1 Introduction

The purpose of this document is to present statistical background and equations for the separable emulator recently developed in our research group and implemented in R programming language. The discussion is tailored to a specific example of Greenland Ice Sheet (GIS) mass anomaly output from ice sheet model SICOPOLIS.

2 Emulator Equations

2.1 Emulator Equations

Let $y_{i,j} \in \mathbb{R}$ be physical model output of at parameter setting $\theta_i$ and time $t_j$. In our case this is SICOPOLIS model output of GIS ice mass anomaly (Gt). The time settings form an $n$-dimensional vector $t = (t_1, ..., t_n)^T$. Each parameter setting is a $m$-dimensional vector: $\theta_i = (\theta_{1,i}, ..., \theta_{m,i})$. In out case, $m=5$. The parameter settings $\theta_i$ form a $p \times m$ parameter matrix $\Theta$. Then $y_j = (y_{1,j}, ..., y_{p,j})^T$ is a $p$-dimensional vector of model outputs for all $p$ parameters for time $t_j$. Consecutively, the stacked $pn \times 1$ column matrix of all model output for times from 1 through $n$ is $Y = (y_{1}^T, ..., y_{n}^T)^T$. Associated with $Y$ is the $pn \times (m + 1)$ design matrix $D$. Its columns are the parameter values and time values of the ensemble. It is calculated as:

$$D = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{n \times 1} \otimes \Theta \otimes \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{p \times 1}$$

(1)

We model the SICOPOLIS output as a Gaussian process such that:

$$Y \sim N(\mu, \Sigma(\xi_y)),$$

(2)

where $\mu$ is a mean function that is linear in time, and $\xi_y$ is a vector of covariance matrix parameters. The mean for parameter setting $\theta_i$ and time $j$ is $\mu_{i,j}$. Specifically, $\mu = X\beta$ where $\beta$ is a column matrix of regression coefficients $\beta$ and $X$ is a matrix of covariates. It includes the column of ones (always the first column), and can also have columns of the design matrix $D$. In our specific case, we are using the mean function that is linear in time. Hence, $\beta$ has dimension of $2 \times 1$, and $X$ is $pm \times 2$. It is calculated as:
Under the assumption of separability, the covariance matrix $\Sigma$ can be represented as a Kroenecker product of a separate covariance matrix in the time $\Sigma_t$ and in parameters $\Sigma_\theta$.

$$\Sigma = \Sigma_t \otimes \Sigma_\theta. \quad (4)$$

The time covariance matrix $\Sigma_t$ $(n \times n)$ has AR(1) covariance. To reduce identifiability issues, we assume that the AR(1) process has innovation standard deviation of 1. Specifically, its $(j, k)$ element is (Shumway and Stoffer, 2006):

$$\gamma_{t,jk} = \rho^{t_j - t_k} \frac{1 - \rho^2}{1 - \rho^2}. \quad (5)$$

where $\rho$ is the lag-1 autocorrelation parameter.

The parameter covariance $\Sigma_\theta = [\gamma_{\theta, ij}]$ $(p \times p)$ is assumed to be squared exponential. Its $(i, j)$ element is:

$$\gamma_{\theta, ij} = \kappa \exp(-\sum_{k=1}^{m} \frac{|\theta_{k,i} - \theta_{k,j}|^2}{\phi_k^2}) + \zeta 1(i = j). \quad (6)$$

Here $\kappa$ is partial sill, $\zeta$ is nugget, and $\phi_k$ is range parameter for $k^{th}$ model input parameter. The range parameters form a vector $\phi = \phi_1, ..., \phi_m$.

Specifically, the total covariance matrix $(np \times np)$ is constructed as:

$$\Sigma = \begin{bmatrix}
\gamma_{t,11} & \cdots & \gamma_{t,n1} \\
\vdots & \ddots & \vdots \\
\gamma_{t,n1} & \cdots & \gamma_{t,nn}
\end{bmatrix} \quad (7)$$

Hence, the covariance parameters are $\xi_y = (\rho, \kappa, \phi, \zeta)^T$. The emulator parameters are $\psi = (\beta^T, \xi_y^T)^T$. The number of emulator parameters will be different depending on the number of model parameters used in the ensemble, and the number of covariates. In the SICOPOLIS case, this is a total of 10 parameters.

### 2.2 Estimating Emulator Parameters

The log-likelihood for the model output $Y$ given the emulator parameters $\psi$ can be written as (Rasmussen and Williams, 2006):

$$\ln L(Y|\psi) = -\frac{1}{2}(Y - \mu_\beta)^T \Sigma^{-1}(Y - \mu_\beta) - \frac{1}{2} \ln |\Sigma| - \frac{np}{2} \ln 2\pi. \quad (8)$$

The emulator parameters $\psi$ can be found by maximizing this likelihood over a reasonable parameter range using one of standard optimization routines. The regression parameters $\beta$ can either be fixed, or optimized along with other emulator parameters.
2.3 Prediction

We are interested in predicting model output for all times for a given parameter vector \( \theta^* \). We denote this output, an \( n \)-dimensional vector, by \( \mathbf{y}^* = (y_{\theta^*,1}, ..., y_{\theta^*,n})^T \). Associated with the prediction points is an \( n \times 1 \) prediction design matrix \( \mathbf{D}^* \) which is constructed in a similar manner to \( \mathbf{D} \). Likewise, matrix \( \mathbf{X}^* \) consists of covariates evaluated at prediction points. It is constructed similarly to \( \mathbf{X} \). To give an example, in the SICOPOLIS case:

\[
\mathbf{X}^* = \begin{bmatrix}
1 & t_1 \\
1 & t_2 \\
\vdots & \vdots \\
1 & t_n
\end{bmatrix}_{n \times 2}
\]  

(9)

The prediction is a multivariate normal distribution (Rasmussen and Williams, 2006):

\[
\mathbf{y}^* \sim N(\mu_{\beta}^*, \Sigma^*)
\]  

(10)

Here:

\[
\mu_{\beta}^* = \mathbf{X}^*\beta + (\Sigma_t \otimes \Sigma_{\theta^*\theta})^{-1}(\mathbf{Y} - \mu_\beta)
\]  

(11)

where \( \Sigma_{\theta^*\theta} \) is a \( 1 \times p \) cross-covariance matrix between the prediction parameter setting, and all the ensemble parameter settings, calculated using the same covariance function as for \( \Sigma_\theta \).

The predictive covariance (an \( n \times n \) matrix) is given by:

\[
\Sigma^* = (\kappa + \zeta)\Sigma_t - \Sigma_t \otimes \Sigma_{\theta^*\theta}(\Sigma_{\theta^*\theta}^{-1}\Sigma_{\theta^*\theta})^T
\]  

(12)

2.4 Computational Technique

Computational techniques can be used to simplify computations of

1. Likelihood, equation 8. Construct a \( p \times n \) matrix \( \mathbf{C} \) where \( \mathbf{Y} - \mu_\beta = \text{vec}(\mathbf{C}) \). The vec operation stacks the columns of a matrix into a column vector, from left to right. Then:

\[
(\mathbf{Y} - \mu_\beta)^T\Sigma^{-1}(\mathbf{Y} - \mu_\beta) = \text{sum}[\mathbf{C} * (\Sigma_{\theta}^{-1}\Sigma_{\theta}^{-1})]
\]  

(13)

2. Equation 11:

\[
\mu_{\beta}^* = \mathbf{X}^*\beta + (\mathbf{I}_{n \times n} \otimes \Sigma_{\theta^*\theta}^{-1})(\mathbf{Y} - \mu_\beta).
\]  

(14)