21.61	24.61	23.03	26.78	25.06	23.1	22.02	23.42	24.5
54	24.36	23.07	28.7	21.7	24.33	22.65	21.7	24.62	23.09	26.76	25.16	23.21	22.09	23.5	24.57
54.5	24.42	23.14	28.72	21.79	24.38	22.75	21.73	24.65	23.15	26.91	25.2	23.23	22.18	23.35	24.61
55	24.48	23.22	28.59	21.86	24.48	22.8	21.8	24.69	23.22	26.92	25.27	23.26	22.24	23.56	24.66

Table A.3: Whirlpool Data

Time	Sub31	Sub32	Sub33	Sub34	Sub35	Sub36	Sub37	Sub38	Sub39	Sub40	Sub41	Sub42	Sub43	Sub44	Sub45
0	26.12	29.75	26.23	24.82	29.01	26.09	28.62	29.44	24.4	26.98	27.18	23.72	28.07	25.83	28.24
0.5	26.08	29.88	26.32	24.87	29.09	26.12	28.72	29.41	24.49	27.09	27.15	23.63	28.13	25.88	28.28
1	25.99	29.99	26.41	24.95	29.18	26.15	28.79	29.54	24.62	27.1	27.14	23.58	28.24	25.99	28.36
1.5	26.06	30.1	26.52	24.97	29.26	26.18	28.85	29.63	24.69	27.18	27.1	23.56	28.28	26.05	28.42
2	26.4	30.15	26.57	25.28	29.32	26.19	28.93	29.75	24.78	27.21	27.21	23.56	28.41	26.14	28.52
2.5	26.31	30.2	26.63	25.18	29.4	26.53	29.06	29.81	24.83	27.28	27.31	23.58	28.44	26.23	28.61
3	26.4	30.22	26.69	25.22	29.4	26.31	29.14	30.01	24.93	27.27	27.32	23.64	28.5	26.48	28.65
3.5	26.57	30.24	26.78	25.55	29.51	26.45	29.22	30.05	25.06	27.34	27.35	23.59	28.53	26.4	28.68
4	26.56	30.01	26.67	25.56	29.58	26.35	29.26	30.07	25.09	27.39	27.35	23.6	28.5	26.47	28.76
4.5	27.21	30.13	26.86	25.54	29.66	26.41	29.31	30.27	25.23	27.4	27.36	23.62	28.61	26.54	28.92
5	26.73	30.24	26.52	25.35	29.83	26.54	29.36	29.98	24.98	27.17	27.43	23.64	28.61	26.61	28.85
5.5	25.14	29.15	23.97	23.25	28.52	25.87	27.33	28.13	20.17	22.96	26.05	21.63	26.98	25.52	26.88
6	22.44	27.02	20.53	20.03	25.38	22.7	25.01	24.58	17.04	20.38	22.94	18.74	23.59	23.19	24.39
6.5	20.62	25.55	18.37	17.95	22.82	20.12	22.15	22.09	15.49	18.85	20.79	16.93	21.21	21.41	21.58
7	19.29	24.71	16.93	16.56	20.96	18.26	20.19	20.34	14.51	17.95	19.02	15.66	19.59	20.08	19.58
7.5	18.19	23.89	15.84	15.57	19.59	16.89	18.71	18.99	13.98	17.34	17.74	14.79	18.38	18.98	18.1
8	17.29	23.06	14.99	14.78	18.47	15.84	17.51	17.94	13.6	16.81	16.76	14.1	17.42	18.08	16.97
8.5	16.54	22.36	14.39	14.14	17.54	15.03	16.64	17.06	13.2	16.29	15.93	13.53	16.72	17.46	16.06
9	15.97	21.86	13.89	13.63	16.81	14.39	15.93	16.4	12.83	15.84	15.39	13.12	16.14	16.86	15.34
9.5	15.66	21.21	13.43	13.2	16.18	13.87	15.35	15.81	12.53	15.46	14.92	12.78	15.58	16.33	14.74
10	15.41	20.6	13.05	12.87	15.65	13.42	14.86	15.33	12.19	15.13	14.55	12.49	15.17	15.89	14.23
10.5	15.07	19.93	12.75	12.57	15.2	13.07	14.38	14.96	12.01	14.84	14.29	12.25	14.78	15.5	13.88
11	14.84	19.42	12.46	12.33	14.83	12.74	14.07	14.64	11.79	14.56	14.02	12.04	14.46	15.15	13.53
11.5	14.64	19.11	12.22	12.1	14.48	12.45	13.76	14.39	11.75	14.27	13.74	11.93	14.18	14.82	13.19
12	14.48	18.83	12.01	11.99	14.18	12.23	13.5	14.15	11.78	14.05	13.52	11.79	13.97	14.54	12.93
12.5	14.39	18.55	11.9	11.82	13.9	12.03	13.27	13.97	11.77	13.83	13.31	11.66	13.74	14.3	12.7

Time	Sub31	Sub32	Sub33	Sub34	Sub35	Sub36	Sub37	Sub38	Sub39	Sub40	Sub41	Sub42	Sub43	Sub44	Sub45
13	14.25	18.27	11.78	11.7	13.67	11.89	13.08	13.76	11.68	13.68	13.12	11.53	13.53	14.07	12.52
13.5	14.06	18.07	11.63	11.6	13.46	11.74	12.89	13.59	11.6	13.54	12.93	11.44	13.35	13.89	12.36
14	13.91	17.86	11.52	11.51	13.28	11.6	12.72	13.45	11.51	13.39	12.75	11.34	13.18	13.71	12.2
14.5	13.76	17.63	11.42	11.42	13.13	11.52	12.57	13.33	11.44	13.34	12.6	11.26	13.01	13.54	12.01
15	13.67	17.45	11.36	11.34	12.96	11.38	12.45	13.24	11.37	13.28	12.46	11.21	12.87	13.38	11.88
15.5	13.6	17.28	11.29	11.27	12.82	11.29	12.34	13.16	11.26	13.24	12.31	11.15	12.74	13.26	11.73
16	13.48	17.09	11.21	11.21	12.68	11.21	12.22	13.04	11.21	13.16	12.22	11.11	12.64	13.16	11.62
16.5	13.35	16.98	11.15	11.18	12.57	11.14	12.15	12.94	11.17	13.11	12.14	11.08	12.56	13.04	11.57
17	13.24	16.89	11.11	11.12	12.45	11.07	12.07	12.89	11.19	13.02	12.04	11.06	12.49	12.93	11.49
17.5	13.09	16.7	11.04	11.08	12.34	11.04	11.99	12.83	11.19	12.96	11.97	10.99	12.41	12.83	11.43
18	12.98	16.57	10.99	11.04	12.26	11	11.97	12.75	11.06	12.86	11.93	10.95	12.33	12.74	11.4
18.5	12.89	16.36	10.97	11.02	12.15	10.96	11.92	12.7	10.99	12.78	11.86	10.92	12.26	12.68	11.33
19	12.82	16.17	10.93	11	12.08	10.92	11.85	12.63	10.95	12.7	11.82	10.89	12.19	12.59	11.31
19.5	12.75	15.97	10.89	10.96	12.01	10.89	11.79	12.57	10.84	12.63	11.74	10.86	12.18	12.53	11.28
20	12.67	15.8	10.86	10.92	12.01	10.86	11.77	12.53	10.78	12.56	11.68	10.84	12.12	12.44	11.23
20.5	12.63	15.64	10.84	10.93	11.94	10.84	11.71	12.49	10.73	12.49	11.63	10.8	12.05	12.38	11.21
21	12.56	15.52	10.82	10.89	11.88	10.82	11.67	12.44	10.67	12.42	11.55	10.8	12.01	12.34	11.17
21.5	12.53	15.41	10.8	10.89	11.83	10.81	11.6	12.36	10.65	12.4	11.52	10.8	11.99	12.26	11.15
22	12.49	15.33	10.77	10.84	11.77	10.78	11.55	12.3	10.59	12.36	11.44	10.76	11.99	12.18	11.12
22.5	12.44	15.2	10.76	10.82	11.71	10.77	11.51	12.25	10.55	12.31	11.42	10.76	11.93	12.11	11.07
23	12.41	15.08	10.73	10.8	11.68	10.76	11.48	12.19	10.52	12.26	11.4	10.71	11.89	12.07	11.05
23.5	12.37	14.97	10.71	10.78	11.64	10.71	11.42	12.15	10.54	12.22	11.37	10.69	11.88	12.03	11.08
24	12.22	15.03	10.69	10.77	11.6	10.7	11.4	12.08	10.52	12.21	11.36	10.67	11.85	12	11.04
24.5	12.15	15.04	10.67	10.76	11.56	10.7	11.36	12.01	10.5	12.16	11.33	10.65	11.79	12	11.01
25	12.07	15.01	10.63	10.84	11.52	10.7	11.33	12	10.5	12.16	11.32	10.65	11.78	11.97	10.97
25.5	12.26	14.64	10.7	10.99	11.51	10.76	11.53	12	10.97	12.3	11.41	10.69	11.79	11.99	11.02
26	12.44	14.71	11.04	11.17	11.67	10.86	11.63	12.21	11.14	12.33	11.42	10.85	11.97	11.99	11.25
26.5	12.77	14.83	11.02	11.37	11.81	11.06	11.86	12.48	11.27	12.59	11.67	11.03	12.16	12.23	11.52
27	13.08	14.95	11.21	11.53	11.97	11.26	12.01	12.74	11.48	12.89	11.94	11.22	12.4	12.41	11.81

Time	Sub31	Sub32	Sub33	Sub34	Sub35	Sub36	Sub37	Sub38	Sub39	Sub40	Sub41	Sub42	Sub43	Sub44	Sub45
27.5	13.34	15.15	11.51	11.73	12.18	11.47	12.27	13.01	11.68	13.15	12.15	11.47	12.63	12.7	12.01
28	13.57	15.31	11.52	11.92	12.37	11.68	12.46	13.28	12	13.41	12.36	11.62	12.85	12.89	12.3
28.5	13.8	15.5	11.66	12	12.56	11.89	12.66	13.54	11.94	13.64	12.57	11.82	13.07	13.09	12.55
29	14.06	15.58	11.79	12.16	12.77	12.03	12.92	13.78	12.26	13.87	12.82	12	13.26	13.3	12.84
29.5	14.27	15.9	11.91	12.31	12.96	12.21	13.04	14.05	12.38	14.09	13.02	12.15	13.49	13.49	13.04
30	14.54	16.03	12.03	12.46	13.11	12.42	13.26	14.25	12.59	14.29	13.18	12.3	13.65	13.68	13.19
30.5	14.7	16.06	12.14	12.66	13.31	12.6	13.45	14.45	12.75	14.48	13.35	12.44	13.83	13.82	13.48
31	14.79	16.19	12.25	12.71	13.45	12.77	13.65	14.59	12.9	14.67	13.5	12.57	14.01	13.99	13.66
31.5	14.91	16.29	12.34	12.85	13.64	12.92	13.84	14.75	13.02	14.83	13.67	12.72	14.17	14.13	13.86
32	15.03	16.41	12.42	13	13.78	13.08	14.02	14.92	13.18	15	13.82	12.85	14.31	14.26	14.04
32.5	15.19	16.46	12.46	13.08	13.91	13.23	13.98	15.08	13.28	15.13	13.99	12.97	14.48	14.38	14.18
33	15.29	16.56	12.56	13.23	14.06	13.39	14.37	15.24	13.39	15.27	14.07	13.08	14.63	14.5	14.33
33.5	15.42	16.64	12.68	13.33	14.18	13.52	14.45	15.37	13.5	15.41	14.23	13.18	14.75	14.63	14.49
34	15.54	16.69	12.77	13.48	14.29	13.6	14.71	15.44	13.67	15.56	14.35	13.31	14.84	14.74	14.66
34.5	15.66	16.78	12.87	13.59	14.41	13.74	14.78	15.57	13.76	15.65	14.46	13.42	14.97	14.84	14.82
35	15.74	16.85	13	13.68	14.51	13.84	14.91	15.68	13.97	15.73	14.54	13.54	15.07	14.94	14.94
35.5	15.84	16.92	13.11	13.78	14.59	13.98	15.03	15.78	14.03	15.85	14.63	13.68	15.16	15.05	15.08
36	15.94	16.98	13.48	13.84	14.7	14.23	15.09	15.86	14.13	15.93	14.74	13.8	15.27	15.15	15.2
36.5	16.22	17.05	13.61	13.91	14.78	14.18	15.23	15.97	14.25	16.02	14.82	13.89	15.36	15.24	15.31
37	16.14	17.08	13.49	14.01	14.91	14.27	15.35	16.05	14.37	16.1	14.92	13.95	15.46	15.32	15.43
37.5	16.23	17.16	13.56	14.1	14.95	14.38	15.45	16.11	14.46	16.19	14.99	14.01	15.56	15.4	15.56
38	16.32	17.2	13.65	14.17	15.07	14.46	15.56	16.14	14.55	16.28	15.09	14.06	15.62	15.49	15.63
38.5	16.41	17.28	13.75	14.26	15.13	14.56	15.66	16.22	14.5	16.37	15.17	14.09	15.68	15.54	15.72
39	16.5	17.32	13.74	14.27	15.19	14.64	15.74	16.28	14.68	16.44	15.27	14.13	15.77	15.62	15.84
39.5	16.57	17.36	13.71	14.26	15.29	14.72	15.8	16.33	14.7	16.5	15.35	14.17	15.84	15.69	15.9
40	16.68	17.41	13.69	14.39	15.32	14.82	15.89	16.38	14.78	16.56	15.41	14.27	15.85	15.76	15.99
40.5	16.69	17.32	13.69	14.46	15.45	14.9	15.95	16.44	14.86	16.62	15.46	14.33	15.89	15.84	16.05
41	16.74	17.37	13.69	14.48	15.52	14.99	16.02	16.53	14.9	16.73	15.54	14.41	15.95	15.91	16.1
41.5	16.8	17.37	13.72	14.56	15.58	15.09	16.13	16.54	14.96	16.74	15.58	14.46	16.02	15.98	16.14

Time	Sub31	Sub32	Sub33	Sub34	Sub35	Sub36	Sub37	Sub38	Sub39	Sub40	Sub41	Sub42	Sub43	Sub44	Sub45
42	16.86	17.38	13.8	14.62	15.74	15.15	16.09	16.6	15.04	16.78	15.66	14.54	16.07	16.03	16.22
42.5	16.9	17.39	13.78	14.7	15.76	15.23	16.1	16.64	15.09	16.98	15.69	14.62	16.13	16.07	16.24
43	17	17.34	13.8	14.74	15.81	15.32	16.11	16.66	15.07	16.86	15.78	14.7	16.18	16.14	16.26
43.5	17.08	17.39	13.87	14.82	15.86	15.37	16.17	16.66	14.99	16.93	15.8	14.79	16.26	16.19	16.32
44	17.17	17.45	13.95	14.86	15.94	15.42	16.23	16.69	15.12	16.96	15.81	14.84	16.3	16.22	16.34
44.5	17.29	17.49	14.02	14.87	16.02	15.46	16.25	16.72	15.21	17.04	15.85	14.88	16.36	16.28	16.4
45	17.39	17.47	14.14	14.95	16.11	15.53	16.26	16.74	15.25	17.08	15.88	14.92	16.4	16.3	16.38
45.5	17.51	17.47	14.18	15.23	16.21	15.6	16.22	16.74	15.28	17.1	15.9	15	16.46	16.34	16.41
46	17.62	17.51	14.22	15.03	16.3	15.65	16.29	16.77	15.33	17.16	15.95	15.01	16.53	16.4	16.42
46.5	17.71	17.51	14.42	15.05	16.42	15.7	16.34	16.78	15.4	17.21	16.02	15.07	16.6	16.44	16.46
47	17.83	17.54	14.29	15.11	16.5	15.76	16.4	16.8	15.42	17.24	16.01	15.11	16.66	16.46	16.5
47.5	17.94	17.53	14.33	15.09	16.61	15.88	16.42	16.81	15.49	17.28	16.05	15.11	16.72	16.52	16.53
48	18.03	17.61	14.39	15.16	16.69	15.84	16.42	16.82	15.58	17.29	16.09	15.11	16.76	16.57	16.56
48.5	18.12	17.57	14.43	15.2	16.81	15.89	16.49	16.82	15.6	17.34	16.1	15.09	16.84	16.86	16.61
49	18.23	17.57	14.47	15.27	16.86	15.93	16.52	16.84	15.66	17.37	16.13	15.21	16.93	16.69	16.63
49.5	18.31	17.54	14.47	15.32	16.92	15.93	16.53	16.84	15.76	17.39	16.15	15.19	16.97	16.76	16.69
50	18.41	17.42	14.51	15.45	17	15.99	16.58	17.06	15.74	17.39	16.18	15.5	17.02	16.82	16.74
50.5	18.5	17.58	14.55	15.35	17.09	16.03	16.69	16.81	15.8	17.41	16.21	15.17	17.09	16.97	16.81
51	18.59	17.53	14.59	15.46	17.14	16.05	16.68	16.76	15.81	17.42	16.25	15.23	17.16	16.93	16.89
51.5	18.65	17.51	14.71	15.45	17.22	16.07	16.7	16.7	15.85	17.43	16.26	15.24	17.22	16.98	16.95
52	18.74	17.5	14.84	15.53	17.29	16.06	16.72	16.72	15.84	17.66	16.21	15.25	17.29	17.06	17.06
52.5	18.83	17.29	14.94	15.56	17.39	16.13	16.7	16.72	15.81	17.53	16.23	15.28	17.37	17.12	17.12
53	18.91	17.47	15.05	15.56	17.46	16.21	16.7	16.72	15.88	17.57	16.29	15.31	17.46	17.17	17.19
53.5	18.97	17.49	15.15	15.58	17.53	16.19	16.72	16.7	15.88	17.57	16.36	15.31	17.53	17.24	17.28
54	18.99	17.46	15.31	15.6	17.57	16.25	16.74	16.69	15.91	17.62	16.29	15.32	17.59	17.55	17.34
54.5	19.05	17.46	15.36	15.6	17.65	16.29	16.74	16.66	15.94	17.63	16.32	15.32	17.62	17.34	17.42
55	19.03	17.45	15.46	15.62	17.73	16.3	16.74	16.68	15.98	17.69	16.3	15.33	17.71	17.39	17.48

B.1 One Group Solution

```
1  setwd('/Users/John/Documents/Thesis/Project/')
2  library(nlme)
3  #rm(list=ls())
4
5
6  index <- rep(1, 15) %x% seq(1, 111, 1)
7  control <- read.csv('control.csv', header=T)
8  control.tendon <- control[,seq(2, 30, 2)]
9
10 png(file="ControlGroup.png", bg="white")
11 plot(index*30/60, c(10, rep(35, 1664)), xlab="Minutes",
12 ylab="Temperature (Degrees Celcius)", main="Control Group",
13 type='n')
14 for(i in 1:15) {
15  lines(seq(0, 55, .5), unlist(control.tendon[i]), lty=i)
16 }
17 dev.off()
18
19 icepack <- read.csv('icepack.csv', header=T)
20 icepack.tendon <- icepack[,seq(2, 30, 2)]
21
22
23 png(file="IcePackGroup.png", bg="transparent")
24 plot(index*30/60, c(10, rep(35, 1664)), xlab="Minutes",
25 ylab="Temperature (Degrees Celcius)", main="Ice Pack Group",
26 type='n')
27 for(i in 1:15) {
28  lines(seq(0, 55, .5), unlist(icepack.tendon[i]), lty=i)
29 }
30 dev.off()
31
32 whirlpool <- read.csv('whirlpool.csv', header=T)
33 whirlpool.tendon <- whirlpool[,seq(2, 30, 2)]
34
35 png(file="WhirlpoolGroup.png", bg="transparent")
36 plot(index*30/60, c(10, rep(35, 1664)), xlab="Minutes",
```

```
ylab="Temperature (Degrees Celcius)", main="Whirlpool Group",
type='n')
for(i in 1:15) {
 lines(seq(0, 55, .5), unlist(whirlpool.tendon[i]), lty=i)
}
dev.off()



#Control
control <- read.csv('control.csv', header=T)
control.tendon <- control[,seq(2, 30, 2)]
control.temperature <- unlist(control.tendon)
subject <- seq(1, 15, 1)%x%rep(1, 111)
index <- rep(1, 15) %x% seq(1, 111, 1)
respondent <- 1:15 %x% rep(1, 111)
dummy <- rep(1, 1665)
control.X <- cbind(rep(1, length(index)), index)
source('../Splines/default.knots.sf')
knots <- default.knots(index)
control.Z <- outer(index, knots, "-")
control.Z <- control.Z*(control.Z>0)
control.grouped <- groupedData(control.temperature~index|dummy,
data=data.frame(index, control.temperature))
control.fit <- lme(control.temperature~-1 + control.X,
random=pdIdent(~-1+control.Z), data=control.grouped)
summary(control.fit)
control.beta.hat <- control.fit$coef$fixed

control.sig.eps.hat <- control.fit$sigma
control.sig.u.hat <-
control.sig.eps.hat*exp(unlist(control.fit$modelStruct))

#Plot of fit
num.grid <- 111
x.grid <- seq(min(index),max(index),length=num.grid)
X.grid <- cbind(rep(1,num.grid),x.grid)
Z.grid <- outer(x.grid,knots,"-")
Z.grid <- Z.grid*(Z.grid>0)
control.fhat.grid <- X.grid%*%control.beta.hat +
Z.grid%*%t(control.u.hat)
plot(index, c(10, rep(35, 1664)), xlab="Minutes",
ylab="Temperature (Degrees Celcius)", pch=subject, type='n')
for(i in 1:15) {
 lines(seq(1, 111, 1), unlist(control.tendon[i]), col='red',
```

```
 lty=i)
}
lines(x.grid,control.fhat.grid, col="red", lwd=3, lty=1)

#Icepack
icepack <- read.csv('icepack.csv', header=T)
icepack.tendon <- icepack[,seq(2, 30, 2)]
icepack.temperature <- unlist(icepack.tendon)
subject <- seq(1, 15, 1)%x%rep(1, 111)
index <- rep(1, 15) %x% seq(1, 111, 1)
dummy <- rep(1, 1665)
icepack.X <- cbind(rep(1, length(index)), index)
source('../Splines/default.knots.sf')
knots <- default.knots(index)
icepack.Z <- outer(index, knots, "-")
icepack.Z <- icepack.Z*(icepack.Z>0)
icepack.grouped <- groupedData(icepack.temperature~index|dummy,
data=data.frame(index, icepack.temperature))
icepack.fit <- lme(icepack.temperature~-1 + icepack.X,
random=pdIdent(~-1+icepack.Z), data=icepack.grouped)
summary(icepack.fit)
icepack.beta.hat <- icepack.fit$coef$fixed
icepack.u.hat <- unlist(icepack.fit$coef$random)
icepack.sig.eps.hat <- icepack.fit$sigma
icepack.sig.u.hat <-
icepack.sig.eps.hat*exp(unlist(icepack.fit$modelStruct))

#Plot of fit
num.grid <- 111
x.grid <- seq(min(index),max(index),length=num.grid)
X.grid <- cbind(rep(1,num.grid),x.grid)
Z.grid <- outer(x.grid,knots,"-")
Z.grid <- Z.grid*(Z.grid>0)
icepack.fhat.grid <- X.grid%*%icepack.beta.hat +
Z.grid%*%icepack.u.hat
plot(index, icepack.temperature, pch=subject, type='n')
for(i in 1:15) {
 lines(1:111, unlist(icepack.tendon[i]), lty=i, col='blue')
}
lines(x.grid,icepack.fhat.grid, col="blue", lwd=3, lty=2)


#whirlpool
whirlpool <- read.csv('whirlpool.csv', header=T)
whirlpool.tendon <- whirlpool[,seq(2, 30, 2)]
```

```
whirlpool.temperature <- unlist(whirlpool.tendon)
subject <- seq(1, 15, 1)%x%rep(1, 111)
time <- rep(1,15) %x% seq(0,55,.5)
dummy <- rep(1, 1665)
whirlpool.X <- cbind(rep(1, length(time)), time)
source('../Splines/default.knots.sf')
knots <- default.knots(time)
knots <- append(knots, 5, after=2)
knots <- append(knots, 25, after=12)
whirlpool.Z <- outer(time, knots, "-")
whirlpool.Z <- whirlpool.Z*(whirlpool.Z>0)
attr(whirlpool.Z, "dimnames") <- NULL
whirlpool.grouped <- groupedData(whirlpool.temperature~time|
dummy, data=data.frame(time, whirlpool.temperature))
whirlpool.fit <- lme(whirlpool.temperature~-1 + whirlpool.X,
random=pdIdent(~-1+whirlpool.Z), data=whirlpool.grouped)
summary(whirlpool.fit)
whirlpool.beta.hat <- whirlpool.fit$coef$fixed
whirlpool.u.hat <- unlist(whirlpool.fit$coef$random)
whirlpool.sig.eps.hat <- whirlpool.fit$sigma
whirlpool.sig.u.hat <-
whirlpool.sig.eps.hat*exp(unlist(whirlpool.fit$modelStruct))

#Plot of fit
num.grid <- 111*2
x.grid <- seq(min(time),max(time),length=num.grid)
X.grid <- cbind(rep(1,num.grid),x.grid)
Z.grid <- outer(x.grid,knots,"-")
Z.grid <- Z.grid*(Z.grid>0)
whirlpool.fhat.grid <- X.grid%*%whirlpool.beta.hat +
Z.grid%*%whirlpool.u.hat
png(file="WhirlpoolFit2.png", bg="transparent")
plot(time, whirlpool.temperature, pch=subject, type='n')
for(i in 1:15) {
 lines(unique(time), unlist(whirlpool.tendon[i]), lty=i)
}
lines(x.grid,whirlpool.fhat.grid, col="blue", lwd=3, lty=1)
abline(v=5)
abline(v = 25)

dev.off()
```

```
1  library(nlme)
2  setwd("~/Documents/Thesis/Project")
3
4  control <- read.csv('control.csv', header=T)
5  icepack <- read.csv('icepack.csv', header=T)
6  whirlpool <- read.csv('whirlpool.csv', header=T)
7
8  control <- control[seq(1, 111, 1),seq(2,30, 2)]
9  icepack <- icepack[seq(1, 111, 1),seq(2,30, 2)]
10 whirlpool <- whirlpool[seq(1, 111, 1),seq(2,30, 2)]
11 names(control) <- rep(" ", 15)
12 names(icepack) <- rep(" ", 15)
13 names(whirlpool) <- rep(" ", 15)
14
15 temperature <- rbind(t(t(unlist(control))),
16 t(t(unlist(icepack))), t(t(unlist(whirlpool))))
17
18 n <- length(temperature)
19
20 time <- rep(seq(0, 55, .5), 45)
21
22
23 K <- 20
24
25 knots <- quantile(unique(time), seq(0,1,length=K+2))[-c(1, K+2)]
26 #Adjust for treatment change points
27
28 knots.1 <- c(2.5, 5)
29 knots.2 <- quantile(c(5, 25), seq(0, 1, length=11))[-c(1)]
30 knots.3 <- quantile(c(25, 55), seq(0,1, length=10))[-c(1, 10)]
31 knots <- c(knots.1, knots.2, knots.3)
32 names(knots) <- NULL
33
34 index.control <- (1:(n/3))
35 index.icepack <- n/3 + (1:(n/3))
36 index.whirlpool <- 2*(n/3) + (1:(n/3))
37
38
39 int.control <- rep(0, n)
40 int.icepack <- rep(0, n)
41 int.whirlpool <- rep(0, n)
```

```
42
43  int.control[index.control] <- 1
44  int.icepack[index.icepack] <- 1
45  int.whirlpool[index.whirlpool] <- 1
46
47  X <- cbind(int.control, int.icepack, int.whirlpool,
48  int.control*time, int.icepack*time, int.whirlpool*time)
49
50  Z.overall <- outer(time, knots, "-")
51  Z.overall <- Z.overall*(Z.overall > 0)
52
53  Z <- cbind(Z.overall*int.control, Z.overall*int.icepack,
54  Z.overall*int.whirlpool)
55
56
57  block.ind <- list(1:K, (K+1):(2*K), (2*K+1):(3*K))
58  Z.block  <-  list()
59
60  for (i in 1:length(block.ind))
61      Z.block[[i]] <- as.formula(paste("~Z[,c(",paste(block.ind
62          [[i]],collapse=","),")]-1"))
63
64  dummy <- rep(1, n)
65  dimnames(temperature)[[1]] <- 1:n
66  dimnames(X)[[2]] <- c("control", "icepack", "whirlpool",
67          "time*control", "time*icepack", "time*whirlpool")
68  tendon <- groupedData(temperature~X|dummy,
69          data=data.frame(X, temperature))
70
71  fit <- lme(temperature~-1+X, data=tendon,
72          random=pdBlocked(Z.block, pdClass="pdIdent"))
73
74
75  beta.hat <- fit$coef$fixed
76  u.hat <- unlist(fit$coef$random)
77
78  sig.sq.eps <- fit$sigma^2
79  sig.sq.u   <- sig.sq.eps*exp(2*unlist(fit$modelStruct))
80
81
82  plot(time, temperature, type='n')
83
84  j <- 1
85  for(i in 1:15){
86          points(time[j:(j+110)], temperature[j:(j+110)], pch=".",
```

```
                cex=3, col="red")
        j <- j+111
}

for(i in 1:15){
        points(time[j:(j+110)], temperature[j:(j+110)],pch=".",
                cex=3, col="blue")
        j <- j+111
}

for(i in 1:15){
        points(time[j:(j+110)], temperature[j:(j+110)], pch=".",
                cex=3, col="green")
        j <- j+111
}

grid.control <- 1:111
grid.icepack <- (15*111+1):(16*111)
grid.whirlpool <- (30*111+1):(31*111)


X.grid <- X[c(grid.control, grid.icepack, grid.whirlpool),]
Z.grid <- Z[c(grid.control, grid.icepack, grid.whirlpool),]

fhat <- X.grid%*%beta.hat + Z.grid%*%u.hat

lines(time[1:111], fhat[1:111], col='red', lwd=3)
lines(time[(111+1):(2*111)], fhat[(111+1):(2*111)],
        col='blue', lwd=3)
lines(time[(2*111+1):(3*111)], fhat[(2*111+1):(3*111)],
        col='green', lwd=3)
```

B.3 Confidence Bands

```
################################
Confidence Bands

G <- sapply(pdMatrix(fit$modelStruct$reStruct), "*", fit$sigma^2)
G <- matrix(G, 60, 60)

G.inv <- solve(G)
```

```
R.inv <- diag(1/sig.sq.eps, 4995)


C11 <- t(X)%*%R.inv%*%X
C21 <- t(Z)%*%R.inv%*%X
C22 <- t(Z)%*%R.inv%*%Z+G.inv

Chat <- cbind(rbind(C11, C21), rbind(t(C21), C22))
Chat <- solve(Chat)

se.control <- sqrt(diag(cbind(X[grid.control,], Z[grid.control,])
%*%Chat%*%t(cbind(X[grid.control,], Z[grid.control,]))))



ul.control <- fhat[1:111] + (sqrt(2)*sqrt(qf(.95, 2, 8))
*se.control)
ll.control <- fhat[1:111] - (sqrt(2)*sqrt(qf(.95, 2, 8))
*se.control)

se.icepack <- sqrt(diag(cbind(X[grid.icepack,], Z[grid.icepack,])
%*%Chat%*%t(cbind(X[grid.icepack,], Z[grid.icepack,]))))

ul.icepack <- fhat[(111+1):(2*111)] + (sqrt(2)*sqrt(qf(.95, 2, 8))*se.icepack)
ll.icepack <- fhat[(111+1):(2*111)] - (sqrt(2)*sqrt(qf(.95, 2, 8))*se.icepack)

se.whirlpool <- sqrt(diag(cbind(X[grid.whirlpool,], Z
[grid.whirlpool,])%*%Chat%*%t(cbind(X[grid.whirlpool,], Z
[grid.whirlpool,]))))

ul.whirlpool <- fhat[(2*111+1):(3*111)] + (sqrt(2)*sqrt(qf(.95, 2, 8))*se.whirl
ll.whirlpool <- fhat[(2*111+1):(3*111)] - (sqrt(2)*sqrt(qf(.95, 2, 8))*se.whirl

lines(seq(0, 55, .5), ll.control, col='red', lwd=3, lty=2)
lines(seq(0, 55, .5), ul.control, col='red', lwd=3, lty=2)
lines(seq(0, 55, .5), ll.icepack, col='blue', lwd=3, lty=2)
lines(seq(0, 55, .5), ul.icepack, col='blue', lwd=3, lty=2)
lines(seq(0, 55, .5), ll.whirlpool, col='green', lwd=3, lty=2)
lines(seq(0, 55, .5), ul.whirlpool, col='green', lwd=3, lty=2)

```