# Computer Experiments

# An Overview

# Computer experiments typically have the following components.

1. We are interested in studying a physical process that cannot easily be investigated directly.

2. We are interested in responses $y_1, y_2, \ldots, y_r$ that can be measured on the physical process. These responses are believed to depend on several factors.

3. We describe the physical process by a mathematical model implemented with code on a computer .

4. The code produces deterministic outputs $y_1(\boldsymbol{x})$, $y_2(\boldsymbol{x})$, …, $y_r(\boldsymbol{x})$ that depend on a set of input variables

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top$$

We assume the input variables are restricted to some subset

$$\mathbf{X} \subset \Re^d$$

5. The inputs include factors that are believed to affect the responses. The inputs may also include “tuning parameters” related to the code itself (for example, mesh size in finite element methods).

6. We use the code to explore or experiment with the physical process, i.e., we try different inputs in order to assess the effect of each on the response. We call this a *computer experiment.*

7. The code runs slowly. One run may take a day or longer. Thus, we can only observe (experiment with) the code a small number of times. The choice of the inputs at which to observe the code must be done carefully. Monte Carlo methods that require many runs of the code are not feasible.

# Differences between physical and computer experiments

1. The code is deterministic. There is no random error (measurement error). As a result, *no replication is needed.* Uncertainty is only due to a lack of knowledge about the true nature of the relationship between the inputs and the outputs. The code is a sort of "black box."

2. Because we know the code, we know all the factors that affect the output. Thus, techniques such as randomization and blocking are not needed, because there is no need to control for the effects of factors that affect the response but have not been included among the experimental factors.

# A Popular Statistical Approach

Observe the code at a relatively small number of runs.

Fit a statistical model to the output.

Use the statistical model as a surrogate for the computer code in the same spirit as using the computer code as a surrogate for the physical process.

Use the statistical model, possibly supplemented with additional observations from the code and/or the physical process, to achieve stated goals.

Goals might include overall model fit, estimating the inputs that optimize the output, determining which inputs have the largest effect on the output, calibrating the computer code (finding values of tuning parameters that bring the code into agreement with physical data).

# What sorts of statistical models are popular?

# Gaussian Process Models (GASP models)
## (popular in spatial statistics and sometimes referred to as kriging models)

View $y(\boldsymbol{x})$ as a realization of the random function

$$Y(\boldsymbol{x}) = \beta_0 + \beta_1 f_1(\boldsymbol{x}) + \ldots + \beta_p f_p(\boldsymbol{x}) + Z(\boldsymbol{x})$$

where $Z(\boldsymbol{x})$ is a mean zero, second-order stationary Gaussian process, and

$$\mathrm{Cov}\left(Y(\boldsymbol{x}_1), Y(\boldsymbol{x}_2)\right) = \sigma_Z^2 R(\boldsymbol{x}_1 - \boldsymbol{x}_2)$$

Here the $\beta_i$ are unknown constants (regression pararmeters) and the $f_i$ are known regression functions.

Notice that if $Z(\boldsymbol{x})$ was replaced by independent random errors, this would be the standard general linear model. However, we have allowed observations to be correlated.

What does second-order stationary Gaussian process mean?

Second-order means that the mean and variance of $Y(\boldsymbol{x})$ are constant, i.e., do not depend on $\boldsymbol{x}$.

Stationary means that the covariance (correlation) between $Y(\boldsymbol{x}_1)$ and $Y(\boldsymbol{x}_2)$ is a function only of the difference $\boldsymbol{x}_1 - \boldsymbol{x}_2$.

Gaussian process means that for any $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, the joint distribution of $Y(\boldsymbol{x}_1)$, $Y(\boldsymbol{x}_2)$, …, $Y(\boldsymbol{x}_n)$ is multivariate normal.

$R$ here is the so-called correlation function.

1. As presented here, the correlation function tells you how correlated two observations $Y(\boldsymbol{x}_1)$ and $Y(\boldsymbol{x}_2)$ are as a function of the difference $\boldsymbol{x}_1 - \boldsymbol{x}_2$ between the two inputs.

In practice, people typically use correlation functions that only depend on some measure of distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

2. Not any old function will do for $R$. $R$ must satisfy several conditions. For example,

a. $R(0) = 1$.

b. For any $n$ values $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, the $n \times n$ matrix whose $i, j$-th entry is $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$ must be a valid correlation matrix (e.g., nonnegative definite).

Two examples of correlation functions are

1. The Gaussian correlation function

$$R(\boldsymbol{x}) = \prod_{i=1}^{d} \exp(-\theta_i x_i^2)$$

2. The cubic correlation function

$$R(\boldsymbol{x}) = \prod_{i=1}^{d} R_{\theta_i}(x_i)$$

where

$$R_\theta(x) = 2(1 - \frac{x}{\theta})^3 I(\frac{\theta}{2} < x < \theta) + \left[1 - 6(\frac{x}{\theta})^2 + 6(\frac{x}{\theta})^3\right] I(x < \frac{\theta}{2})$$

and $I$ is the indicator function.

# Another take on the GASP model.

The GASP model,

$$Y(\boldsymbol{x}) = \beta_0 + \beta_1 f_1(\boldsymbol{x}) + \ldots + \beta_p f_p(\boldsymbol{x}) + Z(\boldsymbol{x})$$

can be viewed as a mixed model in the framework of the general linear model, where all observations are taken on the same subject, hence correlated. $Z(\boldsymbol{x})$ represents the within subject effect.

# Prediction with GASP models

If we observe $y(\boldsymbol{x})$ at $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, and wish to predict the value of $y(\boldsymbol{x})$ at the new input $\boldsymbol{x}_0$, we will use the so-called empirical best linear unbiased predictor, or EBLUP.

$$\hat{y}(\boldsymbol{x}_0) = \boldsymbol{f}_0^\top \hat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \hat{\boldsymbol{R}}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})$$

$$\hat{y}(\boldsymbol{x}_0) = \boldsymbol{f}_0^\top \hat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \hat{\boldsymbol{R}}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})$$

where

$\boldsymbol{f}_0 = (1, f_1(\boldsymbol{x}_0), \ldots, f_p(\boldsymbol{x}_0))^{\mathrm{T}}$

$\boldsymbol{F}$ is the $n \times (p+1)$ (design) matrix whose $i, j$-th entry is $f_{j-1}(\boldsymbol{x}_i)$, for $j > 1$ and 1 if $j = 1$.

$\boldsymbol{r}_0 = (R(\boldsymbol{x}_0 - \boldsymbol{x}_1), R(\boldsymbol{x}_0 - \boldsymbol{x}_2), \ldots, R(\boldsymbol{x}_0 - \boldsymbol{x}_n))^{\mathrm{T}}$

$\boldsymbol{R}$ is the $n \times n$ matrix whose $i, j$-th entry is $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$

$\hat{\boldsymbol{R}}$ is the maximum likelihood estimate of $\boldsymbol{R}$.

$$\hat{y}(\boldsymbol{x}_0) = \boldsymbol{f}_0^\top \hat{\boldsymbol{\beta}} + \boldsymbol{r}_0^\top \hat{\boldsymbol{R}}^{-1}(\boldsymbol{Y}^n - \boldsymbol{F}\hat{\boldsymbol{\beta}})$$

$\boldsymbol{Y}_n = (y(\boldsymbol{x}_1), y(\boldsymbol{x}_2), \ldots, y(\boldsymbol{x}_n))^{\mathrm{T}}$

and

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^\top \hat{\boldsymbol{R}}^{-1} \boldsymbol{F})^{-1} \boldsymbol{F}^\top \hat{\boldsymbol{R}}^{-1} \boldsymbol{Y}^n$$

is the generalized least squares estimate of $\beta$.

Some facts.

1. If $\boldsymbol{x}_0$ is one of the inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, say $\boldsymbol{x}_0 = \boldsymbol{x}_i$, then

$$\hat{y}(\boldsymbol{x}_i) = y(\boldsymbol{x}_i)$$

i.e., the EBLUP interpolates the data.

2. The EBLUP is a complicated nonlinear function of the data because it involves the inverse of the maximum likelihood estimate of $\boldsymbol{R}$.

3. If we use the simple intercept model as the regression part of our model, i.e., we use the model

$$Y(\boldsymbol{x}) = \beta_0 + Z(\boldsymbol{x})$$

the EBLUP is still an interpolator and does a surprisingly good job of fitting observed data.

***In practice, people often just fit this simple form of the GASP model.***

4. The GASP model and the EBLUP should look somewhat familiar. The GASP model looks like a linear model with correlated observations and the EBLUP is related to generalized least squares. In fact, this leads to another way of looking at the GASP model.

5. The mean squared error for the simple intercept EBLUP is estimated by

$$s^2(\boldsymbol{x}_0) = \hat{\sigma}_Z^2 \left( 1 - \boldsymbol{r}_0^\top \hat{\boldsymbol{R}}^{-1} \boldsymbol{r}_0 + \frac{(1 - \mathbf{1}_n^\top \hat{\boldsymbol{R}}^{-1} \boldsymbol{r}_0)^2}{1 - \mathbf{1}_n^\top \hat{\boldsymbol{R}}^{-1} \mathbf{1}_n} \right)$$

where $\hat{\sigma}_Z^2$ is the maximum likelihood estimate of $\sigma_Z^2$

If you want to try out fitting Gaussian Process models, look at the latest version of JMP (JMP 9). This is available for download from the Office of the Chief Information Officer.

You can also use home grown code. Several students have written Matlab programs for fitting these models.

There is also the MPERK program (Matlab version) in the comp_exp directory on the server tibia.

Questions?