Quantifying Climate Change:

Bayesian Model for Antarctic Surface Mass Balance

Philip Andrew White

A capstone report submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Dr. C. Shane Reese, Advisor

Department of Physics and Astronomy

Brigham Young University

March 2014

ABSTRACT

Quantifying Climate Change:
Bayesian Model for Antarctic Surface Mass Balance

Philip Andrew White
Department of Physics and Astronomy
Bachelor of Science

Because of the importance of Antarctic surface mass balance (SMB) in predicting sea level change, models are created to predict SMB on the Antarctic ice sheet. Using Favier et al.'s quality-controlled aggregate data set $N = 3529$, a fully Bayesian spatial model has been utilized to predict Antarctic SMB (Favier et al. 2013). Utilizing Markov random fields constructed through Gaussian process models, SMB is predicted over the entire Antarctic ice sheet. An SMB surface over the Antarctic ice sheet is computed by this model and compared with previous maps. An SMB prediction error surface is created to identify regions of high prediction uncertainty. This model estimates total SMB to be $1.75 \pm 0.335 \cdot 10^{12}\ \mathrm{m}^3 \cdot \mathrm{w.e} \cdot \mathrm{yr}^{-1}$ and mean SMB as $124.80 \pm 23.85\ \mathrm{mm} \cdot \mathrm{w.e.} \cdot \mathrm{yr}^{-1}$. These results suggest lower Antarctic water accumulation than previously purported. The calculated SMB surface showed more negative SMB regions and higher spatial variation than is likely plausible. Lastly, Antarctic boundary regions and areas with little data show high prediction uncertainty by the generated SMB prediction uncertainty surface.

Keywords: Bayesian statistics, climate modeling, Gaussian process, global warming, Metropolis-Hastings, spatial statistics, surface mass balance

ACKNOWLEDGMENTS

First, I want to thank my wife, Laurel. Without her love and support, I would have never been able to complete the research necessary to complete this project. She has been working on her master's degree as I have been in school. She has sacrificed time and energy in supporting me and taking care of our child, when she could have been working on her own endeavors. Any thanks I give her is truly insufficient for what she has done for me. Although my capstone deals with different research, I want to thank Dr. John Colton for helping me to get started with research. Working with him in his lab taught me to enjoy and appreciate research. I also want to thank Dr. Shane Reese for helping me to get involved in statistics research, as well as teaching and mentoring me. I cannot thank him enough for the personal help and attention that he has given me, especially for this project. I also would like to thank him, Dr. Christensen, and Dr. Tolley for encouraging me to pursue Statistics. I would also like to thank all those who read and helped edit my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Surface Mass Balance

Although climate change is one of the most hotly-debated issues in politics, its scientific foundations are well established. Climate change could cause drastic changes in growing seasons and regional climates. Given the consequences of severe climate change, the scientific community attempts to quantify climate change accurately in important areas like polar ice regions.

Antarctica is significant to the global climate because the water stored in its ice sheet would increase the sea level by about 200 feet if it were to melt, which would leave every coastal city under water. Even though radical climate change would not melt the entire Antarctic ice sheet for thousands of years, smaller, more realistic changes would still make a significant impact in the global climate, sea level, and growing seasons. For this reason, climatologists describe the changes in the Antarctic ice sheet in order to measure, predict, and anticipate potential global climate changes. Currently, surface mass balance is the primary measurement used to quantify changes in ice systems and can be used to predict sea level changes (Monaghan et al. 2006a; Van de Berg et al. 2006; Vaughan et al. 1999).

Surface mass balance (SMB) measures the difference between water accumulation and loss in ice systems. Thus, SMB indicates an ice body's net water gain at measurement locations. Ice

stakes are the most common way to measure SMB; they use markers to determine the relative water gain or loss at a location of interest. When coupled with ice density measurements, ice stake measurements accurately and reliably represent true SMB. Because of SMB's importance in predicting sea level change, people traverse Antarctica, normally by snowmobile, to install ice stakes and to take SMB measurements at stake locations (displayed in Figure 1.1 and 1.2). Besides ice stakes, ice core isotope content, volcanic material content, satellite altimetry, and wave propagation are used to measure SMB. The statistical reliability of these measurements is discussed in Section 1.4.

(a) Installing ice core to measure SMB **(b)** SMB measurement taken at ice core

Figure 1.1 Ice core installation for surface mass balance (SMB) measurement is expensive and physically demanding. Because many measurements are taken on site in a harsh climate, there is significant human error in all SMB measurement methods. ***Courtesy:*** *Summer Ruper, Department of Geology, Brigham Young University.*

Though water accumulation and loss vary seasonally because of snow build-up in the winter and snow melt in the summer, measurements taken over long periods of time help to eliminate seasonal variation. Without seasonal noise, one can integrate SMB over the whole system of interest,

$$\int_{system} \text{SMB}\, d\boldsymbol{A} = \text{Total SMB}, \tag{1.1}$$

where $\boldsymbol{A}$ is the area of the ice sheet. In words, the total SMB addresses the net water gain or loss over an entire ice system's surface. Total balance addresses an ice system's total gain and loss, not just what happens on the surface. This analysis is frequently applied to glaciers or ice sheets

(a) Snowmobiles used to traverse Antarctica

(b) Snow pit used to obtain SMB measurements.

Figure 1.2 Travel and measurement in Antarctica is time consuming, labor intensive, and physically demanding. Furthermore, some Antarctic regions cannot be reached for measurement because of difficult terrain. Thus, SMB data is sparse and unevely spaced. ***Courtesy:*** *Summer Ruper, Department of Geology, Brigham Young University.*

to explore whether that system is expanding or shrinking. Because total balance over an ice sheet indicates global changes in ice sheets, total balance allows us to make inferences about climate changes in the area. Antarctica, however, is unique because it loses almost all its ice mass through calving, the breaking off of chunks of ice from the edge of an ice sheet. Furthermore, the Antarctic ice sheet gains all of its water on its surface through snowfall and rain. For this reason, its surface is almost strictly an accumulation region. Because of how Antarctica gains and loses its mass,

$$\text{Total Mass Balance} = \text{Total SMB} - \text{Ice}_c, \tag{1.2}$$

where Ice_c is the ice lost through calving. Because SMB is essential for understanding global changes on the Antarctic ice sheet, scholars consider surface mass balance to be one of the most important measurements in analyzing Antarctic climate changes.

Besides predicting current SMB over the Antarctic ice sheet, climatologists are interested in describing how SMB changes over time. If regions that once accumulated snow now have net water loss, this would signal regional climate change. Furthermore, if scientists can detect significant temporal trends in SMB and predict corresponding global sea-level changes, the scientific

community could explain the rate of climate change more effectively. For these reasons, SMB measurements and modeling have garnered attention from the scientific community.

In this thesis, I first review literature sources that discuss previous SMB models, measurement reliability, and discuss challenges in detecting spatial and temporal trends in SMB. I then survey literature that examines important statistical aspects of our model. Next, I propose a fully Bayesian model for SMB and discuss our methods. Lastly, I contextualize the results of our model, explain their climatological significance, and suggest steps for further research.

1.2 Previous Surface Mass Balance Models

Vaughan et al.'s 1999 paper *Reassessment of the Net Surface Mass Balance in Antarctica* presented the first enduring model for surface mass balance. Because of irregular data spacing, estimating total surface mass balance on the Antarctic ice sheet is problematic. Models for surface mass balance have focused on eclectic factors including elevation, surface slope, air drainage routes, incoming lower-tropospheric flow, distance from the coast, and the distance to the seasonal sea-ice edge (Giovinetto and Bull 1987; Vaughan et al. 1999). For this reason, no unified model has been reached.

Due to developments in satellite imaging and microwave measurement technology, Vaughan et al.'s model attempted to overcome problematic data spacing by creating a background SMB field based on satellite measurements. The generated background field enabled them to interpolate between SMB measurement locations. Specifically, Vaughan et al. modeled surface mass balance MB_z as

$$MB_z = \left(a_o + \frac{a_1}{0.95 - T_b/T_m} \right) \cdot \exp(-5250/T_m) \cdot 10^{10}, \tag{1.3}$$

where T_b is brightness temperature in K, T_m is mean annual surface temperature in K, and $a_0 = -5.50$ and $a_1 = 6.50$. This model was first suggested by Zwally in 1977 and numerical solutions for a_0 and a_1 were suggested by Zwally and Giovinetto in 1995 as the best fit for data (Vaughan et al. 1999; Zwally 1977; Zwally and Giovinetto 1995).

Upon comparison to observed SMB measurements, they found $MB_{obs} = (0.81 \pm .002)MB_z + (35 \pm 5)$, where MB_{obs} is the observed mass balance. Thus, the correlation between MB_{obs} and MB_z was $r = .46$, suggesting that MB_z represents some, but not all, of the variation in the field observations (Vaughan et al. 1999). When compared to in-situ measurements, Vaughan et al.'s model found 5% error, which was more accurate than previous models. They also suggested that integrated net SMB values were higher than previously purported in accumulation regions. They found that the integrated surface mass balance over the grounded ice sheet is 1811 Gton yr^{-1} (149 kg $m^{-2}yr^{-1}$), and over the entire continent (including ice shelves) it is 2288 Gton yr^{-1} (166 kg $m^{-2}yr^{-1}$). These values were about 18% and 7% higher than the then-current estimates (Vaughan et al. 1999).

Beyond proposing much higher total balance than previous models, Vaughan et al.'s model also enabled computerized SMB contour map rendering because the background SMB field could be used to interpolate SMB values in data-poor regions. Prior to this, map rendering of Antarctic surface mass balance could only be done manually. Due to this difficulty, all previous maps portraying Antarctic surface mass balance ignored non-accumulation regions and only considered drainage basins. Vaughan et al.'s background SMB field was one of biggest steps forward in SMB modeling.

Rather than using satellite "brightness temperature" readings to interpolate between data points like Vaughan et al., Van de Berg et al. used weighted averages of nearby SMB measurements to predict SMB values at locations of interest. Van de Berg et al. limited the distance for which a data point could impact an SMB prediction to 55 km. In their model, nearby data were weighted more heavily than distant data. Formally, the weight $w_{g,o}$ of the model grid point, g, at the observation location, o, is

$$w_{g,o} = \alpha_o \cdot \max\left(0, 1 - \frac{dx_{g,o}}{dx_{max}}\right), \tag{1.4}$$

where $dx_{g,o}$ is the distance between model grid point and observation location and dx_{max} is the maximum distance which data would affect predictions. They account for the density of observa-

tions around to grid points, d_g. Using d_g, the model then calculates

$$d_o = \sum_g \max(d_g, 1) \cdot w_{g,o}. \tag{1.5}$$

Then, d_o is scaled by γ to give $\beta_o = d_o^{\gamma}$. Finally, SMB predictions are given by

$$\mathrm{SMB}_g = \frac{\sum_o \beta_o w_{g,o} \mathrm{SMB}_o}{\sum_o \beta_o w_{g,o}}, \tag{1.6}$$

where SMB_g is the predicted SMB at model grid location and SMB_o is the observed SMB at data locations.

Van de Berg et al.'s approach yielded a correlation between model and observation of $r = 0.82$, improving greatly upon Vaughan et al.'s result. When they extended maximum distance at which data would affect predictions to 193 km, they observed $r = 0.84$. Furthermore, they found that integrated SMB over the Antarctic ice sheet was more than 15% larger than previous estimates. However, it must be noted that Van de Berg et al. excluded coastal Marie Byrd Land, the Antarctic Peninsula, and Law Dome in their comparisons between observations and model because these regions have extreme SMB values (Van de Berg et al. 2006). Therefore, Van de Berg et al.'s success using weighted averages must be understood in the context of the problematic regions they excluded.

In 2006, Wingham et al. diverged from others' models by conditioning their SMB model upon satellite radar altimetry. Satellite altimetry enables researchers to observe changes in elevation, particularly changes due to relative annual water accumulation. Beyond annual elevation changes, elevation influences SMB as accumulation basins gain water from locations with higher elevation. Thus, elevation differences between locations are fundamentally important to SMB prediction. Unlike others discussed in this section, Wingham et al. focused on temporal trends in total SMB. Although 0 Gt yr^{-1} is included within their error bounds, (27 ± 29 Gt yr^{-1}), they suggest that this trend was slightly positive (Wingham et al. 2006).

Other models have taken micro, rather than macro, approaches to modeling SMB. To do this, they have utilized regional scale climate models to obtain accurate results when compared to in-situ measurements. However, in using a smaller model they limit the inferences that they can make

about the whole Antarctic ice sheet. Thus, overall trends are challenging to detect when using a smaller model (Gallée et al. 2011; Lenaerts et al. 2012).

When discussing global climate change, we, however, are interested in predictions over the whole continent. For this reason, the field aspects of Vaughan et al., Van de Berg et al.'s conditioning predictions upon nearby data, and elevation aspects similar to Wingham et al. are most applicable to the model we will present (Van de Berg et al. 2006; Vaughan et al. 1999; Wingham et al. 2006).

1.3 Challenges in Detecting Spatial Variability and Temporal Trends

To answer questions about climate change, scholars attempt to identify statistically significant climate trends. Many scholars have proposed incompatible trends and results. Thus, detecting temporal and spatial climate trends has proved elusive.

As discussed in Section 1.2, Wingham et al. performed SMB predictions using satellite altimetry measurements from 1992 to 2003 in an attempt to detect temporal trends in SMB measurements. When integrating over the ice sheet, they found a trend was close to zero (i.e. 27 ± 29 Gt yr^{-1}). Though they suggested that this trend was slightly positive, their result was not statistically significant since 0 Gt yr^{-1} is within their error bounds (Wingham et al. 2006).

In 2006, Monaghan et al. explored temporal trends and spatial patterns in Antarctic SMB. First, they examined Antarctic SMB trends on a regional scale. While they found both positive and negative temporal trends in SMB regionally, but over the whole continent they found no temporal trend in total SMB. They did, however, find substantial spatial and temporal variation in the Antarctic SMB. Furthermore, they attributed a large amount of the variation to seasonal effects (Monaghan et al. 2006b). Then, in a follow-up article, Monaghan et al. used snowfall accumulation to demonstrate that trend identification in certain regions, especially in west Antarctica and Victoria Land, was ineffective (Monaghan et al. 2006a). Since west Antarctica has one of the largest SMB accu-

mulation regions, which greatly affects Antarctic total mass balance, accurately detecting trends in total mass balance over the whole continent is difficult. Thus, if one cannot detect trends in these important regions, predictions over the full continent are likely to be biased.

In a similar analysis based upon regional climate models, Lenaerts et al. found a slightly negative temporal trend of -4.9 ± 0.1 and -5.7 ± 0.3 mm w.e. a^{-1} in total SMB over the Antarctic ice sheet (Lenaerts et al. 2012). Importantly, their results differed from Monaghan et al.'s and Wingham et al.'s conclusions. Because many scholars have suggested different trends, Bromwich et al. compared Antarctic SMB estimates to five data sets and found that many scholars detected incongruent trends because "reanalyses are known to be prone to spurious trends and inhomogeneities caused by changes in the observing system" (Bromwich et al. 2011). Bromwich et al.'s conclusions questioned the validity of many of the results proposed in previous works. Because scholars have made little progress in detecting temporal trends in SMB measurements, they have begun to reanalyze the reliability of their data and methods.

1.4 Reliability of Surface Mass Balance Measurement Methods

In 2007, Magand et al. analyzed SMB measurement reliability criteria to answer questions about SMB measurement methods. In their paper, they describe many SMB measurement methods and each measurement's reliability. They demonstrate the unreliability of short-term stake measurements, natural ^{210}Pb readings, stable isotope content, snow stratigraphy, and precipitation gauges for reasons that are outlined in the table in Table 1.1 (Magand et al. 2007). In the table, "C" rated data are deemed unreliable, "B" rated data are considered conditionally acceptable, and "A" rated data are judged to be reliable. Since many results were based on unreliable measurement methods, Magand et al.'s reliability criteria question the validity of these conclusions.

By applying Magand et al.'s reliability standards for quality control, Favier et al. created a data set that excluded data that were shown to be unreliable due to limited spatial and temporal representativeness, poor measurement accuracy, or quality control issues (Favier et al. 2013). Their

Table 1. Reliability and Applicability Conditions of SMB Measurement Methods[a]

SMB Measurement Methods	Applicability Conditions	Reliability		
		Annual	Multiannual	Decadal[b]
Anthropogenic radionuclides	dry snow facies, little mixing, absolute calibration and dating tool with reference horizon levels	/	A	A
Stake measurements	everywhere, annual and multiyear averaged SMB variability studies	C[c]	A	A
Natural ^{210}Pb	dry snow facies, little mixing, less accurate than anthropogenic radionuclides	/	/	B[d]
Stable isotope content and chemical markers	dry snow facies, Annual and multiyear averaged SMB variability studies, difficulty for clear observations in areas with very low SMB values (central Antarctic plateau), subjectivity in annual layers counting	/	B	B
Snow stratigraphy	dry snow facies, "low" reliability and accuracy	C	C	C
Precipitation gauges	not reliable, not accurate	C	C	C

[a]The methods deemed very reliable are accepted and given an "A." Methods judged reliable are conditionally accepted (marked "B"), while those that are deemed unreliable are marked "C."
[b]The term "decadal" means one or several decades.
[c]Applicable to single stakes and stake networks.
[d]The natural ^{210}Pb SMB method is reliable only from 4 to 5 decades (≈ two half-life time periods).

Table 1.1 Magand et al.'s Quality Rating[a] (Magand et al. 2007). These critiques questioned the reliablity of many previous measurements and the corresponding conclusions.

quality control rejected data taken by the unreliable techniques outlined by Magand et al. which are displayed and explained in Table 1.1. Furthermore, Favier et al. point out the inaccuracy of measurements associated with two snowmobile traversings, altimetry measurements in snow melt regions, and on steep slope terrain (Favier et al. 2013). Upon applying these quality ratings, Favier et al. called "A" rated data reliable and "B" and "C" rated data unreliable. The map in Figure 1.3 depicts data location and reliability.

1.5 Topics in Spatial Statistics

Because SMB measurement location is key to modeling SMB, we must consider how to represent spatial data. Coordinate systems are developed to uniquely grid the Earth's surface. However, whenever one attempts to map spherical locations onto a plane, one limits the way the spatial relationship between locations is interpreted. For example, even though latitudinal and longitudinal pairs uniquely represent points on the earth, these spatial coordinates fail to accurately represent distances between points in polar regions. For instance, compare the Mercator projection in which

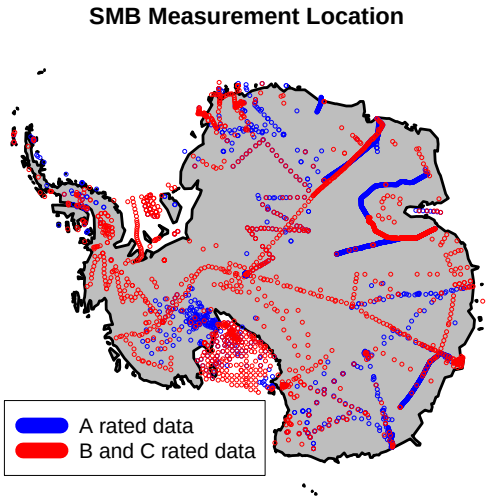


Figure 1.3 This map shows the location of all measurements in Favier et al.'s aggregate data set. All data have been given one of the following reliablity ratings: "A", "B", or "C." "A" rated data are deemed acceptable, "B" rated data are considered conditionally acceptable, while "C" rated are unacceptable. Blue represents "A" rated data and red represents "B" and "C" rated data.

latitude and longitude coordinate axes are linear on a plane, and the stereographic projection, which attempts to preserve the appearance of spherical objects on a plane, of Antarctica (see Figure 1.4).

Importantly, the extreme left and right locations on the Mercator projection are actually adjacent regions. Similarly, the whole bottom line in the Mercator projection represents the south pole, which is a single point. Thus, the Mercator projection obscures true spatial distance and scale for Antarctica. For this reason, when mapping polar regions, stereographic projections represent spatial relationships more accurately than the Mercator projection. To overcome the distortion in distance measurements made using latitudinal and longitudinal coordinates, great circle distance can be calculated between points by means of the Haversine formula, the Spherical Law of Cosines, or equirectangular approximations.

The distance between points of interest is one of the most important aspects of spatial statistics because "one of the fundamental attributes of spatial data is spatial autocorrelation: observations

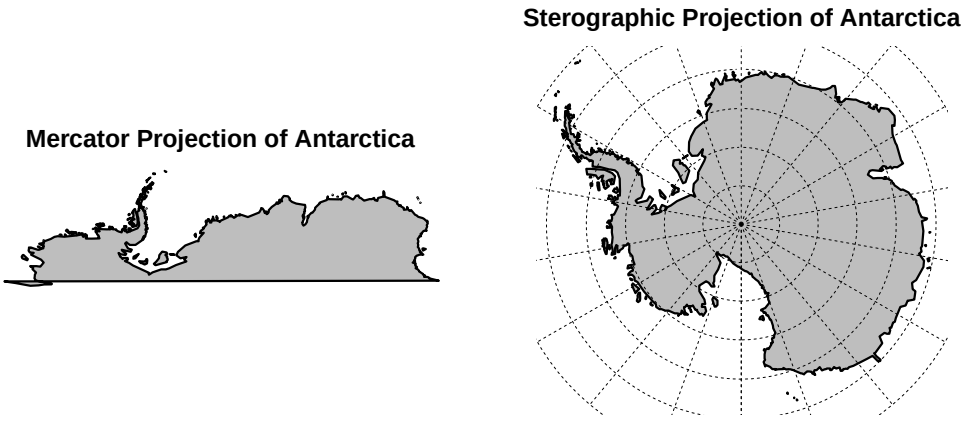


Figure 1.4 Comparison between Mercator and stereographic projections of Antarctica. Note that the Mercator projection obscures true spatial scale near the south pole. For this reason, using latitude longitude pairs to calculate distance inaccurately represent distance.

closer together tend to be more alike than observations farther apart" (Waller and Gotway 2004). In spatial statistics, autocorrelation for a spatial process$Z$, on a subset $S \subseteq \mathbb{R}^2$ is expressed in terms of a semivariogram,

$$\gamma(s_i - s_j) = \frac{\text{Var}(Z(s_i) - Z(s_j))}{2}, \tag{1.7}$$

where s_i, $s_j \in S$. A spatial process is called isotropic if the semivariogram is completely defined by the distance d between points of interest. Because isotropic processes are defined solely by the distance between locations, we write the semivariogram as $\gamma(d)$ instead of $\gamma(s_i - s_j)$.

Semivariograms are characterized by three quantities: the sill, the range, and the nugget. In general, $\lim_{d\to\infty} \gamma(d) = C$, where C is a constant called the sill. The first distance d such that $\gamma(d) = C$ is called the range. All distances greater than or equal to the range yield $\gamma(d) =$ sill and are uncorrelated with the reference point. The value $\gamma(0)$ is called the nugget. The nugget models variation in measurements between nearby locations (Waller and Gotway 2004). Physically, the nugget can be thought of as measurements noise, and importantly, $\gamma(0) \geq 0$. Figure 1.5 illustrates an isotropic semivariogram with a nugget $= 2$, sill $= 9$, and range ≈ 12.

Mathematically, the nugget can be thought of as an adjustment to a model's covariance matrix.

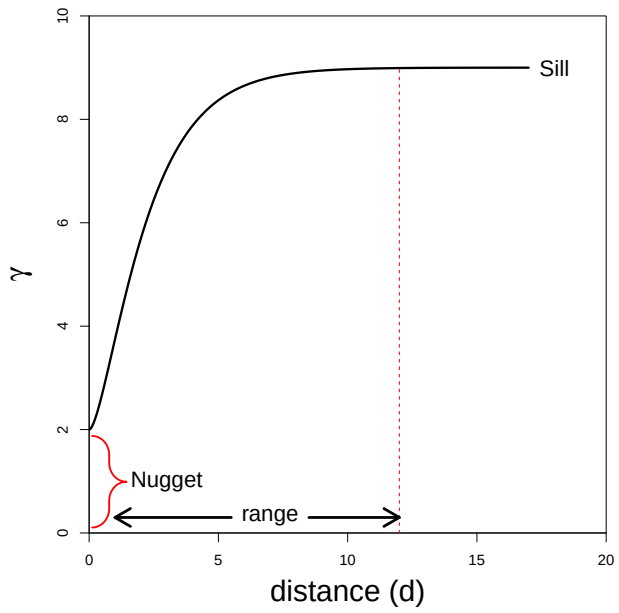


Figure 1.5 Semivariograms γ with arbitrary units of variance are used to represent spatial correlation. This is semivariogram has nugget = 2, sill = 9, range $\approx$ 12. The sill is the maximum value of γ, the nugget is the minimum of γ, and the range is the distance from the nugget to the sill. These terms and their spatial meaning is explained in detail in the text.

Specifically, the nugget is analogous to adding $\gamma(0)$ to the diagonal elements of the covariance matrix. Beyond its spatial interpretation, the nugget is sometimes used as a computational remedy for a non-positive-semidefinite covariance matrix. In this case, it is called a computational nugget. Covariance matrices diverge from positive-semidefiniteness due either to computational precision problems or the design points in the covariance matrix being ill-conditioned (Andrianakis and Challenor 2012). For example, when modeling a spatial process with the squared exponential covariance function

$$k(x,x') = \exp\left(\sum_{i=1}^{p}\left(\frac{d_i}{\delta_i}\right)^2\right), \tag{1.8}$$

where $d_i = |x - x'|$. Andrianakis and Challenor demonstrated that a computational nugget can introduce artificial modes in the estimating smoothing parameter δ when the true value of δ approaches extreme values (Andrianakis and Challenor 2012). However, this issue with the squared exponential covariance function is characteristic of its broader family of covariance functions. The

squared exponential covariance function is a special case of the Matérn covariance function; and the same issues can arise with the Matérn covariance function. Thus, if a model uses a computational nugget with a spatial process modeled with any special case of the Matérn covariance function, the nugget's effects must be monitored as it can induce artificial results in parameters of interest which could lead to problems in inference on target distribution.

Because each spatial process has a unique semivariogram, they are modeled by different covariance functions with appropriately corresponding characteristics. Because any choice of covariance function limits the autocorrelation relationships of the model, selecting a covariance function is vital for an accurate representation of a spatial process. For this reason, when little is known about the spatial process, generalized covariance functions, like the Matérn covariance function, can be used. The Matérn covariance function has the following form:

$$\sigma^2 \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{d}{\rho}\right)^{\nu} K_{\nu}\left(\frac{d}{\rho}\right). \tag{1.9}$$

Though introduced by Bertil Matérn to model autocorrelation in forestry statistics, Minasny and McBratney suggested the Matérn covariance function as a general model for spatial processes as it is a generalization of many covariance functions. The exponential, squared exponential, power, De Wijs, and Whittle covariance functions are special cases of the Matérn covariance function with specific parameters (Minasny and McBratney 2005). For this reason, the Matérn covariance function is a flexible, practical, and general choice for representing autocorrelation of spatial processes.

As discussed above, points separated by distances greater than the range are uncorrelated. So, the entries corresponding to $\gamma(d) =$ sill in a covariance matrix are $\Sigma_{ij} = 0$. Therefore, depending upon the range of the semivariogram, the covariance matrix could be sparse. In other words, there would be many $\Sigma_{ij} = 0$. For large sparse matrices, the Cholsky factorization, which decomposes matrices into 2 lower-triangular matrices, is significantly faster at inverting matrices than the QR decomposition, which uses an orthogonal and upper triangular decomposition and is the standard implementation in the programming language *R*. For this reason, Furrer and Sain argued that different computational methods should be instituted when working with sparse matrices (Furrer and Sain 2010). Therefore, when sparseness is known or expected, computational implementation

methods can significantly quicken computation time.

1.6 Gaussian Processes

Formally, Gaussian processes are "a natural generalization of the Gaussian distribution whose mean and covariance is a vector and matrix, respectively. The Gaussian distribution is over vectors, whereas the Gaussian process is over functions" or distributions (Rasmussen 2006). A Gaussian process is a stochastic process where any finite collection of random variables have a joint Gaussian distribution. A Gaussian process is fully specified by its mean function $m(x)$ and covariance function $k(x,x')$ (Rasmussen 2006). Gaussian processes are normally denoted by $\mathrm{GP}(m(x),k(x,x'))$ where the mean function defines the value about which the Gaussian process is normally distributed and the covariance function specifies the smoothness of the Gaussian process. However, since Gaussian processes are distributed over functions or distributions, Gaussian processes defined by the same $m(x)$ and $k(x,x')$ will be different but will have similar forms (see Figure]1.6). The following is an example of a Gaussian process with $m(x) = \mathbf{0}$ and $k(x,x') = \sigma^2 \cdot e^{\phi|x-x'|^2}$.

Because of their flexibility, Gaussian processes are an effective choice as a prior distribution in non-linear Bayesian models. By selecting Gaussian process priors, draws from the posterior distribution will be Gaussian processes that converge to the data by which they have been conditioned. Importantly, Gaussian process regression performs very well when interpolating and poorly with extrapolation. Gaussian processes are used widely in spatial statistics and machine learning.

1.7 Bayesian Statistics with Gaussian Process Prior

Bayesian statistics is a branch of statistics that makes use of Bayes' rule,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1.10}$$

to calculate the degree of belief in a hypothesis given observations; this is called the Bayesian probability. Bayesian statistics calculates a posteriori probability $P(A|B)$ by utilizing a priori prob-

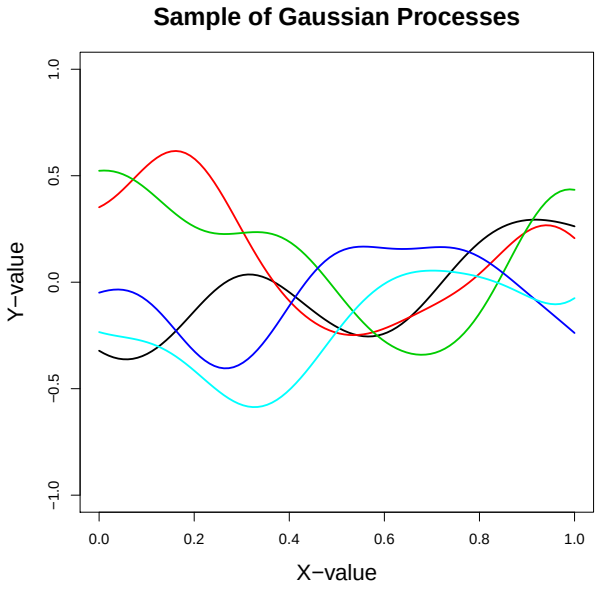


Figure 1.6 This is an example of five Gaussian processes that are generated using the same mean $m(x)$ and covaraince function $k(x,x')$. Note that each realization is different, but similar in form.

ability for A, $P(A)$ and a posteriori probability of B conditioned on A, $P(B|A)$ (Casella and Berger 1990). Rather than limiting its application to events A and B, Bayesian statistics extends Bayes' rule to distribution parameters $\boldsymbol{\theta}$ because Bayesian statistics views distribution parameters as random variables instead of fixed quantities. First, the prior distribution on $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta})$, is the analog to the a priori probability P(A). The prior distribution allows previous knowledge about distribution parameters to be incorporated into the model. The likelihood distribution $\mathscr{L}(\mathbf{y}|\boldsymbol{\theta})$ is the parallel of the a posteriori probability $P(B|A)$. Lastly, the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$, which is our desired quantity, corresponds to the a posteriori probability $P(A|B)$. By Bayes' rule, we arrive at the following:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\mathscr{L}(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{f(\mathbf{y})}, \tag{1.11}$$

where

$$f(\mathbf{y}) = \int_{-\infty}^{+\infty} \mathscr{L}(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{1.12}$$

Because $\int_{-\infty}^{+\infty} \mathscr{L}(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ integrates out all parameters $\boldsymbol{\theta}$, $f(\boldsymbol{y}) = C$, where C is some constant function of $\boldsymbol{y}$. Thus,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \mathscr{L}(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{1.13}$$

Because of this proportionality, we can sample from the joint distribution $\mathscr{L}(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ when the full posterior distribution cannot be sampled from directly.

By partitioning data $w_i \in W$ and prediction points $z_i \in Z$ we arrive at a joint distribution:

$$\begin{pmatrix} \boldsymbol{z}_p \\ \boldsymbol{w}_q \end{pmatrix} \sim \mathrm{MVN}_{p+q} \left[\begin{pmatrix} \boldsymbol{\mu}_p \\ \boldsymbol{\mu}_q \end{pmatrix}, \begin{pmatrix} \Sigma_{zz} & \Sigma_{zw} \\ \Sigma_{wz} & \Sigma_{ww} \end{pmatrix} \right], \tag{1.14}$$

where data and grid points are used to populate our covariance matrices Σ using a covariance function and MVN is the multivariate normal distribution. The form of the multivariate normal is explicitly shown in Appendix A.

With this joint distribution, we can utilize the conditional model that give us the following posterior distribution:

$$p(\boldsymbol{z}|\boldsymbol{w}) \sim MVN_p(\boldsymbol{\mu}_z + \Sigma_{zw}\Sigma_{ww}^{-1}(\boldsymbol{w} - \boldsymbol{\mu}_w), \Sigma_{zz} - \Sigma_{zw}\Sigma_{ww}^{-1}\Sigma_{wz}), \tag{1.15}$$

(Rencher and Schaalje 2008).

Though this is the form of the posterior distribution, if our parameters $\boldsymbol{\theta}$ are unknown, we cannot sample directly from the posterior distribution. Because $p(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto \mathscr{L}(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$, the posterior can be sampled from indirectly using the likelihood and prior distributions on $\boldsymbol{\theta}$ by Markov chain Monte Carlo (MCMC) methods.

In Bayesian Gaussian process models, the same process that was described above is used. We utilize Gaussian process priors such that

$$f(x) \sim \mathrm{GP}(\mathbf{0}, k(x, x')), \tag{1.16}$$

where the smoothness of the Gaussian process is defined by the covariance function $k(x, x')$. A corresponding likelihood function will be generated with the following parameters:

$$\mathscr{L}(\boldsymbol{y} \mid \boldsymbol{\theta}) \sim \mathrm{MVN}(f(x_i), \sigma^2). \tag{1.17}$$

Upon application of the conditional model, the posterior distribution is:

$$p(\boldsymbol{z}|\boldsymbol{w}) \sim \text{MVN}_p(\boldsymbol{\mu}_z + \Sigma_{zw}\Sigma_{ww}^{-1}(\boldsymbol{w} - \boldsymbol{\mu}_w), \Sigma_{zz} - \Sigma_{zw}\Sigma_{ww}^{-1}\Sigma_{wz}), \tag{1.18}$$

where Σs are populated by a covariance function $k(x,x')$ that regulates the covariance between points (Rasmussen 2006).

Figure 1.7 shows the appearance of Gaussian process priors and posteriors. Note how the Gaussian process posteriors converges to data. Importantly, the Gaussian process posteriors are very effective interpolators, but fail if used to extrapolate as can be seen by the boundary behavior of the Gaussian process posteriors.

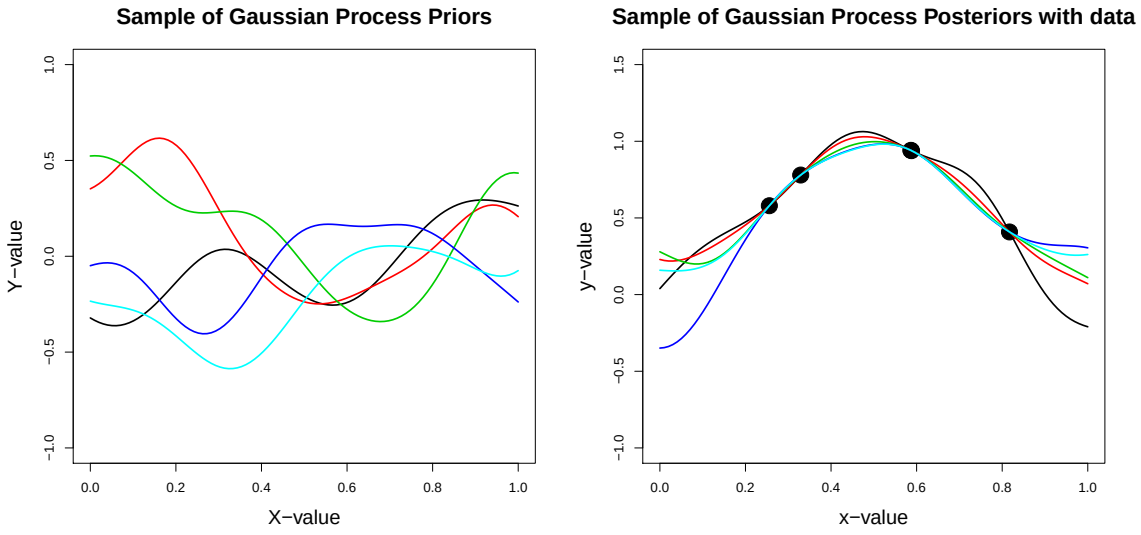


Figure 1.7 Example of five Gaussian process priors and posteriors. Note that the Gaussian process priors has similar form because they are generated from the same mean function $m(x)$ and covariance funtion $k(x,x')$. Gaussian posteriors conditioned on data converge to the data.

To perform inference, the posterior mean and credible interval are accessed using thousands of realizations from the posterior. Specifically, for every prediction location there are thousands of realizations $\boldsymbol{y}$. At each location, the posterior mean $\bar{y} = \text{mean}(\boldsymbol{y})$ and the credible interval is $\bar{y} \pm \text{qnorm}(0.975,0,1) \cdot \sqrt{\text{var}(\boldsymbol{y})}$, where $\text{qnorm}(0.975,0,1)$ is the 0.975 quantile of the standard normal distribution and $\text{var}(\boldsymbol{y})$ is the variance of the posterior realizations (Ebden 2008). To minimize

Monte Carlo error, many posterior realizations should be taken. Figure 1.8 is a complete regression example with posterior mean (blue) and 95% credible interval (red).

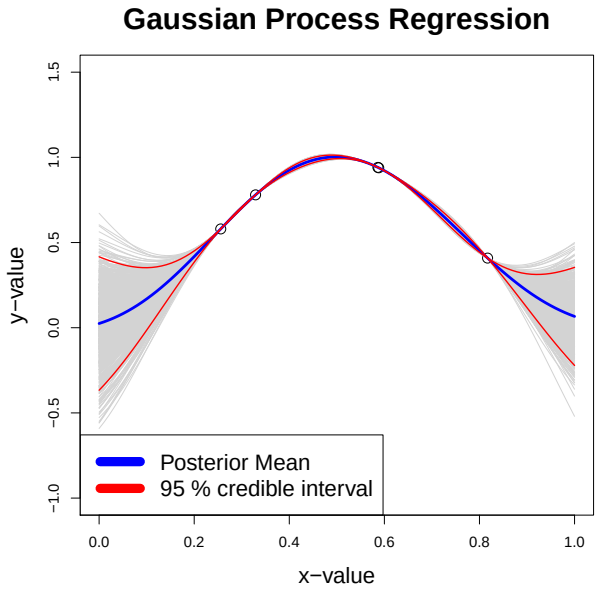


Figure 1.8 In Gaussian process regression, we take many draws from the conditional posterior distribution $p(\boldsymbol{z}|\boldsymbol{w})$ (gray). We then take the mean of the draws (blue) and solve for the 95% credible interval (red).

1.8 Markov Chain Monte Carlo Methods

Markov chain Monte Carlo (MCMC) methods are used to estimate a distribution when it cannot plausibly be sampled from directly. By definition, a Markov chain is a stochastic or random process that undergoes transitions between different states. The state transitions are made by a probabilistic model. Because the current state of a Markov chain affects what the next state will be, each state is statistically dependent and correlated with the previous state. While there are many variations on MCMC methods, we will discuss the Metropolis-Hastings algorithm and the Gibbs sampler.

1.8.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is an MCMC method used to sample from a target distribution which cannot be sampled from directly. Instead of the full posterior, the joint density $f(\boldsymbol{\theta}) = \mathscr{L}(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ is sampled from. A candidate distribution q from which we will make proposals for a variable of interest is selected. This algorithm can be used for many variables, but we will describe it for only one.

For some variable x we select an initial value x^0. Then, for iterations $i = 1, 2, ...$ complete the following steps:

1. Draw a proposal x^{cand} from the candidate distribution $x^{\text{cand}} \sim q(x^i|x^{i-1})$.

2. Calculate the acceptance ratio

$$\alpha = \frac{q(x^{i-1}|x^{cand})f(x^{cand})}{q(x^{cand}|x^{i-1})f(x^{i-1})}. \tag{1.19}$$

3. Draw $u \sim U(0,1)$.

4. If $u < \alpha$, then accept the proposal so $x^i \leftarrow x^{cand}$. Otherwise, reject the proposal such that $x^i \leftarrow x^{i-1}$.

In a special case of the Metropolis-Hastings algorithm, when q is symmetric, $q(x|y) = q(y|x)$, the acceptance ratio is

$$\alpha = \frac{f(x^{cand})}{f(x^{i-1})}. \tag{1.20}$$

This special case is called the Metropolis algorithm.

Unlike some samplers, like rejection sampling, high acceptance rates do not assure optimal mixing and convergence. High acceptance rates may indicate that the sampler is not exploring the entire support of the target distribution, f. Conversely, very low acceptance rates can indicate that the candidate draws are moving across the support too quickly and are missing important features of the target density (Robert and Casella 2004). Roberts et al. suggested that a 0.234 acceptance ratio is asymptotically optimal. It should be noted that each situation is different and that the ideal

ratio is a flexible guideline (Roberts et al. 1997). Interestingly, lower than ideal acceptance rates are preferable to acceptance rates that are higher by the same magnitude (Robert and Casella 2004).

The scale of the candidate distribution also impacts the acceptance rate as it controls the size of the jumps between candidate proposals. If the scale of the candidate distribution is too large, acceptance rates will be low and features of the target distribution may be poorly resolved. In contrast, if the scale of the candidate distribution is too small, the acceptance rate will be too high and the support of the target distribution will likely be left unexplored. Gelman et al. proposed that the ideal scale of the the candidate distribution is $\frac{2.4}{\sqrt{d}}$ times the scale of the target distribution where d is the dimension of the target distribution (Gelman et al. 1996). However, the scale of the target distribution is frequently unknown and can only be approximated prior to the algorithm.

1.8.2 Gibbs Sampler

The Gibbs sampler is a special case of the Metropolis-Hasting algorithm where every draw is accepted. Instead of sampling for all dimensions simultaneously, the Gibbs sampler samples for each dimension separately from the others. Because of the incremental variations, mixing occurs faster for the multidimensional Gibbs sampler than for the Metropolis-Hastings algorithm.

The Gibbs sampler relies on writing the the full or complete conditional for each variable θ_i for $i \in \{1,2,3,...,n\}$. Thus, we must know how the parameter θ_i is distributed. Expressly, the complete conditional is $p(\theta_i|\boldsymbol{\theta_j},\boldsymbol{y})$ where $j \in \{1,2,3,...,n\}$ such that $i \neq j$. The complete conditional is frequently written in the form $[\theta_i]$ where $[\theta_i] \equiv p(\theta_i|\boldsymbol{\theta_j},\boldsymbol{y})$

For all variables, $\boldsymbol{\theta}$ select initial values $\theta_i^{(0)}$ where the super script $^{(k)}$ indicates that the algorithm is on iteration k. For each iteration, k in $k \in 1,2,3,....$ we compute the following:

- generate $\theta_1^{(k)} \sim \left[\theta_1^{(k)}\right]$ where $\left[\theta_1^{(k)}\right]$ is the complete conditional of $\theta_1^{(k)}$.
- generate $\theta_2^{(k)} \sim \left[\theta_2^{(k)}\right]$ where $\left[\theta_2^{(k)}\right]$ is the complete conditional of $\theta_2^{(k)}$.

  $\vdots$

- generate $\theta_n^{(k)} \sim \left[\theta_n^{(k)}\right]$ where $\left[\theta_n^{(k)}\right]$ is the complete conditional of $\theta_n^{(k)}$

This process is repeated many times. Note that the complete conditional updates using the most recent value of θ_i. The convergence of the Gibbs sampler to the target density, in this case the joint distribution, was shown by Casella et al. (Casella and George 1992). Because the Gibbs sampler converges to the joint distribution and $p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \mathscr{L}(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, the Gibbs sampler can be used to generate posterior distribution draws since it is proportional to the joint posterior.

Chapter 2

Methods and Models

Many different groups have modeled Antarctic SMB because SMB is currently the best metric to track changes in ice sheets (Monaghan et al. 2006a; Van de Berg et al. 2006; Vaughan et al. 1999). Most models for SMB, however, have had two major shortcomings: they have not used advanced statistical methods to interpolate SMB in data-poor regions and they have based their conclusions on unreliable data. Because travel, and thus data acquisition, is difficult on the Antarctic Ice Sheet, most data are clustered and there are many data-poor regions (as shown in Figure 2.1). Thus, statistical interpolation is the only way to comprehensively analyze SMB over the whole surface of Antarctica.

Projecting the future of the Antarctic ice sheet is especially important to geophysicists and climatologists because the Antarctic ice sheet can dramatically affect sea levels. However, scientists have not found statistically significant trends (Monaghan et al. 2006a; Wingham et al. 2006). By using more reliable data and new methods, our research goals are to first, to construct Markov random fields through Gaussian process models for the SMB in Antarctica to interpolate between SMB measurement locations, and second, to quantify our prediction uncertainty. These methods enable us to model SMB and our prediction uncertainty over the entire Antarctic ice Sheet.

2.1 Description of Data Set

Using the reliability critiques suggested by Magand et al., Favier et al. compiled about 5500 data from over 90 sources. For all data, a reliability rating of "A," "B," or "C" was given depending upon the method and the duration of measurement (Favier et al. 2013; Magand et al. 2007). To decrease measurement uncertainty, we have chosen to include only "A" rated data points in our model. This selection criterion reduces available data from 5564 to 3529 (the "A-rated" data are plotted in Figure 2.1). Most previous models have compared their model prediction to fewer than 1000 in-situ measurements for verification (Lenaerts et al. 2012). Furthermore, many models have verified their SMB predictions using unreliable data (Bromwich et al. 2011). Also, many models did not compare their predictions to regions with anomalous surface mass balance values (Van de Berg et al. 2006; Vaughan et al. 1999). We, on the other hand, will compare our model predictions with all 3529 "A-rated" data regardless of the region where the measurement was taken. In the data set, important information like elevation was missing. Because elevation is important quantities in projecting SMB values, we will use geospatial databases to query the elevation for coordinates with missing elevation data Wingham et al. (2006).

Using the reliability critiques suggested by Magand et al., Favier et al. compiled about 5000 data from over 90 sources. For all data, a reliability rating of "A," "B" or "C" was given depending upon the method and the duration of measurement (Favier et al. 2013; Magand et al. 2007). To decrease prediction uncertainty, we have chosen to include only "A-rated" data points in our model. This selection criterion reduces available data from 5564 to 3529 (the "A-rated" data is plotted in Figure 2.1).

Most previous models have compared their model prediction to fewer than 1000 in-situ measurement data for verification (Lenaerts et al. 2012). Furthermore, many models have verified their SMB predictions using unreliable data (Bromwich et al. 2011). Also, many models did not compare their predictions to regions with anomalous surface mass balance values (Van de Berg et al. 2006; Vaughan et al. 1999). We, on the other hand, will compare our model predictions with all 3529 "A-rated" regardless of the region where the measurement was taken. In the data set, impor-

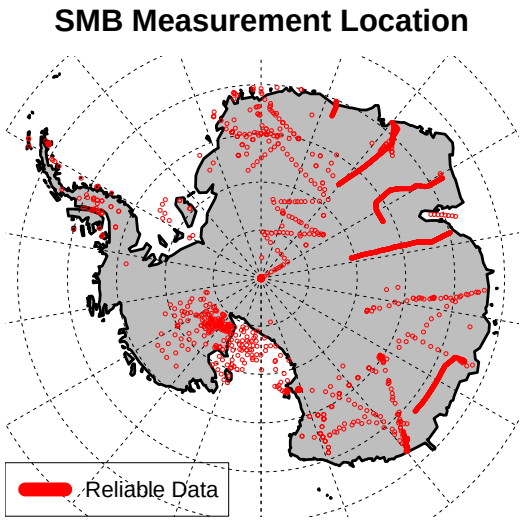


Figure 2.1 This map shows the location of "A" rated measurements in Favier et al.'s aggregate data set. All data in this data set has been given on of the following reliablity ratings: "A," "B," or "C." "A-rated" data is deemed acceptable, "B-rated" data is considered conditionally acceptable, while "C-rated" is unacceptable.

tant information like elevation was missing. Because elevation is important quantities in projecting SMB values, we will use geospatial databases to query the elevation for coordinates with missing elevation data (Wingham et al. 2006).

2.2 Distance Model

Previous SMB models on Antarctica have relied on Euclidean distances when interpolating model-simulated SMB. Utilization of Euclidean distances are potentially problematic when the distance spans bays. When locations are separated by a bay, Euclidean distances will exaggerate the proximity of these locations. If proximity is exaggerated, predictions in peninsular regions will be inaccurate. For this reason, several models excluded peninsular regions in their models to obtain a better fit to measurements on the remainder of the Antarctic ice sheet (Van de Berg et al. 2006; Vaughan et al. 1999). We will attempt to remedy this problem by adjusting euclidean distance by

including elevation in our distance metric.

We will also be using euclidean distance by means of the haversine formula which calculates great circle arc length, however we will augment this distance using difference in elevation between points to more accurately capture the spatial distance between prediction locations. Great circle arc length is the shortest distance between two points on the surface of a sphere, though on a Mercator projection it appears longer than other routes (as illustrated in Figure 2.2).

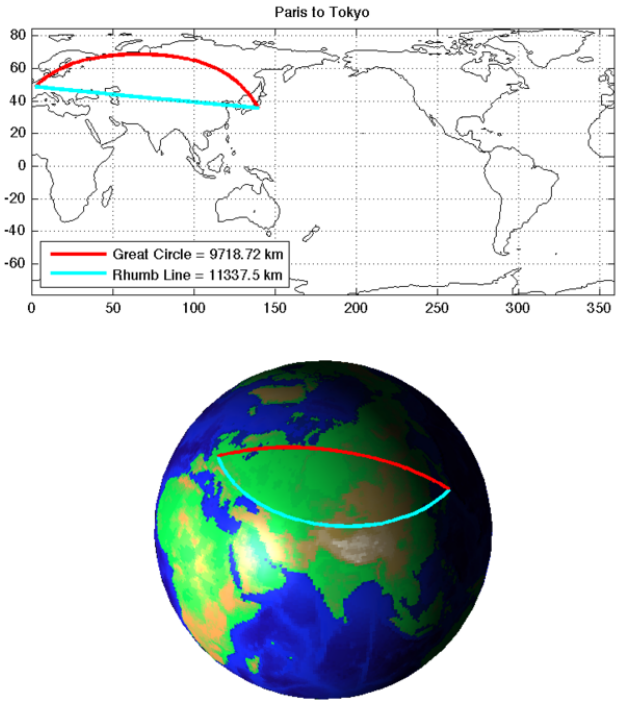


Figure 2.2 Great circle distance is the shortest path between two points on a sphere and is important in modeling distance between points on the earth. The red line represents great circle distance between Paris and Tokyo

The haversine distance d is calculated as follows:

$$d = 2R \arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \tag{2.1}$$

where $a = \sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_1)\cos(\phi_2)\sin(\frac{\Delta\lambda}{2})$

and, $\Delta\phi$ is the change in latitude between points,

$\Delta\lambda$ is the change in longitude between points,

ϕ_1 and ϕ_2 are the latitudes of the first and second points.

As stated above, we will augment the haversine distance to an $\mathbb{R}^3$ distance that it includes change in elevation between data points. For every prediction point, we will query elevation using an elevation database that has 200 m grid-coarseness. Namely, we will find the grid location that is closest to our prediction point using the haversine distance and will then set the elevation of the prediction point to be the elevation of the nearest grid point. Then, by scaling the difference in elevation ΔE and haversine distance d between locations such that both have minimum of 0 and maximum of 1, we will weight the distance and elevation equally. We will then compute the augmented distance ($d_{\mathbb{R}^3}$) as follows:

$$d_{\mathbb{R}^3} = \sqrt{d^2 + \Delta E^2}. \tag{2.2}$$

By including elevation changes in the model, we will more accurately represent spatial relationships between locations.

2.3 Bayesian Spatial Model

We will construct a Markov random field through Gaussian process models to predict SMB over the Antarctic ice sheet. Gaussian Process models allow a flexible non-linear model that allows accurate interpolation in regions where data is unavailable. Though we initially used a squared exponential covariance function, we found that its semivariogram poorly matched the SMB spatial process. Furthermore, the squared exponential covariance function caused the covariance matrix to

become nearly-singular and computationally unstable. For this reason, we use Matérn covariance functions to regulate the smoothness of our Gaussian Processes.

Matérn covariance functions also increase computational stability as they are a general covariance function that can represent nearly any semivariogram by adjusting its parameters ρ and ν In our model, we denote our model's covariance function as k. We argue that the covariance depends only, or mostly, upon the distance between points. So, we will make our covariance function to be a function of d where $d = |x - x'|$ and x, x' are two locations. Because k is only a function of d, it is isotropic. We are working under the following model:

Our Prior distribution will be represented by the following Gaussian process:

$$f(x) \sim \mathrm{GP}(\mathbf{0}, k(x,x')), \tag{2.3}$$

where

$$k(x,x') = \sigma^2 \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{d}{\rho}\right)^{\nu} K_{\nu}\left(\frac{d}{\rho}\right), \tag{2.4}$$

and d is distance between points x and x', K_{ν} is a Modified Bessel function of the second kind, and σ^2, ν, and ρ are unknown parameters. Our unknown parameter ρ is the smoothing parameter, ν alters the the order of the Bessel K function which alters the structure of the covariance function, and σ^2 is a scale parameter.

To account for surface mass balance variability between nearby measurements, we will use a spatial nugget, δ_s, to adjust the minimum autocorrelation of our spatial process. Since we make no assumption that σ^2, ν, ρ, and δ_s are known, we will set prior distributions on σ^2, ν, ρ, and δ_s. We will use sampling, in this case the Metropolis algorithm that we will update parameter-wise, to solve for σ^2, ν, ρ, and δ_s.

We will use the following priors on ν, ρ, σ^2, and δ_s:

$$\begin{aligned} \sigma^2 &\sim \mathrm{Gamma}(10,50), \\ \rho &\sim \mathrm{Gamma}(12.5,4), \\ \nu &\sim \mathrm{Gamma}(5,.1), \\ \delta_s &\sim \mathrm{Gamma}(8,1), \end{aligned} \tag{2.5}$$

where we are using the shape-scale parameterization of the Gamma distribution (this pdf is also shown explicitly in Appendix A).

We selected these priors based upon the anticipated semivariogram. Using random draws from the prior distributions on σ^2, ν, ρ, and δ_s, we can develop a prior semivariogram using the prior predictive distribution

$$p(y_{new} \mid \theta) = \int_{\theta} \mathscr{L}(y_{new} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{2.6}$$

where $\pi(\boldsymbol{\theta})$ represents the prior distributions on σ^2, ν, ρ, and δ_s. Based on our prior predictive distribution, we would expect to see the semivariogram displayed in Figure 2.3.

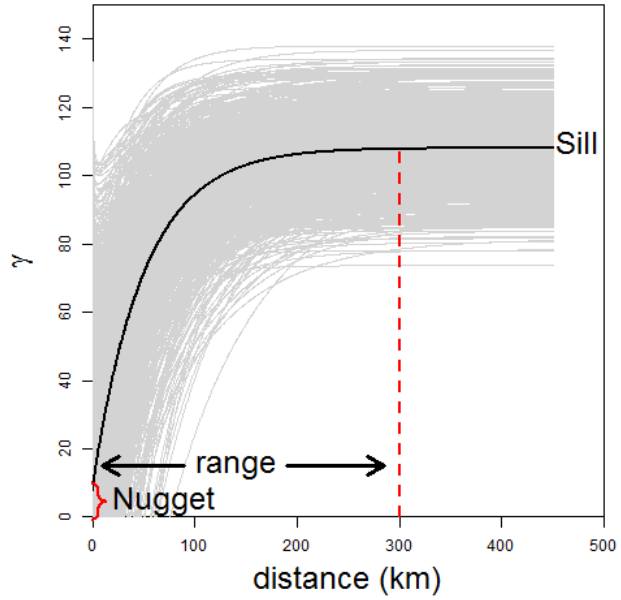


Figure 2.3 Semivariograms γ with units of variance are used to represent spatial correlation. We plot semivariograms corresponding to prior distribution selection (gray lines are possible semivariograms). Black in the mean of the possibilities The sill is the maximum value of γ, the nugget is the minimum of γ, and the range is the distance from the nugget to the sill.

Note that correlation is high within 75 km and almost is 0 by 200 km. However, the range $\approx$ 300 km, but the correlation is extremely weak at this distance. The large range will enable weak conditioning over data poor regions so that all grid predictions will be informed by reliable measurements. Van de berg et al. proposed using 55 km as the conditioning boundary, but showed their best fit enabled weighting data within 193 km (Van de Berg et al. 2006). Thus, our prior

selection has a range similar to van de berg et al.'s best fit model with strong correlation within 75 km, moderate correlation between 75 km and 150 km, and very weak correlation from 150 km to 300 km. Importantly, as demonstrated by the prior semivariogram, our semivariogram is very flexible and will allow the estimated process to converge to the true semivariogram of the Antarctic SMB process.

The Likelihood function will be generated as follows:

$$\mathscr{L}(\boldsymbol{y} \mid \boldsymbol{\theta}) \sim MVN(f(x), \sigma^2). \tag{2.7}$$

Given this prior distribution and Likelihood function, we arrive at the following posterior distribution:

$$p(\boldsymbol{x}|\boldsymbol{y}) \sim MVN_p(\boldsymbol{\mu}_x + \Sigma_{xy}\Sigma_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}), \tag{2.8}$$

where our Σs are populated by $k(x, x')$ using data $\boldsymbol{y}$ and grid points $\boldsymbol{x}$ where we make predictions.

Instead of the full posterior, we will sample from the joint density $f(\boldsymbol{\theta}) = \mathscr{L}(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. As discussed previously, we will use a Gibbs sampler where the parameters are iteratively updated using the Metropolis-Hastings algorithm to sample from the posterior distribution.

We will evaluate our SMB model on Antarctica by selecting four grids that will be evauluated using four data partitions discussed previously. Though our model can be evaluated at any location on Antarctica, the model must be evaluated at a finite number of locations. We will evaluate our model at a completely spatially random (CSR) grid such that the whole ice sheet is well represented. We will use a stereographic projection of Antarctica to create a polygon that will be used to create a CSR grid. An example of such a grid is displayed in Figure 2.4.

We will use the model described to draw from the posterior distribution

$$p(y_{new} \mid \theta) = \int_{\theta} \mathscr{L}(\boldsymbol{y}_{new} \mid \theta) p(\boldsymbol{\theta}, \boldsymbol{y}), \tag{2.9}$$

to predict SMB at all prediction locations on the CSR grid. For every prediction location, we will calculate the posterior mean using thousands of realizations $\boldsymbol{y}$ from the posterior predictive distribution

$$\bar{y} = \text{mean}(\boldsymbol{y}) \tag{2.10}$$

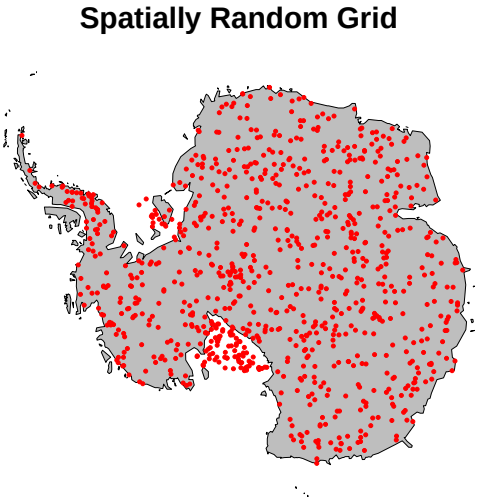


Figure 2.4 Selecting the grid we evaluate our model at is important because we must make global conclusions based on a finite number of grid points. For this reason, we want to select a grid that will fill the Antarctic ice sheet. We plot a complete spatially random (CSR) grid to demonstratet that it fills the ice sheet well.

and the 95% credible interval

$$\bar{y} \pm \text{qnorm}(.975, 0, 1) \cdot \sqrt{\text{var}(\boldsymbol{y})}, \tag{2.11}$$

where $\text{qnorm}(.975, 0, 1)$ is the .975 quantile of the standard Normal distribution and $\text{var}(\boldsymbol{y})$ is the variance of the posterior realizations (Ebden 2008). The credible interval will be used to quantify our prediction uncertainty and quantify the convergence of our predictions. We will create an uncertainty surface over the Antarctic ice sheet to visualize what spatial regions have high uncertainty.

Using the predicted SMB grid values, we will estimate the mean SMB ($\overline{\text{SMB}}$) and a 95% credible interval about $\overline{\text{SMB}}$ using the means of all posterior draws:

$$\overline{\text{SMB}} \pm \text{qt}(.975, n-1) \cdot \frac{s}{\sqrt{n}} \tag{2.12}$$

where $\text{qt}(.975, n-1)$ is the 0.975 quantile of the t distribution with $n-1$ degrees of freedom, n is the number of grid predictions, and s is the sample standard deviation. We estimate total surface

mass balance by integrating over the predicted SMB with respect to our surface

$$\int_{system} \text{SMB}\, d\boldsymbol{A} = \text{total SMB}. \tag{2.13}$$

We then quantify our uncertainty using a 95% credible interval about total balance and compare our results for the SMB surface and total balance to previous results to contextualize our results.

2.4 Computational Considerations

2.4.1 Data Partitions for Computational Speed

Because $N = 3529$, using all of our data in Gaussian process regression would require us to invert a 3529×3529 covariance matrix Σ multiple times. Because large matrix inversion is computationally expensive, we ran diagnostic speed tests on our model. We found that to complete enough iterations to model SMB effectively, we would need to run our model for more than 500 days. Because 1.5 years of computation was unfeasible for this project, we decided to partition our data into four subsets and run these four models separately.

Because Gaussian process regression is interpolative, we decided to not exclude data at boundary locations. Thus, to prevent interpolative issues, we chose 503 boundary data points to include in every partition, displayed in Figure 2.5. We then partitioned the remaining 3026 data points by dividing each point's row index by 4 modularly and placed that data into four groups depending upon whether the modular division result was zero, one, two, or three. This process selects every fourth data point from the remaining data. Because data set is grouped by location, modular division will space the data in the four partitions such that each partition will fill the continent which ensures that all predictions will be well-conditioned by data.

Thus, instead of running a single algorithm that would take about 500 days to complete, we run four algorithms simultaneously that each take about 30 days to complete. We then combine and compare the results from the four algorithms. Although an ad-hoc approach, this solution will save time while still allowing us to use all of the "A" rated data.

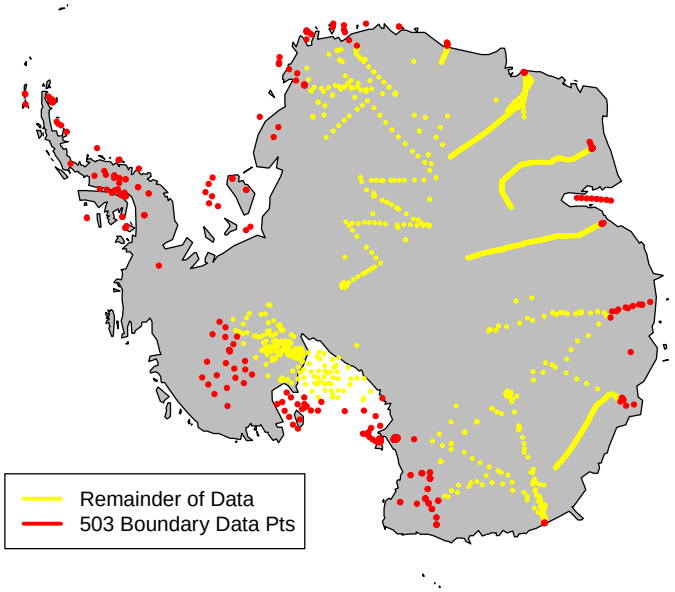


Figure 2.5 We select data points near the boundary of the continent for every data partition so that we prevent extrapolation issues for all algorithm runs.

2.4.2 Inclusion of Computational Nugget in Model

Because our data are unevenly spaced and clustered, it is likely our covariance matrix will be nearly singular. As discussed in section 2.4, a computational nugget can be used to increase computational stability due to near-singular matrices. We add a small computational nugget to assure that this matrix remains invertible for all values of ρ, ν, and σ^2. The computational nugget will be between approximately 5 and 7 orders of magnitude smaller than our spatial nugget δ_s, so we do not expect it to introduce any artificial modes. Thus, it should have no confounding effects upon our posterior distributions.

Chapter 3

Results and Discussion

This results section will discuss preliminary results using a random sample of 1259 "A" rated data points from Favier et al.'s database. Using these data, we have predicted SMB at 601 spatially random grid points over the continental Antarctic ice sheet. Our grid does not cover ice shelves over bays, however future analyses will include these regions.

3.1 Metropolis-Hastings Algorithm Result

We used the Metropolis-Hastings algorithm to solve for the spatial nugget δ_s, the range parameter ρ, the spatial variance σ^2, and the smoothing parameter ν of the Matérn covariance function used to model our spatial process. We have iterated over the algorithm 20 000 times to assure that we have sufficient draws from the posterior distribution for each parameter. To insure that the algorithm has converged before using solutions, we will discard the first 2000 algorithm iterations; these iterations are called the "burn-in." The results for each parameter are summarized in Table 3.1 below.

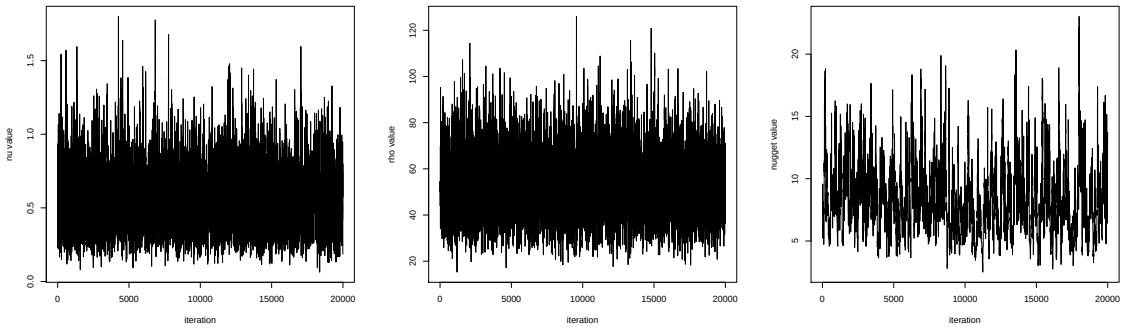
As discussed in Section 1.8.1, acceptance rates between 0.234 and 0.5 suggest that the parameters are mixing well, that is that the support of each parameter is being explored. Note that all parameters have acceptance rates between 0.234 and 0.5 (see Table 3.1), which suggests that the parameters have converged well. We illustrate parameter values with trace plot that plot the value

	Parameter Mean	Acceptance Rates	Candidate Distribution Variance	Initial Value
ρ	56.3	0.42	100	50
δ_s	10.2	0.29	8.00	8.00
ν	0.47	0.43	0.005	0.5

Using Gibbs sampler, we approximate $\sigma^2 = 59.2$.

Table 3.1 Results from Metropolis-Hastings algorith for each parameter

of a parameter at every iteration. These plots are used to show whether the algorithm has converged for a parameter of interest. For our parameters, the algorithm has clearly converged to a range of values (see Figure 3.1).



(a) Trace plot of ν **(b)** Trace plot of ρ **(c)** Trace plot of δ_s

Figure 3.1 Trace plots for algorithm parameter values. Note that all of the trace plots have converged to a specific range suggesting that the whole parameter support is explored and represented.

Given our trace plots and acceptance rates, we infer that our algorithm has successfully converged to posterior distribution of each parameter. Thus, the parameters found in the algorithm can be used to predict SMB.

3.2 Surface Mass Balance and Error Surface Results

Using the posterior draws from the Metropolis-Hastings algorithm, we use the posterior predictive distribution to predict SMB over the Antarctic ice sheet. Specifically, we get predictions using

$$p(\boldsymbol{y}_{\text{new}} \mid \boldsymbol{\theta}) = \int_{\theta} \mathscr{L}(\boldsymbol{y}_{\text{new}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}, \boldsymbol{y}), \tag{3.1}$$

where $\boldsymbol{\theta}$ are draws from the posterior distribution and y_{new} are SMB predictions at grid locations. We predict SMB for all 18 000 post-burn-in draws from the posterior distribution. With all 18 000 predictions, we predict total SMB, average SMB, and create SMB surfaces. We obtain the following SMB predictions:

	Estimate	C.I. Lower Bound	C.I. Upper Bound	Units
Total SMB	$1.75{\cdot}10^{12}$	$1.41{\cdot}10^{12}$	$2.08{\cdot}10^{12}$	$\text{m}^3 \cdot \text{w.e.} \cdot \text{yr}^{-1}$
Average SMB	124.80	100.95	148.66	$\text{mm} \cdot \text{w.e.} \cdot \text{yr}^{-1}$

Average 95% Credible Interval Margin for SMB predictions: 114.66 $\text{mm} \cdot \text{w.e.} \cdot \text{yr}^{-1}$

Table 3.2 Total and average SMB estimates and 95% credible interval results. Note that C.I. signifies a 95 % credible interval.

Using the mean of each grid prediction, we create a SMB heat map over the Antarctic ice sheet. With calculated credible interval margins, we map our prediction uncertainty spatially to identify regions where we have high prediction uncertainty. We map both high SMB and high uncertainty with red and low SMB and low uncertainty in blue (see Figure 3.2).

Our SMB surface yielded higher spatial variation and more negative values than expected; however, we believe that this will be remedied when we use more grid predictions. As expected, using Gaussian process regression yielded highest uncertainty at the boundaries.

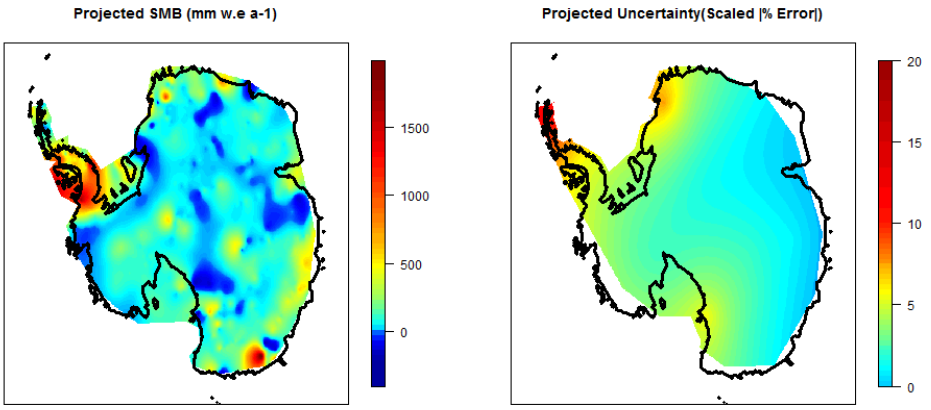


Figure 3.2 SMB surface plotted in units of mm w.e yr^{-1} and SMB error surface is in units of scaled percent error. Note that high values in SMB and uncertainty are in red while low values are in blue. As expected, we have high uncertainty at the ice sheet boundaries.

3.3 Discussion and Conclusions

Because the algorithm results have acceptance rates in the ideal range, 23.4%–50.0%, and the trace plots have no clear maxima or minima, we conclude that our Metropolis-Hastings algorithm has converged to the solution. Even though the results suggest convergence, the parameter ν does not diverge from the prior distribution meaning that the data had no impact on ν. For this reason, we conclude that the data actually does not inform the parameter ν and we will simply fix $\nu = 0.5$ in the future.

Our predicted average SMB, 124.80 mm $\cdot$ w.e. $\cdot$ yr^{-1}, is lower than previously estimated values by Vaughan et al., 146 mm $\cdot$ w.e. $\cdot$ yr^{-1}, and Van de Berg et al., 167.9 mm $\cdot$ w.e. $\cdot$ yr^{-1}. Specifically, Vaughan et al.'s prediction is 17.00% higher than our estimate and Van de Berg et al.'s estimate is 34% larger (Van de Berg et al. 2006; Vaughan et al. 1999). Vaughan et al.'s estimate is within our prediction error bounds, so we cannot say that our prediction varies significantly. Van de Berg et al.'s result, however, is far larger than ours, but their larger result is expected because they include ice shelves in their analysis which generally have higher SMB than continental regions. Thus, our result is reasonable in the context of previous results.

When compared to other SMB surfaces, our SMB surface has higher spatial resolution. Ad-

vantageously, this surface enables us to identify small variations which is helpful in identifying the borders of SMB accumulation regions. However, due to high spatial variation, our model produced more negative SMB regions than is likely plausible. Unlike any previous analyses, we have created a prediction uncertainty surface which allows us to identify regions where predictions are more likely to be biased. We will focus on gathering data in regions of high prediction uncertainty to minimize future prediction error. In future analyses, we will create a much finer grid with approximately 6000 grid points which will reduce total SMB and average SMB prediction uncertainty.

To make more general climate conclusions, we will meld our results with modeled Antarctic calving. Because

$$\text{Total Mass Balance} \approx \text{Total SMB} - \text{calving}, \tag{3.2}$$

we will be able to discuss whether the Antarctic ice sheet is growing or shrinking. Modeling total mass balance will allow us to predict sea level change.

3.4 Further Research

In the future we have four primary goals: First, we will implement a full data model and reanalyze the results. Second, we will analyze how SMB is changing over time. In analyzing how SMB is changing over time, we will need to develop a new model that integrates time into our spatial model. Third, we will consider including "B-rated" or "conditionally acceptable" data in our model in regions where "A- rated" data is sparse (Magand et al. 2007). Because "B-rated" data would have higher variance, we would have adjust prediction uncertainty that uses "B" rated data. Lastly, we will propose new measurement locations using integrated mean square error (IMSE); if carried out, then we can monitor how these measurements stabilize prediction errors. We will write a grant proposal to fund an expedition to Antarctica to install these stakes.

Bibliography

Andrianakis, I., and Challenor, P. G. (2012), "The effect of the nugget on Gaussian process emulators of computer models," *Computational Statistics & Data Analysis*, 56, 4215–4228.

Bromwich, D. H., Nicolas, J. P., and Monaghan, A. J. (2011), "An Assessment of Precipitation Changes over Antarctica and the Southern Ocean since 1989 in Contemporary Global Reanalyses," *Journal of Climate*, 24, 4189–4209.

Casella, G., and Berger, R. L. (1990), *Statistical inference* (Vol. 70), Duxbury Press Belmont, CA.

Casella, G., and George, E. I. (1992), "Explaining the Gibbs sampler," *The American Statistician*, 46, 167–174.

Ebden, M. (2008), "Gaussian processes for regression: A quick introduction," *Available on Internet: http://www. robots. ox. ac. uk/mebden/reports/GPtutorial.pdf*.

Favier, V., Agosta, C., Parouty, S., Durand, G., Delaygue, G., Gallée, H., Drouet, A.-S., Trouvilliez, A., and Krinner, G. (2013), "An updated and quality controlled surface mass balance dataset for Antarctica," *The Cryosphere*, 7, 583–597.

Furrer, R., and Sain, S. R. (2010), "spam: A sparse matrix R package with emphasis on MCMC methods for Gaussian Markov random fields," *Journal of Statistical Software*, 36, 1–25.

Gallée, H., Agosta, C., Gential, L., Favier, V., and Krinner, G. (2011), "A downscaling approach toward high-resolution surface mass balance over Antarctica," *Surveys in Geophysics*, 32, 507–518.

Gelman, A., Roberts, G., and Gilks, W. (1996), "Efficient metropolis jumping rules," *Bayesian statistics*, 5, 599–608.

Giovinetto, M. B., and Bull, C. (1987), "Summary and analyses of surface mass balance compilations for Antarctica, 1960-1985," *Byrd Polar Research Center, The Ohio State University.*

Lenaerts, J., den Broeke, M., Berg, W., Meijgaard, E., and Kuipers Munneke, P. (2012), "A new, high-resolution surface mass balance map of Antarctica (1979–2010) based on regional atmospheric climate modeling," *Geophysical Research Letters*, 39.

Magand, O., Genthon, C., Fily, M., Krinner, G., Picard, G., Frezzotti, M., and Ekaykin, A. A. (2007), "An up-to-date quality-controlled surface mass balance data set for the 90Âř–180Âř E Antarctica sector and 1950–2005 period," *Journal of Geophysical Research: Atmospheres (1984–2012)*, 112.

Minasny, B., and McBratney, A. B. (2005), "The Matérn function as a general model for soil variograms," *Geoderma*, 128, 192–207.

Monaghan, A. J., Bromwich, D. H., Fogt, R. L., Wang, S.-H., Mayewski, P. A., Dixon, D. A., Ekaykin, A., Frezzotti, M., Goodwin, I., Isaksson, E., et al. (2006a), "Insignificant change in Antarctic snowfall since the International Geophysical Year," *Science*, 313, 827–831.

Monaghan, A. J., Bromwich, D. H., and Wang, S.-H. (2006b), "Recent trends in Antarctic snow accumulation from Polar MM5 simulations," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364, 1683–1708.

Rasmussen, C. E. (2006), "Gaussian processes for machine learning," *Citeseer*.

Rencher, A. C., and Schaalje, G. B. (2008), *Linear models in statistics*, Wiley. com.

Robert, C. P., and Casella, G. (2004), *Monte Carlo statistical methods* (Vol. 319), Citeseer.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997), "Weak convergence and optimal scaling of random walk Metropolis algorithms," *The annals of applied probability*, 7, 110–120.

Van de Berg, W., Van den Broeke, M., Reijmer, C., and Van Meijgaard, E. (2006), "Reassessment of the Antarctic surface mass balance using calibrated output of a regional atmospheric climate model," *Journal of Geophysical Research: Atmospheres (1984–2012)*, 111.

Vaughan, D. G., Bamber, J. L., Giovinetto, M., Russell, J., and Cooper, A. P. R. (1999), "Reassessment of net surface mass balance in Antarctica," *Journal of Climate*, 12, 933–946.

Waller, L. A., and Gotway, C. A. (2004), *Applied spatial statistics for public health data* (Vol. 368), John Wiley & Sons.

Wingham, D., Shepherd, A., Muir, A., and Marshall, G. (2006), "Mass balance of the Antarctic ice sheet," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364, 1627–1635.

Zwally, H. J. (1977), "Microwave emissivity and accumulation rate of polar firn," *J. Glaciol*, 18, 195–215.

Zwally, H. J., and Giovinetto, M. B. (1995), "Accumulation in Antarctica and Greenland derived from passive-microwave data: A comparison with contoured compilations," *Annals of Glaciology*, 21, 123–130.

Appendix A

Referenced Distributions

A.1 The Gamma Distribution

We will use the shape (α) and scale (β) parameterization of the gamma distribution:

$$\text{PDF:} \qquad f(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha-1} e^{-\frac{x}{\beta}}$$

$$\text{CDF:} \qquad F(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)} \gamma(\alpha, \frac{x}{\beta})$$

$$\mathrm{E}(X) = \alpha\beta$$

$$\mathrm{Var}(X) = \alpha\beta^2$$

where $\alpha > 0$, $\beta > 0$, $x \in (0,\infty)$, $\gamma(\cdot)$ is the lower incomplete gamma function, and Γ is the gamma function.

A.2 The Inverse Gamma Distribution

We will use the shape (α) and scale (β) parameterization of the Inverse-gamma distribution:

$$\text{PDF:} \qquad f(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{-\alpha-1} e^{-\frac{1}{x\beta}}$$

$$\text{CDF:} \qquad F(x|\alpha,\beta) = \frac{\Gamma_i(\alpha, \frac{\beta}{x})}{\Gamma(\alpha)})$$

$$\mathrm{E}(X) = \frac{1}{\beta(\alpha-1)}$$

$$\mathrm{Var}(X) = \frac{1}{\beta^2(\alpha-1)^2(\alpha-2)}$$

where $\alpha > 0$, $\beta > 0$, $x \in (0,\infty)$, Γ is the gamma function, and Γ_i is the incomplete upper gamma function.

A.3 The Multivariate Normal Distribution

We will use a parameterization with a k-dimensional location parameter ($\boldsymbol{\mu}$) and $k \times k$-dimensional covariance matrix Σ:

$$\text{PDF:} \qquad f(\boldsymbol{x}|\alpha,\beta) = (2\pi)^{-\frac{k}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

$$\mathrm{E}(X) = \boldsymbol{\mu}$$

$$\mathrm{Var}(X) = \Sigma$$

where $\boldsymbol{\mu} \in \mathscr{R}^k$, $\Sigma \in \mathscr{R}^{k\times k}$, and $x \in (-\infty,\infty)$.

Appendix B

Referenced R Code

B.1 Metropolis-Hastings Algorithm to solve for ρ, ν, and δ_s

```
library(MASS)
library(graphics)
library(maps)
library(mapdata)
library(mapproj)
library(pscl)
library(splancs)
library(geoR)
library(fields)
library(corpcor)
rm(list=ls(all=TRUE))

## define distance and covariance functions
hd <- function(long1, lat1, long2, lat2,len) {
  R <- 6371 # Earth mean radius in km - I should change to something else
  d_long <- (long2 - long1)*pi/180
  d_lat <- (lat2 - lat1)*pi/180
  a <- sin(d_lat/2)^2 + cos(lat1*pi/180) * cos(lat2*pi/180) * sin(d_long/2)^2
  c <- 2 * atan2(sqrt(a),sqrt(1-a))
  d <- R * c
  return(d) # Distance in km
}
Matern <- function (d, scale = 1, range = 1, alpha = 1/range, smoothness = 0.5,
    nu = smoothness, phi = scale) {
  if (any(d < 0))  stop("distance argument must be nonnegative")
  ind <- which(d == Inf)
  d <- d * alpha
  d[d == 0] <- 1e-10
  con <- (2^(nu - 1)) * gamma(nu)
  cov <- phi * con * (d^nu) * besselK(d, nu)
  cov[ind] <- 0
  return(cov)
}
write("",file="reps.txt")
#### read in data
ice <- read.csv("ant_loc.csv")
names(ice) <- c("latitude","longitude","SMB","year","method","pub","temp","el")
cord <- cbind(ice$longitude,ice$latitude,ice$SMB)

## create grid and parse out data
group1<-cord[cord[,1]>=39.4 & cord[,1]<=50 & cord[,2]>=(-69.028) & cord[,2]<=(-65),]
cord_left<-cord[(cord[,1]<39.4 | cord[,1]>50) | (cord[,2]<(-69.028) | cord[,2]>(-65)),]

group2<-cord[cord[,1]>=50 & cord[,1]<=67 & cord[,2]>=(-68.76) & cord[,2]<=(-65),]
```

```
cord_left<-cord_left[(cord_left[,1]<50 | cord_left[,1]>67) | (cord_left[,2]<(-68.76) | cord_left[,2]>(-65)),]

group3<-cord[cord[,1]>=75 & cord[,1]<=80 & cord[,2]>=(-70.09) & cord[,2]<=(-65),]
cord_left<-cord_left[(cord_left[,1]<75 | cord_left[,1]>80) | (cord_left[,2]<(-70.09) | cord_left[,2]>(-65)),]

tiny<-cord_left[(cord_left[,1]<=(74) & cord_left[,1]>=(67)) & (cord_left[,2]<=(-65) & cord_left[,2]>=(-73)),]
cord_left<-cord_left[(cord_left[,1]>74 | cord_left[,1]<67) | (cord_left[,2]>(-65) | cord_left[,2]<(-73)),]

group4<-cord[cord[,1]>=20 & cord[,1]<=27 & cord[,2]>=(-70.37) & cord[,2]<=(-69),]
cord_left<-cord_left[(cord_left[,1]<20 | cord_left[,1]>27) | (cord_left[,2]<(-70.37) | cord_left[,2]>(-69)),]

group6<-cord[cord[,1]>=110 & cord[,1]<=134 & cord[,2]>=(-67.55) & cord[,2]<=(-66),]
cord_left<-cord_left[(cord_left[,1]<110 | cord_left[,1]>134) | (cord_left[,2]<(-67.55) | cord_left[,2]>(-66)),]

group7<-cord[cord[,1]>=(-180) & cord[,1]<=(-100) & cord[,2]>=(-81) & cord[,2]<=(-75),]
cord_left<-cord_left[(cord_left[,1]<(-180) | cord_left[,1]>(-100)) | (cord_left[,2]<(-81) | cord_left[,2]>(-75)),]
group7b<-cord[cord[,1]>=(159) & cord[,1]<=(180) & cord[,2]>=(-81) & cord[,2]<=(-75),]
cord_left<-cord_left[(cord_left[,1]<159 | cord_left[,1]>180) | (cord_left[,2]<(-81) | cord_left[,2]>(-75)),]
group7<-rbind(group7,group7b)

group9<-cord[cord[,1]>=134 & cord[,1]<=142 & cord[,2]>=(-66.706) & cord[,2]<=(-65),]
cord_left<-cord_left[(cord_left[,1]<134 | cord_left[,1]>142) | (cord_left[,2]<(-66.706) | cord_left[,2]>(-65)),]

group10<-cord[cord[,1]>=(-90) & cord[,1]<=(-38) & cord[,2]>=(-85) & cord[,2]<=(-55),]
cord_left<-cord_left[(cord_left[,1]<(-90) | cord_left[,1]>(-38)) | (cord_left[,2]<(-85) | cord_left[,2]>(-55)),]

group11a<-cord_left[(cord_left[,1]<=(20) & cord_left[,1]>=(0) & cord_left[,2]>=(-72) & cord_left[,2]<=(-55)) ,]
cord_left<-cord_left[(cord_left[,1]>20 | cord_left[,1]<0 | cord_left[,2]<(-72) | cord_left[,2]>(-55)),]
group11b<-cord_left[(cord_left[,1]<=(0) & cord_left[,1]>=(-10) & cord_left[,2]>=(-72) & cord_left[,2]<=(-55) ),]
cord_left<-cord_left[(cord_left[,1]>0 | cord_left[,1]<(-10) | cord_left[,2]<(-72) | cord_left[,2]>(-55) ),]
group11c<-cord_left[(cord_left[,1]<=(-10) & cord_left[,1]>=(-20)& cord_left[,2]>=(-75) & cord_left[,2]<=(-50)),]
cord_left<-cord_left[(cord_left[,1]>(-10) | cord_left[,1]<(-20) | cord_left[,2]<(-75) | cord_left[,2]>(-50) ),]
group11d<-cord_left[(cord_left[,1]<=(-20) & cord_left[,1]>=(-40) & cord_left[,2]>=(-80) & cord_left[,2]<=(-50)),]
cord_left<-cord_left[(cord_left[,1]>(-20) | cord_left[,1]<(-40) | cord_left[,2]<(-80) | cord_left[,2]>(-50)),]
group11<-rbind(group11a,group11b,group11c,group11d)

group8a<-cord_left[(cord_left[,1]<=(110) & cord_left[,1]>=(80) & cord_left[,2]>=(-70) & cord_left[,2]<=(-50)),]
cord_left<-cord_left[(cord_left[,1]>(110) | cord_left[,1]<(80) | cord_left[,2]<(-70) | cord_left[,2]>(-50)),]
group8b<-cord_left[(cord_left[,1]<=(160) & cord_left[,1]>=(155) & cord_left[,2]>=(-78) & cord_left[,2]<=(-50)),]
cord_left<-cord_left[(cord_left[,1]>(160) | cord_left[,1]<(155) | cord_left[,2]<(-78) | cord_left[,2]>(-50)),]
group8c<-cord_left[(cord_left[,1]<=(180) & cord_left[,1]>=(160) & cord_left[,2]>=(-75) & cord_left[,2]<=(-50)),]
cord_left<-cord_left[(cord_left[,1]>(180) | cord_left[,1]<(160) | cord_left[,2]<(-75) | cord_left[,2]>(-50)),]
group8 <- rbind(group8a,group8b,group8c)

group <- rbind(group1,group2,group3,group4,group6,group7,group8,group9,group10,group11,tiny)
ind <- which( 1:length(cord_left[,3]) %% 4 == 0)

griddy <- data_use <- rbind(group,cord_left[ind,])

#### allot memory for distance matrices # note the weird notation phi= sigma^2  # range =rho , and nu=nu
sigma_mat <- d_mat_gg <- matrix(1,nrow=length(griddy[,1]),ncol=length(griddy[,1]))
d_mat_gd <- matrix(1,nrow=length(griddy[,1]),ncol=length(data_use[,1]))
d_mat_dd <- matrix(1,nrow=length(data_use[,1]),ncol=length(data_use[,1]))

## populade distance matrices
for(i in 1:length(griddy[,1])){
  d_mat_gg[,i] <- hd(griddy[i,1],griddy[i,2],griddy[,1],griddy[,2],length(griddy[,1]))
}
for(i in 1:length(griddy[,1])){
  d_mat_gd[i,] <- hd(griddy[i,1],griddy[i,2],data_use[,1],data_use[,2],length(data_use[,1]))
}
for(i in 1:length(data_use[,1])){
  d_mat_dd[,i] <- hd(data_use[i,1],data_use[i,2],data_use[,1],data_use[,2],length(data_use[,1]))
}

### throw away distance that are too long - create sparseness
d_mat_gg <- d_mat_dd <- ifelse(d_mat_gg > 1200,Inf,d_mat_gg)
d_mat_gd <- ifelse(d_mat_gd > 1200,Inf,d_mat_gd)
# d_mat_dd <- ifelse(d_mat_dd > 1200,Inf,d_mat_dd)
d_mat_dg<-t(d_mat_gd)

y <- data_use[,3]

##### metropolis-hastings
## set initial values
runs <- 20000
nu <- rho <- sigma <- nug <- numeric(runs)
```

```
nu[1] <- .5
rho[1] <- 50
sigma[1] <- 100
nug[1] <- 8
var_n <- 0.15
var_r <- 1200
var_nug <- 2

cnug <- countn <- countr <- 0

n <- length(griddy[,1])
astar <- n/2
bstar <- 9e-6
sol <- matrix(0,nrow=length(griddy[,1]),ncol=runs)
nug1 <- 4e-1
nug2 <- 2e-4
for( i in 2:runs)
{
  draw <- diag(2)
  while(!is.vector(draw) || !is.matrix(invdd)){
    comgg <- comdd <- Matern(d_mat_gg,phi=sigma[i-1],nu=nu[i-1],range=rho[i-1])
    #comdd <- Matern(d_mat_dd,phi=sigma[i-1],nu=nu[i-1],range=rho[i-1])
    comdg <- Matern(d_mat_dg,phi=sigma[i-1],nu=nu[i-1],range=rho[i-1])
    comgd <- t(comdg)
    invdd <- try(solve(comdd+nug2*diag(length(comdd[,1]))),silent=TRUE)
    post_mean <- try(comgd %*% invdd %*% (y),silent=TRUE)
    post_sig <- try( comgg - comgd %*% invdd %*% comdg , silent =TRUE)
    post_sig <- try((post_sig+t(post_sig))/2 + (nug1+nug[i-1])*diag(length(comgg[,1])),silent=TRUE)
    draw <- try(mvrnorm(1,post_mean,post_sig,tol=1e-2),silent=TRUE)
    sol[,i-1] <- draw
    bstar <- ((sum((y-sol[,i-1])^2)/2)^(-1))
    sigma[i] <- 1/rgamma(1,astar,scale=bstar)
    log.norm <- -.5*sum(log(eigen(post_sig,only.values=T)$values))-
                0.5*t(sol[,i-1]-post_mean)%*%solve(post_sig)%*%(sol[,i-1]-post_mean)
  }

  met.rat <- NaN  ## loop entrance condition
  ######### draw on nu #########
  while(is.nan(met.rat) || !is.matrix(invdd) || !is.numeric(rat_num) || !is.numeric(met.rat) ){
    n.draw <- rgamma(1 , shape = (nu[i-1])^2 / var_n , scale = var_n/nu[i-1] )
    rat_den <- (dgamma(nu[i-1],5,scale=.1,log=T)+
                dgamma(n.draw,shape =(nu[i-1])^2/var_n,scale=var_n/nu[i-1],log=T)+log.norm)
    comgg <- comdd <- Matern(d_mat_gg,phi=sigma[i],nu=n.draw,range=rho[i-1])
    # comdd <- Matern(d_mat_dd,phi=sigma[i],nu=n.draw,range=rho[i-1])
    comdg <- Matern(d_mat_dg,phi=sigma[i],nu=n.draw,range=rho[i-1])
    comgd <- t(comdg)
    invdd <- try(solve(comdd+nug2*diag(length(comdd[,1]))),silent=TRUE)
    post_mean_nu <- try(comgd %*% invdd %*% y,silent=TRUE)
    post_sig_nu <- try(comgg - comgd %*% invdd %*% comdg,silent=TRUE)
    post_sig_nu <-try( (post_sig_nu+t(post_sig_nu))/2 + (nug1+nug[i-1])*diag(length(comgg[,1])),silent=TRUE)
    rat_num <- try((dgamma(n.draw,5,scale=.1,log=T)+dgamma(nu[i-1],shape=(n.draw)^2/var_n,scale=var_n/n.draw,log=T)-
.5*sum(log(eigen(post_sig_nu,only.values=T)$values))-
0.5*t(sol[,i-1]-post_mean_nu)%*%solve(post_sig_nu)%*%(sol[,i-1]-post_mean_nu)),silent=TRUE)
    nu[i] <- nu[i-1]
    met.rat <- try(rat_num - rat_den,silent=TRUE)
    try(if( met.rat >= log(runif(1)) ) { nu[i] <- n.draw ; countn <- countn+1 },silent=TRUE)
  }

  met.rat <- NaN  ## loop entrance condition
  ######## draw on rho ##########
  while(is.nan(met.rat) || !is.matrix(invdd) || !is.numeric(rat_num) || !is.numeric(met.rat) ){
    r.draw <- rgamma(1 , shape = (rho[i-1])^2 / var_r , scale = var_r/rho[i-1] )
    rat_den <- (dgamma(rho[i-1],12.5,scale=4,log=T)+
                dgamma(r.draw,shape=(rho[i-1])^2/var_r,scale=var_r/rho[i-1],log=T)+log.norm)
    comgg <- comdd <- Matern(d_mat_gg,phi=sigma[i],nu=nu[i],range=r.draw)
    # comdd <- Matern(d_mat_dd,phi=sigma[i],nu=nu[i],range=r.draw)
    comdg <- Matern(d_mat_dg,phi=sigma[i],nu=nu[i],range=r.draw)
    comgd <- t(comdg)
    invdd <- try(solve(comdd+nug2*diag(length(comdd[,1]))),silent=TRUE)
    post_mean_rho <- try( comgd %*% invdd %*% (y),silent=TRUE)
    post_sig_rho <- try( comgg - comgd %*% invdd %*% comdg,silent=TRUE)
    post_sig_rho <- try( (post_sig_rho+t(post_sig_rho))/2 + (nug1+nug[i-1])*diag(length(comgg[,1])),silent=TRUE)
    rat_num <- try((dgamma(r.draw,12.5,scale=4,log=T)+dgamma(rho[i-1],shape=(r.draw)^2/var_r,scale=var_r/r.draw,log=T)-
.5*sum(log(eigen(post_sig_rho,only.values=T)$values))-
0.5*t(sol[,i-1]-post_mean_rho)%*%solve(post_sig_rho)%*%(sol[,i-1]-post_mean_rho)),silent=TRUE)
    rho[i] <- rho[i-1]
    met.rat <- try(rat_num - rat_den,silent=TRUE)
```

```
    try(if( met.rat >= log(runif(1))) { rho[i] <- r.draw ; countr <- countr+1 },silent=TRUE)
  }

  met.rat <- NaN  ## loop entrance condition
  ######## draw on nug ########
  while(is.nan(met.rat) || !is.matrix(invdd) || !is.numeric(rat_num) || !is.numeric(met.rat) ){
    nug.draw <- rgamma(1 , shape = (nug[i-1])^2 / var_nug , scale = var_nug/nug[i-1] )
    rat_den <- (dgamma(nug[i-1],8,scale=1,log=T)+
                dgamma(nug.draw,shape=(nug[i-1])^2/var_nug,scale=var_nug/nug[i-1],log=T)+log.norm)
    comgg <- comdd <- Matern(d_mat_gg,phi=sigma[i],nu=nu[i],range=rho[i])
    # comdd <- Matern(d_mat_dd,phi=sigma[i],nu=nu[i],range=rho[i])
    comdg <- Matern(d_mat_dg,phi=sigma[i],nu=nu[i],range=rho[i])
    comgd <- t(comdg)
    invdd <- try(solve(comdd+nug2*diag(length(comdd[,1]))),silent=TRUE)
    post_mean_nug <- try(comgd %*% invdd %*% (y),silent=TRUE)
    post_sig_nug <- try(comgg - comgd %*% invdd %*% comdg,silent=TRUE)
    post_sig_nug <- try((post_sig_nug+t(post_sig_nug))/2 + (nug1+nug.draw)*diag(length(comgg[,1])),silent=TRUE)
    rat_num <- try((dgamma(nug.draw,8,scale=1,log=T)+
                      dgamma(nug[i-1],shape=(nug.draw)^2/var_nug,scale=var_nug/nug.draw,log=T ) -
.5*sum(log(eigen(post_sig_nug,only.values=T)$values))-
 0.5*t(sol[,i-1]-post_mean_nug)%*%solve(post_sig_nug)%*%(sol[,i-1]-post_mean_nug)),silent=TRUE)
    nug[i] <- nug[i-1]
    met.rat <- try(rat_num - rat_den,silent=TRUE)
    try(if( met.rat >= log( runif(1) ) ){ nug[i] <- nug.draw ; cnug <- cnug+1 },silent=TRUE)
  }

  if(i %% 1000 ==0){
    write(c("we're on: ",paste(i)),file="reps.txt",append=TRUE)
  }

  if(i %% 1000 ==0){
    # mean
    sol_tot <- sol[,1:i]
    sol_mean <- apply(sol_tot,1,mean)
    cred_lo_95 <- cred_hi_95<-seq(0,0,length=length(griddy[,1]))
    for(i in 1:length(griddy[,1])){
      lo_hi <- qnorm(c(.025,.975),mean=sol_mean[i],sd=sd(sol_tot[i,]))
      cred_lo_95[i] <- lo_hi[1]
      cred_hi_95[i] <- lo_hi[2]
    }
    per_error <- ((sol_mean-griddy[,3])/griddy[,3])*100
    per_error[per_error==Inf | per_error==-Inf] <- 0
    avg_error <- mean(abs(per_error))
    griddy_out <- cbind(griddy,cred_lo_95,sol_mean,cred_hi_95,per_error,avg_error)
    colnames(griddy_out) <- c("longitude","latitude","measured_SMB","95_cred_int_low",
"pred_mean","95_cred_int_hi","percent error", "avg error" )
    accept_nu <- countn/(i-1) # acceptance for nu
    accept_rho <- countr/(i-1) # acceptance for rho
    accept_nug <- cnug/(i-1)
    parameter <- cbind(sigma,rho,nu,nug,accept_rho,accept_nu,accept_nug)
    write.csv(griddy_out,file="d_pred.csv")
    write.csv(parameter,file="parameter.csv")
    write.csv(sol,file="full_sol.csv")
    write(c("we're on: ",paste(i)),file="reps.txt",append=TRUE)
  }
}
# compute the final solution for the algorithm
draw <- diag(2)
while(!is.vector(draw) || !is.matrix(invdd)){
  comgg <- comdd <- Matern(d_mat_gg,phi=sigma[i],nu=nu[i],range=rho[i])
  # comdd <- Matern(d_mat_dd,phi=sigma[i],nu=nu[i],range=rho[i])
  comdg <- Matern(d_mat_dg,phi=sigma[i],nu=nu[i],range=rho[i])
  comgd <- t(comdg)
    invdd <- try(solve(comdd+nug2*diag(length(comdd[,1]))),silent=TRUE)
  post_mean <- try(comgd %*% invdd %*% (y),silent=TRUE)
  post_sig <- try(comgg - comgd %*% invdd %*% comdg,silent=TRUE)
  post_sig <- try((post_sig+t(post_sig))/2 + nug[runs]*diag(length(comgg[,1])),silent=TRUE)
  draw <- try(mvrnorm(1,post_mean,post_sig,tol=1e-2),silent=TRUE)
  sol[,runs] <- draw
}

sol_mean <- apply(sol,1,mean) # mean
cred_lo_95 <- cred_hi_95<-seq(0,0,length=length(griddy[,1]))
for(i in 1:length(griddy[,1])){ #credible interval
  lo_hi <- qnorm(c(.025,.975),mean=sol_mean[i],sd=sd(sol[i,]))
  cred_lo_95[i] <- lo_hi[1]
  cred_hi_95[i] <- lo_hi[2]
```

```
}
per_error <- ((sol_mean-griddy[,3])/griddy[,3])*100  #percent error
per_error[per_error==Inf | per_error==-Inf] <- 0
avg_error <- mean(abs(per_error)) #absolute mean per. error
griddy_out <- cbind(griddy,cred_lo_95,sol_mean,cred_hi_95,per_error,avg_error)
colnames(griddy_out) <- c("longitude","latitude","measured_SMB","95_cred_int_low",
"pred_mean","95_cred_int_hi","percent error", "avg error" )
accept_nu <- countn/(runs-1) # acceptance for nu
accept_rho <- countr/(runs-1) # acceptance for rho
accept_nug <- cnug/(runs-1)   # acceptance for nugget

### output results
parameter <- cbind(sigma,rho,nu,nug,accept_rho,accept_nu,accept_nug)
colnames(parameter) <- c ("sigma","rho","nu","nugget","accept_rho","accept_nu","accept_nug")
write.csv(griddy_out,file="d_pred.csv")
write.csv(parameter,file="parameter.csv")
write.csv(sol,file="full_sol.csv")
```

B.2 Creating SMB and SMB Uncertainty Surface on Antarctica

```
library(MASS)
library(graphics)
library(maps)
library(mapdata)
library(splancs)
library(mapproj)

rm(list=ls())
gridp<-read.csv("C:/Users/Phil/Dropbox/Stats/stats research/ice/SMB prediction/g_pred.csv")
gridp<-gridp[gridp$pred_mean<2000 & gridp$pred_mean>(-200),]
xg<-gridp$longitude
yg<-gridp$latitude
zg<-gridp$pred_mean
gridxy<-mapproject(xg,yg,projection="stereographic",orientation=c(-90,0,0))

datap<-read.csv("C:/Users/Phil/Dropbox/Stats/stats research/ice/SMB prediction/d_pred.csv")
xd<-datap$longitude
yd<-datap$latitude
zd<-datap$pred_mean
dataxy<-mapproject(xd,yd,projection="stereographic",orientation=c(-90,0,0))
xp<-dataxy$x
yp<-dataxy$y
zp<-abs(datap$percent.error)
data<-datap$measured_SMB
scale.needed<-which(abs(zd)<abs(30))
zp[scale.needed]<-zp[scale.needed]*data[scale.needed]/30
x<-c(gridxy$x,dataxy$x)
y<-c(gridxy$y,dataxy$y)
z<-c(zg,zd)

require(akima)
require(fields)
b1<-(14:0)
b2<-1:47
bb<-c(-5000,-500,-(b1^2),b2^1.95,10000000)

this<- Krig(cbind(x,y),z,cov.function="stationary.cov")
png("smb_heat_krig.png")
surface.Krig(this,type="I",nx=500,ny=500,breaks=bb,zlim=c(-400,1990),
legend.args=list( text="",col="black", cex=1, side=3, line=1),
ylim=c(-.4,.4),xlim=c(-.5,.5),yaxt="n",xaxt="n",xlab="",ylab="",
main="Projected SMB (mm w.e a-1)",cex.main=1.7)
map("worldHires","Antarctica",xlim=c(-180,180),ylim=c(-90,-60),
col="black",lwd=3,fill=F,projection="stereographic",orientation=c(-90,0,0),add=T)
dev.off()
```

```
bbe2<-c(seq(-1000,-0,length=20),seq(0,10,length=15),seq(11,20,length=10),
seq(21,100,length=15),seq(100,100000,length=5))
bbe1<-c(seq(-500,-0,length=20),seq(0,10,length=35),seq(11,20,length=8),
seq(21,100,length=1),seq(100,100000,length=1))

this2<- Krig(cbind(xp,yp),zp,cov.function="stationary.cov")
png("error_heat_krig.png")
surface.Krig(this2,type="I",nx=500,ny=500,breaks=bbe1,zlim=c(0,20),
legend.args=list( text="",col="black", cex=1, side=3, line=1),
ylim=c(-.4,.4),xlim=c(-.5,.5),yaxt="n",xaxt="n",xlab="",ylab="",
main="Projected Uncertainty(Scaled |% Error|)",cex.main=1.7)
map("worldHires","Antarctica",xlim=c(-180,180),ylim=c(-90,-60),
col="black",lwd=3,fill=F,projection="stereographic",orientation=c(-90,0,0),add=T)
dev.off()
```

B.3 Plots on Antarctica

```
library(MASS)
library(graphics)
library(maps)
library(mapdata)
library(splancs)
library(mapproj)
graphics.off()
rm(list=ls(all=TRUE))
ice.bad <- read.csv("C:/Users/Phil/Dropbox/Stats/stats research/ice/ant_bad.csv")
names(ice.bad)<-c("latitude","longitude","el","SMB","method","pub")
ice<-read.csv("C:/Users/Phil/Dropbox/Stats/stats research/ice/ant_loc.csv")
names(ice)<-c("latitude","longitude","SMB","year","method","pub","temp","el")
long<-ice$longitude
lat<-ice$latitude
cord<-cbind(ice$longitude,ice$latitude)
grid_l<-15
lat_grid<-seq(-90,-62,length=grid_l)
long_grid<-seq(-180,180,length=grid_l)
ant<-map("worldHires","Antarctica",fill=TRUE,plot=FALSE)
ant_mat<-cbind(ant$x,ant$y)
br<-which(is.na(ant_mat[,2]))
mainland<-ant_mat[(br[1]+1):(br[2]-1),1:2]
l<-length(mainland[,2])
bottom<-cbind(seq(-180,-180,length=500),seq(-90,-84.305,length=500))
top<-cbind(seq(180,180,length=500),seq(-84.3023,-90,length=500))
left<-cbind(seq(180,-180,length=1000),seq(-90,-90,length=1000))
mainland<-rbind(mainland[1:11973,1:2],top,left,bottom,mainland[11974:l,1:2])
# this recreates mainland so that is has a full border along the south pole.

# 1. Plot with all measurements with grid
pdf("ant_good.pdf")
map("worldHires","Antarctica",xlim=c(-180,180),lwd=3,ylim=c(-90,-60),col="gray",fill=T,
projection="stereographic",orientation=c(-90,0,0))
map.grid(c(-180,180,-90,-60),col="black",nx=20,ny=10,labels=F)
all_obs<-mapproject(ice$longitude,ice$latitude,projection="stereographic",orientation=c(-90,0,0))
points(all_obs,pch=1,cex=.6,col="red")
title(main="SMB Measurement Location",cex.main=2.1)
legend("bottomleft",c("Reliable Data"),lty=c(1,1),lwd=c(15,15)
,col=c("red"),cex=1.5)
dev.off()
```

B.4 Create Trace Plots and Data Analysis

```
graphics.off()
data<-read.csv("C:/Users/Phil/Dropbox/Stats/stats research/ice/SMB prediction/data_sim/d_pred.csv")
parameter<-read.csv("C:/Users/Phil/Dropbox/Stats/stats research/ice/SMB prediction/data_sim/parameter.csv"
```

```
,header=T)
rho <- parameter$rho
nu <- parameter$nu
nugget <- parameter$nug

pe <- data$percent.error
smb <- data$measured_SMB
cat("Absolute percent error is: ", mean(abs(pe)),"% \n")

scale.needed<-which(abs(smb)<abs(30))
pe[scale.needed]<-pe[scale.needed]*smb[scale.needed]/30
cat("Scaled absolute percent error is: " , mean(abs(pe)),"% \n")

cat("correlation between predicted and measured SMB is: ",
cor(data$measured_SMB,data$pred_mean)," \n")

cor(nu,nugget)
cor(rho,nugget)
cor(nu,rho)

parameter$accept_nu[1]
parameter$accept_nug[1]
parameter$accept_rho[1]

x11()
#pdf("trace_rho.pdf")
plot(parameter$X,rho,xlab="iteration",ylab="rho value",type="l")
#dev.off()

x11()
#pdf("trace_nu.pdf")
plot(parameter$X,nu,xlab="iteration",ylab="nu value",type="l")
#dev.off()

x11()
#pdf("trace_nug.pdf")
plot(parameter$X,nugget,xlab="iteration",ylab="nugget value",type="l")
#dev.off()
```

B.5 Create Semivariograms

```
graphics.off()
library(MASS)
library(graphics)
library(ggplot2)
library(grid)

curly <- function(N = 100, Tilt = 1, Long = 2, scale = 0.1, xcent = 0.5,
                  ycent = 0.5, theta = 0, col = 1, lwd = 1, grid = FALSE){
           ymin <- scale / Tilt
           y2 <- ymin * Long
           i <- seq(0, pi/2, length.out = N)

           x <- c(ymin * Tilt * (sin(i)-1),
                  seq(0,0, length.out = 2),
                  ymin * (Tilt * (1 - sin(rev(i)))),
                  ymin * (Tilt * (1 - sin(i))),
                  seq(0,0, length.out = 2),
                  ymin * Tilt * (sin(rev(i)) - 1))

           y <- c(-cos(i) * ymin,
                  c(0,y2),
                  y2 + (cos(rev(i))) * ymin,
                  y2 + (2 - cos(i)) * ymin,
                  c(y2 + 2 * ymin, 2 * y2 + 2 * ymin),
                  2 * y2 + 2 * ymin + cos(rev(i)) * ymin)

           x <- x + xcent
           y <- y + ycent - ymin - y2
```

```
          x1 <- cos(theta) * (x - xcent) - sin(theta) * (y - ycent) + xcent
          y1 <- cos(theta) * (y - ycent) + sin(theta) * (x - xcent) + ycent

          ##For grid library:
          if(grid){
             grid.lines(unit(x1,"npc"), unit(y1,"npc"),gp=gpar(col=col,lwd=lwd))
          }

          ##Uncomment for base graphics
          else{
             par(xpd=TRUE)
             points(x1,y1,type='l',col=col,lwd=lwd)
             par(xpd=FALSE)
          }
}
c0 <- 8
ck <- 100
a <- .5
ak <- 50

semi <- function(h,nug,var,nu,rho){
nug + var * (1 - (1/(2^(nu-1)*gamma(nu)))*(h/ak)^a*besselK(h/rho,nu))
}

nug <- rgamma(1000,4,scale=2)
var <- rnorm(1000,100,10)
nu <- rgamma(1000,5,scale=.1)
rho <- rgamma(1000,12.5,scale=4)

h <- seq(0.000001,450,length=1000)
gam <- semi(h,c0,ck,a,ak)
gam.pos <- matrix(0,ncol=1000,nrow=1000)
for(i in 1:1000){
 gam.pos[,i] <- semi(h,nug[i],var[i],nu[i],rho[i])
}

par(mar=c(5,5,1,1))
plot(h,gam,type="l",ylim=c(0,150),xlim=c(0,500),xaxs="i",yaxs="i",ylab="",xlab="",cex.lab=1.75,lwd=2.5)
title(ylab=expression(gamma),xlab="distance (km)",cex.lab=2)
for(i in 1:1000){
  lines(h,gam.pos[,i],lwd=.5,col="lightgray")
}
lines(h,gam,lwd=2.5)
text(475,110,"Sill",cex=2,col="black")
text(70,5,"Nugget",cex=2,col="black")
arrows(300,0,300,gam[710],length = 0.0,lty=2,lwd=2,col="red")

text(140,15,"range",cex=2,col="black")
arrows(190,15,285,15,length = 0.2,lty=1,lwd=3,col="black")
arrows(90,15,10,15,length = 0.2,lty=1,lwd=3,col="black")
curly(N=100,Tilt=.9,Long=0.4,scale=0.011,xcent=0.16
      ycent=.175,theta=0,col="red",lwd=2,grid=TRUE)
```

Index