Multi-Fidelity surrogate modeling

Claire Cannamela & **Baptiste Kerleguer**

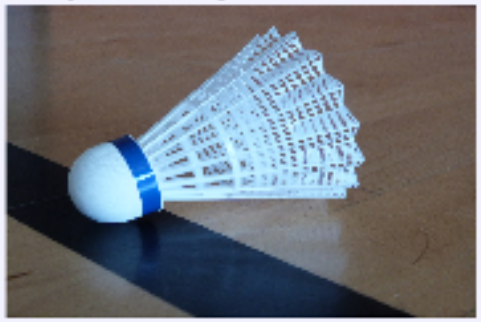
16 Juin 2021

Numerical analysis Summer school 2021 - CEA EDF INRIA

Metamodel/surrogate model

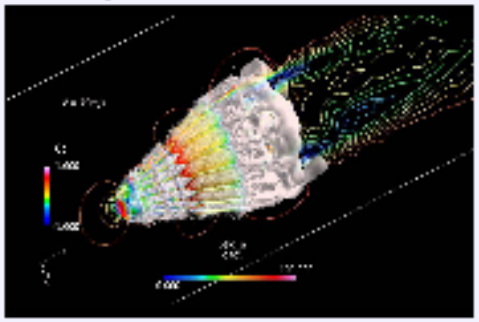
Input parameters x

Physical system



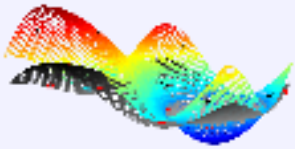
$\approx$

Computer code



$\approx$

Surrogate model

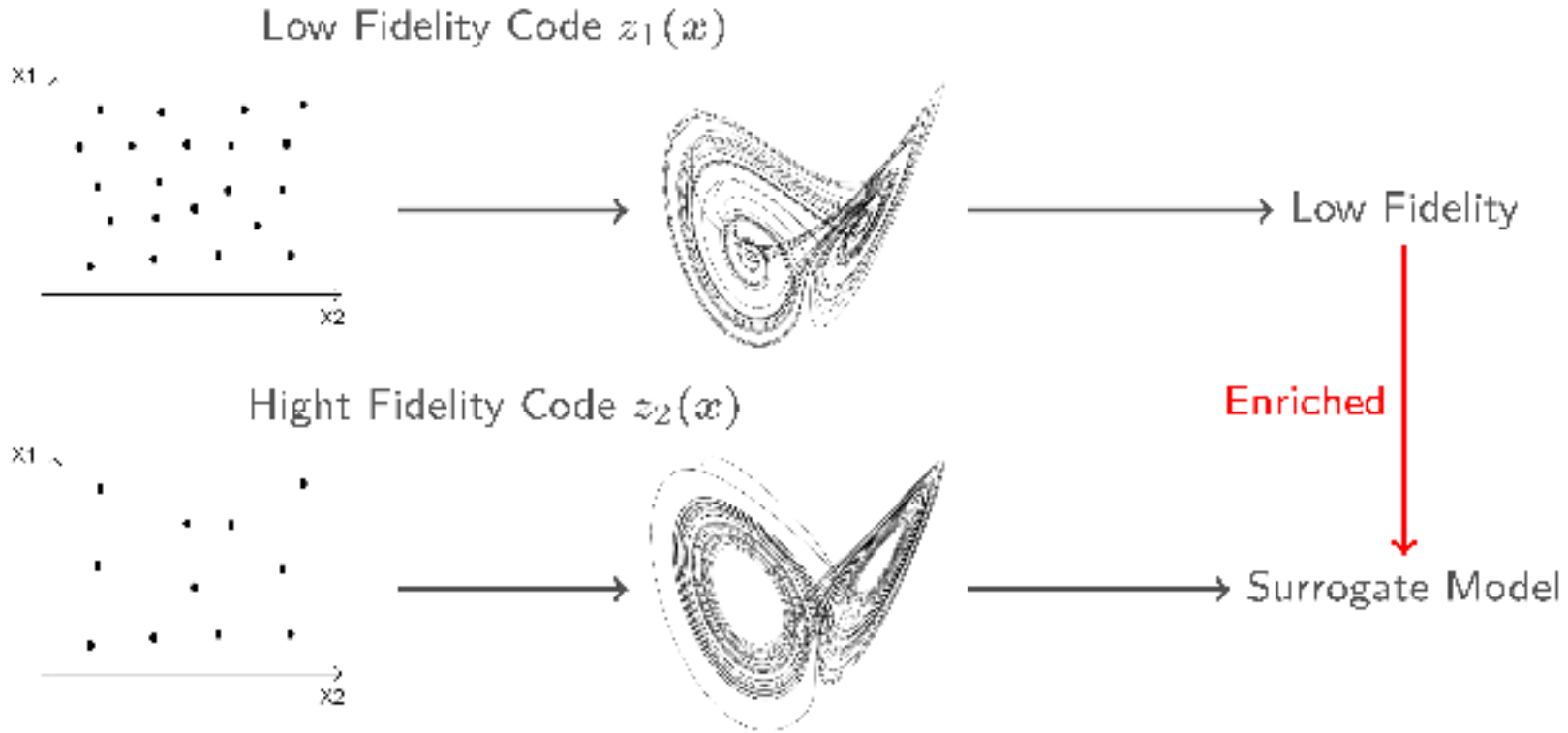


$g_{\text{exp}}(x)$ $g_{\text{num}}(x)$ $\hat{g}(x)$

Multi-fidelity surrogate model

- Objective : Replace the output of a code, called $z_2(x)$, by a metamodel.

$$z_2(x) : x \in Q \subset \mathbb{R}^d \mapsto \mathbb{R}$$



- Principle :B build a metamodel of $z_2(x)$ which integrates as well observations of the coarse code output $z_1(x)$.

Some potential applications of multi-fidelity surrogate modeling

Sea surface temperature estimation from satellite observations and in-situ measurements

Prempraneerac P., Perdikaris P., Karniadakis G. E. and Chryssostomidis C. (2017), "Sea Surface Temperature estimation from satellite observations and in-situ measurements using multifidelity Gaussian Process regression," 2017 International Conference on Digital Arts, Media and Technology.

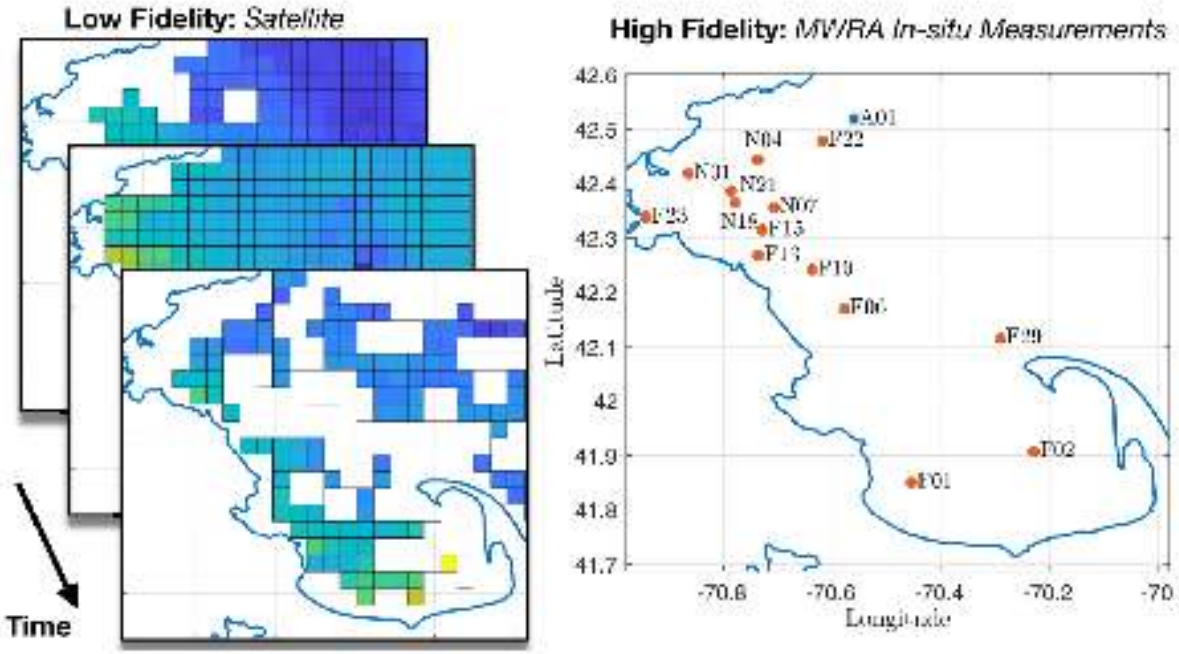


Figure 1. Multifidelity data sources: Left: The MODerate-resolution Imaging Spectroradiometer (MODIS) Terra on board NASA satellite with the spatial resolution of 0.041 degrees (latitude) × 0.041 degrees (longitude), which is equivalent to the resolution of 1 × 4 km (Wardell et al., 2013). Right: Massachusetts Water Resource Authority (MWRA) in situ measurements shown in red. The NERACOOS buoy A01 is shown in blue, and it is used only for validation purposes.

Some potential applications of multi-fidelity surrogate modeling

Infection rate of Plasmodium falciparum among African children.

Cutajar K, Pullin M, Damianou A, Lawrence N, González J (2019) Deep Gaussian processes for multi-fidelity modeling. arXiv preprint arXiv :1903.07320

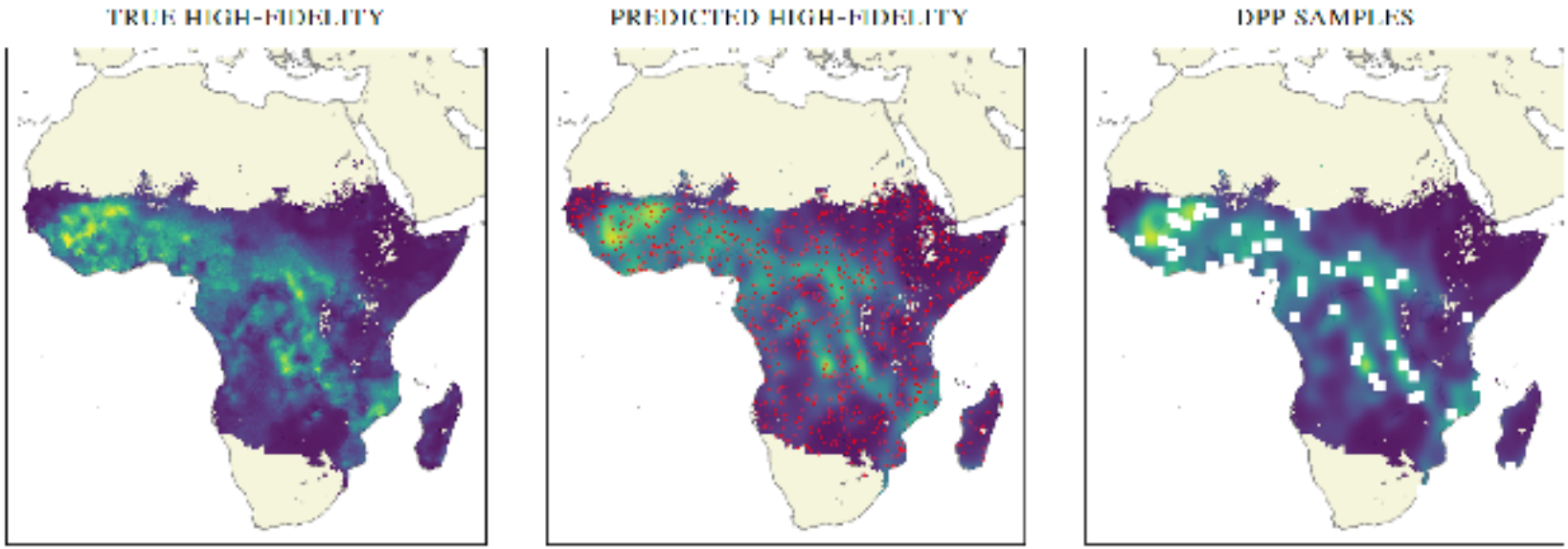
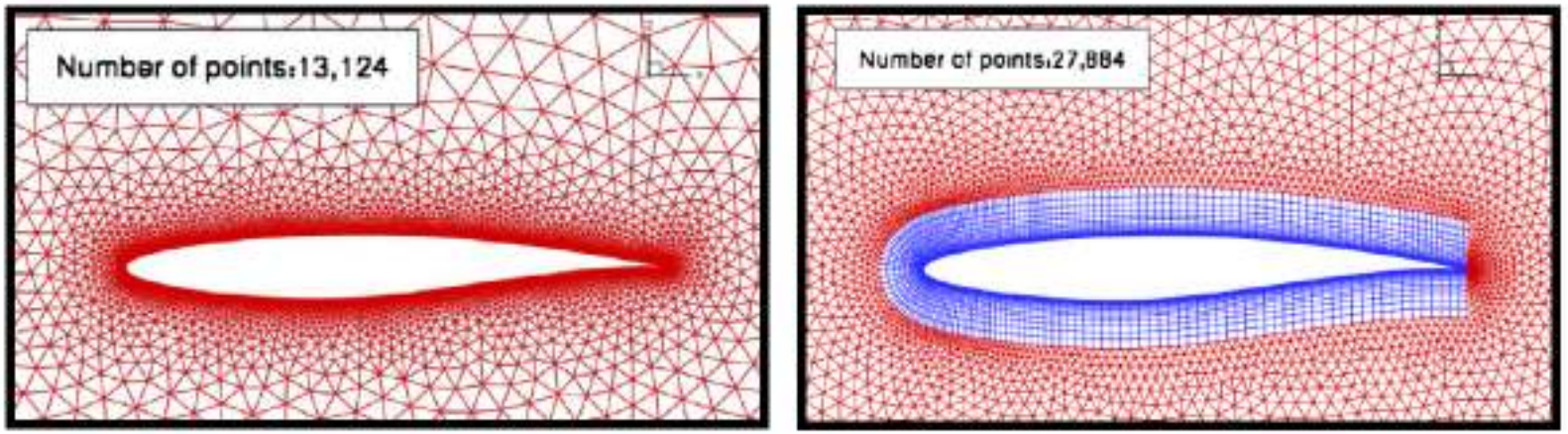


Figure 5: Real-world experiment indicating the infection rate of *Plasmodium falciparum* among African children. Lighter-shaded regions denote higher infection rates in that area of the continent. *Left:* True infection rates recorded for the year 2015. *Center:* MF-DGP predictions given low-fidelity data from 2005 and limited high-fidelity training points (marked in red) from 2015. *Right:* White squares show the samples drawn from a DPP using the posterior covariance of the MF-DGP model as its kernel.

Some potential applications of multi-fidelity surrogate modeling

Flow around an RAE 2822 airfoil.

Han, Z.H., Görtz, S. (2012), Hierarchical kriging model for variable-fidelity surrogate modeling, AIAA J. 50(9).



a) Grid for inviscid flow computations –low fidelity (26 samples)

b) Grid for viscous flow computations – high fidelity (4 samples)

Fig. 6 Computational grids for the low- and high-fidelity computations of the flow around an RAE 2822 airfoil.

Some potential applications of multi-fidelity surrogate modeling

Rayleigh-Bénard instability, or natural convection

Parussini L, Venturi D., Perdikaris P. and Karniadakis G.E. (2017), Multi-fidelity Gaussian process regression for prediction of random fields, Journal of Computational Physics

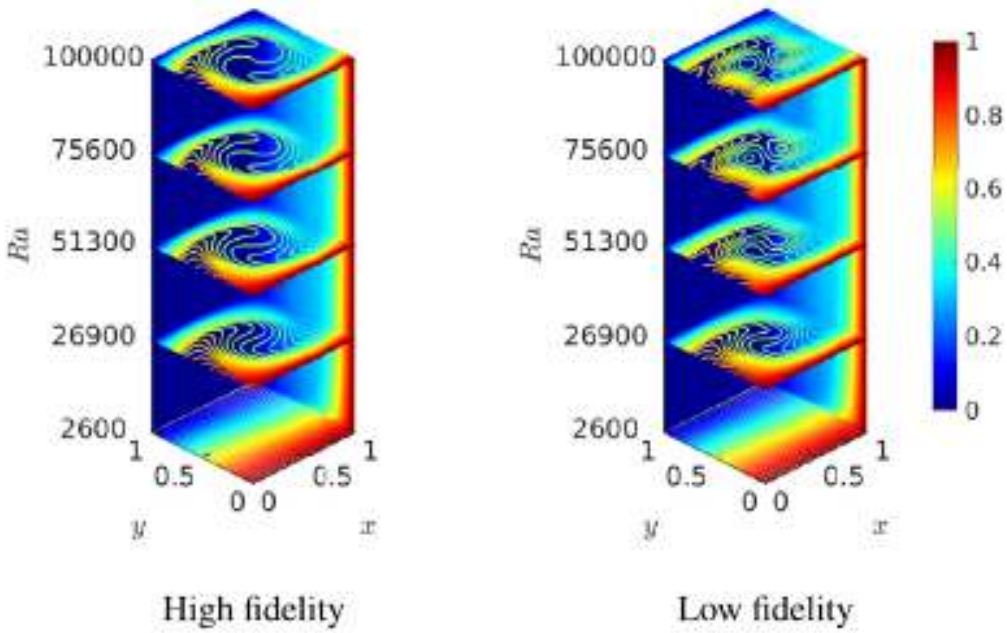


Fig. 10. Stochastic Rayleigh–Bénard convection. Steady-state temperature field corresponding to one-roll convection patterns within a wide rang numbers, i.e., between 2.6×10^3 and 10^5. Shown are simulations results with different resolutions in physical space.

Outline

Gaussian Vector

- A random vector $\boldsymbol{Z} = (Z_1, \ldots, Z_n)'$ is said to be **Gaussian** if its PDF $f_{\boldsymbol{Z}}$ is written as :

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma_{\boldsymbol{Z}}}} \exp\left\{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_{\boldsymbol{Z}})^T \Sigma_{\boldsymbol{Z}}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_{\boldsymbol{Z}})\right\},$$

- Analogously to a Gaussian random variable, $\boldsymbol{Z}$ is characterized by :
  - its mean $\boldsymbol{\mu}_{\boldsymbol{Z}} = \mathbb{E}(\boldsymbol{Z}) = (\mathbb{E}(Z_1), \ldots, \mathbb{E}(Z_n))^T$
  - its covariance matrix $\Sigma_{\boldsymbol{Z}} = \mathbb{E}\left((\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}])^T (\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}])\right)$

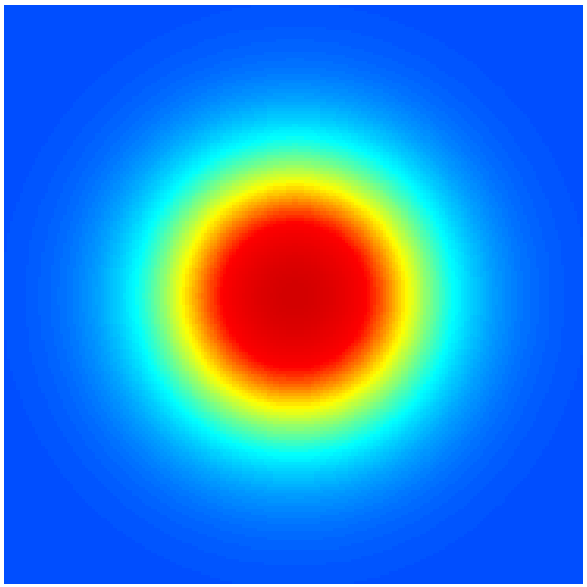
$$\Sigma_{\boldsymbol{Z}} = \begin{pmatrix} \mathbb{V}\mathrm{ar}(Z_1) & \mathrm{cov}(Z_1, Z_2) & \ldots & \mathrm{cov}(Z_1, Z_n) \\ \mathrm{cov}(Z_2, Z_1) & \mathbb{V}\mathrm{ar}(Z_2) & \ldots & \mathrm{cov}(Z_2, Z_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(Z_n, Z_1) & \mathrm{cov}(Z_n, Z_2) & \ldots & \mathbb{V}\mathrm{ar}(Z_n) \end{pmatrix}$$

$\Sigma_{\boldsymbol{Z}}$ is symmetric since $\mathrm{cov}(Z_i, Z_j) = \mathrm{cov}(Z_j, Z_i)$.

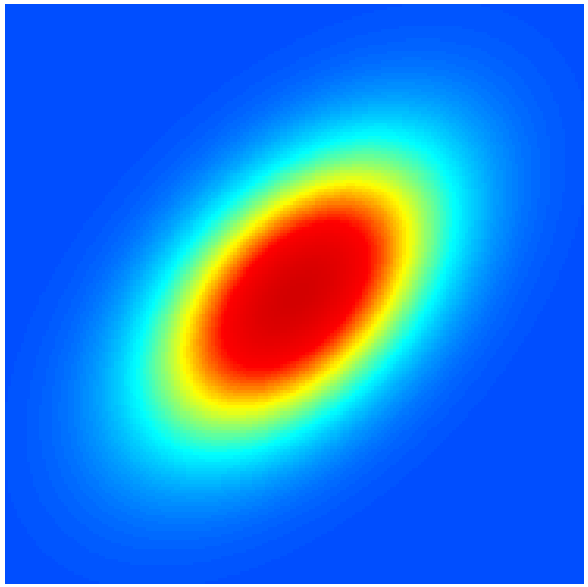
Let's assume $\Sigma_{\boldsymbol{Z}}$ is invertible.

2D graphic visualization

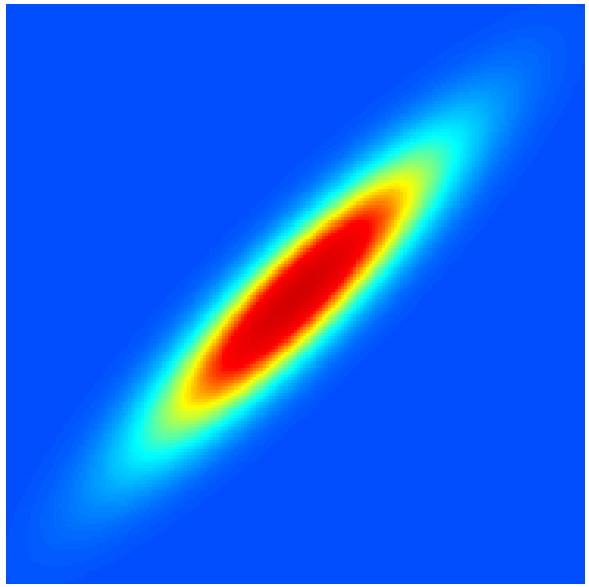
$\mathrm{cov}(Z_1, Z_2) = 0$



$\mathrm{cov}(Z_1, Z_2) = 0.5$



$\mathrm{cov}(Z_1, Z_2) = 0.9$



Reminders : Gaussian vectors and Gaussian processes

Définition

- $\boldsymbol{Z}$ is a n-dimensional **Gaussian vector** if any **linear combination of its components follows a Gaussian distribution**

$$\forall \boldsymbol{a} \in \mathbb{R}^n \qquad \boldsymbol{a}^T \boldsymbol{Z} = \sum_{i=1}^{n} a_i Z_i \quad \text{is Gaussian}$$

Properties

- The components of a Gaussian vector are Gaussian (note that the converse is not true).
- The components of a Gaussian vector $\boldsymbol{Z}$ are independent **if** they are uncorrelated ($\Sigma_{\boldsymbol{Z}}$ is diagonal).
- If the components of a vector are **Gaussian and independent** then this vector is Gaussian.
- The sum of two independent Gaussian vectors is a Gaussian vector.

Two remarkable properties

- **Stability by affine transformation**

  If $\boldsymbol{Z}$ is a Gaussian vector of dimension n, mean $\boldsymbol{Z}$ and covariance matrix Σ_Z, then for any matrix M of size m and any vector $\boldsymbol{y}$ of size m, $\boldsymbol{X} = M\boldsymbol{Z} + \boldsymbol{y}$ is also a Gaussian vector with :

$$\mathbb{E}(\boldsymbol{X}) = M\boldsymbol{\mu_Z} + \boldsymbol{y}, \quad \mathrm{cov}(\boldsymbol{X}) = M\Sigma_{\boldsymbol{Z}}M'.$$

- **Generation of independent realizations**

  To generate independent realizations of a Gaussian vector $\boldsymbol{Z}$ from a reduced centered Gaussian variable, we use the following result :

$$\boldsymbol{Z} \text{ et } R\boldsymbol{\xi} + \boldsymbol{\mu_Z} \text{ have the same law}$$

  - $\boldsymbol{Z}$ Gaussian vector of mean $\boldsymbol{\mu_Z}$ and covariance matrix $\Sigma_{\boldsymbol{Z}}$,
  - $\boldsymbol{\xi}$ centered Gaussian vector whose components are independent and of variance 1.
  - R a matrix such that $RR' = \Sigma_{\boldsymbol{Z}}$ (Cholesky decomposition).

An essential property : stability by conditioning

- **Gaussian conditioning theorem**

  Let $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ be two Gaussian vectors of sizes n_1 and n_2 such that : :

$$\begin{pmatrix} \boldsymbol{Z_1} \\ \boldsymbol{Z_2} \end{pmatrix} \sim \mathcal{N}_{n_1+n_2}\left(\begin{pmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}} \end{pmatrix}\right)$$

  with $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ mean vectors of size n_1 and n_2 respectively, the covariance matrices $\boldsymbol{\Sigma_{11}}$ of size $n_1 \times n_1$, $\boldsymbol{\Sigma_{12}}$ of size $n_1 \times n_2$, $\boldsymbol{\Sigma_{21}} = \boldsymbol{\Sigma_{12}^t}$ and $\boldsymbol{\Sigma_{22}}$ of size $n_2 \times n_2$.

  Then the distribution of $\boldsymbol{Z_1}$ **conditionally** at $\boldsymbol{Z_2}$ is also Gaussian :

$$(\boldsymbol{Z_1} \mid \boldsymbol{Z_2} = \boldsymbol{z_2}) \sim \mathcal{N}(\boldsymbol{\mu}^{\text{cond}}(\boldsymbol{z_2}), [\Sigma^{\text{cond}}(\boldsymbol{z_2})]),$$

$$\begin{cases} \boldsymbol{\mu}^{\text{cond}}(\boldsymbol{z_2}) = \boldsymbol{\mu_1} + \boldsymbol{\Sigma_{12}}\boldsymbol{\Sigma_{22}^{-1}}(\boldsymbol{z_2} - \boldsymbol{\mu_2}), \\ \Sigma^{\text{cond}}(\boldsymbol{z_2}) = \boldsymbol{\Sigma_{11}} - \boldsymbol{\Sigma_{12}}\boldsymbol{\Sigma_{22}^{-1}}\boldsymbol{\Sigma_{21}}. \end{cases}$$

Illustration of Gaussian conditioning

Let $X \sim \mathcal{N}(0,1)$, $Z \sim \mathcal{N}(0,1)$, $\mathbb{E}[XZ] = \rho \in [-1,1]$. Quantify the influence of the observation of $X = x$ on the distribution of Z.

Law of Z conditional on X :

Illustration of Gaussian conditioning

Let $X \sim \mathcal{N}(0,1)$, $Z \sim \mathcal{N}(0,1)$, $\mathbb{E}[XZ] = \rho \in [-1,1]$. Quantify the influence of the observation of $X = x$ on the distribution of Z.

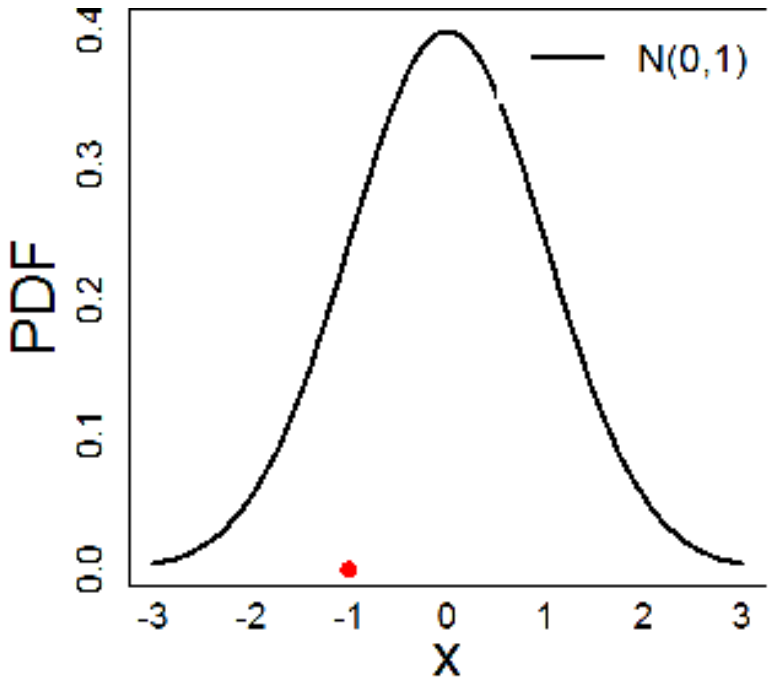
Law of Z conditional on X : $(Z \mid X = x) \sim \mathcal{N}(\rho x, 1 - \rho^2)$.

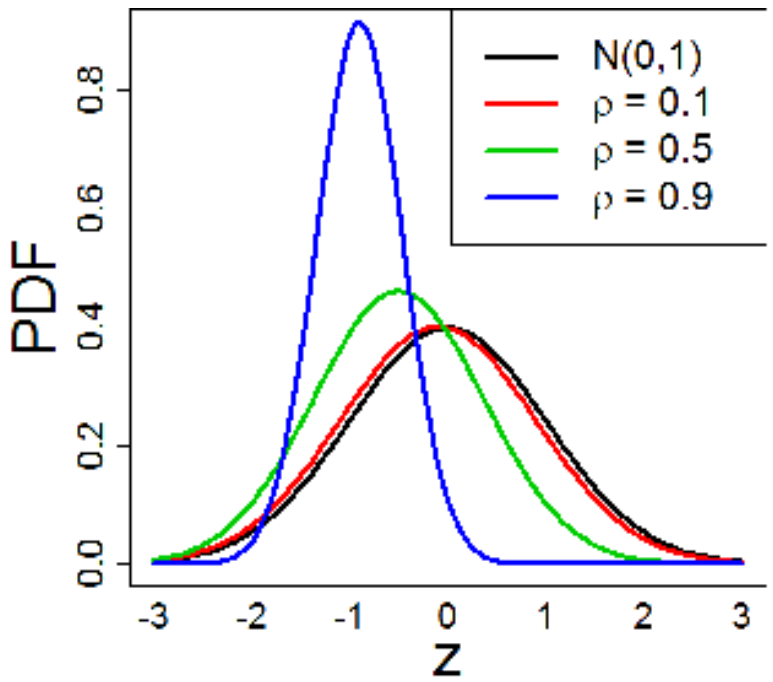
Illustration of Gaussian conditioning

Let $X \sim \mathcal{N}(0,1)$, $Z \sim \mathcal{N}(0,1)$, $\mathbb{E}[XZ] = \rho \in [-1,1]$. Quantify the influence of the observation of $X = x$ on the distribution of Z.

Law of Z conditional on X : $(Z \mid X = x) \sim \mathcal{N}(\rho x, 1-\rho^2)$.

We condition Z by the fact that $X = -1$.





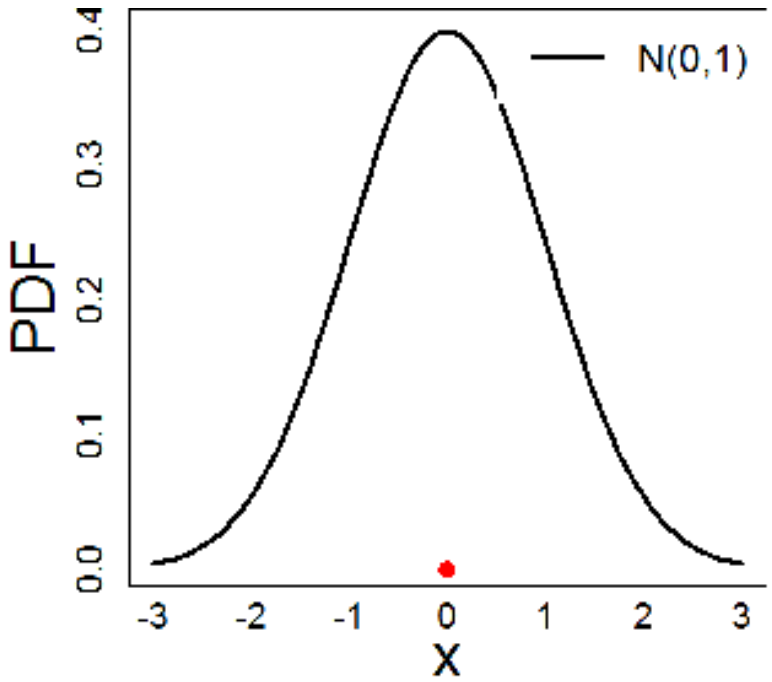
Conditioning changes the mean and reduces the variance !

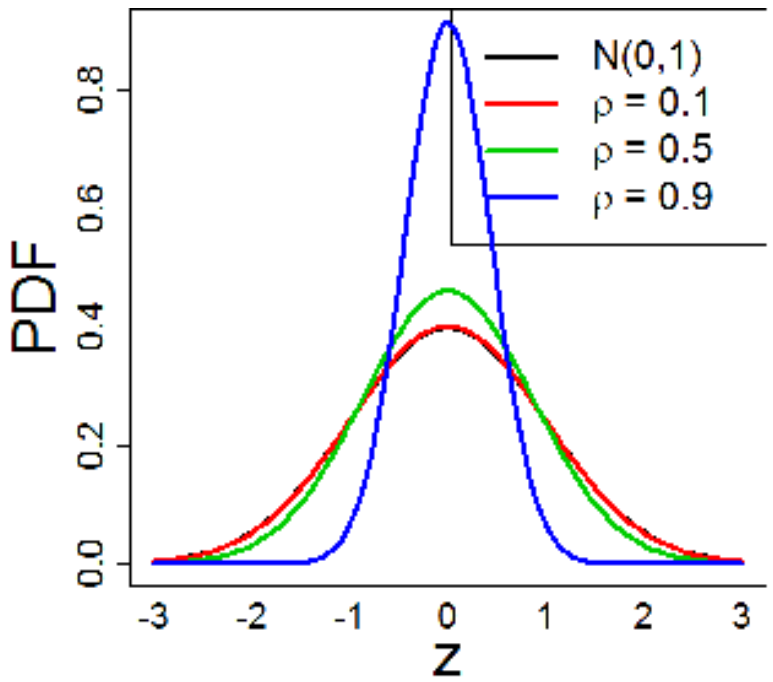
Illustration of Gaussian conditioning

Let $X \sim \mathcal{N}(0,1)$, $Z \sim \mathcal{N}(0,1)$, $\mathbb{E}[XZ] = \rho \in [-1,1]$. Quantify the influence of the observation of $X = x$ on the distribution of Z.

Law of Z conditional on X : $(Z \mid X = x) \sim \mathcal{N}(\rho x, 1-\rho^2)$.

We condition Z by the fact that $X = 0$





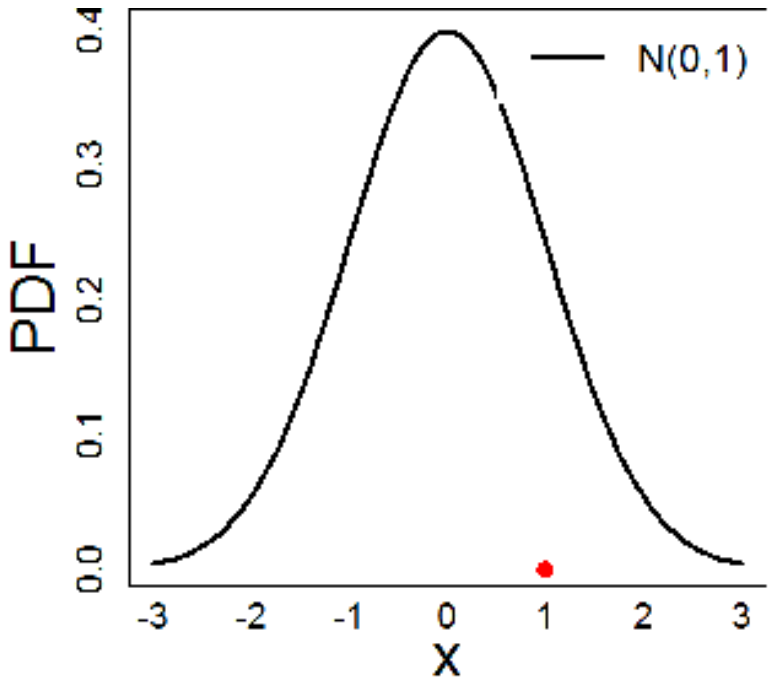
Conditioning changes the mean and reduces the variance !

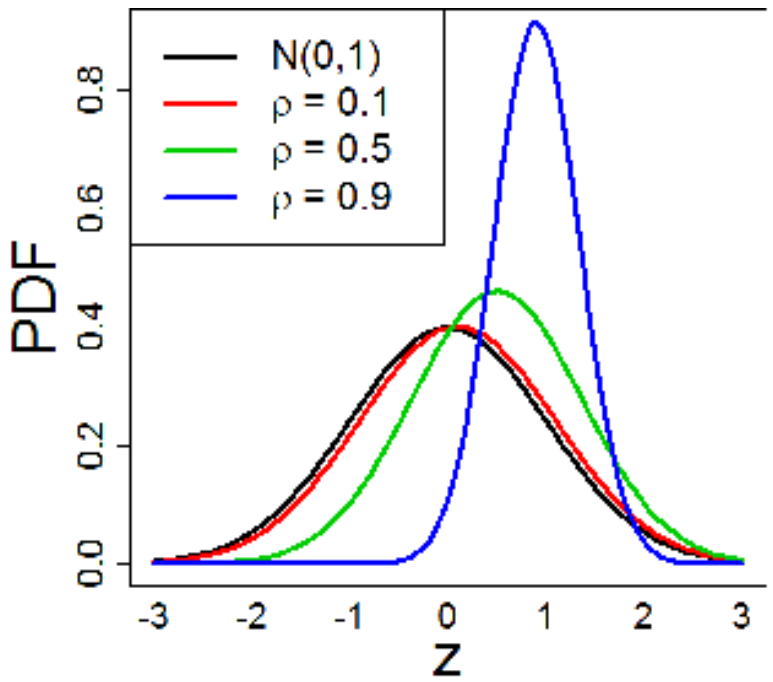
Illustration of Gaussian conditioning

Let $X \sim \mathcal{N}(0,1)$, $Z \sim \mathcal{N}(0,1)$, $\mathbb{E}[XZ] = \rho \in [-1,1]$. Quantify the influence of the observation of $X = x$ on the distribution of Z.

Law of Z conditional on X : $(Z \mid X = x) \sim \mathcal{N}(\rho x, 1-\rho^2)$.

We condition Z by the fact that $X = 1$





Conditioning changes the mean and reduces the variance !

Reminders : Gaussian vectors and Gaussian processes

Gaussian processes

- A random process (or field) $Z(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^d$, is a random variable with values in a **functional space**.
- A realization of a random process is a **function** of $\mathbb{R}^d$ in $\mathbb{R}$.
- The Gaussian process can be considered as **natural extension** of the Gaussian vector in infinite dimension.
- A random field is said to be **Gaussian** if for all $N \in \mathbb{N}^*$ and for all $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\right\}$, the random vector $\left(Z(\boldsymbol{x}^{(1)}), \ldots, Z(\boldsymbol{x}^{(n)})\right)$ is Gaussian.
- A Gaussian process is then **completely** defined by :

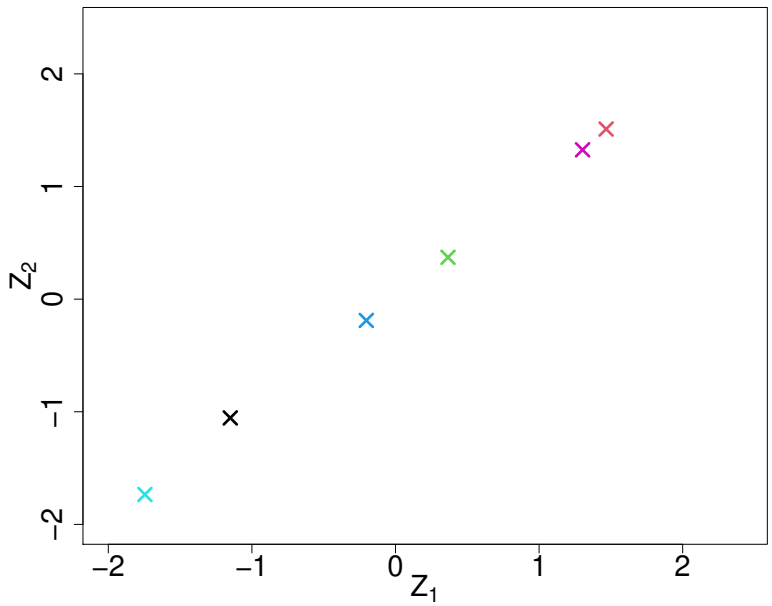
Gaussian processes

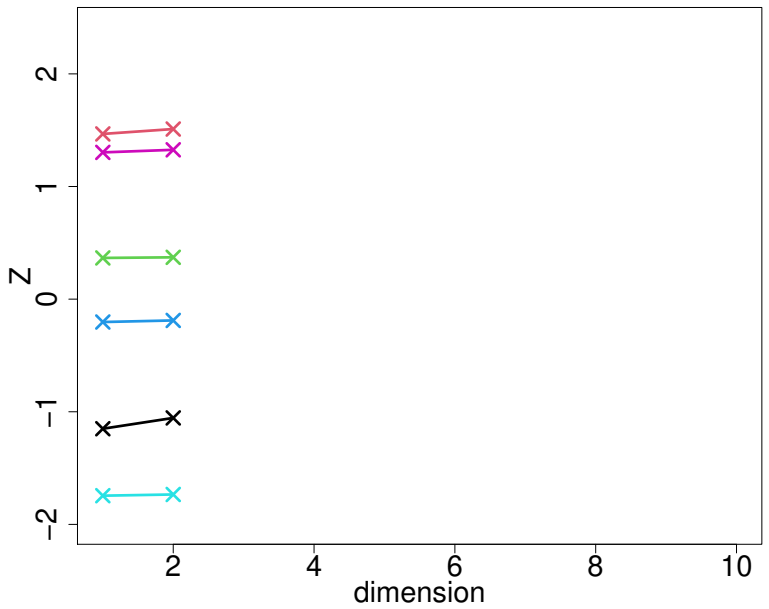
- A random process (or field) $Z(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^d$, is a random variable with values in a **functional space**.
- A realization of a random process is a **function** of $\mathbb{R}^d$ in $\mathbb{R}$.
- The Gaussian process can be considered as **natural extension** of the Gaussian vector in infinite dimension.
- A random field is said to be **Gaussian** if for all $N \in \mathbb{N}^*$ and for all $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\right\}$, the random vector $\left(Z(\boldsymbol{x}^{(1)}), \ldots, Z(\boldsymbol{x}^{(n)})\right)$ is Gaussian.
- A Gaussian process is then **completely** defined by :
- its **mean function** : $\mu(\boldsymbol{x}) = \mathbb{E}\left[Z(\boldsymbol{x})\right]$,
  it represents the trend of the Gaussian process,
- its **covariance function** : $C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \mathbb{E}\left[\left(Z(\boldsymbol{x}) - \mu(\boldsymbol{x})\right)\left(Z(\tilde{\boldsymbol{x}}) - \mu(\tilde{\boldsymbol{x}})\right)\right]$,
- we note : $Z(\boldsymbol{x}) \sim \mathcal{GP}\left(\mu(\boldsymbol{x}), C(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$.

From Gaussian vectors to Gaussian processes - illustration

Gaussian vectors are difficult to visualize in dimensions > 2, we try to represent the points next to each other.

For 2 components instead of this representation, we have this one





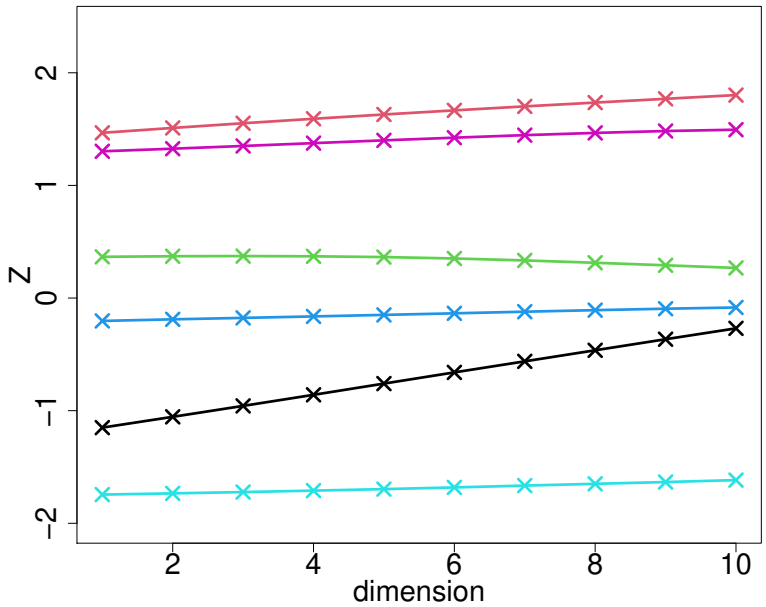
Each line is one realization of the Gaussian vector.

Z has a mean equal to zero and a given covariance matrix.

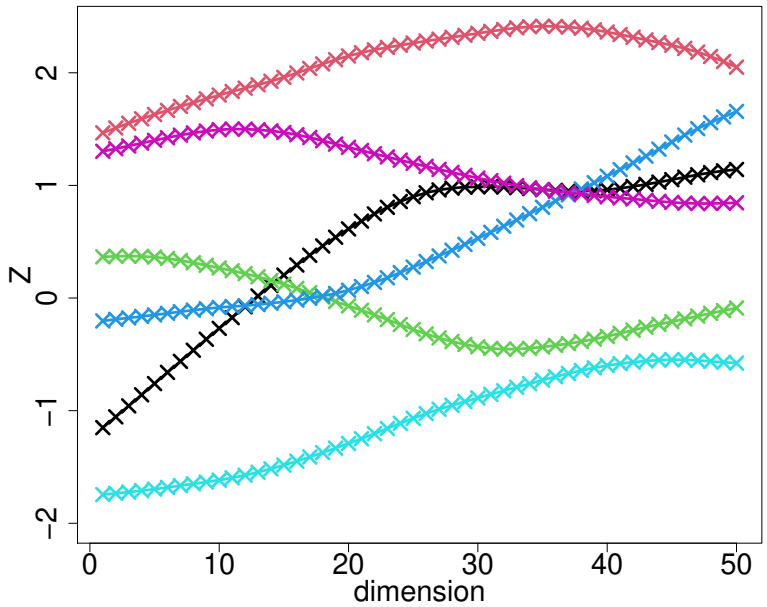
From Gaussian vectors to Gaussian processes - illustration

Gaussian vectors are difficult to visualize in dimensions > 2, we try to represent the points next to each other.

For 10 components



For 50 components



Each line is one realization of the Gaussian vector in the corresponding dimension

To think about Gaussian process, we have just to change the indexation (for $x \in \mathbb{R}$)

Covariance functions, also called covariance kernels

- $C(x, \tilde{x})$ must be **symmetric**.

Covariance functions, also called covariance kernels

- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **symmetric**.
- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **positive definite** : for all $(a_i)_{i=1,\dots,n} \in \mathbb{R}$ and distinct $(\boldsymbol{x}^{(i)})_{i=1,\dots,n} \in \mathbb{R}^d$, it satisfies the following property :

$$\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \geq 0$$

and $\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = 0$ if and only if $a_i = 0$ for all $i = 1, \dots, n$.

Reminders : Gaussian vectors and Gaussian processes

Covariance functions, also called covariance kernels

- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **symmetric**.
- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **positive definite** : for all $(a_i)_{i=1,\dots,n} \in \mathbb{R}$ and distinct $(\boldsymbol{x}^{(i)})_{i=1,\dots,n} \in \mathbb{R}^d$, it satisfies the following property :

$$\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \geq 0$$

  and $\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = 0$ if and only if $a_i = 0$ for all $i = 1, \dots, n$.
- The covariance kernel describe **relations between** $Z(\boldsymbol{x})$ and $Z(\tilde{\boldsymbol{x}})$:

Covariance functions, also called covariance kernels

- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **symmetric**.
- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **positive definite** : for all $(a_i)_{i=1,\dots,n} \in \mathbb{R}$ and distinct $(\boldsymbol{x}^{(i)})_{i=1,\dots,n} \in \mathbb{R}^d$, it satisfies the following property :

$$\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \geq 0$$

  and $\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = 0$ if and only if $a_i = 0$ for all $i = 1, \dots, n$.
- The covariance kernel describe **relations between** $Z(\boldsymbol{x})$ and $Z(\tilde{\boldsymbol{x}})$:

  Let $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = 1$, a Gaussian process $Z(\boldsymbol{x})$ with mean zero and covariance function C_1 is constant : $\forall \boldsymbol{x} \in \mathbb{R}^d$, $Z(\boldsymbol{x}) = X$, where $X \sim \mathcal{N}(0, 1)$

Reminders : Gaussian vectors and Gaussian processes

Covariance functions, also called covariance kernels

- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **symmetric**.
- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **positive definite** : for all $(a_i)_{i=1,\dots,n} \in \mathbb{R}$ and distinct $(\boldsymbol{x}^{(i)})_{i=1,\dots,n} \in \mathbb{R}^d$, it satisfies the following property :

$$\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \geq 0$$

  and $\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = 0$ if and only if $a_i = 0$ for all $i = 1, \dots, n$.
- The covariance kernel describe **relations between** $Z(\boldsymbol{x})$ and $Z(\tilde{\boldsymbol{x}})$:

  Let $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = 1$, a Gaussian process $Z(\boldsymbol{x})$ with mean zero and covariance function C_1 is constant : $\forall \boldsymbol{x} \in \mathbb{R}^d$, $Z(\boldsymbol{x}) = X$, where $X \sim \mathcal{N}(0, 1)$

  Let $C_2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \mathbf{1}_{\{\boldsymbol{x}=\tilde{\boldsymbol{x}}\}}$, a Gaussian process $Z(\boldsymbol{x})$ with mean zero and covariance function C_2 is composed of independant Gaussian values.

Reminders : Gaussian vectors and Gaussian processes

Covariance functions, also called covariance kernels

- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **symmetric**.
- $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must be **positive definite** : for all $(a_i)_{i=1,\dots,n} \in \mathbb{R}$ and distinct $(\boldsymbol{x}^{(i)})_{i=1,\dots,n} \in \mathbb{R}^d$, it satisfies the following property :

$$\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \geq 0$$

and $\sum_{i,j=1}^{n} a_i a_j C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = 0$ if and only if $a_i = 0$ for all $i = 1, \dots, n$.

- The covariance kernel describe **relations between** $Z(\boldsymbol{x})$ and $Z(\tilde{\boldsymbol{x}})$:

  Let $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = 1$, a Gaussian process $Z(\boldsymbol{x})$ with mean zero and covariance function C_1 is constant : $\forall \boldsymbol{x} \in \mathbb{R}^d$, $Z(\boldsymbol{x}) = X$, where $X \sim \mathcal{N}(0, 1)$

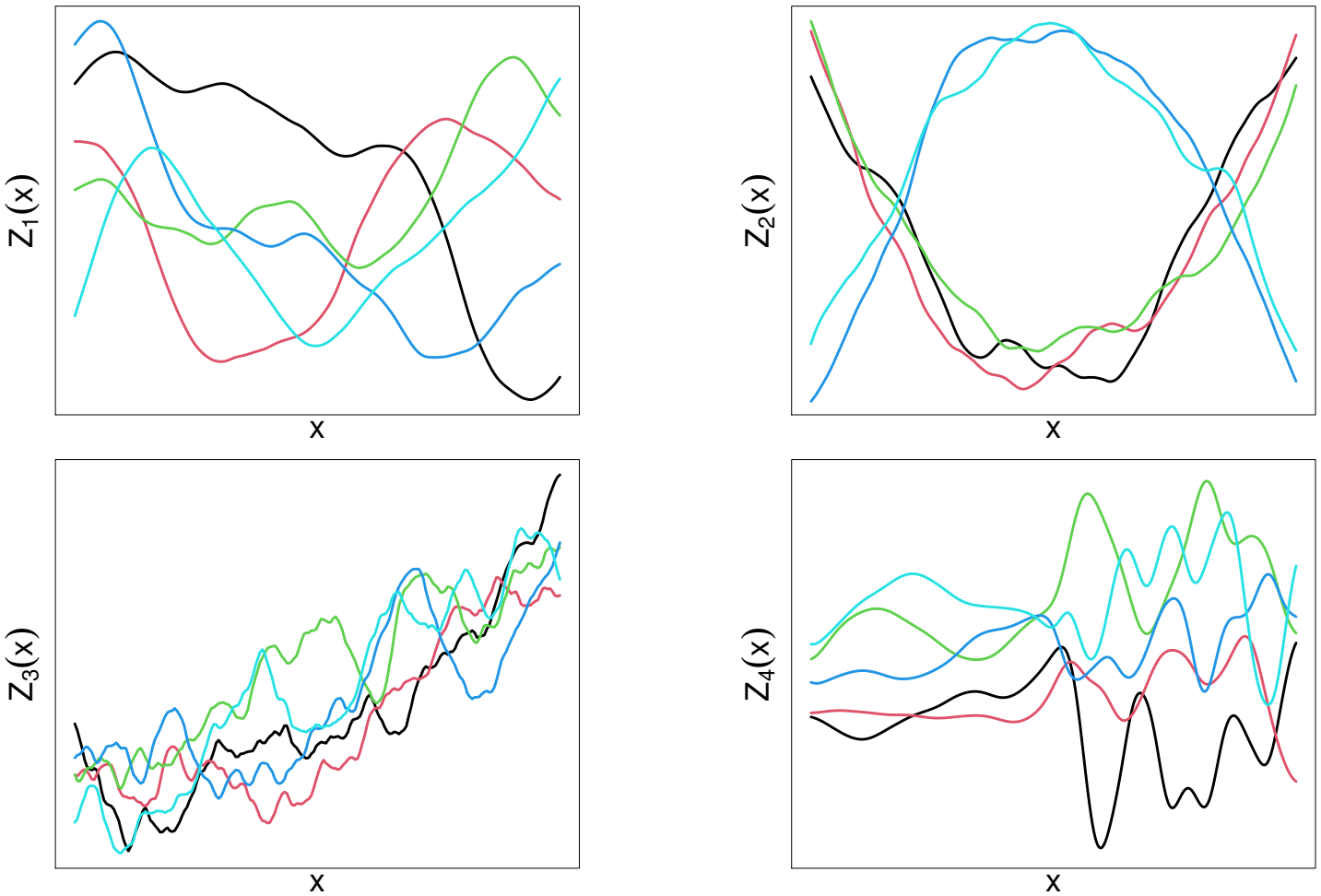
  Let $C_2(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \mathbf{1}_{\{\boldsymbol{x}=\tilde{\boldsymbol{x}}\}}$, a Gaussian process $Z(\boldsymbol{x})$ with mean zero and covariance function C_2 is composed of independant Gaussian values.

- **Stationarity covariance function** :

  $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is said to be **stationary** if it is a function of $(\boldsymbol{x} - \tilde{\boldsymbol{x}})$. The relation between $Z(\boldsymbol{x})$ and $Z(\tilde{\boldsymbol{x}})$ is uniquely determined by the distance between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$. A GP with a stationary covariance functions is said to be stationary.

Gaussian processes - illustration

From the representations of five of their realizations, can we say that $Z_1(x)$, $Z_2(x)$, $Z_3(x)$ and $Z_4(x)$ are Gaussian processes ? Is their covariance stationary ?



Some standard stationary covariance kernel on $\mathbb{R}$

We focus on stationary covariance kernel which can be written $C(x,\tilde{x}) = \sigma^2 R(x,\tilde{x})$ where $R(x,\tilde{x})$ is a correlation function.

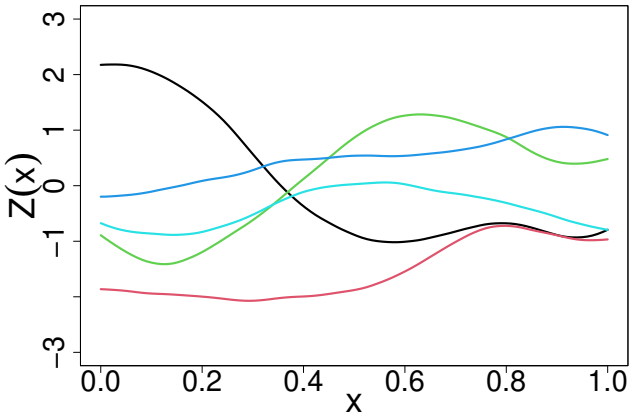
- "Nugget" kernels : $C(x,\tilde{x}) = \sigma^2 \delta_0(x-\tilde{x})$,
- Linear kernels : $C(x,\tilde{x}) = \sigma^2 \max\left(0, 1 - \frac{|x-\tilde{x}|}{\theta}\right)$,
- Exponential kernels : $C(x,\tilde{x}))\sigma^2 \exp\left(-\frac{|x-\tilde{x}x|}{\theta}\right)$,
- Gaussian kernels : $C(x,\tilde{x}) = \sigma^2 \exp\left(-\frac{(x-\tilde{x})^2}{\theta^2}\right)$,
- Matern kernels, with Γ the Euler function and $\mathcal{B}_\nu^{III}$ the Bessel function of the third type : $C(x,\tilde{x}) = \sigma^2 \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(2\sqrt{\nu}\frac{|x-\tilde{x}|}{\theta}\right)^\nu \mathcal{B}_\nu^{III}\left(2\sqrt{\nu}\frac{|x-\tilde{x}|}{\theta}\right)$.

→ parameterized by **three** parameters : **variance** σ^2, **correlation lengths** θ_i, **power** ν.

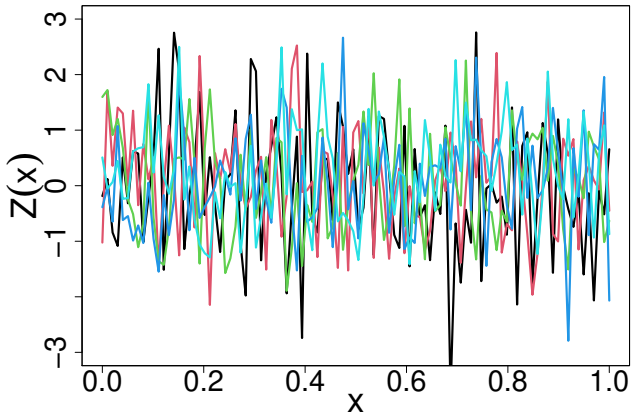
Parametric stationary covariance kernel - illustration on $\mathbb{R}$

Comment the influence of the choice of the covariance kernel and the power

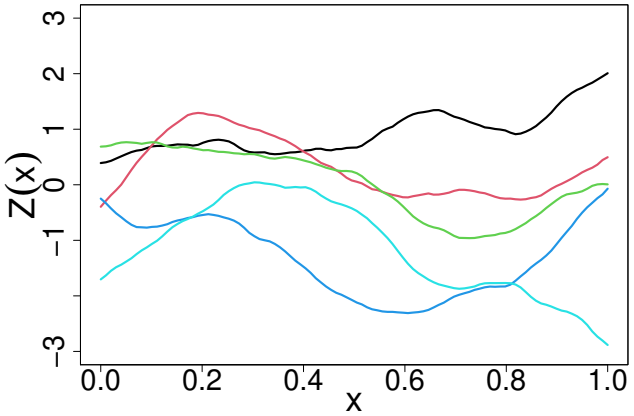
Matern : $\nu = 5/2, \sigma = 1, \theta = 0.4$



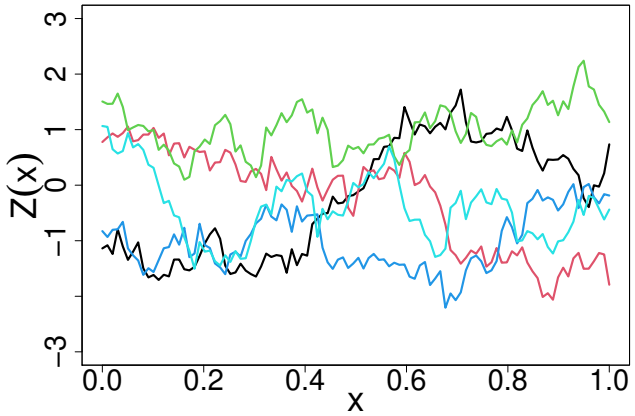
"Nugget" : $\sigma = 1$



Matern : $\nu = 3/2, \sigma = 1, \theta = 0.4$



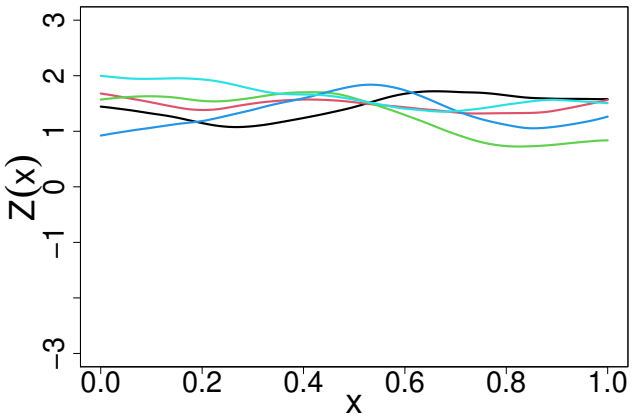
Exponential : $\sigma = 1, \theta = 0.4$



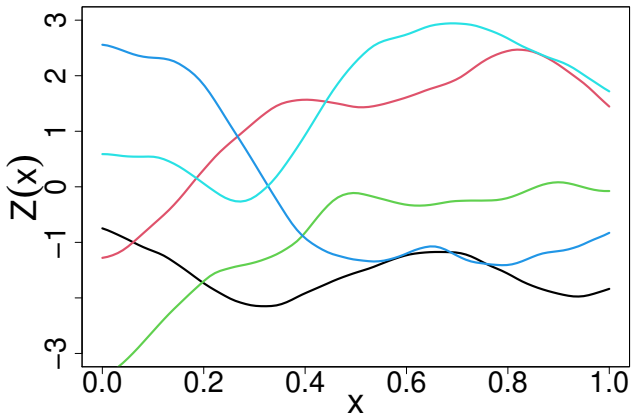
Parametric stationary covariance kernel - illustration on $\mathbb{R}$

Comment the influence of the variance parameter. What is the GP mean ? We choose a Matern kernel with $\nu = 5/2$

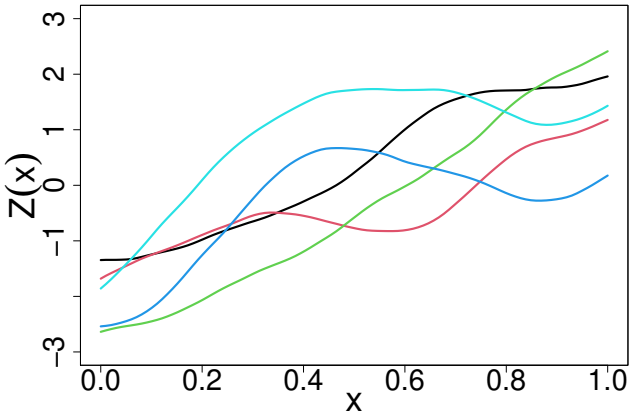
$\sigma = 0.3, \theta = 0.3$



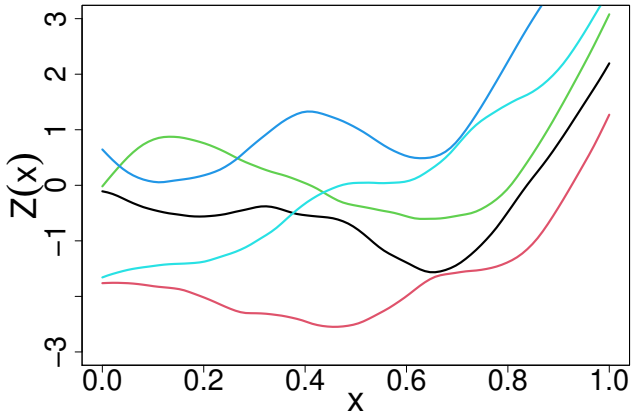
$\sigma = 1.3, \theta = 0.3$



$\sigma = 0.6, \theta = 0.3$



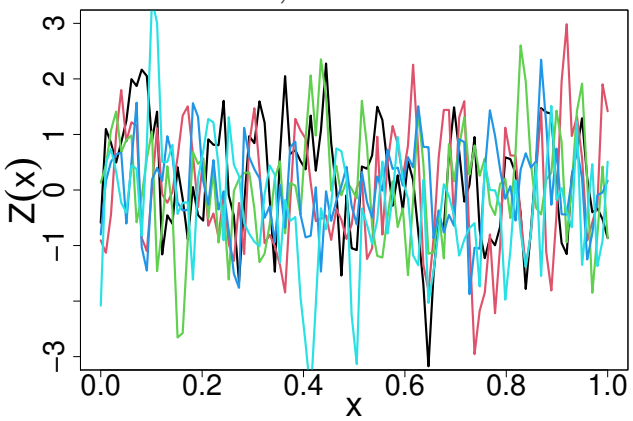
$\sigma = 1, \theta = 0.3$



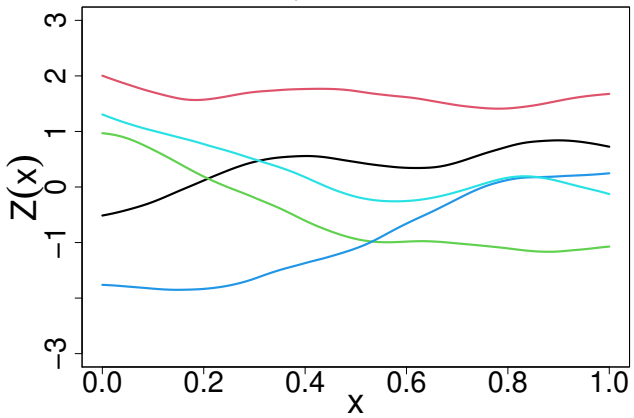
Parametric stationary covariance kernel - illustration on $\mathbb{R}$

Comment the influence of correlation length parameter.We choose a Matern kernel with $\nu = 5/2$. We choose a Matern kernel with $\nu = 5/2$

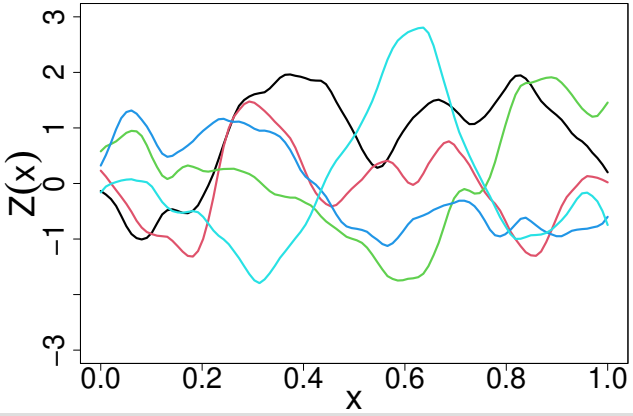
$\sigma = 1, \theta = 0.01$



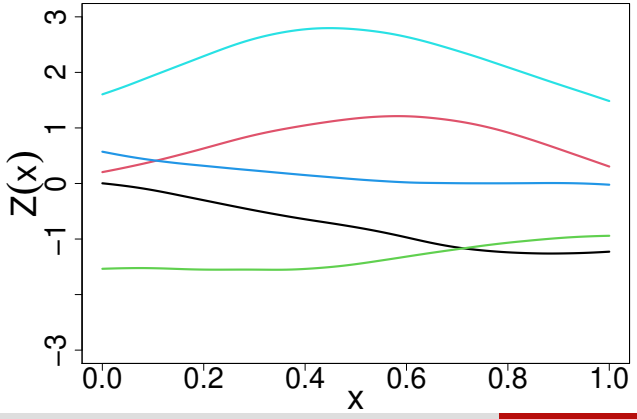
$\sigma = 1, \theta = 0.5$



$\sigma = 1, \theta = 0.1$



$\sigma = 1, \theta = 1$



Still about covariance kernels

- **Sum and product of covariance kernels** :

  Let $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ and $C_2(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ be two covariance kernels on $\mathcal{X}$, then :

  $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + C_2(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ or $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})C_2(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is a covariance kernel on $\mathcal{X}$.

- **Product with a deterministic function** :

  Let $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ be a covariance kernel on $\mathcal{X}$ and $f : \mathcal{X} \to \mathbb{R}$ a deterministic function :

  $f(\boldsymbol{x})C(\boldsymbol{x}, \tilde{\boldsymbol{x}})f(\tilde{\boldsymbol{x}})$ is a covariance kernel on $\mathcal{X}$.

- **Kernel mapping** :

  Let $C_2(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ be a covariance kernels on $\mathcal{X}_2$ and $f : \mathcal{X}_2 \to \mathcal{X}_1$ a function :

  $C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = C_2(f(\boldsymbol{x}), f(\tilde{\boldsymbol{x}}))$ is a covariance kernel on $\mathcal{X}_1$.

- **Tensorization** :

  Let $C_1, \ldots, C_d$ be covariance kernels on $\mathbb{R}$:

  $$C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = C_1(x_1, \tilde{x}_1) \times \cdots \times C_d(x_d, \tilde{x}_d),\ \boldsymbol{x} = (x_1, \ldots, u_d)$$

  is a covariance kernel on $\mathbb{R}^d$.

Standard covariance kernel on $\mathbb{R}^d$

- We can use **isotropic** covariance kernel :

$$C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma^2 \psi(||\boldsymbol{x} - \tilde{\boldsymbol{x}}||/\theta)$$

The correlation length θ is common to all directions. It controls how fast covariance changes with distance.

- We can also use **tensorized** covariance kernels on $\mathbb{R}^d$:

$$C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma^2 \prod_{i=1}^{d} \psi(|x_i - \tilde{x}_i|/\theta_i)$$

For $i = 1, \ldots, d$,θ_i is the correlation length for the variable i. It control how fast covariance changes in the direction i. θ_i small means that the variable plays an important role in the covariance changes.

- For example, for the Matern covariance kernel on $\mathbb{R}^d$, we use :

$$\psi(t) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(2\sqrt{\nu}t\right)^{\nu} \mathcal{B}_{\nu}^{III}\left(2\sqrt{\nu}t\right)$$

.

Remarks on Gaussian processes

- To create a Gaussian process, it is sufficient to create a mean function and a covariance function.
- Any function can be a mean function
- For the covariance function, the crux is to create a definite positive function.
- We presented a catalog of available covariance function on $\mathbb{R}^d$.
- Covariance function controls the order of magnitude (with σ^2) and the speed of variation of the Gaussian process (with θ)
- The regularity of the Gaussian process is directly related to the differentiability of the covariance function (parameter ν for Matern covariance for example)

Gaussian process conditioning

- Let $x \mapsto Z(x)$ be a Gaussian process with mean $\mu(x)$ and covariance $C(x, \tilde{x})$.
- We search the distribution of $Z(x)$ conditioned by values at points in $\mathbf{D} = \{x^{(1)}, \ldots, x^{(n)}\}$: $(Z(x) \mid Z(x^{(1)}) = z_1, \ldots, Z(x^{(n)}) = z_n) = (Z(x) \mid \mathbf{Z} = \mathbf{z})$.

Gaussian process conditioning

- Let $\boldsymbol{x} \mapsto Z(\boldsymbol{x})$ be a Gaussian process with mean $\mu(\boldsymbol{x})$ and covariance $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$.
- We search the distribution of $Z(\boldsymbol{x})$ conditioned by values at points in $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$: $(Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{(1)}) = z_1, \ldots, Z(\boldsymbol{x}^{(n)}) = z_n) = (Z(\boldsymbol{x}) \mid \mathbf{Z} = \mathbf{z})$.
- First, we express the joint distribution of $Z(\boldsymbol{x})$ and $\mathbf{Z}$:

$$\begin{pmatrix} Z(\boldsymbol{x}) \\ \mathbf{Z} \end{pmatrix} = \mathcal{N}_{1+n}\left(\begin{pmatrix} \mu(x) \\ \boldsymbol{\mu} \end{pmatrix}, \begin{pmatrix} C(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{r}'(\boldsymbol{x}) \\ \boldsymbol{r}(\boldsymbol{x}) & \boldsymbol{C}_{\mathbf{D}} \end{pmatrix}\right)$$

with $\boldsymbol{\mu} = (\mu(\boldsymbol{x}^{(1)}), \ldots, \mu(\boldsymbol{x}^{(n)}))$, $\boldsymbol{r}(\boldsymbol{x}) = \left(C(\boldsymbol{x}, \boldsymbol{x}^{(1)}), \ldots, C(\boldsymbol{x}, \boldsymbol{x}^{(n)})\right)$ and $\boldsymbol{C}_{\mathbf{D}} = \left(C(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})\right)_{i,j=1\ldots,n}$.

Gaussian process conditioning

- Let $x \mapsto Z(x)$ be a Gaussian process with mean $\mu(x)$ and covariance $C(x, \tilde{x})$.
- We search the distribution of $Z(x)$ conditioned by values at points in $\mathbf{D} = \{x^{(1)}, \ldots, x^{(n)}\}$: $(Z(x) \mid Z(x^{(1)}) = z_1, \ldots, Z(x^{(n)}) = z_n) = (Z(x) \mid \mathbf{Z} = \mathbf{z})$.
- First, we express the joint distribution of $Z(x)$ and $\mathbf{Z}$:

$$\begin{pmatrix} Z(x) \\ \mathbf{Z} \end{pmatrix} = \mathcal{N}_{1+n}\left(\begin{pmatrix} \mu(x) \\ \boldsymbol{\mu} \end{pmatrix}, \begin{pmatrix} C(x,x) & r'(x) \\ r(x) & C_{\mathbf{D}} \end{pmatrix}\right)$$

with $\boldsymbol{\mu} = (\mu(x^{(1)}), \ldots, \mu(x^{(n)}))$, $r(x) = \left(C(x, x^{(1)}), \ldots, C(x, x^{(n)})\right)$ and $C_{\mathbf{D}} = \left(C(x^{(i)}, x^{(j)})\right)_{i,j=1\ldots,n}$.

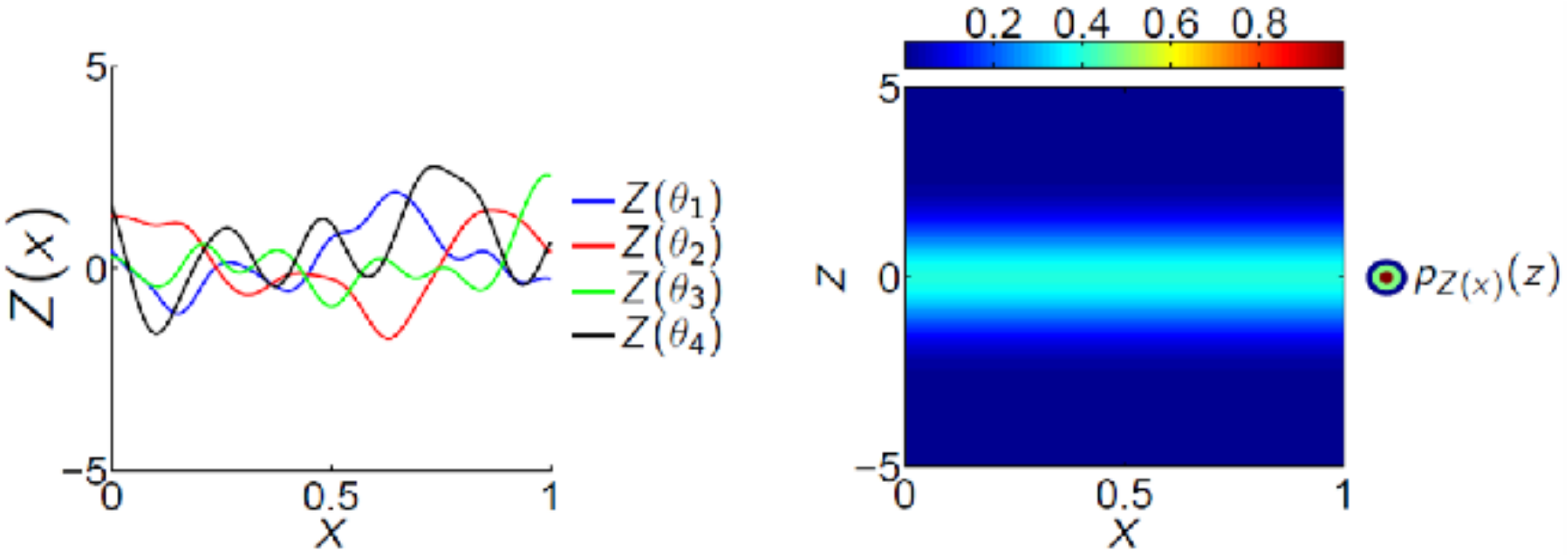
- By **Gaussian conditioning** theorem, we obtain :
  $Z^{\text{cond}}(x) = (Z(x) \mid Z(x^{(1)}) = z_1, \ldots, Z(x^{(n)}) = z_n)$ is a **Gaussian** process, with mean and covariance conditioned $\mu^{\text{cond}}(x)$ et $C^{\text{cond}}(x, \tilde{x})$ such that :

$$\mu^{\text{cond}}(x) = \mu(x) + r'(x) C_{\mathbf{D}}^{-1}(\mathbf{z} - \boldsymbol{\mu})$$

$$C^{\text{cond}}(x, \tilde{x}) = C(x, \tilde{x}) - r'(x) C_{\mathbf{D}}^{-1} r(\tilde{x})$$

Illustration of the conditioning of a Gaussian process

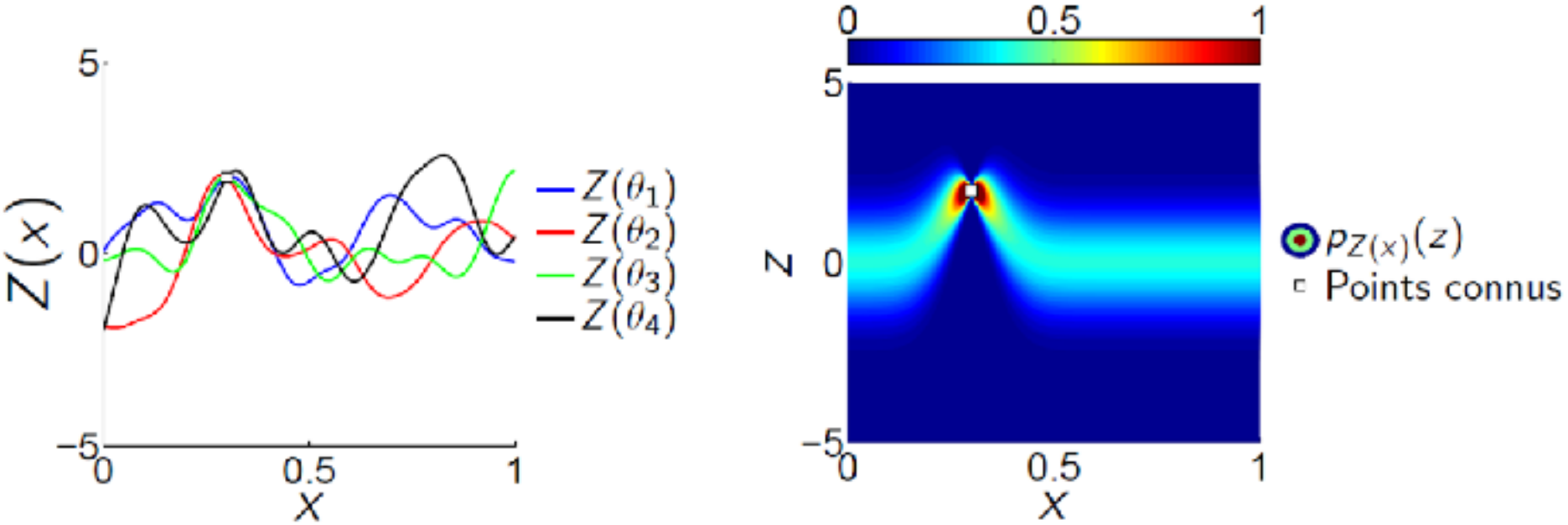
It is assumed that $Z(x)$ is a stationary Gaussian process with mean $\mu(x) = 0$ and covariance function $C(x, x') = \exp\left(-(x - x')^2/100\right)$.



Z initially unconditioned

Illustration of the conditioning of a Gaussian process

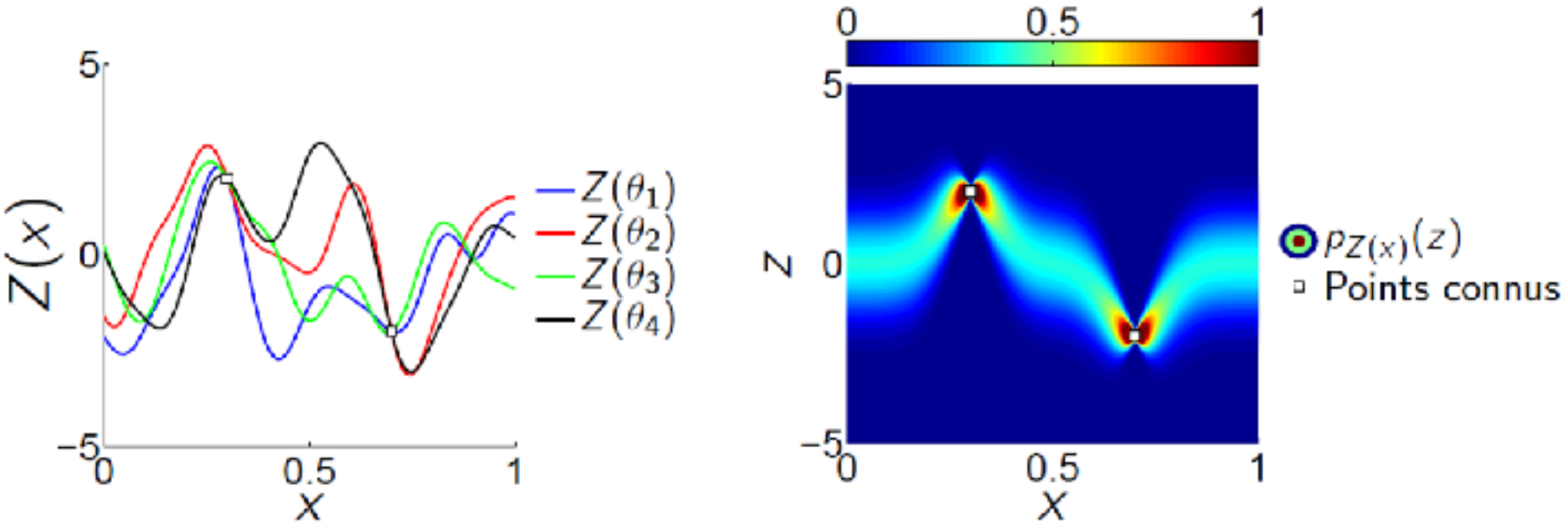
It is assumed that $Z(x)$ is a stationary Gaussian process with mean $\mu(x) = 0$ and covariance function $C(x, x') = \exp\left(-(x - x')^2/100\right)$.



Z conditioned at 1 point

Illustration of the conditioning of a Gaussian process

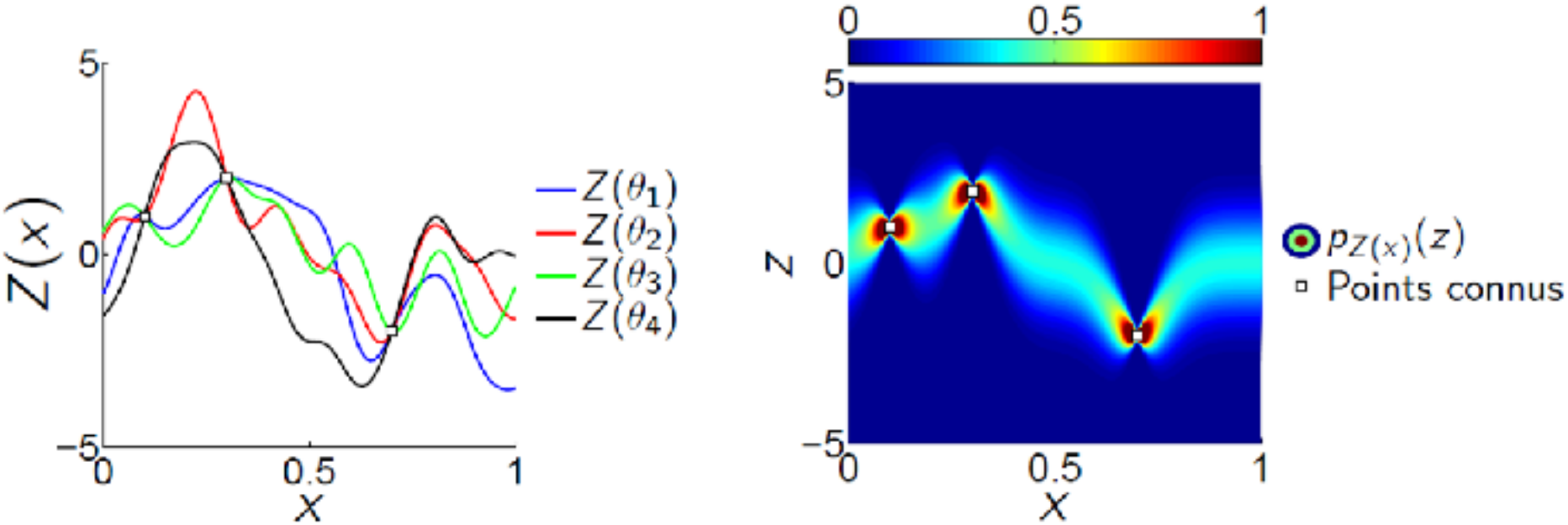
It is assumed that $Z(x)$ is a stationary Gaussian process with mean $\mu(x) = 0$ and covariance function $C(x, x') = \exp\left(-(x - x')^2/100\right)$.



Z conditioned at 2 points

Illustration of the conditioning of a Gaussian process

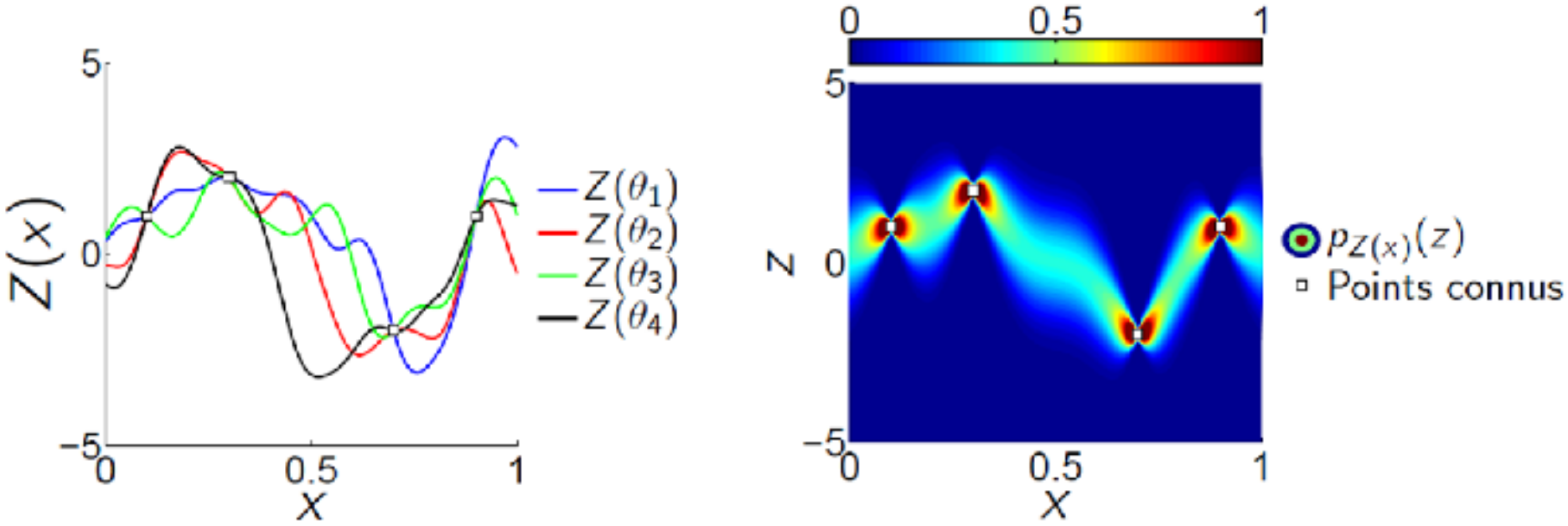
It is assumed that $Z(x)$ is a stationary Gaussian process with mean $\mu(x) = 0$ and covariance function $C(x, x') = \exp\left(-(x - x')^2/100\right)$.



Z conditioned at 3 points

Illustration of the conditioning of a Gaussian process

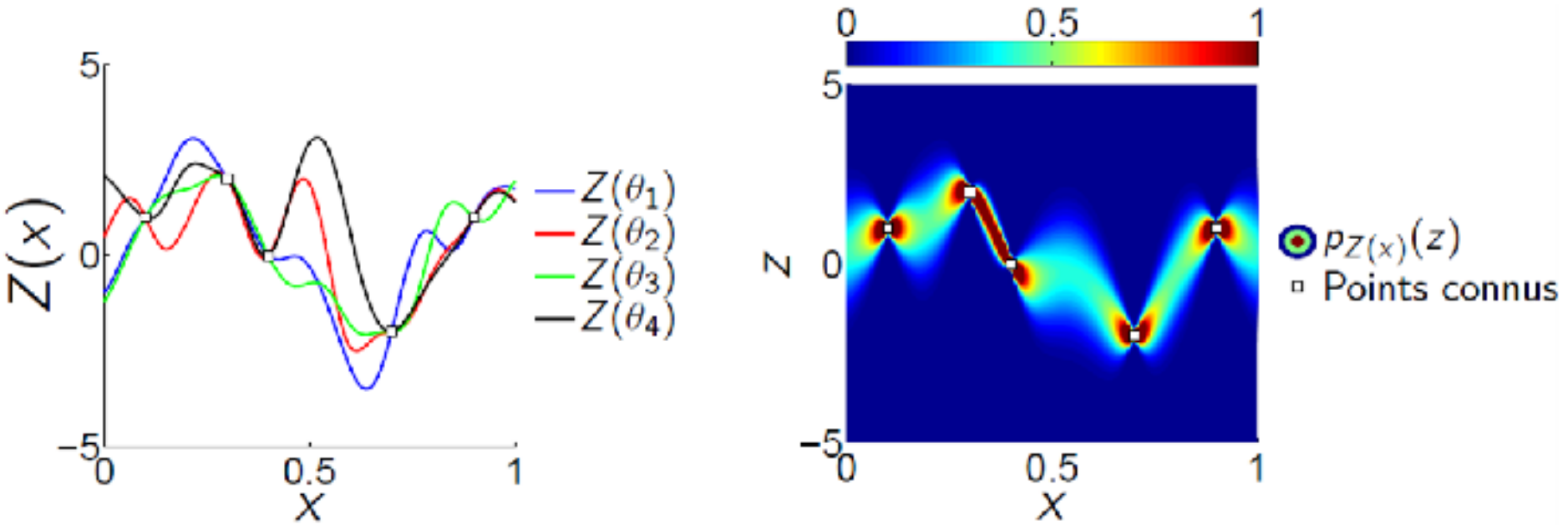
It is assumed that $Z(x)$ is a stationary Gaussian process with mean $\mu(x) = 0$ and covariance function $C(x, x') = \exp\left(-(x - x')^2/100\right)$.



Z conditioned at 4 points

Illustration of the conditioning of a Gaussian process

It is assumed that $Z(x)$ is a stationary Gaussian process with mean $\mu(x) = 0$ and covariance function $C(x, x') = \exp\left(-(x - x')^2/100\right)$.



Z conditioned at 5 points

Sampling of a conditioned Gaussian process

We want to create realizations of $Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{(1)}) = z_1, \ldots, Z(\boldsymbol{x}^{(n)}) = z_n)$ at points in $\mathcal{X}$.

We note $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$ and we write $Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\mathcal{D}) = \mathbf{z})$

We have seen previously that $Z^{\text{cond}}(\boldsymbol{x}) \sim \mathcal{GP}\left(\mu^{\text{cond}}(\boldsymbol{x}), C^{\text{cond}}(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$.

We note $\boldsymbol{C}_{\boldsymbol{\mathcal{X}}}^{\text{cond}} = [C^{\text{cond}}(\acute{\boldsymbol{x}}^{(i)}, \acute{\boldsymbol{x}}^{(j)})]_{i=1,\ldots,l}$, $(\acute{\boldsymbol{x}}^{(i)})_{i=1,\ldots,l} \in \mathcal{X}$.

Sampling of a conditioned Gaussian process

We want to create realizations of $Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{(1)}) = z_1, \ldots, Z(\boldsymbol{x}^{(n)}) = z_n)$ at points in $\mathcal{X}$.

We note $\mathcal{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$ and we write $Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\mathcal{D}) = \mathbf{z})$

We have seen previously that $Z^{\text{cond}}(\boldsymbol{x}) \sim \mathcal{GP}\left(\mu^{\text{cond}}(\boldsymbol{x}), C^{\text{cond}}(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$.

We note $\boldsymbol{C}_{\boldsymbol{\mathcal{X}}}^{\text{cond}} = [C^{\text{cond}}(\acute{\boldsymbol{x}}^{(i)}, \acute{\boldsymbol{x}}^{(j)})]_{i=1,\ldots,l}$, $(\acute{\boldsymbol{x}}^{(i)})_{i=1,\ldots,l} \in \mathcal{X}$.

- **Cholesky's decomposition of the covariance matrix** : $\boldsymbol{C}_{\mathcal{X}}^{\text{cond}} = \mathbf{L}_{\mathcal{X}}\mathbf{L}'_{\mathcal{X}}$

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at points in $\mathcal{X}$ can be obtained by sampling a noise $\boldsymbol{\xi} = [\xi_i]_{i=1,\ldots,l}$ where $(\xi_i)_{i=1,\ldots,l} \sim \mathcal{N}(0,1)$ are independant with the following equation :

  $$Z^{\text{cond}}(\mathcal{X}) = \mathbf{L}_{\mathcal{X}}\boldsymbol{\xi} + \mu^{\text{cond}}(\mathcal{X}).$$

  Can obviously be use to sample a unconditioned GP : take $\mu(\boldsymbol{x})$ and $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$.

  However, when we want to deal with large l, the cholesky decomposition can be coslty → numerically easier to sample of a unconditioned GP.

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

  $Y(\boldsymbol{x}) = \mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \tilde{\mu}^{\text{cond}}(\boldsymbol{x})$ is Gaussian

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

  $Y(\boldsymbol{x}) = \mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \tilde{\mu}^{\text{cond}}(\boldsymbol{x})$ is Gaussian

  $\mathbb{E}\left[Y(\boldsymbol{x})\right] = \mathbb{E}\left[\mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})\right] = \mu^{\text{cond}}(\boldsymbol{x})$

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

  $Y(\boldsymbol{x}) = \mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \tilde{\mu}^{\text{cond}}(\boldsymbol{x})$ is Gaussian

  $$\mathbb{E}\left[Y(\boldsymbol{x})\right] = \mathbb{E}\left[\mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})\right] = \mu^{\text{cond}}(\boldsymbol{x})$$

  $$\mathrm{cov}\left(Y(\boldsymbol{x}), Y(\tilde{\boldsymbol{x}})\right) = \mathrm{cov}\left(\tilde{Z}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) + \mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) - 2\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right)$$

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

  $Y(\boldsymbol{x}) = \mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \tilde{\mu}^{\text{cond}}(\boldsymbol{x})$ is Gaussian

  $$\mathbb{E}\left[Y(\boldsymbol{x})\right] = \mathbb{E}\left[\mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})\right] = \mu^{\text{cond}}(\boldsymbol{x})$$

  $$\mathrm{cov}\left(Y(\boldsymbol{x}), Y(\tilde{\boldsymbol{x}})\right) = \mathrm{cov}\left(\tilde{Z}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) + \mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) - 2\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right)$$

  $$\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \mathrm{cov}\left(\tilde{Z}(\mathcal{D}), \tilde{Z}(\mathcal{D})\right) \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\tilde{\boldsymbol{x}}) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\boldsymbol{x})$$

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

  $Y(\boldsymbol{x}) = \mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \tilde{\mu}^{\text{cond}}(\boldsymbol{x})$ is Gaussian

  $$\mathbb{E}\left[Y(\boldsymbol{x})\right] = \mathbb{E}\left[\mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})\right] = \mu^{\text{cond}}(\boldsymbol{x})$$

  $$\mathrm{cov}\left(Y(\boldsymbol{x}), Y(\tilde{\boldsymbol{x}})\right) = \mathrm{cov}\left(\tilde{Z}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) + \mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) - 2\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right)$$

  $$\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \mathrm{cov}\left(\tilde{Z}(\mathcal{D}), \tilde{Z}(\mathcal{D})\right) \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\tilde{\boldsymbol{x}}) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\boldsymbol{x})$$

  $$\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \mathrm{cov}\left(\tilde{Z}(\mathcal{D}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\tilde{\boldsymbol{x}})$$

Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

  Why does it work ?

  $Y(\boldsymbol{x}) = \mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \tilde{\mu}^{\text{cond}}(\boldsymbol{x})$ is Gaussian

  $$\mathbb{E}\left[Y(\boldsymbol{x})\right] = \mathbb{E}\left[\mu^{\text{cond}}(\boldsymbol{x}) + \tilde{Z}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})\right] = \mu^{\text{cond}}(\boldsymbol{x})$$

  $$\mathrm{cov}\left(Y(\boldsymbol{x}), Y(\tilde{\boldsymbol{x}})\right) = \mathrm{cov}\left(\tilde{Z}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) + \mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) - 2\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right)$$

  $$\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{\mu}^{\text{cond}}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \mathrm{cov}\left(\tilde{Z}(\mathcal{D}), \tilde{Z}(\mathcal{D})\right) \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\tilde{\boldsymbol{x}}) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\boldsymbol{x})$$

  $$\mathrm{cov}\left(\tilde{\mu}^{\text{cond}}(\boldsymbol{x}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \mathrm{cov}\left(\tilde{Z}(\mathcal{D}), \tilde{Z}(\tilde{\boldsymbol{x}})\right) = \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\tilde{\boldsymbol{x}})$$

  $$\rightarrow \mathrm{cov}\left(Y(\boldsymbol{x}), Y(\tilde{\boldsymbol{x}})\right) = C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{C}_{\mathcal{D}}^{-1} \boldsymbol{r}(\boldsymbol{x}) = C^{\text{cond}}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$$

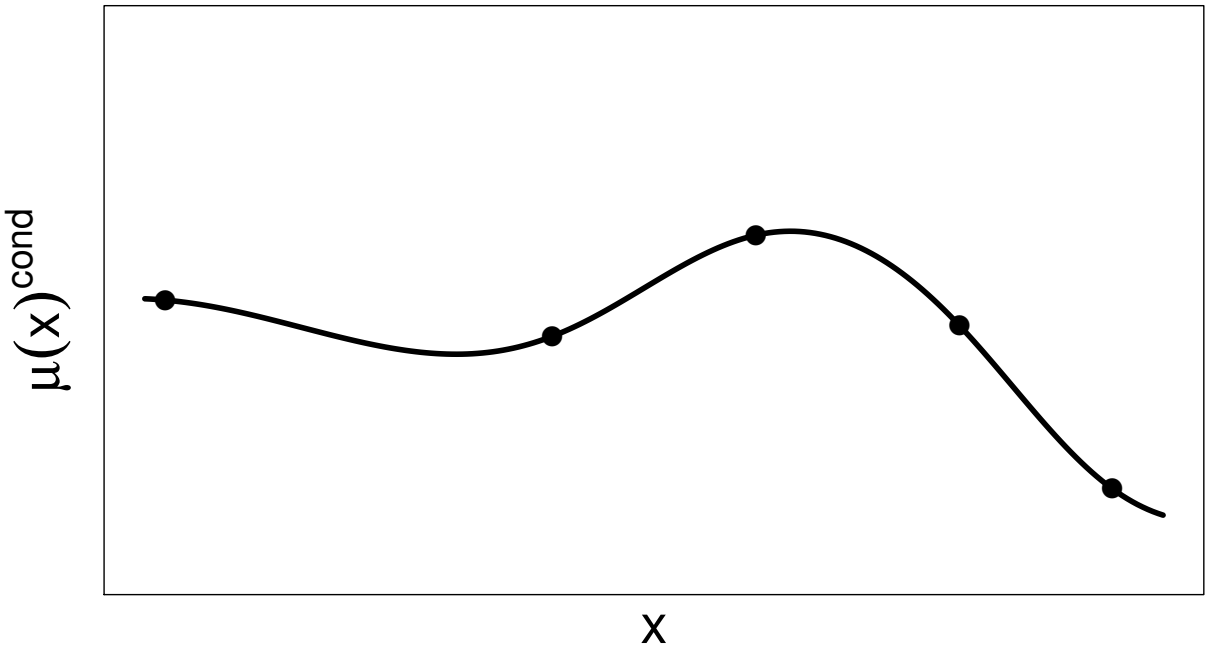
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



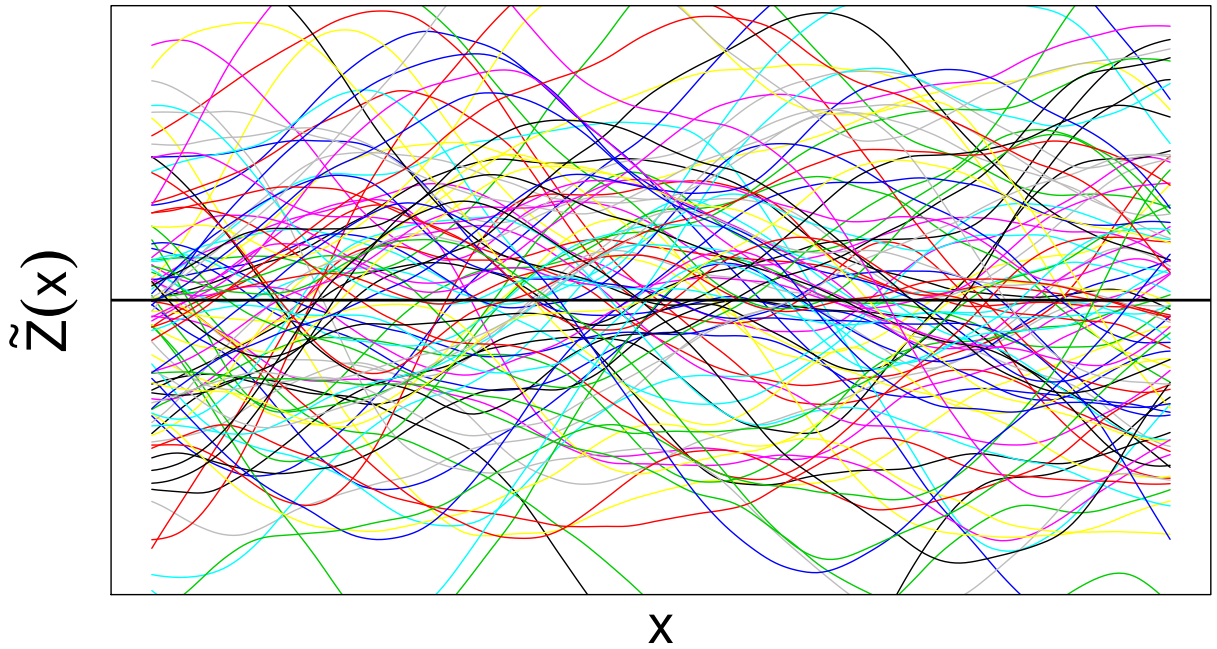
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



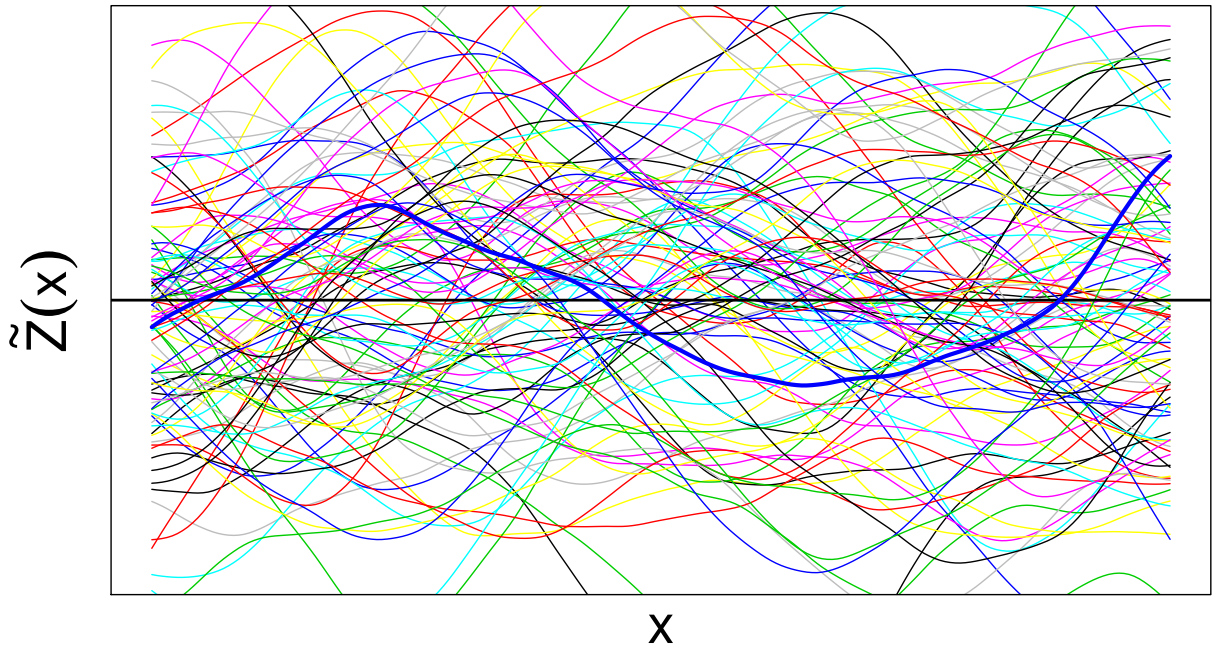
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



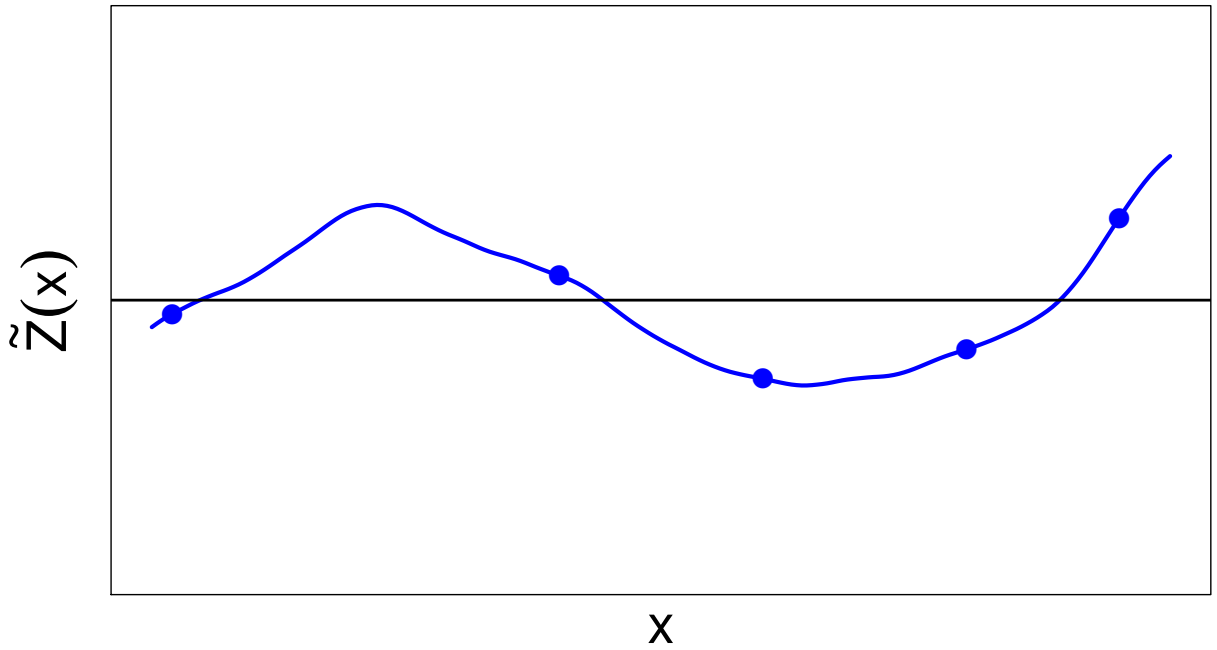
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



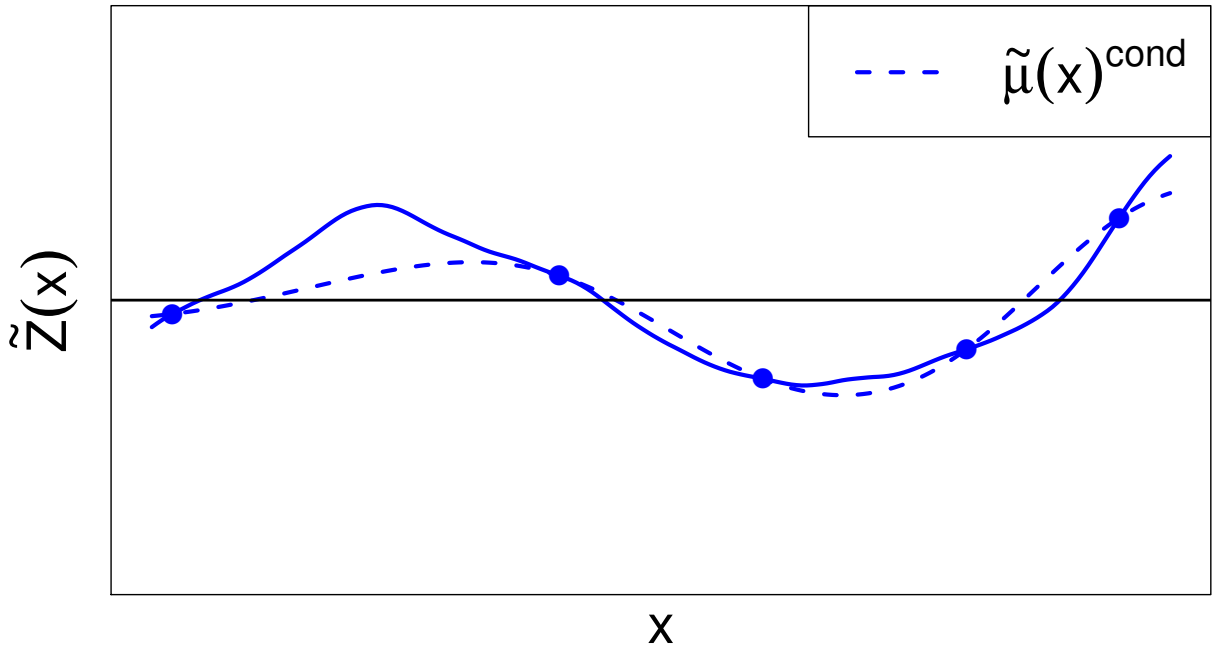
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



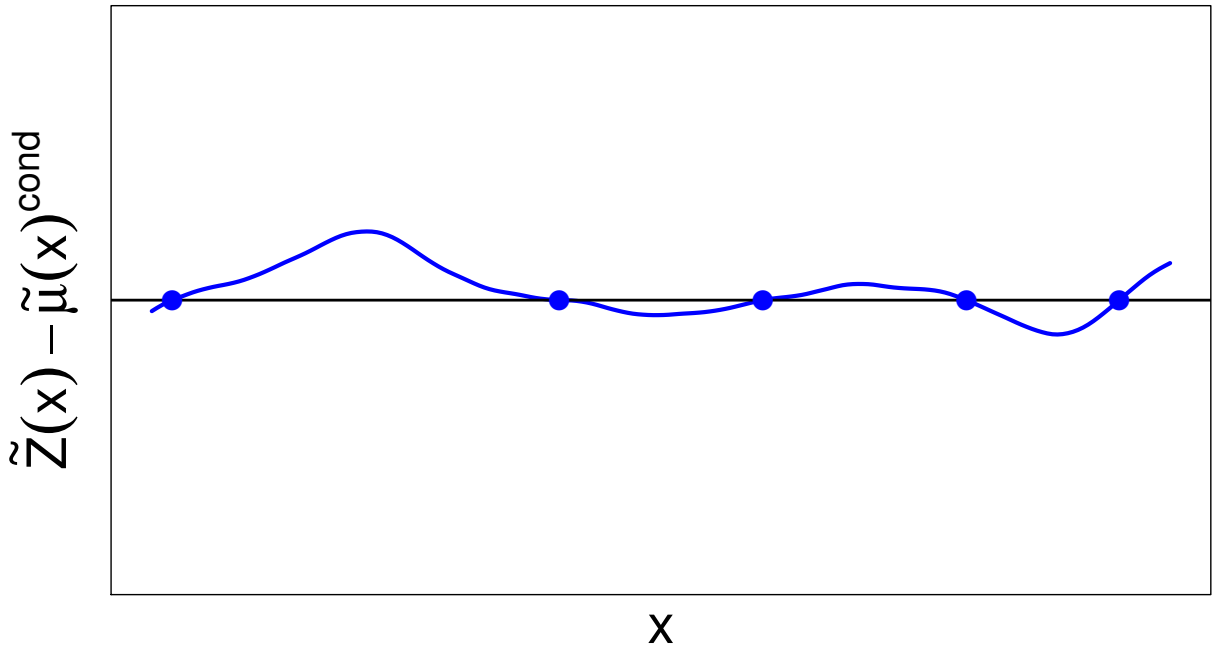
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



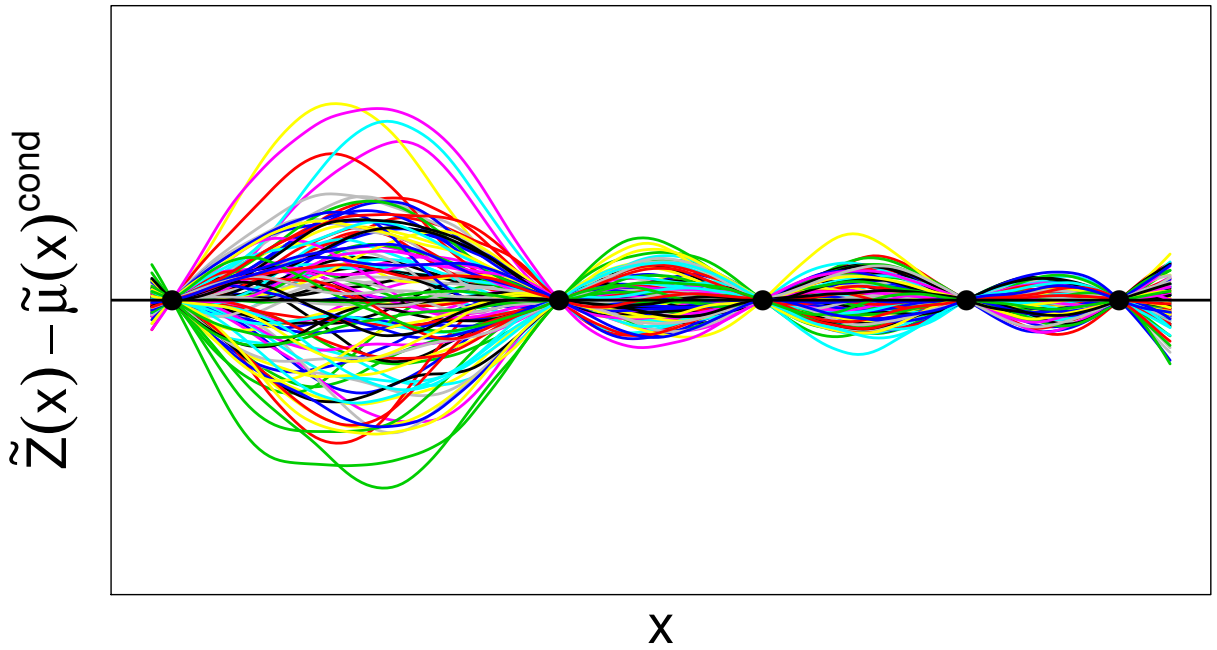
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



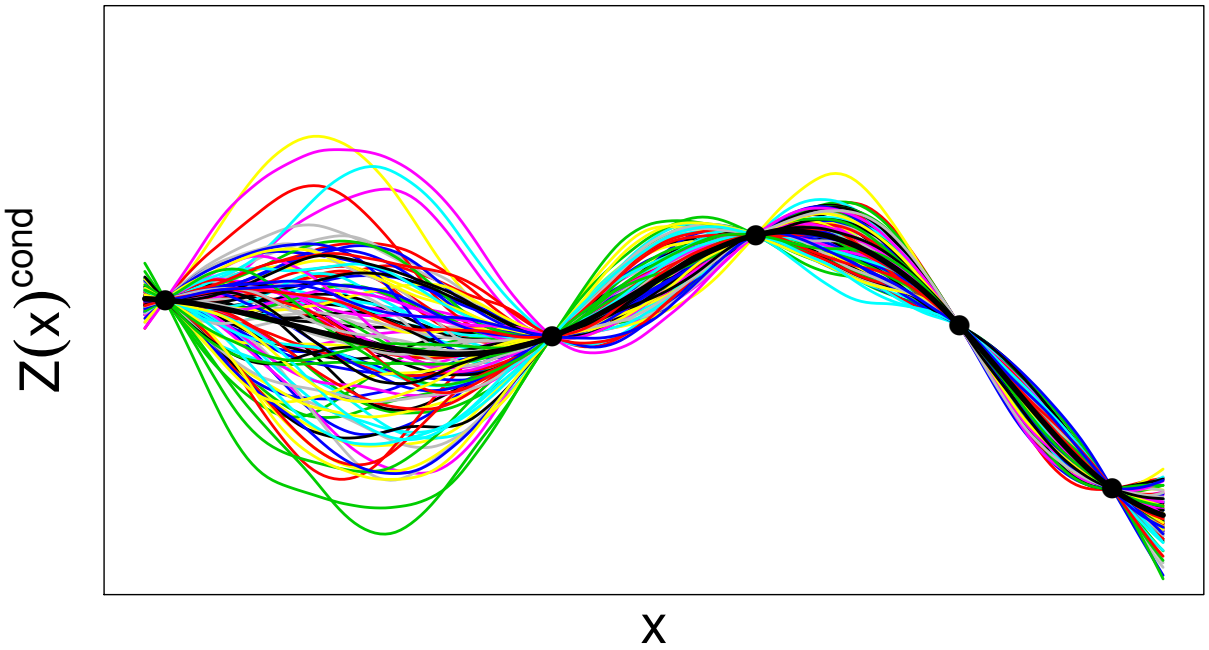
Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \mathrm{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

$$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$



Samping of a conditioned Gaussian process

- **Sample a conditioned GP from a unconditioned GP** (Chiles & Delfiner, 1999) :

  A realization of $Z^{\text{cond}}(\boldsymbol{x})$ at $\mathcal{X}$ can be obtained by sampling a unconditioned GP $\tilde{Z}(\boldsymbol{x}) \sim \text{GP}(0, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ at $\mathcal{X}$ and following the result :

  $$Z^{\text{cond}}(\mathcal{X}) = \mu^{\text{cond}}(\mathcal{X}) + \tilde{Z}(\mathcal{X}) - \tilde{\mu}^{\text{cond}}(\mathcal{X})$$

  with $\tilde{\mu}^{\text{cond}}(\mathcal{X}) = \boldsymbol{r}(\mathcal{X})' \boldsymbol{C}_{\mathcal{D}}^{-1} \tilde{Z}(\mathcal{D})$ corresponds to the mean of $\tilde{Z}(\boldsymbol{x})$ conditioned by its own values at $\mathcal{D}$, $\boldsymbol{r}(\mathcal{X}) = \left(C(\mathcal{X}, \boldsymbol{x}^{(1)}), \ldots, C(\mathcal{X}, \boldsymbol{x}^{(n)})\right)$

- To sample an unconditioned GP, several methods could be mentioned :

  – Fourier spectral decomposition

  Stein, M. (1999), Interpolation of Spatial Data, New York : Springer Series in Statistics.

  – Karhunen-Loeve spectral decomposition

  Rasmussen, C. and Williams C. (2006), Gaussian Processes for Machine Learning, Cambridge : MIT Press.

  – Propagative version of the Gibbs sampler

  Lantuéjoul, C. and Desassis N. (2012), Simulation of a Gaussian random vector : A propagative version of the Gibbs sampler, In The 9th International Geostatistics Congress, Norway.

Outline

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.
- We are interested in approximating an function $z(\boldsymbol{x}) \in \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^d$ from few of its observations at points in the DOE $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$. We note $\mathbf{z} = (z_1, \ldots, z_n)$ these observations.

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.
- We are interested in approximating an function $z(\boldsymbol{x}) \in \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^d$ from few of its observations at points in the DOE $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$. We note $\mathbf{z} = (z_1, \ldots, z_n)$ these observations.
- **Gaussian process model hypothesis** : representing the unknown fonction $z(\boldsymbol{x})$ by a realization of a Gaussian process (it is the prior)

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.
- We are interested in approximating an function $z(\boldsymbol{x}) \in \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^d$ from few of its observations at points in the DOE $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$. We note $\mathbf{z} = (z_1, \ldots, z_n)$ these observations.
- **Gaussian process model hypothesis** : representing the unknown fonction $z(\boldsymbol{x})$ by a realization of a Gaussian process (it is the prior)
- The approximation entirely based on the theory of conditioned Gaussian processes except that neither the mean function nor the covariance function is known.

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.
- We are interested in approximating an function $z(\boldsymbol{x}) \in \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^d$ from few of its observations at points in the DOE $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$. We note $\mathbf{z} = (z_1, \ldots, z_n)$ these observations.
- **Gaussian process model hypothesis** : representing the unknown fonction $z(\boldsymbol{x})$ by a realization of a Gaussian process (it is the prior)
- The approximation entirely based on the theory of conditioned Gaussian processes except that neither the mean function nor the covariance function is known.
- When the mean is known, it is called simple kriging, and when the mean is unknown, it is called universal kriging.

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.
- We are interested in approximating an function $z(\boldsymbol{x}) \in \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^d$ from few of its observations at points in the DOE $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$. We note $\mathbf{z} = (z_1, \ldots, z_n)$ these observations.
- **Gaussian process model hypothesis** : representing the unknown fonction $z(\boldsymbol{x})$ by a realization of a Gaussian process (it is the prior)
- The approximation entirely based on the theory of conditioned Gaussian processes except that neither the mean function nor the covariance function is known.
- When the mean is known, it is called simple kriging, and when the mean is unknown, it is called universal kriging.
- A common choice for **the mean** is to take a **linear form**.

  $z(\boldsymbol{x})$ is a realization of $Z(\boldsymbol{x})$ with : $Z(\boldsymbol{x}) \sim \mathcal{PG}(\mathbf{f}'(\boldsymbol{x})\boldsymbol{\beta}, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ where $\mathbf{f}(\boldsymbol{x})$ is a vector of p known functions and $\boldsymbol{\beta}$ the unknown coefficients of the trend.

Gaussian process regression (GPR)

Gaussian process model principle

- GPR is also known as **kriging models** in geostatistics.
- We are interested in approximating an function $z(\boldsymbol{x}) \in \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^d$ from few of its observations at points in the DOE $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$. We note $\mathbf{z} = (z_1, \ldots, z_n)$ these observations.
- **Gaussian process model hypothesis** : representing the unknown fonction $z(\boldsymbol{x})$ by a realization of a Gaussian process (it is the prior)
- The approximation entirely based on the theory of conditioned Gaussian processes except that neither the mean function nor the covariance function is known.
- When the mean is known, it is called simple kriging, and when the mean is unknown, it is called universal kriging.
- A common choice for **the mean** is to take a **linear form**.

  $z(\boldsymbol{x})$ is a realization of $Z(\boldsymbol{x})$ with : $Z(\boldsymbol{x}) \sim \mathcal{PG}(\mathbf{f}'(\boldsymbol{x})\boldsymbol{\beta}, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$ where $\mathbf{f}(\boldsymbol{x})$ is a vector of p known functions and $\boldsymbol{\beta}$ the unknown coefficients of the trend.

Rasmussen, C.E. and Williams,C.K.I. (2006), Gaussian Processes for Machine Learning, The MIT Press.

Santner, T.J., Williams, B.J. and Notz, W.I. (2003), The design and Analysis of Computer Experiments, Springer Series in Statistics.

Gaussian process prediction - known parameters

- if we know $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ and $\boldsymbol{\beta}$, we obtain same result as before :

$Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{(1)}) = z_1, \ldots, Z(\boldsymbol{x}^{(n)}) = z_n)$ is a **Gaussian** variable, with mean and variance conditioned $\mu_C(\boldsymbol{x})$ and $\sigma_C^2(\boldsymbol{x})$ such that :

$$\mu_C(\boldsymbol{x}) = \mathbf{f}'(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{r}'(\boldsymbol{x})\boldsymbol{C}_{\mathbf{D}}^{-1}(\mathbf{z} - \mathbf{F}\boldsymbol{\beta})$$

$$\sigma_C^2(\boldsymbol{x}) = C(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}'(\boldsymbol{x})\boldsymbol{C}_{\mathbf{D}}^{-1}\boldsymbol{r}(\boldsymbol{x})$$

where $\mathbf{F} = \begin{pmatrix} \mathrm{f}_1(\boldsymbol{x}^{(1)}) & \cdots & \mathrm{f}_p(\boldsymbol{x}^{(1)}) \\ \vdots & & \vdots \\ \mathrm{f}_1(\boldsymbol{x}^{(n)}) & \cdots & \mathrm{f}_p(\boldsymbol{x}^{(n)}) \end{pmatrix}$ is the matrix of regressors.

Gaussian process prediction - known parameters

- if we know $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ and $\boldsymbol{\beta}$, we obtain same result as before :

  $Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{(1)}) = z_1, \ldots, Z(\boldsymbol{x}^{(n)}) = z_n)$ is a **Gaussian** variable, with mean and variance conditioned $\mu_C(\boldsymbol{x})$ and $\sigma_C^2(\boldsymbol{x})$ such that :

$$\mu_C(\boldsymbol{x}) = \mathbf{f}'(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{r}'(\boldsymbol{x})\boldsymbol{C}_{\mathbf{D}}^{-1}(\mathbf{z} - \mathbf{F}\boldsymbol{\beta})$$

$$\sigma_C^2(\boldsymbol{x}) = C(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}'(\boldsymbol{x})\boldsymbol{C}_{\mathbf{D}}^{-1}\boldsymbol{r}(\boldsymbol{x})$$

  where $\mathbf{F} = \begin{pmatrix} \mathrm{f}_1(\boldsymbol{x}^{(1)}) & \cdots & \mathrm{f}_p(\boldsymbol{x}^{(1)}) \\ \vdots & & \vdots \\ \mathrm{f}_1(\boldsymbol{x}^{(n)}) & \cdots & \mathrm{f}_p(\boldsymbol{x}^{(n)}) \end{pmatrix}$ is the matrix of regressors.

- Remarks :
  - The conditioned mean interpolates the observations.
    To release this, simply take $\boldsymbol{C}_{\mathbf{D}} + \tau^2\mathbf{I}_n$ instead of $\boldsymbol{C}_{\mathbf{D}}$ in $\mu_C(\boldsymbol{x})$ and $\sigma_C^2(\boldsymbol{x})$ (τ^2 is the noise variance)
  - The conditioned mean does not depend on the variance σ^2
  - The conditioned variance does not depend on the observations $\mathbf{z}$.
  - These equations refers to simple kriging.

Gaussian process prediction - unknown parameters - MLE estimation

- if only $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is known, we obtain the following likelihood for parameters $\boldsymbol{\beta}$:

$$f(\mathbf{z}|\boldsymbol{\beta}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \boldsymbol{C}_{\mathbf{D}}}} \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{F}\boldsymbol{\beta})' \boldsymbol{C}_{\mathbf{D}}^{-1} (\mathbf{z}-\mathbf{F}\boldsymbol{\beta})\right),$$

Gaussian process prediction - unknown parameters - MLE estimation

- if only $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is known, we obtain the following likelihood for parameters $\boldsymbol{\beta}$:

$$f(\mathbf{z}|\boldsymbol{\beta}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \boldsymbol{C}_{\mathbf{D}}}} \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{F}\boldsymbol{\beta})' \boldsymbol{C}_{\mathbf{D}}^{-1} (\mathbf{z}-\mathbf{F}\boldsymbol{\beta})\right),$$

- The maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ is given by :

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}'\boldsymbol{C}_{\mathbf{D}}^{-1}\mathbf{F})^{-1}\mathbf{F}'\boldsymbol{C}_{\mathbf{D}}^{-1}\mathbf{z}.$$

  It corresponds to its generalized least squares estimate.

Gaussian process prediction - unknown parameters - MLE estimation

- if only $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is known, we obtain the following likelihood for parameters $\boldsymbol{\beta}$:

$$f(\mathbf{z}|\boldsymbol{\beta}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \boldsymbol{C}_{\mathbf{D}}}} \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{F}\boldsymbol{\beta})' \boldsymbol{C}_{\mathbf{D}}^{-1} (\mathbf{z}-\mathbf{F}\boldsymbol{\beta})\right),$$

- The maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ is given by :

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}'\boldsymbol{C}_{\mathbf{D}}^{-1}\mathbf{F})^{-1}\mathbf{F}'\boldsymbol{C}_{\mathbf{D}}^{-1}\mathbf{z}.$$

  It corresponds to its generalized least squares estimate.

- $Z^{\text{cond}}(\boldsymbol{x}) = (Z(\boldsymbol{x}) \mid Z(\boldsymbol{x}^{(1)}) = z_1, \ldots, Z(\boldsymbol{x}^{(n)}) = z_n)$ is then a **Gaussian** variable, with mean and variance conditioned $\mu_C(\boldsymbol{x})$ and $\sigma_C^2(\boldsymbol{x})$ such that :

$$\mu_C(\boldsymbol{x}) = \mathbf{f}'(\boldsymbol{x})\hat{\boldsymbol{\beta}} + \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{C}_{\mathbf{D}}^{-1}(\mathbf{z} - \mathbf{F}\hat{\boldsymbol{\beta}})$$

$$\sigma_C^2(\boldsymbol{x}) = C(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{C}_{\mathbf{D}}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{u}'(\boldsymbol{x})\left(\mathbf{F}'\boldsymbol{C}_{\mathbf{D}}^{-1}\mathbf{F}\right)^{-1}\boldsymbol{u}(\boldsymbol{x})$$

  with $\boldsymbol{u}(\boldsymbol{x}) = \mathbf{F}'\boldsymbol{C}_{\mathbf{D}}^{-1}\boldsymbol{r}(\boldsymbol{x}) - \mathbf{f}(\boldsymbol{x})$

  The variance incorporates an additional part due to the estimation of $\boldsymbol{\beta}$.

  These equations refer to universal kriging.

Gaussian process prediction - unknown parameters - MLE

- If the covariance function is also unknown, a common choice is a **parametric form** such that $C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma^2 R(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \boldsymbol{\theta})$.
- To estimate σ^2, we substitute the value of $\hat{\beta}$ in the likelihood and maximize it, we obtain :

$$\hat{\sigma}^2 = \frac{(\mathbf{z} - \mathbf{F}\hat{\beta})' \boldsymbol{R}_{\mathbf{D}}^{-1} (\mathbf{z} - \mathbf{F}\hat{\beta}) \sigma^2}{n}.$$

  σ^2 can also be estimated with a restricted maximum likelihood method.

Gaussian process prediction - unknown parameters - MLE

- If the covariance function is also unknown, a common choice is a **parametric form** such that $C(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma^2 R(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \boldsymbol{\theta})$.

- To estimate σ^2, we substitute the value of $\hat{\beta}$ in the likelihood and maximize it, we obtain :

$$\hat{\sigma}^2 = \frac{(\mathbf{z} - \mathbf{F}\hat{\boldsymbol{\beta}})' \boldsymbol{R}_{\mathbf{D}}^{-1} (\mathbf{z} - \mathbf{F}\hat{\boldsymbol{\beta}}) \sigma^2}{n}.$$

  σ^2 can also be estimated with a restricted maximum likelihood method.

- The estimation of $\boldsymbol{\theta}$, is conducted by substituting β and σ^2 by their MLE in the likelihood :

$$f(\mathbf{z}|\boldsymbol{\theta}) = (2\pi\hat{\sigma}^2)^{-n/2} (\det \boldsymbol{R}_{\mathbf{D}})^{1/2} \exp\left(-\frac{n}{2}\right)$$

  $\boldsymbol{\theta}$ can be estimates by minimizing the opposite of this log-likelihood (called the concentrated log-likelihood)

Gaussian process prediction - unknown parameters - bayesian estimate

- For the mean, it is of course possible to prior distribution

  For Gaussian prior on β, $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is a Gaussian process. For $p(\beta) \propto 1$, it corresponds to the MLE.

Gaussian process regression (GPR)

Gaussian process prediction - unknown parameters - bayesian estimate

- For the mean, it is of course possible to prior distribution

  For Gaussian prior on β, $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is a Gaussian process. For $p(\beta) \propto 1$, it corresponds to the MLE.

- For the σ^2 parameter of the covariance function, if we give inverse gamma prior, the distributions $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is tractable after integrating the posterior distribution of the variance parameter σ^2 : $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is a t-process. In practice, for reasonable n, this is indistinguishable from a GP.

Gaussian process regression (GPR)

Gaussian process prediction - unknown parameters - bayesian estimate

- For the mean, it is of course possible to prior distribution

  For Gaussian prior on β, $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is a Gaussian process. For $p(\beta) \propto 1$, it corresponds to the MLE.

- For the σ^2 parameter of the covariance function, if we give inverse gamma prior, the distributions $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is tractable after integrating the posterior distribution of the variance parameter σ^2 : $(Z(\boldsymbol{x}) \mid \mathbf{D})$ is a t-process. In practice, for reasonable n, this is indistinguishable from a GP.

- For θ, in general case, we have to use sampling techniques (Markov Chain Monte Carlo for example) to estimate conditioned distribution which is not necessary Gaussian

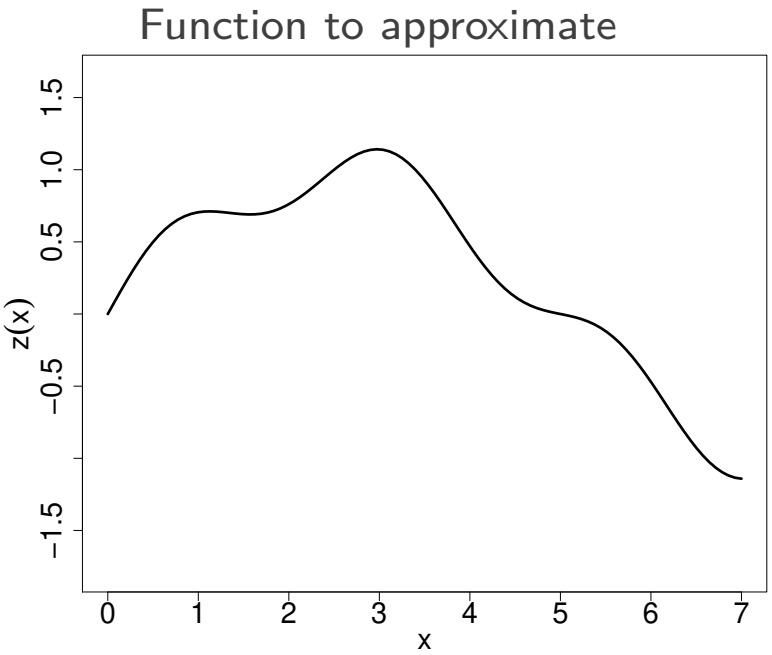
  This is called a full Bayesian approach

- For more information :

J. O. Berger, V. De Oliveira, and B. Sansó ,(2001), Objective Bayesian analysis of spatially correlated data, Journal of the American Statistical Association.

Muré, J. (2018), Objective Bayesian analysis of Kriging models with anisotropic correlation kernel, PhD

Illustration of Gaussian process prediction - the noisy-free case



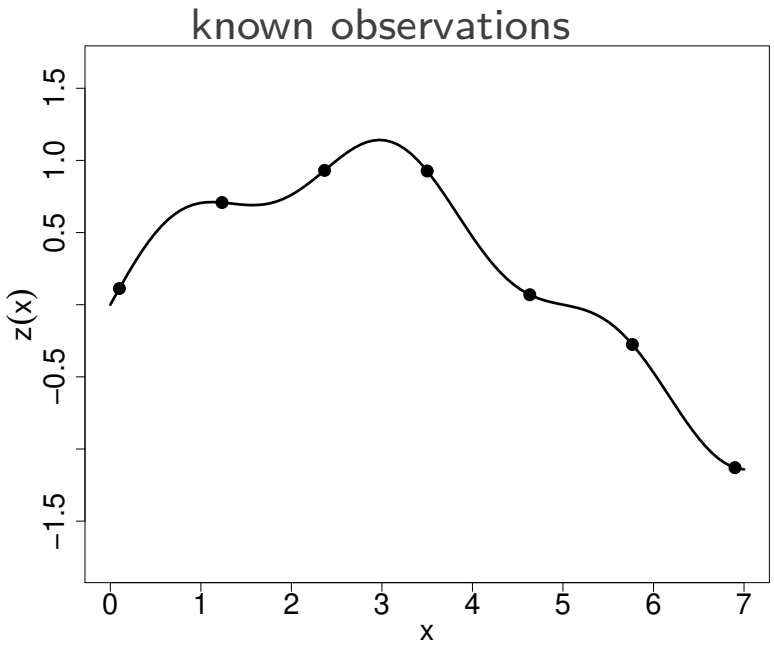
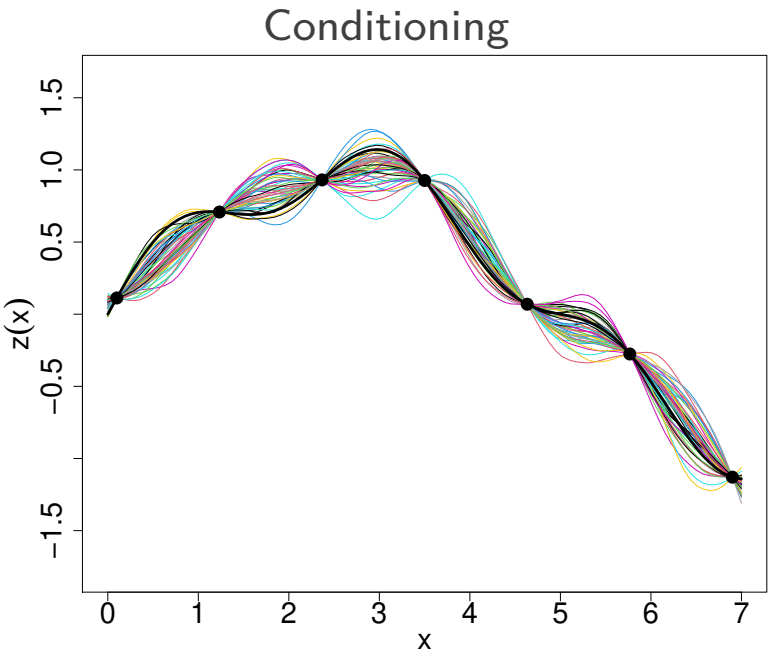


Illustration of Gaussian process prediction - the noisy-free case



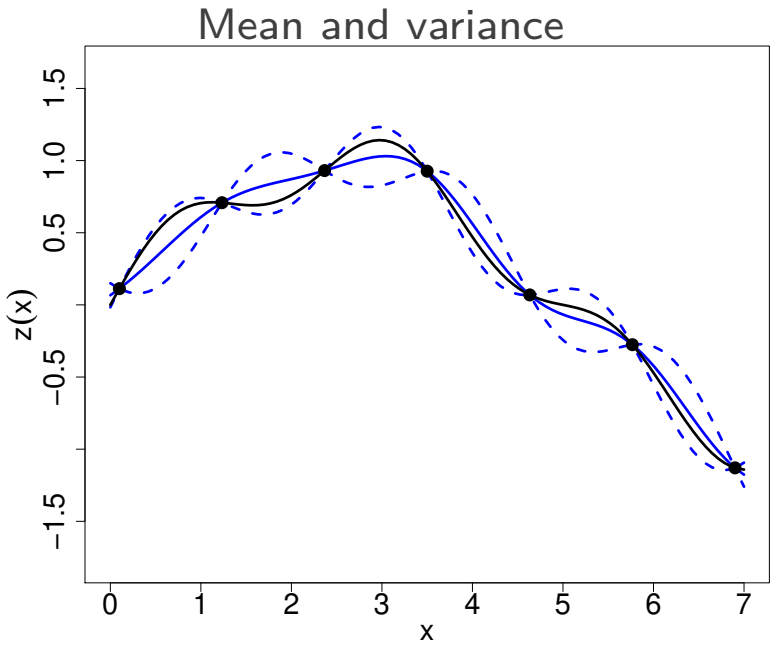
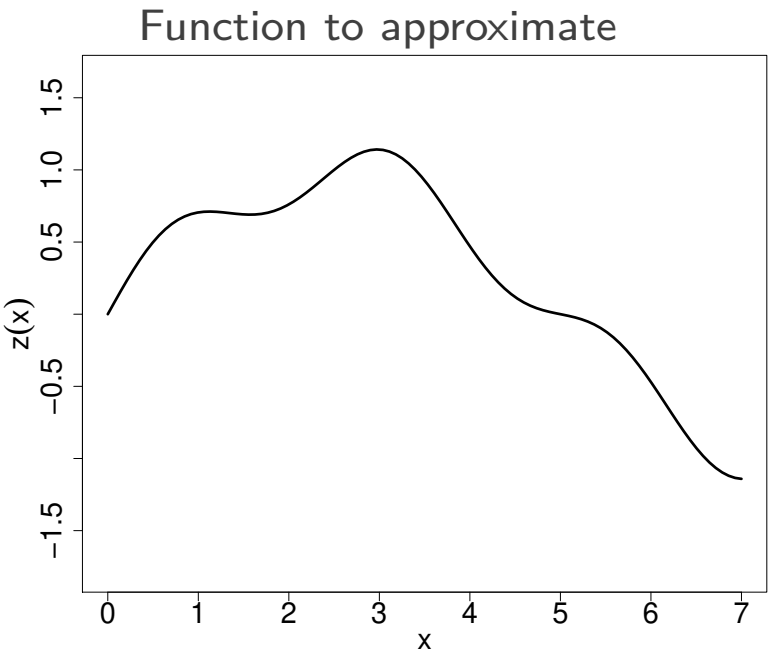


Illustration of Gaussian process prediction - the noisy case



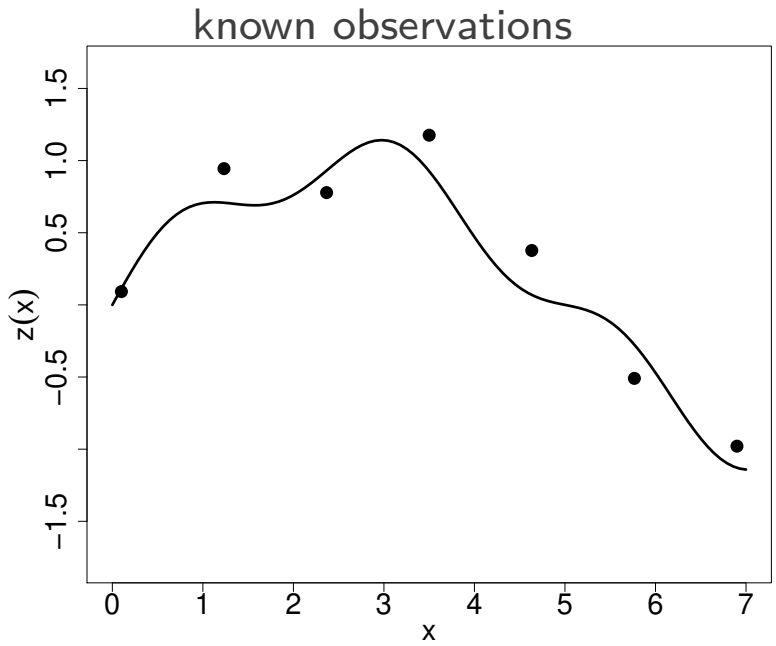
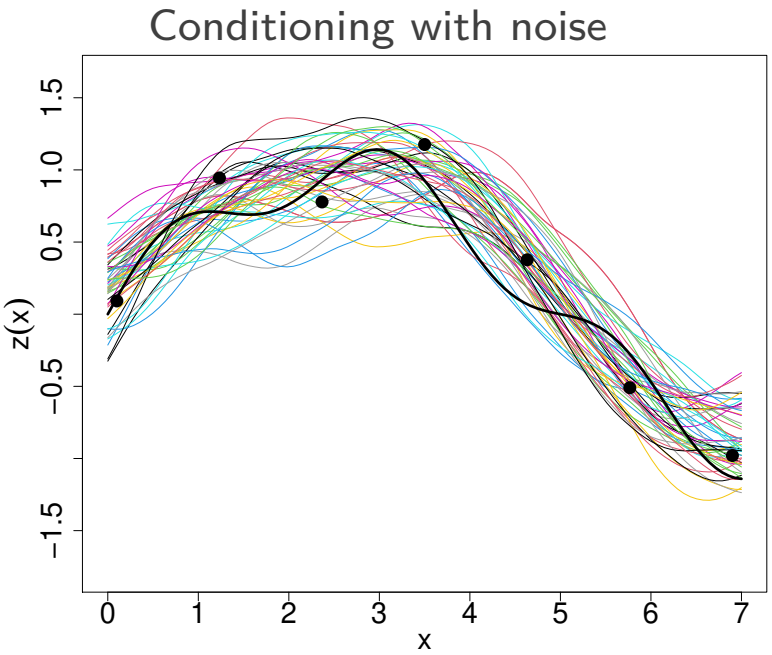
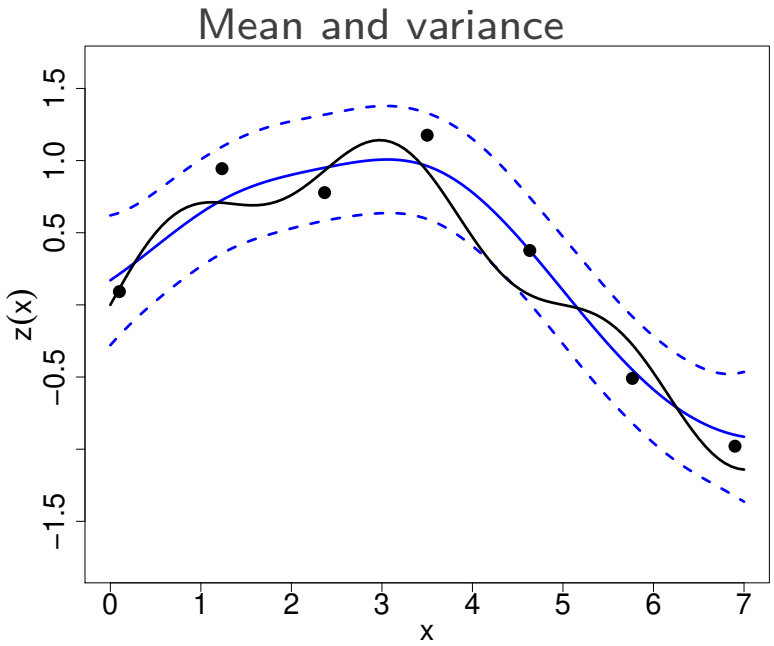


Illustration of Gaussian process prediction - the noisy case





One word on the validation (Leave-One-Out case)

- Let $\mu_{C(-i)}(\boldsymbol{x})$ and $\sigma^2_{C(-i)}(\boldsymbol{x})$ the conditioned mean and variance functions obtained without the observation $z_i = z(\boldsymbol{x}^{(i)})$.

  It is possible to have these quantities directly without having to build the n models

  Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. Mathematical Geology.

One word on the validation (Leave-One-Out case)

- Let $\mu_{C(-i)}(\boldsymbol{x})$ and $\sigma^2_{C(-i)}(\boldsymbol{x})$ the conditioned mean and variance functions obtained without the observation $z_i = z(\boldsymbol{x}^{(i)})$.

  It is possible to have these quantities directly without having to build the n models

  Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. Mathematical Geology.

- Evaluation of the predictive qualities of the conditioned mean :

$$Q_2 = 1 - \frac{\mathrm{var}(z(\boldsymbol{x}) - \mu_C(\boldsymbol{x}))}{\mathrm{var}(z(\boldsymbol{x}))} \simeq 1 - \frac{\sum_{i=1}^{n} \left(z(\boldsymbol{x}^{(i)}) - \mu_{C(-i)}(\boldsymbol{x}^{(i)})\right)^2}{\sum_{i=1}^{n} (z(\boldsymbol{x}^{(i)}) - \bar{\mathbf{z}})^2}, \; \bar{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} z_i$$

One word on the validation (Leave-One-Out case)

- Let $\mu_{C(-i)}(\boldsymbol{x})$ and $\sigma^2_{C(-i)}(\boldsymbol{x})$ the conditioned mean and variance functions obtained without the observation $z_i = z(\boldsymbol{x}^{(i)})$.

  It is possible to have these quantities directly without having to build the n models

  Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. Mathematical Geology.

- Evaluation of the predictive qualities of the conditioned mean :

$$Q_2 = 1 - \frac{\mathrm{var}(z(\boldsymbol{x}) - \mu_C(\boldsymbol{x}))}{\mathrm{var}(z(\boldsymbol{x}))} \simeq 1 - \frac{\sum_{i=1}^{n} \left(z(\boldsymbol{x}^{(i)}) - \mu_{C(-i)}(\boldsymbol{x}^{(i)})\right)^2}{\sum_{i=1}^{n} (z(\boldsymbol{x}^{(i)}) - \bar{\mathbf{z}})^2}, \ \bar{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} z_i$$

- Evaluation of the globale predictive qualities :

$$\varepsilon^2 = \frac{\mathrm{var}(z(\boldsymbol{x}) - \mu_C(\boldsymbol{x}))}{\sigma^2_C(x)} \simeq \frac{1}{n}\sum_{i=1}^{n} \frac{\left(z(\boldsymbol{x}^{(i)}) - \mu_{C(-i)}(\boldsymbol{x}^{(i)})\right)^2}{\sigma^2_{C(-i)}(\boldsymbol{x})}$$

  The target value is 1.

One word on the validation (Leave-One-Out case)

- Let $\mu_{C(-i)}(\boldsymbol{x})$ and $\sigma^2_{C(-i)}(\boldsymbol{x})$ the conditioned mean and variance functions obtained without the observation $z_i = z(\boldsymbol{x}^{(i)})$.

  It is possible to have these quantities directly without having to build the n models

  Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. Mathematical Geology.

- Evaluation of the predictive qualities of the conditioned mean :

$$Q_2 = 1 - \frac{\mathrm{var}(z(\boldsymbol{x}) - \mu_C(\boldsymbol{x}))}{\mathrm{var}(z(\boldsymbol{x}))} \simeq 1 - \frac{\sum_{i=1}^{n} \left(z(\boldsymbol{x}^{(i)}) - \mu_{C(-i)}(\boldsymbol{x}^{(i)})\right)^2}{\sum_{i=1}^{n} (z(\boldsymbol{x}^{(i)}) - \bar{\mathbf{z}})^2}, \ \bar{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} z_i$$

- Evaluation of the globale predictive qualities :

$$\varepsilon^2 = \frac{\mathrm{var}(z(\boldsymbol{x}) - \mu_C(\boldsymbol{x}))}{\sigma^2_C(x)} \simeq \frac{1}{n}\sum_{i=1}^{n} \frac{\left(z(\boldsymbol{x}^{(i)}) - \mu_{C(-i)}(\boldsymbol{x}^{(i)})\right)^2}{\sigma^2_{C(-i)}(\boldsymbol{x})}$$

  The target value is 1.

- Just for information : it is possible to use LOO results for estimating the parameters σ^2 ant $\boldsymbol{\theta}$.Bachoc, F. 2013, Parametric estimation of covariance function in Gaussian-process based Kriging models. Application to uncertainty quantification for computer experiments, PhD.

Outline

The bivariate case - objective

- We want now to approximate the last component $z_2(\boldsymbol{x})$ of $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), z_2(\boldsymbol{x})) \in \mathbb{R}^2$ with $x \in \mathbb{R}^d$.

  We assume that $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$ are observed without measurement error

The bivariate case - objective

- We want now to approximate the last component $z_2(\boldsymbol{x})$ of $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), z_2(\boldsymbol{x})) \in \mathbb{R}^2$ with $x \in \mathbb{R}^d$.

  We assume that $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$ are observed without measurement error

- This extension is called **co-kriging** and was first developed in geostatistics

  Chilès, J. and Delfiner, P. (1999). Geostatistics : modeling spatial uncertainty. Wiley series in probability and statistics.

  Wackernagel, H. (2003). Multivariate Geostatistics. Springer-Verlag.

The bivariate case - objective

- We want now to approximate the last component $z_2(\boldsymbol{x})$ of $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), z_2(\boldsymbol{x})) \in \mathbb{R}^2$ with $x \in \mathbb{R}^d$.

  We assume that $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$ are observed without measurement error

- This extension is called **co-kriging** and was first developed in geostatistics

  Chilès, J. and Delfiner, P. (1999). Geostatistics : modeling spatial uncertainty. Wiley series in probability and statistics.

  Wackernagel, H. (2003). Multivariate Geostatistics. Springer-Verlag.

- For the approximation, we have $\mathbf{z}_1 = (z_1(\boldsymbol{x}_1^{(1)}), \ldots, z_1(\boldsymbol{x}_1^{(n_1)}))'$ the observations of $z_1(\boldsymbol{x})$ at points in $\mathbf{D}_1 = \{\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_1^{(n_1)}\}$ and $\mathbf{z}_2 = (z_2(\boldsymbol{x}_1^{(1)}), \ldots, z_2(\boldsymbol{x}_1^{(n_2)}))'$ observations of $z_2(\boldsymbol{x})$ at points in $\mathbf{D}_2 = \{\boldsymbol{x}_2^{(1)}, \ldots, \boldsymbol{x}_2^{(n_2)}\}$.

  So, the column vector of observations is written $\mathbf{z}^{(2)} = ((\mathbf{z}_1)', (\mathbf{z}_2)')'$.

The bivariate case - objective

- We want now to approximate the last component $z_2(\boldsymbol{x})$ of $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), z_2(\boldsymbol{x})) \in \mathbb{R}^2$ with $x \in \mathbb{R}^d$.

  We assume that $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$ are observed without measurement error

- This extension is called **co-kriging** and was first developed in geostatistics

  Chilès, J. and Delfiner, P. (1999). Geostatistics : modeling spatial uncertainty. Wiley series in probability and statistics.

  Wackernagel, H. (2003). Multivariate Geostatistics. Springer-Verlag.

- For the approximation, we have $\mathbf{z}_1 = (z_1(\boldsymbol{x}_1^{(1)}), \ldots, z_1(\boldsymbol{x}_1^{(n_1)}))'$ the observations of $z_1(\boldsymbol{x})$ at points in $\mathbf{D}_1 = \{\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_1^{(n_1)}\}$ and $\mathbf{z}_2 = (z_2(\boldsymbol{x}_1^{(1)}), \ldots, z_2(\boldsymbol{x}_1^{(n_2)}))'$ observations of $z_2(\boldsymbol{x})$ at points in $\mathbf{D}_2 = \{\boldsymbol{x}_2^{(1)}, \ldots, \boldsymbol{x}_2^{(n_2)}\}$.

  So, the column vector of observations is written $\mathbf{z}^{(2)} = ((\mathbf{z}_1)', (\mathbf{z}_2)')'$.

- $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$ can be dependent

  If we want to approximate $z_2(\boldsymbol{x})$, it is important to take into account $z_1(\boldsymbol{x})$ too.

  We want to approximate $z_2(\boldsymbol{x})$ by taking into account the observations $\mathbf{z}^{(2)}$

The bivariate case - assumptions and notations

- Analogously to the GPR, we suppose that $\boldsymbol{z}(\boldsymbol{x})$ is a realization of a bivariate GP $\boldsymbol{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ with mean $\mathbf{m}(\boldsymbol{x})$ and covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$:

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} m_1(\boldsymbol{x}) \\ m_2(\boldsymbol{x}) \end{pmatrix} \text{ and } \mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} C_{11}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & C_{12}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ C_{21}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & C_{22}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix},$$

where $C_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \mathrm{cov}\left(Z_i(\boldsymbol{x}), Z_j(\tilde{\boldsymbol{x}})\right)$ and $m_i(\boldsymbol{x}) = \mathbb{E}\left[Z_i(\boldsymbol{x})\right]$, $i, j = 1, 2$:

The bivariate case - assumptions and notations

- Analogously to the GPR, we suppose that $\boldsymbol{z}(\boldsymbol{x})$ is a realization of a bivariate GP $\boldsymbol{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ with mean $\mathbf{m}(\boldsymbol{x})$ and covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$:

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} m_1(\boldsymbol{x}) \\ m_2(\boldsymbol{x}) \end{pmatrix} \text{ and } \mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} C_{11}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & C_{12}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ C_{21}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & C_{22}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix},$$

  where $C_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \mathrm{cov}\left(Z_i(\boldsymbol{x}), Z_j(\tilde{\boldsymbol{x}})\right)$ and $m_i(\boldsymbol{x}) = \mathbb{E}\left[Z_i(\boldsymbol{x})\right]$, $i, j = 1, 2$:

- We note $\mathbf{Z}^{(2)} = ((\mathbf{Z}_1)', (\mathbf{Z}_2)')'$ the values of $Z_1(\boldsymbol{x})$ and $Z_2(\boldsymbol{x})$ at points in $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively.

The bivariate case - assumptions and notations

- Analogously to the GPR, we suppose that $\boldsymbol{z}(\boldsymbol{x})$ is a realization of a bivariate GP $\boldsymbol{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ with mean $\mathbf{m}(\boldsymbol{x})$ and covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$:

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} m_1(\boldsymbol{x}) \\ m_2(\boldsymbol{x}) \end{pmatrix} \text{ and } \mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} C_{11}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & C_{12}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ C_{21}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & C_{22}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix},$$

  where $C_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \mathrm{cov}\left(Z_i(\boldsymbol{x}), Z_j(\tilde{\boldsymbol{x}})\right)$ and $m_i(\boldsymbol{x}) = \mathbb{E}\left[Z_i(\boldsymbol{x})\right]$, $i, j = 1, 2$:

- We note $\mathbf{Z}^{(2)} = ((\mathbf{Z}_1)', (\mathbf{Z}_2)')'$ the values of $Z_1(\boldsymbol{x})$ and $Z_2(\boldsymbol{x})$ at points in $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively.

- We suppose (as in a univariate case) that the i^{th} component of $\mathbf{m}(x)$ is of the form $m_i(\boldsymbol{x}) = \mathbf{f}_i'(\boldsymbol{x})\boldsymbol{\beta}_i$ with $\mathbf{f}_i'(\boldsymbol{x})$ a vector of functions of size p_i.

  We note $\mathbf{M}^{(s)} = (\mathbf{M}_1, \mathbf{M}_2)$ the values of $m_1(\boldsymbol{x})$ and $m_2(\boldsymbol{x})$ at points in $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively.

  We note $\mathbf{M}_i = \mathbf{f}_i'(\mathbf{D}_i)\boldsymbol{\beta}_i := \mathbf{F}_i\boldsymbol{\beta}_i$ with $\mathbf{F}_i$ a matrix of size $n_i \times p_i$, $i = 1, 2$.

The bivariate case - joint distribution

- As in the univariate case, we search the distribution of $Z_2(\boldsymbol{x})$ conditionnally to the observations of the two components $\mathbf{z}^{(2)}$. We note it $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$

The bivariate case - joint distribution

- As in the univariate case, we search the distribution of $Z_2(\boldsymbol{x})$ conditionnally to the observations of the two components $\mathbf{z}^{(2)}$. We note it $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$
- First, we consider the joint distribution of $Z_2(\boldsymbol{x})$ and $\mathbf{Z}^{(2)}$ given by :

$$\begin{pmatrix} Z_2(\boldsymbol{x}) \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}_2'(\boldsymbol{x})\boldsymbol{\beta}_2 \\ \mathbf{F}_1\boldsymbol{\beta}_1 \\ \mathbf{F}_2\boldsymbol{\beta}_2 \end{pmatrix}\begin{pmatrix} C_{22}(\boldsymbol{x},\boldsymbol{x}) & \boldsymbol{r}_{21}'(\boldsymbol{x}) & \boldsymbol{r}_{22}'(\boldsymbol{x}) \\ \boldsymbol{r}_{12}(\boldsymbol{x}) & \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{r}_{22}(\boldsymbol{x}) & \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}\right),$$

with, for $j = 1, 2$, $\boldsymbol{r}_{2j}(\boldsymbol{x}) = \left(C_{2j}(\boldsymbol{x}, \boldsymbol{x}_j^{(1)}), \ldots, C_{2j}(\boldsymbol{x}, \boldsymbol{x}_j^{(n_j)})\right)$ and $\boldsymbol{r}_{j2}(\boldsymbol{x}) = \left(C_{j2}(\boldsymbol{x}_j^{(1)}, \boldsymbol{x}), \ldots, C_{j2}(\boldsymbol{x}_j^{(n_j)}, \boldsymbol{x})\right)$ are column vectors of size n_j, $\boldsymbol{C}_{ij} = \left(C_{ij}(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(l)})\right)_{\substack{k=1,\ldots,n_i \\ l=1,\ldots,n_j}}$ are matrix of size $n_i \times n_j$.

The bivariate case - joint distribution

- As in the univariate case, we search the distribution of $Z_2(\boldsymbol{x})$ conditionnally to the observations of the two components $\mathbf{z}^{(2)}$. We note it $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$
- First, we consider the joint distribution of $Z_2(\boldsymbol{x})$ and $\mathbf{Z}^{(2)}$ given by :

$$\begin{pmatrix} Z_2(\boldsymbol{x}) \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}_2'(\boldsymbol{x})\boldsymbol{\beta}_2 \\ \mathbf{F}_1\boldsymbol{\beta}_1 \\ \mathbf{F}_2\boldsymbol{\beta}_2 \end{pmatrix}\begin{pmatrix} C_{22}(\boldsymbol{x},\boldsymbol{x}) & \boldsymbol{r}_{21}'(\boldsymbol{x}) & \boldsymbol{r}_{22}'(\boldsymbol{x}) \\ \boldsymbol{r}_{12}(\boldsymbol{x}) & \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{r}_{22}(\boldsymbol{x}) & \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}\right),$$

  with, for $j = 1, 2$, $\boldsymbol{r}_{2j}(\boldsymbol{x}) = \left(C_{2j}(\boldsymbol{x}, \boldsymbol{x}_j^{(1)}), \ldots, C_{2j}(\boldsymbol{x}, \boldsymbol{x}_j^{(n_j)})\right)$ and $\boldsymbol{r}_{j2}(\boldsymbol{x}) = \left(C_{j2}(\boldsymbol{x}_j^{(1)}, \boldsymbol{x}), \ldots, C_{j2}(\boldsymbol{x}_j^{(n_j)}, \boldsymbol{x})\right)$ are column vectors of size n_j, $\boldsymbol{C}_{ij} = \left(C_{ij}(\boldsymbol{x}_i^{(k)}, \boldsymbol{x}_j^{(l)})\right)_{\substack{k=1,\ldots,n_i \\ l=1,\ldots,n_j}}$ are matrix of size $n_i \times n_j$.

- Although in general $C_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \neq C_{ji}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, we have the equality $\boldsymbol{r}_{2j}(\boldsymbol{x}) = \boldsymbol{r}_{j2}(\boldsymbol{x})$ and $\boldsymbol{C_{ij}} = \boldsymbol{C_{ji}'}$.

  Indeed, the equality $\operatorname{cov}(Z_i(\boldsymbol{x}), Z_j(\tilde{\boldsymbol{x}})) = \operatorname{cov}(Z_j(\tilde{x}), Z_i(\boldsymbol{x}))$ implies that $C_{sj}(x, \tilde{x}) = C_{js}(\tilde{x}, x)$ and thus $\boldsymbol{r}_{ij}(x) = \boldsymbol{r}_{ji}(x)$ and $\boldsymbol{C}_{ij} = \boldsymbol{C}_{ji}'$.

The bivariate case - predictive distribution

- If the mean $\mathbf{m}(\boldsymbol{x})$ and covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ are known, the conditional distribution $[Z_2(\boldsymbol{x}) | \mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$ is gaussian with conditioned mean $\mu_{Z_2}(\boldsymbol{x})$ and variance $\sigma^2_{Z_2}(\boldsymbol{x})$ functions given by :

$$\begin{cases} \mu_{Z_2}(\boldsymbol{x}) = \mathbf{f}_2'(\boldsymbol{x})\boldsymbol{\beta}_2 + \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\left(\mathbf{z}^{(2)} - \mathbf{M}^{(2)}\right), \\ \sigma^2_{Z_2}(\boldsymbol{x}) = C_{22}(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\boldsymbol{r}_2(\boldsymbol{x}), \end{cases}$$

where $\boldsymbol{r}_2'(\boldsymbol{x}) = (\boldsymbol{r}_{21}'(\boldsymbol{x}), \boldsymbol{r}_{22}'(\boldsymbol{x}))$ and $\mathbf{V}_2 = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$.

The bivariate case - predictive distribution

- If the mean $\mathbf{m}(\boldsymbol{x})$ and covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ are known, the conditional distribution $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$ is gaussian with conditioned mean $\mu_{Z_2}(\boldsymbol{x})$ and variance $\sigma^2_{Z_2}(\boldsymbol{x})$ functions given by :

$$\begin{cases} \mu_{Z_2}(\boldsymbol{x}) = \mathbf{f}_2'(\boldsymbol{x})\boldsymbol{\beta}_2 + \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\left(\mathbf{z}^{(2)} - \mathbf{M}^{(2)}\right), \\ \sigma^2_{Z_2}(\boldsymbol{x}) = C_{22}(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\boldsymbol{r}_2(\boldsymbol{x}), \end{cases}$$

  where $\boldsymbol{r}_2'(\boldsymbol{x}) = (\boldsymbol{r}_{21}'(\boldsymbol{x}), \boldsymbol{r}_{22}'(\boldsymbol{x}))$ and $\mathbf{V}_2 = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$.

- The predictive mean $\mu_{Z_2}(\boldsymbol{x})$ is the surrogate model for the component $z_2(\boldsymbol{x})$ of $z(\boldsymbol{x})$ and the predictive variance $\sigma^2_{Z_2}(\boldsymbol{x})$ represents the model mean squared error.
- $\mu_{Z_2}(\boldsymbol{x})$ interpolates $z_2(\boldsymbol{x})$ at points of the experimental design $\mathbf{D}_2$ and $\sigma^2_{Z_2}(\boldsymbol{x})$ equals zero at these points. But, we can integrate a noise variance in the model as presented for the univariate case.

The bivariate case - predictive distribution

- If the mean $\mathbf{m}(\boldsymbol{x})$ and covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ are known, the conditional distribution $[Z_2(\boldsymbol{x}) | \mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$ is gaussian with conditioned mean $\mu_{Z_2}(\boldsymbol{x})$ and variance $\sigma^2_{Z_2}(\boldsymbol{x})$ functions given by :

$$\begin{cases} \mu_{Z_2}(\boldsymbol{x}) = \mathbf{f}_2'(\boldsymbol{x})\boldsymbol{\beta}_2 + \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\left(\mathbf{z}^{(2)} - \mathbf{M}^{(2)}\right), \\ \sigma^2_{Z_2}(\boldsymbol{x}) = C_{22}(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\boldsymbol{r}_2(\boldsymbol{x}), \end{cases}$$

  where $\boldsymbol{r}_2'(\boldsymbol{x}) = (\boldsymbol{r}_{21}'(\boldsymbol{x}), \boldsymbol{r}_{22}'(\boldsymbol{x}))$ and $\mathbf{V}_2 = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$.

- The predictive mean $\mu_{Z_2}(\boldsymbol{x})$ is the surrogate model for the component $z_2(\boldsymbol{x})$ of $z(\boldsymbol{x})$ and the predictive variance $\sigma^2_{Z_2}(\boldsymbol{x})$ represents the model mean squared error.
- $\mu_{Z_2}(\boldsymbol{x})$ interpolates $z_2(\boldsymbol{x})$ at points of the experimental design $\mathbf{D}_2$ and $\sigma^2_{Z_2}(\boldsymbol{x})$ equals zero at these points. But, we can integrate a noise variance in the model as presented for the univariate case.
- We note that the matrix $\mathbf{V}_2$ must be positive definite. We will present, after an illustration, covariance structures which ensure this property.

The bivariate case - illustration

- Let the bivariate Gaussian process $(Z_1(x), Z_2(x))$, $x \in \mathbb{R}$ defined by :

$$\begin{cases} Z_1(x) = a_1\delta_1(x) & + & a_2\delta_2(x) \\ Z_2(x) = b_1\delta_1(x) & + & b_2\delta_2(x) \end{cases},$$

where $\delta_1(x)$ and $\delta_2(x)$ are two independent Gaussian processes with means zero and covariances $k_1(x, \tilde{x})$ and $k_2(x, \tilde{x})$ such that :

  - $k_1(x, \tilde{x})$ is a $5/2$-Matérn kernel with $\sigma^2 = 1$ and $\theta = 0.2$,
  - $k_2(x, \tilde{x})$ is a $3/2$-Matérn kernel with $\sigma^2 = 1$ and $\theta = 0.3$.

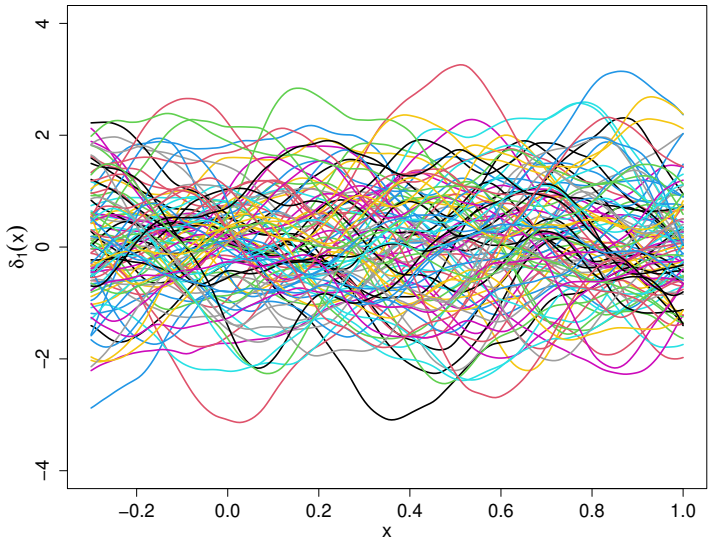
The bivariate case - illustration

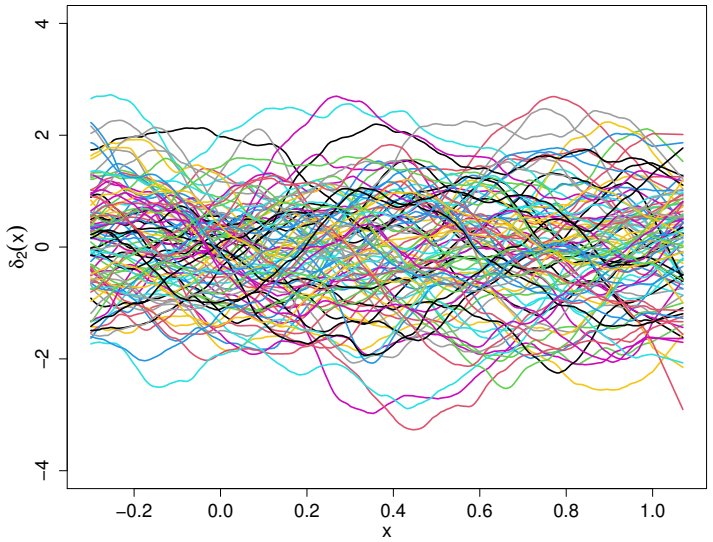
- Let the bivariate Gaussian process $(Z_1(x), Z_2(x))$, $x \in \mathbb{R}$ defined by :

$$\begin{cases} Z_1(x) = a_1\delta_1(x) & + & a_2\delta_2(x) \\ Z_2(x) = b_1\delta_1(x) & + & b_2\delta_2(x) \end{cases},$$

where $\delta_1(x)$ and $\delta_2(x)$ are two independent Gaussian processes with means zero and covariances $k_1(x, \tilde{x})$ and $k_2(x, \tilde{x})$ such that :

  - $k_1(x, \tilde{x})$ is a $5/2$-Matérn kernel with $\sigma^2 = 1$ and $\theta = 0.2$,
  - $k_2(x, \tilde{x})$ is a $3/2$-Matérn kernel with $\sigma^2 = 1$ and $\theta = 0.3$.





The bivariate case - illustration

- $(Z_1(x), Z_2(x))$ is Gaussian (linear combination of $(\delta_1(x), \delta_2(x))$.
- $(Z_1(x), Z_2(x))$ has zero mean and covariance structure :

The bivariate case - illustration

- $(Z_1(x), Z_2(x))$ is Gaussian (linear combination of $(\delta_1(x), \delta_2(x))$.
- $(Z_1(x), Z_2(x))$ has zero mean and covariance structure :

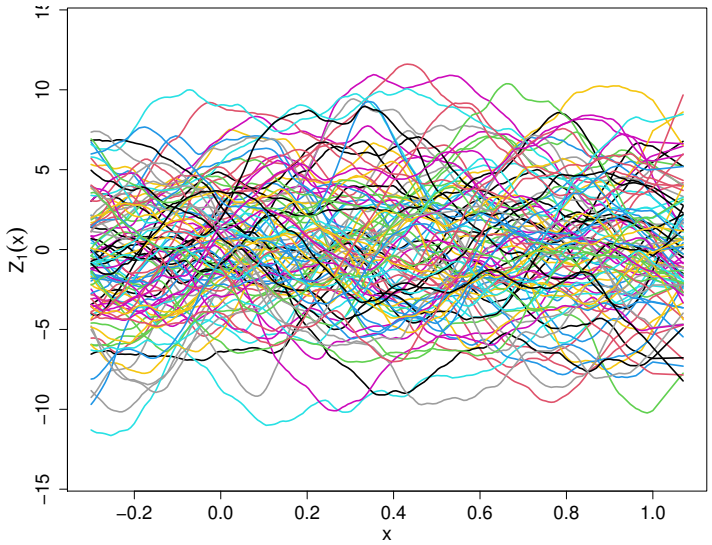
$$\mathbf{V}(x, \tilde{x}) = \begin{pmatrix} a_1^2 k_1(x, \tilde{x}) + a_2^2 k_2(x, \tilde{x}) & a_1 b_1 k_1(x, \tilde{x}) + a_2 b_2 k_2(x, \tilde{x}) \\ a_1 b_1 k_1(x, \tilde{x}) + a_2 b_2 k_2(x, \tilde{x}) & b_1^2 k_1(x, \tilde{x}) + b_2^2 k_2(x, \tilde{x}) \end{pmatrix}.$$

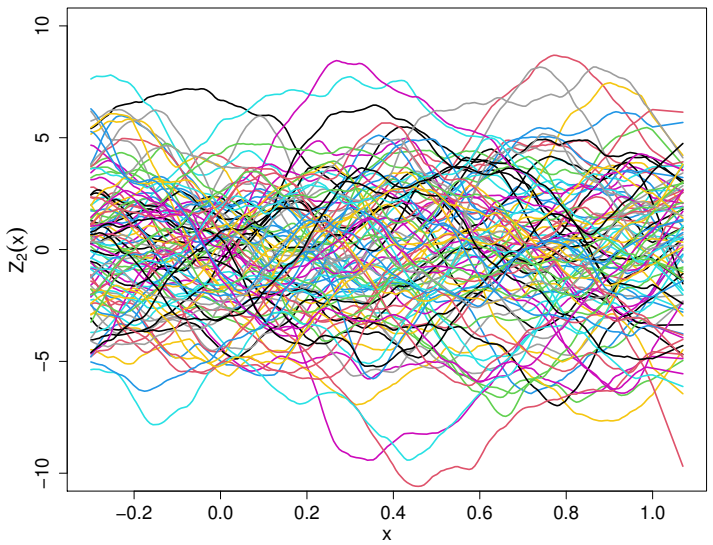
The bivariate case - illustration

- $(Z_1(x), Z_2(x))$ is Gaussian (linear combination of $(\delta_1(x), \delta_2(x))$.
- $(Z_1(x), Z_2(x))$ has zero mean and covariance structure :

$$\mathbf{V}(x, \tilde{x}) = \begin{pmatrix} a_1^2 k_1(x, \tilde{x}) + a_2^2 k_2(x, \tilde{x}) & a_1 b_1 k_1(x, \tilde{x}) + a_2 b_2 k_2(x, \tilde{x}) \\ a_1 b_1 k_1(x, \tilde{x}) + a_2 b_2 k_2(x, \tilde{x}) & b_1^2 k_1(x, \tilde{x}) + b_2^2 k_2(x, \tilde{x}) \end{pmatrix}.$$

- For $a_1 = 0.5$, $a_2 = 3$, $b_1 = 1$ and $b_2 = -4$:



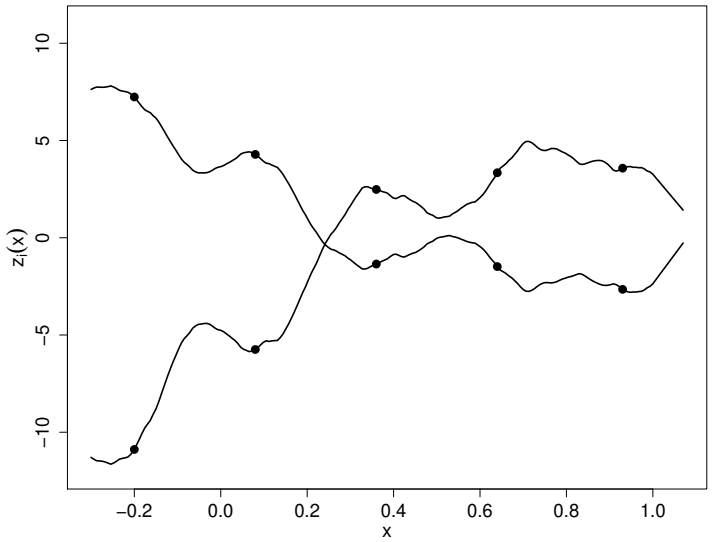


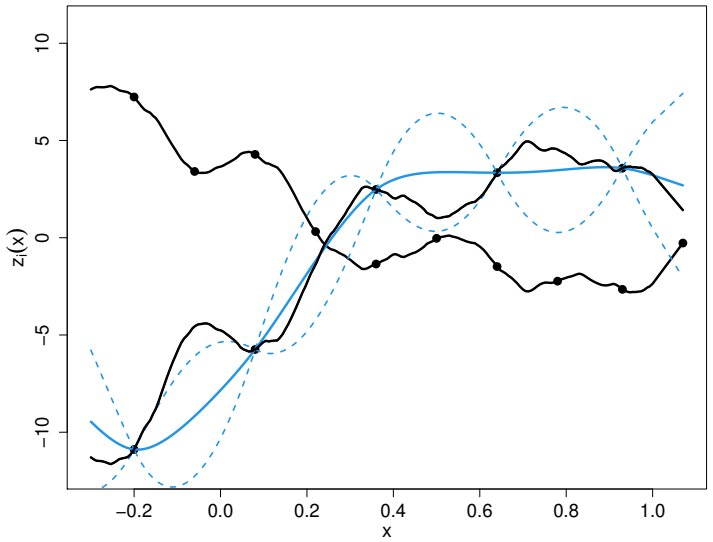
The bivariate case - illustration

- We consider a realization of $Z_1(x)$ and $Z_2(x)$ (noted $z_1(x)$ and $z_2(x)$)
- We want to reconstruct $z_2(x)$ from its values at points in $\mathbf{D}_2$ and the values of $z_1(x)$ at points in $\mathbf{D}_1$.
- If we consider the same points in $\mathbf{D}_1$ and $\mathbf{D}_2$, taking into account $z_1(x)$ does not bring any additional information than that provided by $z_2(x)$ information.

The bivariate case - illustration

- We consider a realization of $Z_1(x)$ and $Z_2(x)$ (noted $z_1(x)$ and $z_2(x)$)
- We want to reconstruct $z_2(x)$ from its values at points in $\mathbf{D}_2$ and the values of $z_1(x)$ at points in $\mathbf{D}_1$.
- If we consider the same points in $\mathbf{D}_1$ and $\mathbf{D}_2$, taking into account $z_1(x)$ does not bring any additional information than that provided by $z_2(x)$ information.

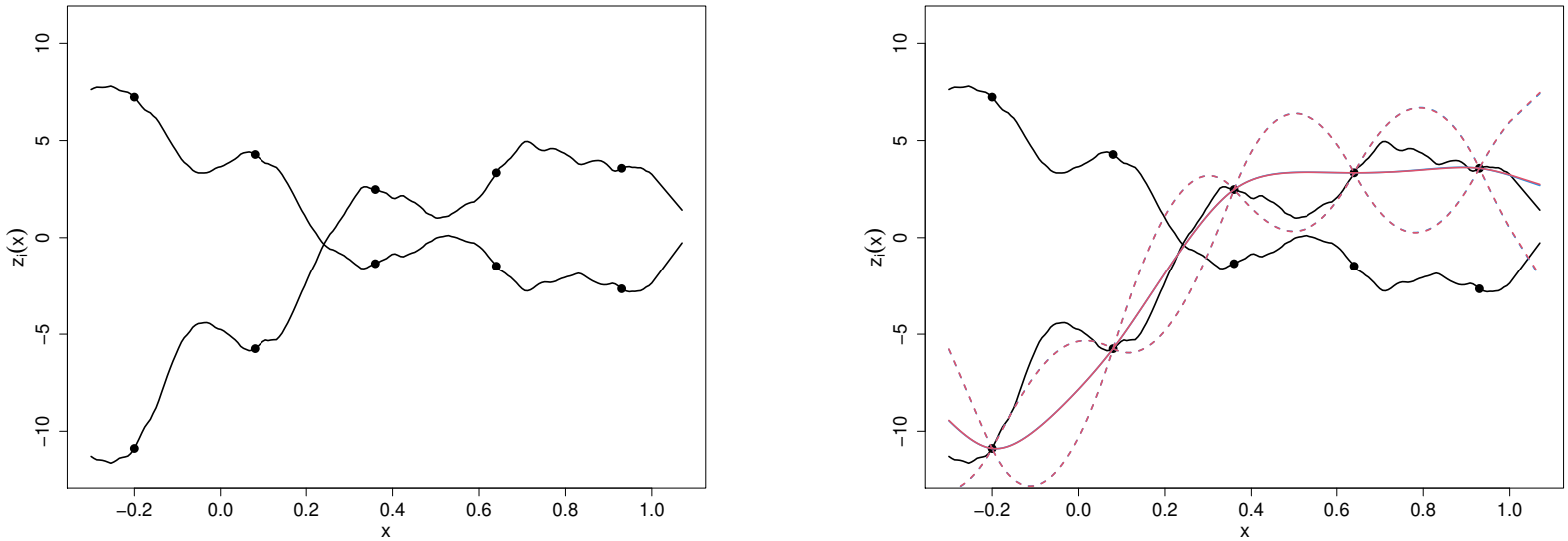




in blue : GPR on $z_2(x)$

The bivariate case - illustration

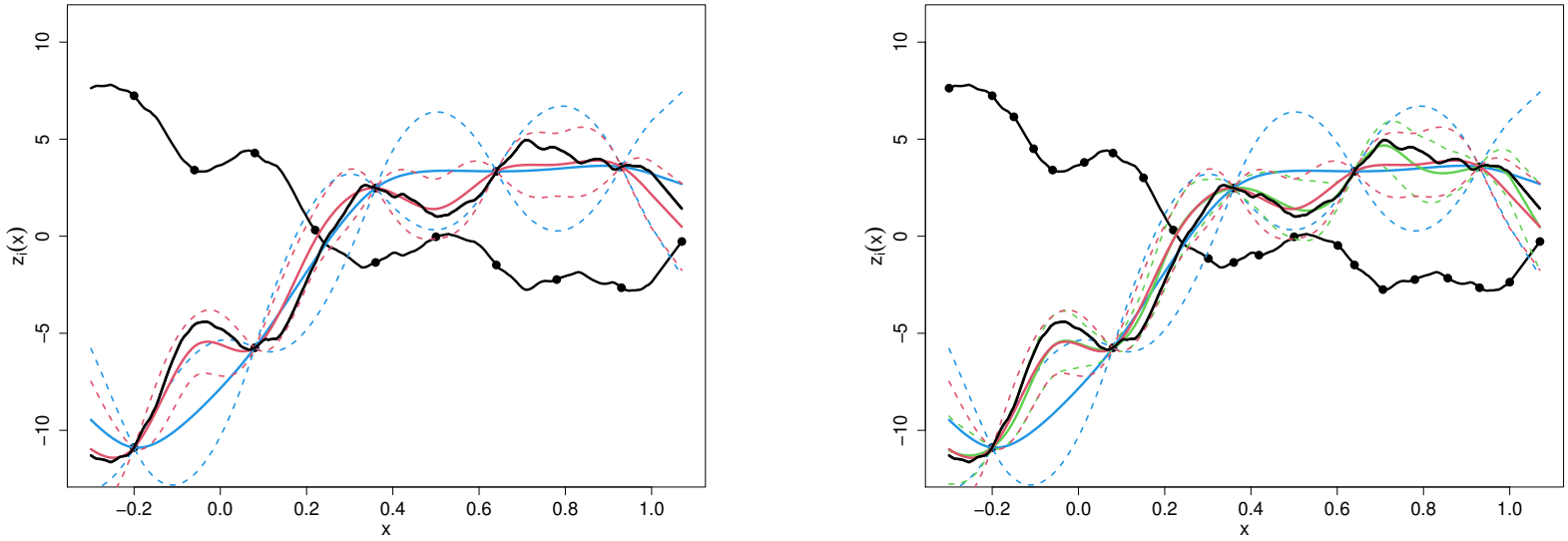
- We consider a realization of $Z_1(x)$ and $Z_2(x)$ (noted $z_1(x)$ and $z_2(x)$)
- We want to reconstruct $z_2(x)$ from its values at points in $\mathbf{D}_2$ and the values of $z_1(x)$ at points in $\mathbf{D}_1$.
- If we consider the same points in $\mathbf{D}_1$ and $\mathbf{D}_2$, taking into account $z_1(x)$ does not bring any additional information than that provided by $z_2(x)$ information.



in blue : univariate GPR, in red : bivariate GPR

The bivariate case - illustration

- We consider a realization of $Z_1(x)$ and $Z_2(x)$ (noted $z_1(x)$ and $z_2(x)$)
- We want to reconstruct $z_2(x)$ from its values at points in $\mathbf{D}_2$ and the values of $z_1(x)$ at points in $\mathbf{D}_1$.
- If we consider more points in $\mathbf{D}_1$ than $\mathbf{D}_2$, the information provided by $z_1(x)$ allows us to build a more predictive and accurate model for $z_2(x)$.



in blue : univariate GPR, in red and green : bivariate GPR

The bivariate case - Linear model of coregionalization (LMC)

- LMC, widely used in geostatistics, is an approach to construct an admissible matrix-valued covariance.
- A valid covariance structure $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must satisfy the condition of positive definiteness. For any $(\mathbf{D}_i)_{i=1,2}$ the following covariance matrix

$$\mathbf{V}_2 = \begin{pmatrix} C_{11}(\mathbf{D}_1, \mathbf{D}_1) & C_{12}(\mathbf{D}_1, \mathbf{D}_2) \\ C_{21}(\mathbf{D}_2, \mathbf{D}_1) & C_{22}(\mathbf{D}_2, \mathbf{D}_2) \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$$ has to be positive definite.

The bivariate case - Linear model of coregionalization (LMC)

- LMC, widely used in geostatistics, is an approach to construct an admissible matrix-valued covariance.
- A valid covariance structure $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must satisfy the condition of positive definiteness. For any $(\mathbf{D}_i)_{i=1,2}$ the following covariance matrix

$$\mathbf{V}_2 = \begin{pmatrix} C_{11}(\mathbf{D}_1, \mathbf{D}_1) & C_{12}(\mathbf{D}_1, \mathbf{D}_2) \\ C_{21}(\mathbf{D}_2, \mathbf{D}_1) & C_{22}(\mathbf{D}_2, \mathbf{D}_2) \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix} \text{ has to be positive definite.}$$

- In the LMC, the components of $\mathbf{Z}(x)$ are expressed as linear combinations of t independent Gaussian processes $\delta_j(x)$ with covariance kernel $C_j(\boldsymbol{x}, \tilde{\boldsymbol{x}})$:

$$Z_i(\boldsymbol{x}) = \sum_{j=1}^{t} \alpha_j^i \delta_j(x), i = 1, 2.$$

The bivariate case - Linear model of coregionalization (LMC)

- LMC, widely used in geostatistics, is an approach to construct an admissible matrix-valued covariance.
- A valid covariance structure $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ must satisfy the condition of positive definiteness. For any $(\mathbf{D}_i)_{i=1,2}$ the following covariance matrix

$$\mathbf{V}_2 = \begin{pmatrix} C_{11}(\mathbf{D}_1, \mathbf{D}_1) & C_{12}(\mathbf{D}_1, \mathbf{D}_2) \\ C_{21}(\mathbf{D}_2, \mathbf{D}_1) & C_{22}(\mathbf{D}_2, \mathbf{D}_2) \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{pmatrix}$$ has to be positive definite.

- In the LMC, the components of $\mathbf{Z}(x)$ are expressed as linear combinations of t independent Gaussian processes $\delta_j(x)$ with covariance kernel $C_j(\boldsymbol{x}, \tilde{\boldsymbol{x}})$:

$$Z_i(\boldsymbol{x}) = \sum_{j=1}^{t} \alpha_j^i \delta_j(x), i = 1, 2.$$

- $Z(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ is then a bivariate Gaussian process and we have :

$$\operatorname{cov}\left(Z_i(\boldsymbol{x}), Z_j(\tilde{\boldsymbol{x}})\right) = \sum_{k=1}^{t} \alpha_k^i \alpha_k^j C_k(\boldsymbol{x}, \tilde{\boldsymbol{x}})$$

The bivariate case - Linear model of coregionalization (LMC)

- The covariance structure of $\boldsymbol{Z}(\boldsymbol{x})$ is : $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sum_{k=1}^{t} \left[\alpha_k^i \alpha_k^j\right]_{i,j=1,2} C_k(\boldsymbol{x}, \tilde{\boldsymbol{x}})$
- The matrix $\left[\alpha_k^i \alpha_k^j\right]_{i,j=1,2}$ is nonnegative definite, for all $k = 1, \ldots, t$:

$$\begin{pmatrix} \alpha_k^1 \alpha_k^1 & \alpha_k^1 \alpha_k^2 \\ \alpha_k^2 \alpha_k^1 & \alpha_k^2 \alpha_k^2 \end{pmatrix} = \begin{pmatrix} \alpha_k^1 \\ \alpha_k^2 \end{pmatrix} \begin{pmatrix} \alpha_k^1 & \alpha_k^2 \end{pmatrix}.$$

The t matrices $\left[\alpha_k^i \alpha_k^j\right]_{i,j=1,2}$ are known as the coregionalization matrices.

The bivariate case - Linear model of coregionalization (LMC)

- The covariance structure of $\boldsymbol{Z}(\boldsymbol{x})$ is : $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sum_{k=1}^{t} \left[\alpha_k^i \alpha_k^j\right]_{i,j=1,2} C_k(\boldsymbol{x}, \tilde{\boldsymbol{x}})$
- The matrix $\left[\alpha_k^i \alpha_k^j\right]_{i,j=1,2}$ is nonnegative definite, for all $k = 1, \ldots, t$:

$$\begin{pmatrix} \alpha_k^1 \alpha_k^1 & \alpha_k^1 \alpha_k^2 \\ \alpha_k^2 \alpha_k^1 & \alpha_k^2 \alpha_k^2 \end{pmatrix} = \begin{pmatrix} \alpha_k^1 \\ \alpha_k^2 \end{pmatrix} \begin{pmatrix} \alpha_k^1 & \alpha_k^2 \end{pmatrix}.$$

  The t matrices $\left[\alpha_k^i \alpha_k^j\right]_{i,j=1,2}$ are known as the coregionalization matrices.
- The regularity of any Gaussian process $Z_i(\boldsymbol{x})$ is the one of the roughest latent process $\delta_j(\boldsymbol{x})$.
- The LMC considers that all outputs provide the same level of information.
- For more details :
  - Goulard, M. and Voltz, M. (1992). Linear coregionalization model :tools for estimation and choice of cross-variogram matrix. Mathematical Geology, 24(3) :269-286.
  - Wackernagel, H. (2003). Multivariate Geostatistics. Springer-Verlag, Berlin.
- LMC is not the only method for building admissible matrix-valued covariance
  - Bonilla, E., Ming, K., Chai, A. and Williams, C. Multi-task Gaussian process prediction (2007). Proceedings of the 20th International Conference on Neural Information Processing Systems, 153–160.
  - Higdon, D. (2002). Space and space-time modeling using process convolutions. In Quantitative methods for current environmental issues, 37-56, Springer.
  - ...

The multivariate case (just to get the formulas ...)

- The general form of GPR for vectorial function is easily generalized for s outputs, the notations are the same as before but a little be more arduous...
- We want to approximate $z_s(\boldsymbol{x})$ of $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), \ldots, z_s(\boldsymbol{x})) \in \mathbb{R}^s$ by taking into account $(z_t(\boldsymbol{x}))_{t=1,\ldots,s-1}$
- We consider that $\boldsymbol{z}(\boldsymbol{x})$ is a realization of a multivariate Gaussian process $\boldsymbol{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), \ldots, Z_s(\boldsymbol{x}))$ with mean $\mathbf{m}(\boldsymbol{x})$ and matrix-valued covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ such that :

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 \\ \vdots \\ \mathbf{f}_s'(\boldsymbol{x})\boldsymbol{\beta}_s \end{pmatrix} \quad \text{and} \quad \mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} C_{11}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \ldots & c_{1s}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ \vdots & \ddots & \vdots \\ C_{s1}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \ldots & C_{ss}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix}$$

- Notations : $\mathbf{D}_t = \{\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(n_t)}\}$ the DOE for the component t

  $\mathbf{z}_t = z_t(\mathbf{D}_t) = (z_t(\boldsymbol{x}_t^{(1)}), \ldots, z_t(\boldsymbol{x}_t^{(n_t)}))'$; $\mathbf{z}^{(s)} = (\mathbf{z}_1', \ldots, \mathbf{z}_s')'$

  $\mathbf{Z}_t = Z_t(\mathbf{D}_t) = (Z_t(\boldsymbol{x}_t^{(1)}), \ldots, Z_t(\boldsymbol{x}_t^{(n_t)}))'$; $\mathbf{Z}^{(s)} = (\mathbf{Z}_1', \ldots, \mathbf{Z}_s')'$

  $\mathbf{M}_t = \mathbf{f}_t'(\mathbf{D}_t)\boldsymbol{\beta}_t := \mathbf{F}_t\boldsymbol{\beta}_t$; $\mathbf{M}^{(s)} = (\mathbf{M}_1, \ldots, \mathbf{M}_s)$

The multivariate case (just to get the formulas ...)

- We have the following joint distribution :

$$\begin{pmatrix} Z_s(\boldsymbol{x}) \\ \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_s \end{pmatrix} \sim \mathbb{N}\begin{pmatrix} \mathbf{f}_s'(\boldsymbol{x})\boldsymbol{\beta}_s \\ \mathbf{F}_1\boldsymbol{\beta}_1 \\ \vdots \\ \mathbf{F}_s\boldsymbol{\beta}_s \end{pmatrix}\begin{pmatrix} C_{ss}(\boldsymbol{x},\boldsymbol{x}) & \boldsymbol{r}_{s1}'(\boldsymbol{x}) & \dots & \boldsymbol{r}_{ss}'(\boldsymbol{x}) \\ \boldsymbol{r}_{1s}(\boldsymbol{x}) & \boldsymbol{C}_{11} & \dots & \boldsymbol{C}_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{r}_{ss}(\boldsymbol{x}) & \boldsymbol{C}_{s1} & \dots & \boldsymbol{C}_{ss} \end{pmatrix},$$

with $\boldsymbol{r}_{sj}(\boldsymbol{x}) = [C_{sj}(\boldsymbol{x}, x_j^{(k)})]_{k=1,\dots,n_j}$, $\boldsymbol{r}_{js}(x) = [C_{js}(x_j^{(k)}, \boldsymbol{x})]_{k=1,\dots,n_j}$ and $\boldsymbol{C}_{ij} = [C_{ij}(x_i^{(k)}, x_j^{(l)})]_{\substack{k=1,\dots,n_i \\ l=1,\dots,n_j}}$.

- For $(\boldsymbol{\beta}_t)_{t=1,\dots,s}$ and $\mathbf{V}(\boldsymbol{x},\tilde{\boldsymbol{x}})$ known, the predictive distribution $(Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)})$ is Gaussian with mean $\mu_s(\boldsymbol{x})$ and variance $\sigma_s^2(\boldsymbol{x})$ given by :

$$\begin{cases} \mu_{Z_s}(\boldsymbol{x}) = \mathbf{f}_s'(\boldsymbol{x})\boldsymbol{\beta}_s + \boldsymbol{r}_s'(x)\mathbf{V}_s^{-1}\left(\mathbf{z}^{(s)} - \mathbf{M}^{(s)}\right), \\ \sigma_{Z_s}^2(\boldsymbol{x}) = C_{ss}(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{r}_s'(x)\mathbf{V}_s^{-1}\boldsymbol{r}_s(x), \end{cases}$$

where $\boldsymbol{r}_s'(x) = \begin{pmatrix} \boldsymbol{r}_{s1}'(x) & \dots & \boldsymbol{r}_{ss}'(x) \end{pmatrix}$ and $\mathbf{V}_s = \begin{pmatrix} \boldsymbol{C}_{11} & \dots & \boldsymbol{C}_{1s} \\ \vdots & \ddots & \vdots \\ \boldsymbol{C}_{s1} & \dots & \boldsymbol{C}_{ss} \end{pmatrix}$.

A particular case : Co-kriging models using function derivatives

- The objective here is not to approximate a component of a vectorial function but approximate a univariate function $z(\boldsymbol{x})$ by using its derivatives function.

  But the formalism is near the GPR formalism for vectorial functions.

- As usual, we suppose that $z(\boldsymbol{x})$ is a realization of $Z(\boldsymbol{x}) \sim \mathcal{PG}(\boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\beta}, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$

A particular case : Co-kriging models using function derivatives

- The objective here is not to approximate a component of a vectorial function but approximate a univariate function $z(\boldsymbol{x})$ by using its derivatives function.

  But the formalism is near the GPR formalism for vectorial functions.

- As usual, we suppose that $z(\boldsymbol{x})$ is a realization of $Z(\boldsymbol{x}) \sim \mathcal{PG}(\boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\beta}, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$
- First, we have the following result : $\partial Z(\boldsymbol{x})/\partial x_i$, $\boldsymbol{x} = (x_1, \ldots, x_d)$ exists if and only if its covariance kernel $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is twice differentiable with respect to x_i.

A particular case : Co-kriging models using function derivatives

- The objective here is not to approximate a component of a vectorial function but approximate a univariate function $z(\boldsymbol{x})$ by using its derivatives function.

  But the formalism is near the GPR formalism for vectorial functions.

- As usual, we suppose that $z(\boldsymbol{x})$ is a realization of $Z(\boldsymbol{x}) \sim \mathcal{PG}(\boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\beta}, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$
- First, we have the following result : $\partial Z(\boldsymbol{x})/\partial x_i$, $\boldsymbol{x} = (x_1, \ldots, x_d)$ exists if and only if its covariance kernel $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is twice differentiable with respect to x_i.
- Secondly, as the differential operator is linear, if the covariance kernels are well defined, then the stochastic process $(Z(\boldsymbol{x}), (\partial Z(\boldsymbol{x})/\partial x_i)_{i=1,\ldots,d})$ is Gaussian.

A particular case : Co-kriging models using function derivatives

- The objective here is not to approximate a component of a vectorial function but approximate a univariate function $z(\boldsymbol{x})$ by using its derivatives function.

  But the formalism is near the GPR formalism for vectorial functions.

- As usual, we suppose that $z(\boldsymbol{x})$ is a realization of $Z(\boldsymbol{x}) \sim \mathcal{PG}(\boldsymbol{f}'(\boldsymbol{x})\boldsymbol{\beta}, C(\boldsymbol{x}, \tilde{\boldsymbol{x}}))$
- First, we have the following result : $\partial Z(\boldsymbol{x})/\partial x_i$, $\boldsymbol{x} = (x_1, \ldots, x_d)$ exists if and only if its covariance kernel $C(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is twice differentiable with respect to x_i.
- Secondly, as the differential operator is linear, if the covariance kernels are well defined, then the stochastic process $(Z(\boldsymbol{x}), (\partial Z(\boldsymbol{x})/\partial x_i)_{i=1,\ldots,d})$ is Gaussian.
- We have the following cross covariances, for $i, j = 1, \ldots, d$:

$$\begin{aligned} \operatorname{cov}\left(Z(\boldsymbol{x}), \frac{\partial Z(\tilde{\boldsymbol{x}})}{\partial \tilde{x}_i}\right) &= \frac{\partial C(\boldsymbol{x}, \tilde{\boldsymbol{x}})}{\partial \tilde{x}_i}, \\ \operatorname{cov}\left(\frac{\partial Z(\boldsymbol{x})}{\partial x_i}, \frac{\partial Z(\tilde{\boldsymbol{x}})}{\partial \tilde{x}_j}\right) &= \frac{\partial^2 C(\boldsymbol{x}, \tilde{\boldsymbol{x}})}{\partial x_i \partial \tilde{x}_j}. \end{aligned}$$

A particular case : Co-kriging models using function derivatives

- Let $\mathbf{Z}$ the values of $Z(\boldsymbol{x})$ at points in $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(1)}\}$, $\boldsymbol{x}^{(j)} = (x_1^{(j)}, \ldots, x_d^{(j)}) \in \mathbb{R}^d$, $j = 1, \ldots, n$.
- Similarly, $\mathbf{z}$ and $\mathbf{z}_{(i)}$ are the values of $z(\boldsymbol{x})$ and $\partial z(\boldsymbol{x})/\partial x_i$ at points in $\mathbf{D}$.

A particular case : Co-kriging models using function derivatives

- Let $\mathbf{Z}$ the values of $Z(\boldsymbol{x})$ at points in $\mathbf{D} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(1)}\}$, $\boldsymbol{x}^{(j)} = (x_1^{(j)}, \ldots, x_d^{(j)}) \in \mathbb{R}^d$, $j = 1, \ldots, n$.
- Similarly, $\mathbf{z}$ and $\mathbf{z}_{(i)}$ are the values of $z(\boldsymbol{x})$ and $\partial z(\boldsymbol{x})/\partial x_i$ at points in $\mathbf{D}$.
- We have just to express the predictive distribution of $(Z(x), \mathbf{Z}, (\mathbf{Z}_{(i)})_{i=1,\ldots,d})$ by following the same formalism presented before.
  - Express the joint distribution $(Z(x), \mathbf{Z}, (\mathbf{Z}_{(i)})_{i=1,\ldots,d})$
  - Use the Gaussian conditioning theorem
- For more details about GPR with derivatives :

  Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments : use of derivatives in surface prediction. Technometrics, 35(3) :243–255.

  Mitchell, T., Morris, M., and Ylvisaker, D. (1994). Asymptotically optimum experimental designs for prediction of deterministic functions given derivative information. Journal of statistical planning and inference, 41(3) :377–389.

Co-kriging models using function derivatives - illustration

- Let $Z(x)$ be a Gaussian process with mean zero and Gussian covariance kernel $C(x,\tilde{x}) = 4\exp\left(-(x-\tilde{x})^2/2\theta^2\right)$ with $\theta = 0.1$ and $x \in [0,1]$.

- We find :

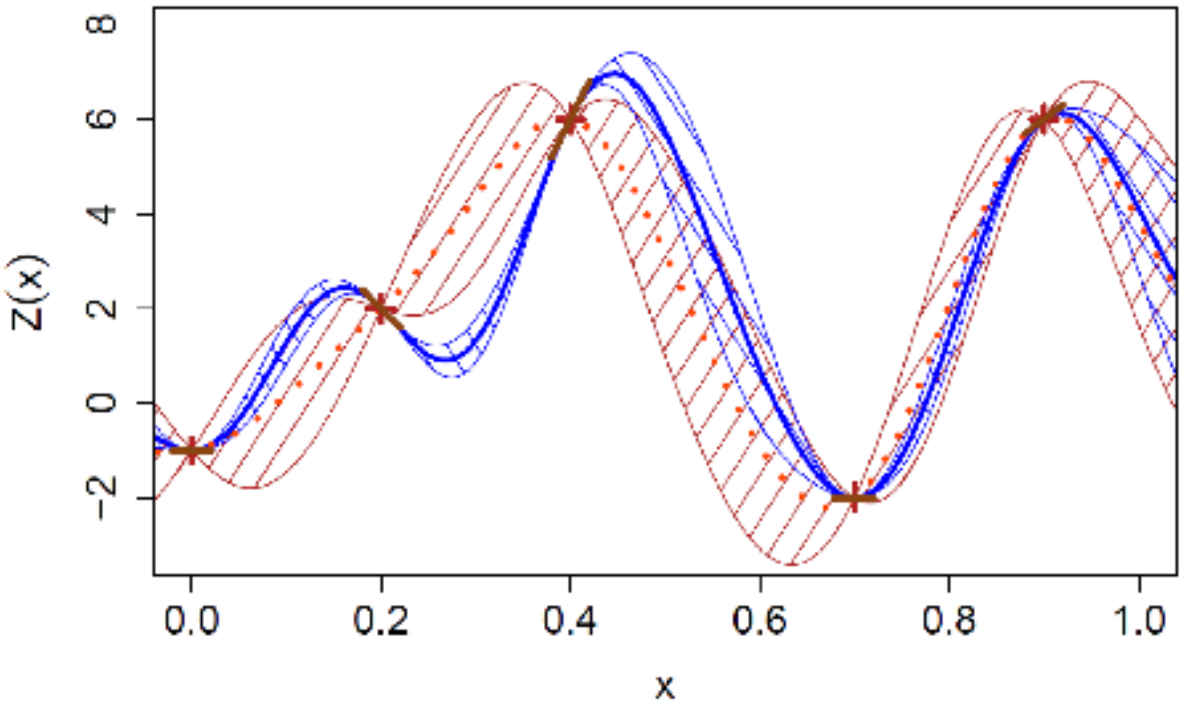
$$\mathrm{cov}\left(Z(x), \frac{dZ}{d\tilde{x}}(\tilde{x})\right) = 4\frac{(x-\tilde{x})}{\theta^2}\exp\left(-\frac{(x-\tilde{x})^2}{2\theta^2}\right)$$

$$\mathrm{cov}\left(\frac{dZ}{dx}(x), \frac{dZ}{d\tilde{x}}(\tilde{x})\right) = 4\left(\frac{1}{\theta^2} - \frac{(x-\tilde{x})^2}{\theta^4}\right)\exp\left(-\frac{(x-\tilde{x})^2}{2\theta^2}\right).$$

- Now let us condition $Z(x)$ at points $\mathbf{D} = (0.0, 0.2, 0.4, 0.7, 0.9)$ with $z(\mathbf{D}) = (-1, 2, 6, -2, 6)$ and $(dz/dx)(\mathbf{D}) = (0, -20, 40, 0, 15)$.

Co-kriging models using function derivatives - illustration

Illustration of the predictive means and confidence intervals obtained with a simple kriging (dotted line) and a simple co-kriging using the derivatives (solid line).



Outline

Linear model with 2 levels of code

Multi-fidelity framework

- We have seen the GPR formalism for vectorial functions in a general framework, where the **covariance structure** $\mathbf{V}(x, \tilde{x})$ is of **primary importance**.
- When neither $z_1(x)$ nor $z_2(x)$ are known, it is difficult or impossible to know (or assume) a covariance structure.

Linear model with 2 levels of code

Multi-fidelity framework

- We have seen the GPR formalism for vectorial functions in a general framework, where the **covariance structure** $\mathbf{V}(x, \tilde{x})$ is of **primary importance**.
- When neither $z_1(x)$ nor $z_2(x)$ are known, it is difficult or impossible to know (or assume) a covariance structure.
- We need to **assume a relationship between the two outputs** !

Linear model with 2 levels of code

Multi-fidelity framework

- We have seen the GPR formalism for vectorial functions in a general framework, where the **covariance structure** $\mathbf{V}(x, \tilde{x})$ is of **primary importance**.
- When neither $z_1(x)$ nor $z_2(x)$ are known, it is difficult or impossible to know (or assume) a covariance structure.
- We need to **assume a relationship between the two outputs** !
- Recall the **multi-fidelity context** : $z_1(x)$ and $z_2(x)$ are the outputs of the code modelling the same phenomenon with different computation times and accuracies. We want to approximate $z_2(x)$ which is the most accurate and costly code.

  For example $z_2(x)$ can be the output of a 3D modeling and $z_1(x)$ the output of a 2D or 1D modeling.

  For example, $z_2(x)$ may be the output of an expensive numerical reference code and $z_1(x)$ may be an earlier version of it, neglecting certain phenomena.

Linear model with 2 levels of code

Multi-fidelity framework

- We have seen the GPR formalism for vectorial functions in a general framework, where the **covariance structure** $\mathbf{V}(x, \tilde{x})$ is of **primary importance**.
- When neither $z_1(x)$ nor $z_2(x)$ are known, it is difficult or impossible to know (or assume) a covariance structure.
- We need to **assume a relationship between the two outputs** !
- Recall the **multi-fidelity context** : $z_1(x)$ and $z_2(x)$ are the outputs of the code modelling the same phenomenon with different computation times and accuracies. We want to approximate $z_2(x)$ which is the most accurate and costly code.

  For example $z_2(x)$ can be the output of a 3D modeling and $z_1(x)$ the output of a 2D or 1D modeling.

  For example, $z_2(x)$ may be the output of an expensive numerical reference code and $z_1(x)$ may be an earlier version of it, neglecting certain phenomena.
- A **linear relationship** between the two outputs appears to be appropriate in a multifidelity framework : the expensive code can be seen as the sum of the light code at a scale factor and an error term.

Linear model with 2 levels of code

The case where one knows perfectly $z_1(x)$ - linear regression

- The simplest model is to approximate the costly code $z_2(x)$ by **linear regression** assuming that the cheaper code $z_1(x)$ is a regression function :

$$\hat{z}_2(\boldsymbol{x}) = \rho z_1(\boldsymbol{x}) + \sum_{i=1}^{p} \mathrm{f}_i(\boldsymbol{x})\beta_i$$

Linear model with 2 levels of code

The case where one knows perfectly $z_1(x)$ - linear regression

- The simplest model is to approximate the costly code $z_2(x)$ by **linear regression** assuming that the cheaper code $z_1(x)$ is a regression function :

$$\hat{z}_2(\boldsymbol{x}) = \rho z_1(\boldsymbol{x}) + \sum_{i=1}^{p} \mathrm{f}_i(\boldsymbol{x})\beta_i$$

- In matrix form, linear regression model is given by : $\mathbf{Z}_2 = \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

  $\mathbf{Z}_2 = \left(Z_2(\boldsymbol{x}^{(1)}), \ldots, Z_2(\boldsymbol{x}^{(n_2)})\right)'$, the random vector of observations

  $(Z_2(\boldsymbol{x}^{(i)}))_{i=1,\ldots,n_2}$ are assumed to be i.id.

  $\mathbf{F} = \begin{pmatrix} z_1(\boldsymbol{x}^{(1)}) & \mathrm{f}_1(\boldsymbol{x}^{(1)}) & \cdots & \mathrm{f}_p(\boldsymbol{x}^{(1)}) \\ \vdots & & \vdots & \\ z_1(\boldsymbol{x}^{(n_2)}) & \mathrm{f}_1(\boldsymbol{x}^{(n_2)}) & \cdots & \mathrm{f}_p(\boldsymbol{x}^{(n_2)}) \end{pmatrix}$ is the matrix of regressors,

  $\boldsymbol{\beta} = (\rho, \beta_1, \ldots, \beta_p)'$ is the vector of coefficients,

  $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_{n_2}$ the vector of the random vector of the residuals such that $(\varepsilon_i)_{i=1,\ldots,n_2}$ are i.id, $\mathbb{E}(\varepsilon_i) = 0$ and $\mathrm{var}(\varepsilon_i) = \sigma^2$.

- Least-squares estimate of the regression coefficient β : $\hat{\boldsymbol{\beta}} = \left(\mathbf{F}'\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{z_2}$.

- No more details : it's a **classical linear model** !

Linear model with 2 levels of code

The case where one knows perfectly $z_1(x)$ - Gaussian process regression

- We suppose now that $z_2(\boldsymbol{x})$ is a realization of a Gaussian process $Z_2(\boldsymbol{x})$.
- The model become : $Z_2(\boldsymbol{x}) = \rho z_1(\boldsymbol{x}) + \delta(\boldsymbol{x})$
  where $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$, $\mathbf{f}_\delta(\boldsymbol{x})$ is a vector of p known fonctions.

Linear model with 2 levels of code

The case where one knows perfectly $z_1(x)$ - Gaussian process regression

- We suppose now that $z_2(\boldsymbol{x})$ is a realization of a Gaussian process $Z_2(\boldsymbol{x})$.
- The model become : $Z_2(\boldsymbol{x}) = \rho z_1(\boldsymbol{x}) + \delta(\boldsymbol{x})$
  where $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$, $\mathbf{f}_\delta(\boldsymbol{x})$ is a vector of p known fonctions.
- Then we have : $Z_2(\boldsymbol{x}) \sim \mathcal{GP}\left(\begin{pmatrix} z_1(x) & \mathbf{f}_\delta'(\boldsymbol{x}) \end{pmatrix}\begin{pmatrix} \rho \\ \boldsymbol{\beta}_\delta \end{pmatrix}, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$

Linear model with 2 levels of code

The case where one knows perfectly $z_1(x)$ - Gaussian process regression

- We suppose now that $z_2(\boldsymbol{x})$ is a realization of a Gaussian process $Z_2(\boldsymbol{x})$.
- The model become : $Z_2(\boldsymbol{x}) = \rho z_1(\boldsymbol{x}) + \delta(\boldsymbol{x})$
  where $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$, $\mathbf{f}_\delta(\boldsymbol{x})$ is a vector of p known fonctions.
- Then we have : $Z_2(\boldsymbol{x}) \sim \mathcal{GP}\left(\begin{pmatrix} z_1(x) & \mathbf{f}_\delta'(\boldsymbol{x}) \end{pmatrix}\begin{pmatrix} \rho \\ \boldsymbol{\beta}_\delta \end{pmatrix}, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$
- We are in the case of **GPR for univariate functions** !
  $\rightarrow$ universal kriging with the conditioned mean and variance functions of $Z_2(\boldsymbol{x})$ incorporates the estimation of $\boldsymbol{\beta} = (\rho, \boldsymbol{\beta}_\delta')'$ done by (MLE) :

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}' \boldsymbol{R}_{\delta,\mathbf{D}_2}^{-1} \mathbf{F})^{-1} \mathbf{F}' \boldsymbol{R}_{\delta,\mathbf{D}_2}^{-1} \mathbf{z}_{2\cdot},$$

  $\mathbf{F}$ is already definied in the previous slide.

Linear model with 2 levels of code

The case where one knows perfectly $z_1(x)$ - Gaussian process regression

- We suppose now that $z_2(\boldsymbol{x})$ is a realization of a Gaussian process $Z_2(\boldsymbol{x})$.
- The model become : $Z_2(\boldsymbol{x}) = \rho z_1(\boldsymbol{x}) + \delta(\boldsymbol{x})$
  where $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(\boldsymbol{x})\beta_\delta, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$, $\mathbf{f}_\delta(\boldsymbol{x})$ is a vector of p known fonctions.
- Then we have : $Z_2(\boldsymbol{x}) \sim \mathcal{GP}\left(\begin{pmatrix} z_1(x) & \mathbf{f}_\delta'(\boldsymbol{x}) \end{pmatrix}\begin{pmatrix} \rho \\ \boldsymbol{\beta}_\delta \end{pmatrix}, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$
- We are in the case of **GPR for univariate functions** !
  $\rightarrow$ universal kriging with the conditioned mean and variance functions of $Z_2(\boldsymbol{x})$ incorporates the estimation of $\boldsymbol{\beta} = (\rho, \boldsymbol{\beta}_\delta')'$ done by (MLE) :
  $$\hat{\boldsymbol{\beta}} = (\mathbf{F}' \boldsymbol{R}_{\delta,\mathbf{D}_2}^{-1} \mathbf{F})^{-1} \mathbf{F}' \boldsymbol{R}_{\delta,\mathbf{D}_2}^{-1} \mathbf{z}_{2\cdot},$$
  $\mathbf{F}$ is already definied in the previous slide.
- Finally, one could almost see Gaussian process regression for univariate functions as multifidelity with the trend as coarse code :
  $$Z_2(\boldsymbol{x}) = \mathbf{f}'\boldsymbol{\beta} + \delta(\boldsymbol{x}) \text{ with } \delta(\boldsymbol{x}) \sim \mathcal{GP}\left(0, \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$$

AR(1) model with 2 levels of code

Notations and assumptions

- As usual in the GPR framework, we suppose that $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), z_2(\boldsymbol{x}))$ is a realization of a bivariate Gaussian proces $(Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$

- We keep the same notations as before.

  $\mathbf{D}_t = \{\boldsymbol{x}_t^{(1)}, \dots, \boldsymbol{x}_t^{(n_t)}\}$ the DOE for the component t

  $\mathbf{z}_t = z_t(\mathbf{D}_t) = (z_t(\boldsymbol{x}_t^{(1)}), \dots, z_t(\boldsymbol{x}_t^{(n_t)}))'$; $\mathbf{z}^{(s)} = (\mathbf{z}_1', \dots, \mathbf{z}_s')'$

  $\mathbf{Z}_t = Z_t(\mathbf{D}_t) = (Z_t(\boldsymbol{x}_t^{(1)}), \dots, Z_t(\boldsymbol{x}_t^{(n_t)}))'$; $\mathbf{Z}^{(s)} = (\mathbf{Z}_1', \dots, \mathbf{Z}_s')'$

  $\mathbf{M}_t = \mathbf{f}_t'(\mathbf{D}_t)\boldsymbol{\beta}_t := \mathbf{F}_t\boldsymbol{\beta}_t$; $\mathbf{M}^{(s)} = (\mathbf{M}_1, \dots, \mathbf{M}_s)$

  $\mathbf{f}_t(\boldsymbol{x}) = (\mathrm{f}_{1,t}, \dots, \mathrm{f}_{p_t,t})$

- We add an assumption : the two DOE are **nested** $\mathbf{D}_2 \subseteq \mathbf{D}_1$

  This is not a big assumption : as coarse code is less expensive than reference code, it is assumed that observations can facilement be added to coarse code.

AR(1) model with 2 levels of code

The AR(1) model

- In the AR(1) model, the dependency between the two levels of code is assumed to be as follows : M. Kennedy and A. O'Hagan (2000), Predicting the output from a complex computer code when fast approximations are available, Biometrika. Forrester A.I.J., Sobester, A. and A.J. Keane (2007), Multi-fidelity optimization via surrogate modelling, Proc. R. Soc.

$$Z_2(\boldsymbol{x}) = \rho Z_1(\boldsymbol{x}) + \delta(\boldsymbol{x})$$

where $\delta(\boldsymbol{x})$ is a Gaussian process independent of $Z_1(\boldsymbol{x})$,

$$\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(m_\delta(\boldsymbol{x}) = \mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta, C_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$$

$$Z_1(x) \sim \mathcal{GP}\left(m_1(\boldsymbol{x}) = \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1, C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$$

ρ represents a scale factor between $Z_2(\boldsymbol{x})$ and $Z_1(\boldsymbol{x})$.

AR(1) model with 2 levels of code

The AR(1) model

- In the AR(1) model, the dependency between the two levels of code is assumed to be as follows : M. Kennedy and A. O'Hagan (2000), Predicting the output from a complex computer code when fast approximations are available, Biometrika. Forrester A.I.J., Sobester, A. and A.J. Keane (2007), Multi-fidelity optimization via surrogate modelling, Proc. R. Soc.

$$Z_2(\boldsymbol{x}) = \rho Z_1(\boldsymbol{x}) + \delta(\boldsymbol{x})$$

where $\delta(\boldsymbol{x})$ is a Gaussian process independent of $Z_1(\boldsymbol{x})$,

$$\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(m_\delta(\boldsymbol{x}) = \mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta, C_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$$

$$Z_1(x) \sim \mathcal{GP}\left(m_1(\boldsymbol{x}) = \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1, C_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$$

ρ represents a scale factor between $Z_2(\boldsymbol{x})$ and $Z_1(\boldsymbol{x})$.

- In fact, the AR(1) model is derived from the following Markov property :

$$\mathrm{Cov}(Z_2(\boldsymbol{x}), Z_1(\tilde{\boldsymbol{x}}) | Z_1(\boldsymbol{x})) = 0 \qquad \forall \boldsymbol{x} \neq \tilde{\boldsymbol{x}}.$$

The property means that if $Z_1(\boldsymbol{x})$ is known, then nothing more can be learn about $Z_2(\boldsymbol{x})$ from any other run of the cheaper code $Z_1(\tilde{\boldsymbol{x}})$ for $\tilde{\boldsymbol{x}} \neq \boldsymbol{x}$.

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ (**do it**) :

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(x) = (Z_1(x), Z_2(x))$ (**do it**) :

$$\mathbf{m}(x) =$$

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(x) = (Z_1(x), Z_2(x))$ (**do it**) :

$$\mathbf{m}(x) = \begin{pmatrix} \mathbf{f}_1'(x)\beta_1 \\ \rho\mathbf{f}_1'(x)\beta_1 + \mathbf{f}_\delta'(x)\beta_\delta \end{pmatrix}$$

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ (**do it**) :

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 \\ \rho\mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 + \mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta \end{pmatrix}$$

$$\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) =$$

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ (**do it**) :

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 \\ \rho\mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 + \mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta \end{pmatrix}$$

$$\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \rho^2\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix},$$

- We want to express the joint distribution of $Z_2(\boldsymbol{x})$ and $\mathbf{Z}^{(2)}$ (**do it again**) :

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ (**do it**) :

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 \\ \rho\mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 + \mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta \end{pmatrix}$$

$$\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \rho^2\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix},$$

- We want to express the joint distribution of $Z_2(\boldsymbol{x})$ and $\mathbf{Z}^{(2)}$ (**do it again**) :
  - First, $\mathbf{Z}^{(2)} = (\mathbf{Z}_1', \mathbf{Z}_2')'$ is a gaussian vector of size $n_1 + n_2$ with :

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(x) = (Z_1(x), Z_2(x))$ (**do it**) :

$$\mathbf{m}(x) = \begin{pmatrix} \mathbf{f}_1'(x)\beta_1 \\ \rho\mathbf{f}_1'(x)\beta_1 + \mathbf{f}_\delta'(x)\beta_\delta \end{pmatrix}$$

$$\mathbf{V}(x, \tilde{x}) = \begin{pmatrix} \sigma_1^2 R_1(x, \tilde{x}) & \rho\sigma_1^2 R_1(x, \tilde{x}) \\ \rho\sigma_1^2 R_1(x, \tilde{x}) & \rho^2\sigma_1^2 R_1(x, \tilde{x}) + \sigma_\delta^2 R_\delta(x, \tilde{x}) \end{pmatrix},$$

- We want to express the joint distribution of $Z_2(x)$ and $\mathbf{Z}^{(2)}$ (**do it again**) :

  – First, $\mathbf{Z}^{(2)} = (\mathbf{Z}_1', \mathbf{Z}_2')'$ is a gaussian vector of size $n_1 + n_2$ with :

$$\mathbb{E}\left[\mathbf{Z}^{(2)}\right] = \begin{pmatrix} \mathbf{f}_1'(\mathbf{D}_1)\beta_1 \\ \rho\mathbf{f}_1'(\mathbf{D}_2)\beta_1 + \mathbf{f}_\delta'(\mathbf{D}_2)\beta_\delta \end{pmatrix}, \text{ with } \mathbf{f}_i'(\mathbf{D}_j) = \begin{pmatrix} \mathbf{f}_i'(\boldsymbol{x}_j^{(1)}) \\ \vdots \\ \mathbf{f}_i'(\boldsymbol{x}_j^{(n_j)}) \end{pmatrix} \text{ and}$$

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(x) = (Z_1(x), Z_2(x))$ (**do it**) :

$$\mathbf{m}(x) = \begin{pmatrix} \mathbf{f}_1'(x)\beta_1 \\ \rho\mathbf{f}_1'(x)\beta_1 + \mathbf{f}_\delta'(x)\beta_\delta \end{pmatrix}$$

$$\mathbf{V}(x, \tilde{x}) = \begin{pmatrix} \sigma_1^2 R_1(x, \tilde{x}) & \rho\sigma_1^2 R_1(x, \tilde{x}) \\ \rho\sigma_1^2 R_1(x, \tilde{x}) & \rho^2\sigma_1^2 R_1(x, \tilde{x}) + \sigma_\delta^2 R_\delta(x, \tilde{x}) \end{pmatrix},$$

- We want to express the joint distribution of $Z_2(x)$ and $\mathbf{Z}^{(2)}$ (**do it again**) :

  – First, $\mathbf{Z}^{(2)} = (\mathbf{Z}_1', \mathbf{Z}_2')'$ is a gaussian vector of size $n_1 + n_2$ with :

$$\mathbb{E}\left[\mathbf{Z}^{(2)}\right] = \begin{pmatrix} \mathbf{f}_1'(\mathbf{D}_1)\beta_1 \\ \rho\mathbf{f}_1'(\mathbf{D}_2)\beta_1 + \mathbf{f}_\delta'(\mathbf{D}_2)\beta_\delta \end{pmatrix}, \text{ with } \mathbf{f}_i'(\mathbf{D}_j) = \begin{pmatrix} \mathbf{f}_i'(x_j^{(1)}) \\ \vdots \\ \mathbf{f}_i'(x_j^{(n_j)}) \end{pmatrix} \text{ and}$$

$$\mathbf{V}_2 = \mathrm{Cov}(\mathbf{Z}_1, \mathbf{Z}_2) = \begin{pmatrix} \sigma_1^2 R_1(\mathbf{D}_1, \mathbf{D}_1) & \rho\sigma_1^2 R_1(\mathbf{D}_1, \mathbf{D}_2) \\ \rho\sigma_1^2 R_1(\mathbf{D}_2, \mathbf{D}_1) & \rho^2\sigma_1^2 R_1(\mathbf{D}_2, \mathbf{D}_2) + \sigma_\delta^2 R_\delta(\mathbf{D}_2, \mathbf{D}_2) \end{pmatrix}$$

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now express the mean and the covariance $\mathbf{Z}(\boldsymbol{x}) = (Z_1(\boldsymbol{x}), Z_2(\boldsymbol{x}))$ (**do it**) :

$$\mathbf{m}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 \\ \rho\mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 + \mathbf{f}_\delta'(\boldsymbol{x})\boldsymbol{\beta}_\delta \end{pmatrix}$$

$$\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \begin{pmatrix} \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \\ \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) & \rho^2\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + \sigma_\delta^2 R_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \end{pmatrix},$$

- We want to express the joint distribution of $Z_2(\boldsymbol{x})$ and $\mathbf{Z}^{(2)}$ (**do it again**) :

  – First, $\mathbf{Z}^{(2)} = (\mathbf{Z}_1', \mathbf{Z}_2')'$ is a gaussian vector of size $n_1 + n_2$ with :

$$\mathbb{E}\left[\mathbf{Z}^{(2)}\right] = \begin{pmatrix} \mathbf{f}_1'(\mathbf{D}_1)\boldsymbol{\beta}_1 \\ \rho\mathbf{f}_1'(\mathbf{D}_2)\boldsymbol{\beta}_1 + \mathbf{f}_\delta'(\mathbf{D}_2)\boldsymbol{\beta}_\delta \end{pmatrix}, \text{ with } \mathbf{f}_i'(\mathbf{D}_j) = \begin{pmatrix} \mathbf{f}_i'(\boldsymbol{x}_j^{(1)}) \\ \vdots \\ \mathbf{f}_i'(\boldsymbol{x}_j^{(n_j)}) \end{pmatrix} \text{ and}$$

$$\mathbf{V}_2 = \mathrm{Cov}(\mathbf{Z}_1, \mathbf{Z}_2) = \begin{pmatrix} \sigma_1^2 R_1(\mathbf{D}_1, \mathbf{D}_1) & \rho\sigma_1^2 R_1(\mathbf{D}_1, \mathbf{D}_2) \\ \rho\sigma_1^2 R_1(\mathbf{D}_2, \mathbf{D}_1) & \rho^2\sigma_1^2 R_1(\mathbf{D}_2, \mathbf{D}_2) + \sigma_\delta^2 R_\delta(\mathbf{D}_2, \mathbf{D}_2) \end{pmatrix}$$

Thus, $\mathbf{Z}^{(2)} \sim \mathcal{N}_{n_1+n_2}(\mathbf{H}\boldsymbol{\beta}, \mathbf{V}_2)$ with $\mathbf{H} = \begin{pmatrix} \mathbf{f}_1'(\mathbf{D}_1) & 0 \\ \rho\mathbf{f}_1'(\mathbf{D}_2) & \mathbf{f}_\delta'(\mathbf{D}_2) \end{pmatrix}$ and $\beta = \begin{pmatrix} \beta_1 \\ \beta_\delta \end{pmatrix}$.

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now easily write the desired joint distribution :

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now easily write the desired joint distribution :

$$\begin{pmatrix} Z_2(\boldsymbol{x}) \\ \mathbf{Z}^{(2)} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{h}'(\boldsymbol{x})\boldsymbol{\beta} \\ \mathbf{H}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \rho^2\sigma_1^2 + \sigma_\delta^2 & \boldsymbol{r}_2'(\boldsymbol{x}) \\ \boldsymbol{r}_2(\boldsymbol{x}) & \mathbf{V}_2 \end{pmatrix} \right) \text{ where } \mathbf{h}'(x) = (\rho \mathbf{f}_1'(\boldsymbol{x}), \mathbf{f}_\delta'(\boldsymbol{x}))$$

and :
$$\begin{aligned} \boldsymbol{r}_2'(\boldsymbol{x}) &= \mathrm{Cov}(Z_2(\boldsymbol{x}), \mathbf{Z}^{(2)}) \\ &= \left(\rho\sigma_1^2 R_1(\{\boldsymbol{x}\}, \mathbf{D}_1), \rho^2\sigma_1^2 R_1(\{\boldsymbol{x}\}, \mathbf{D}_2) + \sigma_\delta^2 R_\delta(\{\boldsymbol{x}\}, \mathbf{D}_2)\right) \end{aligned}$$

with $R_i(\{\boldsymbol{x}\}, \mathbf{D}_j) = \left(R_i(\boldsymbol{x}, \boldsymbol{x}_j^{(1)}), \ldots, R_i(\boldsymbol{x}, \boldsymbol{x}_j^{(n_j)})\right)$

Conditional distribution of $Z_2(x)$ given $\mathbf{Z}^{(2)}$ (do it again)

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now easily write the desired joint distribution :

$$\begin{pmatrix} Z_2(\boldsymbol{x}) \\ \mathbf{Z}^{(2)} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{h}'(\boldsymbol{x})\boldsymbol{\beta} \\ \mathbf{H}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \rho^2\sigma_1^2 + \sigma_\delta^2 & \boldsymbol{r}_2'(\boldsymbol{x}) \\ \boldsymbol{r}_2(\boldsymbol{x}) & \mathbf{V}_2 \end{pmatrix} \right) \text{ where } \mathbf{h}'(x) = (\rho \mathbf{f}_1'(\boldsymbol{x}), \mathbf{f}_\delta'(\boldsymbol{x}))$$

and :

$$\begin{aligned} \boldsymbol{r}_2'(\boldsymbol{x}) &= \mathrm{Cov}(Z_2(\boldsymbol{x}), \mathbf{Z}^{(2)}) \\ &= \left(\rho\sigma_1^2 R_1(\{\boldsymbol{x}\}, \mathbf{D}_1), \rho^2\sigma_1^2 R_1(\{\boldsymbol{x}\}, \mathbf{D}_2) + \sigma_\delta^2 R_\delta(\{\boldsymbol{x}\}, \mathbf{D}_2)\right) \end{aligned}$$

with $R_i(\{\boldsymbol{x}\}, \mathbf{D}_j) = \left(R_i(\boldsymbol{x}, \boldsymbol{x}_j^{(1)}), \ldots, R_i(\boldsymbol{x}, \boldsymbol{x}_j^{(n_j)})\right)$

Conditional distribution of $Z_2(x)$ given $\mathbf{Z}^{(2)}$ (do it again)

- As before, we suppose that we know the mean $\mathbf{m}(\boldsymbol{x})$ and the covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$. That is to say, that we know $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_\delta$, ρ, σ_1, σ_δ, $R_1(\cdot,\cdot)$ and $R_\delta(\cdot,\cdot)$

AR(1) model with 2 levels of code

Joint distribution of $Z_2(x)$ and observations

- We can now easily write the desired joint distribution :

$$\begin{pmatrix} Z_2(\boldsymbol{x}) \\ \mathbf{Z}^{(2)} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{h}'(\boldsymbol{x})\boldsymbol{\beta} \\ \mathbf{H}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \rho^2\sigma_1^2 + \sigma_\delta^2 & \boldsymbol{r}_2'(\boldsymbol{x}) \\ \boldsymbol{r}_2(\boldsymbol{x}) & \mathbf{V}_2 \end{pmatrix} \right) \text{ where } \mathbf{h}'(x) = (\rho \mathbf{f}_1'(\boldsymbol{x}), \mathbf{f}_\delta'(\boldsymbol{x}))$$

and :
$$\begin{aligned} \boldsymbol{r}_2'(\boldsymbol{x}) &= \mathrm{Cov}(Z_2(\boldsymbol{x}), \mathbf{Z}^{(2)}) \\ &= \left(\rho\sigma_1^2 R_1(\{\boldsymbol{x}\}, \mathbf{D}_1), \rho^2\sigma_1^2 R_1(\{\boldsymbol{x}\}, \mathbf{D}_2) + \sigma_\delta^2 R_\delta(\{\boldsymbol{x}\}, \mathbf{D}_2)\right) \end{aligned}$$

with $R_i(\{\boldsymbol{x}\}, \mathbf{D}_j) = \left(R_i(\boldsymbol{x}, \boldsymbol{x}_j^{(1)}), \ldots, R_i(\boldsymbol{x}, \boldsymbol{x}_j^{(n_j)})\right)$

Conditional distribution of $Z_2(x)$ given $\mathbf{Z}^{(2)}$ (do it again)

- As before, we suppose that we know the mean $\mathbf{m}(\boldsymbol{x})$ and the covariance function $\mathbf{V}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$. That is to say, that we know $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_\delta$, ρ, σ_1, σ_δ, $R_1(\cdot,\cdot)$ and $R_\delta(\cdot,\cdot)$
- By Gaussian conditionning, the conditional distribution $[Z_2(\boldsymbol{x}) | \mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$ is gaussian with mean $\mu_{Z_2}(\boldsymbol{x})$ and variance $\sigma_{Z_2}^2(\boldsymbol{x})$ given by :

$$\begin{cases} \mu_{Z_2}(\boldsymbol{x}) = \mathbf{h}'(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\left(\mathbf{z}^{(2)} - \mathbf{H}\boldsymbol{\beta}\right), \\ \sigma_{Z_2}^2(\boldsymbol{x}) = \rho^2\sigma_1^2 + \sigma_\delta^2 - \boldsymbol{r}_2'(\boldsymbol{x})\mathbf{V}_2^{-1}\boldsymbol{r}_2(\boldsymbol{x}), \end{cases}$$

Known as the simple multi-fidelity co-kriging.

AR(1) model with 2 levels of code

About the scale factor ρ

- By definition, we have $\mathrm{Cov}(Z_1(\boldsymbol{x}), Z_1(\tilde{\boldsymbol{x}})) = \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ and we have seen that $\mathrm{Cov}(Z_2(\boldsymbol{x}), Z_1(\tilde{\boldsymbol{x}})) = \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, so we can express ρ as :

$$\rho = \frac{\mathrm{Cov}(Z_2(\boldsymbol{x}), Z_1(\boldsymbol{x}))}{\mathrm{var}(Z_1(\boldsymbol{x}))}$$

  It both represents the **correlation degree** and the **scale factor** between two successive levels of code :

AR(1) model with 2 levels of code

About the scale factor ρ

- By definition, we have $\mathrm{Cov}(Z_1(\boldsymbol{x}), Z_1(\tilde{\boldsymbol{x}})) = \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ and we have seen that $\mathrm{Cov}(Z_2(\boldsymbol{x}), Z_1(\tilde{\boldsymbol{x}})) = \rho\sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, so we can express ρ as :

$$\rho = \frac{\mathrm{Cov}(Z_2(\boldsymbol{x}), Z_1(\boldsymbol{x}))}{\mathrm{var}(Z_1(\boldsymbol{x}))}$$

  It both represents the **correlation degree** and the **scale factor** between two successive levels of code :

- If $Z_1(\boldsymbol{x})$ and $Z_2(\boldsymbol{x}))$ are uncorellated, i.e. $\rho = 0$, the equations simplify and this amounts to approximating $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$ by Gaussian process regression in an independent way.

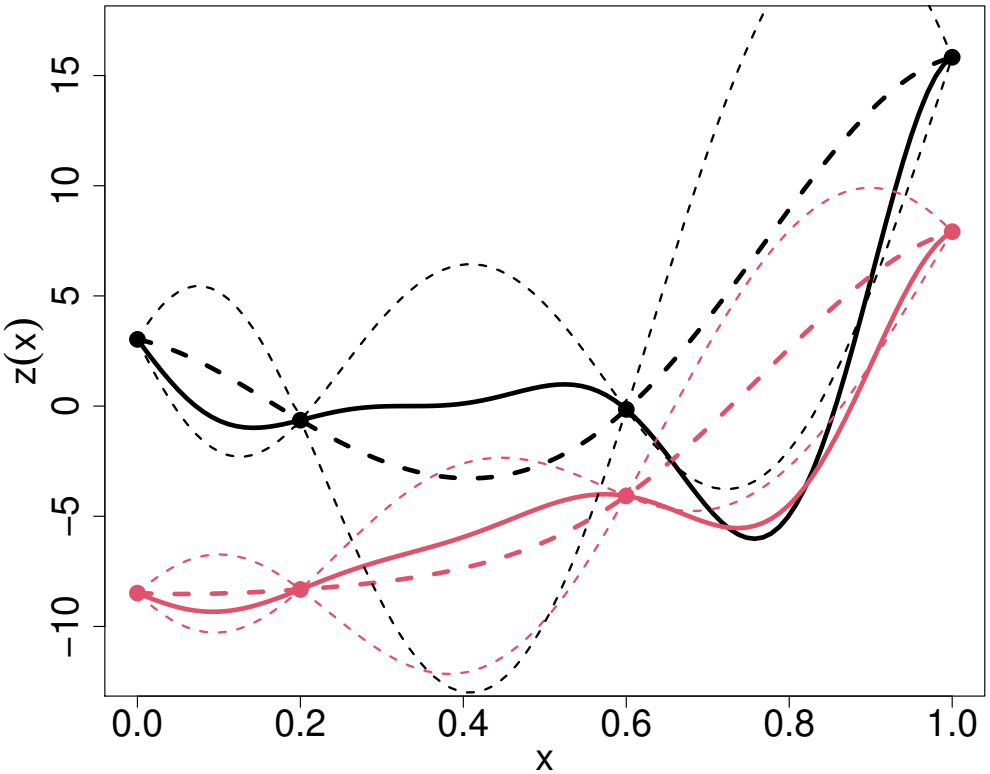
- The addition of information from $z_1(\boldsymbol{x})$ cannot degrade the GPR approximation of $z_2(\boldsymbol{x})$.

AR(1) model with 2 levels of code

Academic 1D illutration

The "cheap" code (in red) : $z_1(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$

The expensive code (in black) : $z_2(x) = 2z_1(x) - 20x + 20$



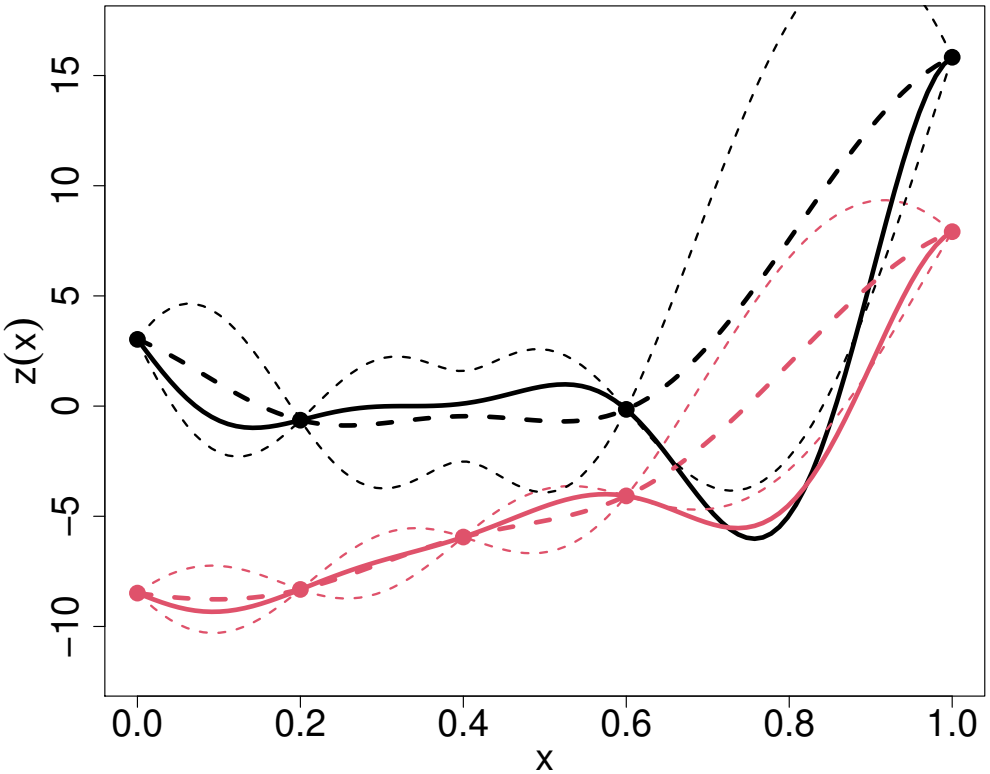
Same DOE : we learn nothing more about $z_2(x)$.

AR(1) model with 2 levels of code

Academic 1D illutration

The "cheap" code (in red) : $z_1(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$

The expensive code (in black) : $z_2(x) = 2z_1(x) - 20x + 20$



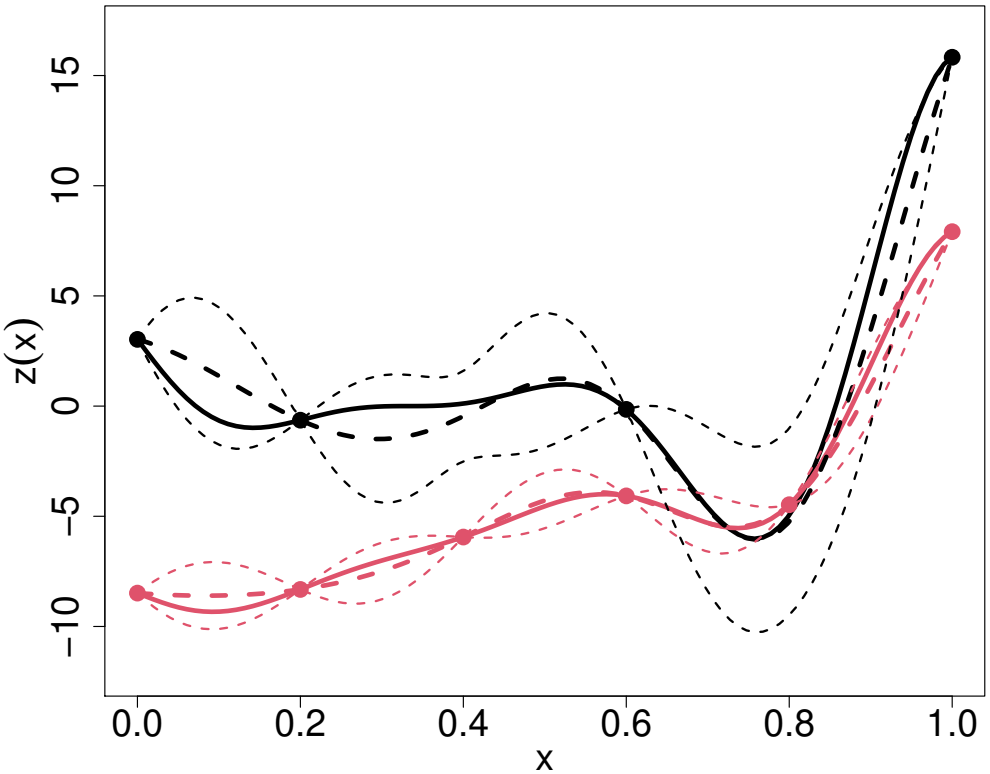
Nested DOE : knowledge of $z_1(x)$ gives us information about $z_2(x)$.

AR(1) model with 2 levels of code

Academic 1D illutration

The "cheap" code (in red) : $z_1(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$

The expensive code (in black) : $z_2(x) = 2z_1(x) - 20x + 20$



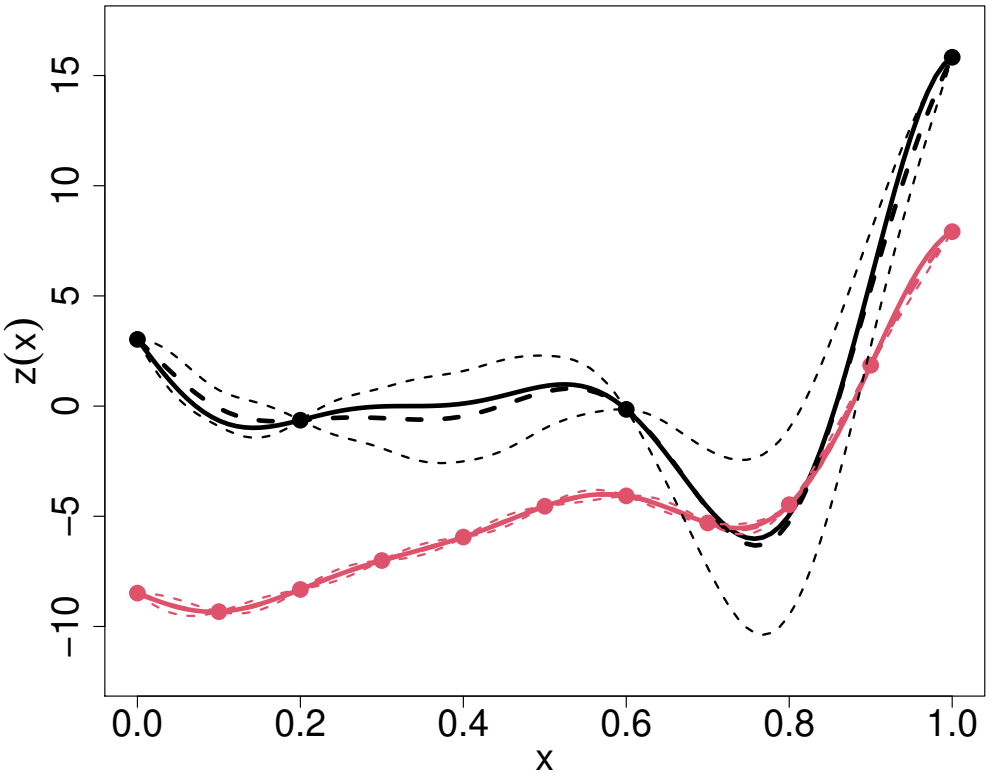
Nested DOE : knowledge of $z_1(x)$ gives us information about $z_2(x)$.

AR(1) model with 2 levels of code

Academic 1D illutration

The "cheap" code (in red) : $z_1(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$

The expensive code (in black) : $z_2(x) = 2z_1(x) - 20x + 20$



Nested DOE : knowledge of $z_1(x)$ gives us information about $z_2(x)$.

AR(1) model with 2 levels of code

AR(1) multi-fidelity model parameters estimation

- Reminder : for the GPR in the univariate case, the universal kriging equations allow to take into account the uncertainties due to the estimation of the trend coefficients β.

AR(1) model with 2 levels of code

AR(1) multi-fidelity model parameters estimation

- Reminder : for the GPR in the univariate case, the universal kriging equations allow to take into account the uncertainties due to the estimation of the trend coefficients β.
- We must also take into account of estimation of the scaling parameter ρ in the AR(1) model.

AR(1) model with 2 levels of code

AR(1) multi-fidelity model parameters estimation

- Reminder : for the GPR in the univariate case, the universal kriging equations allow to take into account the uncertainties due to the estimation of the trend coefficients $\boldsymbol{\beta}$.
- We must also take into account of estimation of the scaling parameter ρ in the AR(1) model.
- According to the hypothesis of independence between $Z_1(\boldsymbol{x})$ and $\delta(\boldsymbol{x})$, we can estimate $(\beta_1, \sigma_1^2, \theta_1)$ and $(\rho, \beta_\delta, \sigma_\delta^2, \theta_\delta)$ separately.

AR(1) model with 2 levels of code

AR(1) multi-fidelity model parameters estimation

- Reminder : for the GPR in the univariate case, the universal kriging equations allow to take into account the uncertainties due to the estimation of the trend coefficients $\boldsymbol{\beta}$.
- We must also take into account of estimation of the scaling parameter ρ in the AR(1) model.
- According to the hypothesis of independence between $Z_1(\boldsymbol{x})$ and $\delta(\boldsymbol{x})$, we can estimate $(\beta_1, \sigma_1^2, \theta_1)$ and $(\rho, \beta_\delta, \sigma_\delta^2, \theta_\delta)$ separately.
- For $Z_1(\boldsymbol{x})$ we proceed as in the univariate case : MLE or Bayesian procedure for $\boldsymbol{\beta}_1$, then σ_1^2 and then θ_1 or full-bayesian estimation.

AR(1) model with 2 levels of code

AR(1) multi-fidelity model parameters estimation

- Reminder : for the GPR in the univariate case, the universal kriging equations allow to take into account the uncertainties due to the estimation of the trend coefficients $\boldsymbol{\beta}$.
- We must also take into account of estimation of the scaling parameter ρ in the AR(1) model.
- According to the hypothesis of independence between $Z_1(\boldsymbol{x})$ and $\delta(\boldsymbol{x})$, we can estimate $(\beta_1, \sigma_1^2, \theta_1)$ and $(\rho, \beta_\delta, \sigma_\delta^2, \theta_\delta)$ separately.
- For $Z_1(\boldsymbol{x})$ we proceed as in the univariate case : MLE or Bayesian procedure for $\boldsymbol{\beta}_1$, then σ_1^2 and then θ_1 or full-bayesian estimation.
- For $Z_2(\boldsymbol{x})$ we need to estimate ρ and $\boldsymbol{\beta}_\delta$ together, indeed we can't suppose ρ and β to be independant.

  MLE or Bayesian procedure for $(\boldsymbol{\beta}_\delta, \rho)$, then σ_δ^2 and then θ_δ or full-bayesian estimation.

Le Gratiet, L. (2013), Bayesian analysis of hierarchical multifidelity codes, SIAM/ASA J. on Uncertain. Quantif.

MA, P. (2019), Objective Bayesian Analysis of a Cokriging Model for Hierarchical Multi delity Codes

AR(1) model with s levels of code

The model is easily generalized to s levels

- We have s levels of code $(z_t(\boldsymbol{x}))_{t=1,\ldots,s}$ sorted by increasing order of fidelity and modeled by Gaussian processes $(Z_t(\boldsymbol{x}))_{t=1,\ldots,s}$, $\boldsymbol{x} \in \mathbb{R}^d$. We want to approximate $z_s(x)$, the most accurate and costly code.
- AR(1) model for s levels : $Z_t(\boldsymbol{x}) = \rho_{t-1} Z_{t-1}(\boldsymbol{x}) + \delta_t(\boldsymbol{x}) \quad t = 2, \ldots, s$
  where $\delta_t(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_t'(\boldsymbol{x})\boldsymbol{\beta}_t, \sigma_t^2 R_t(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$ is independent of $Z_{t-1}(\boldsymbol{x}), \ldots, Z_1(\boldsymbol{x})$
  and $Z_1(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1, \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$

AR(1) model with s levels of code

The model is easily generalized to s levels

- We have s levels of code $(z_t(\boldsymbol{x}))_{t=1,\dots,s}$ sorted by increasing order of fidelity and modeled by Gaussian processes $(Z_t(\boldsymbol{x}))_{t=1,\dots,s}$, $\boldsymbol{x} \in \mathbb{R}^d$. We want to approximate $z_s(x)$, the most accurate and costly code.
- AR(1) model for s levels : $Z_t(\boldsymbol{x}) = \rho_{t-1} Z_{t-1}(\boldsymbol{x}) + \delta_t(\boldsymbol{x}) \quad t = 2, \dots, s$
  where $\delta_t(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_t'(\boldsymbol{x})\boldsymbol{\beta}_t, \sigma_t^2 R_t(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$ is independent of $Z_{t-1}(\boldsymbol{x}), \dots, Z_1(\boldsymbol{x})$
  and $Z_1(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1, \sigma_1^2 R_1(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$
- Proceeding in the same way as for 2 levels, one may find that the conditioned distribution $[Z_s(x) | \mathbf{Z}_1 = \mathbf{z}_1, \dots \mathbf{Z}_s = \mathbf{z}_s]$ (with all parameters known) is Gaussian with :

$$\mu_{Z_s}(\boldsymbol{x}) = \mathbf{h}_s'(\boldsymbol{x})\boldsymbol{\beta} + \mathbf{k}_s'(\boldsymbol{x})\mathbf{V}_s^{-1}(\mathbf{z} - \mathbf{H}_s\boldsymbol{\beta})$$

$$\sigma_{Z_s}^2(\boldsymbol{x}) = \sigma_{Z_s}^2 - \mathbf{k}_s'(\boldsymbol{x})\mathbf{V}_s^{-1}\mathbf{k}_s(\boldsymbol{x}).$$

- We don't define the notations but we can see that the formulation is similar to the one with 2 levels. For instance : $\mathbf{V}_s^{-1}$ is a $\sum_{t=1}^s n_t \times \sum_{t=1}^s n_t$ matrix , $\mathbf{H}_s$ is a $\sum_{t=1}^s n_t \times \sum_{t=1}^s p_t$ matrix (p_t the size of $\mathbf{f}_1'(\boldsymbol{x})$), ...

Recursive AR(1) model with 2 levels of code

Multi-fidelity co-kriging model : recursive formulation

- Reminders : We have 2 levels of code output $z_1(\boldsymbol{x}))$ and $(z_2(\boldsymbol{x})$ modeled by Gaussian processes $Z_1(\boldsymbol{x}))$ and $(Z_2(\boldsymbol{x})$. We want approximate $z_2(\boldsymbol{x})$, the most accurate and costly code, using observations from both code outputs.

Recursive AR(1) model with 2 levels of code

Multi-fidelity co-kriging model : recursive formulation

- Reminders : We have 2 levels of code output $z_1(\boldsymbol{x}))$ and $(z_2(\boldsymbol{x})$ modeled by Gaussian processes $Z_1(\boldsymbol{x}))$ and $(Z_2(\boldsymbol{x})$. We want approximate $z_2(\boldsymbol{x})$, the most accurate and costly code, using observations from both code outputs.
- The recursive AR(1) model is done by : Le Gratiet L. & Garnier J. (2014), Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, Int. J. Uncertain. Quantif.

$$\begin{cases} Z_2(\boldsymbol{x}) = \rho \tilde{Z}_1(\boldsymbol{x}) + \delta(\boldsymbol{x}) \\ \tilde{Z}_1(\boldsymbol{x}) \perp \delta_t(\boldsymbol{x}) \end{cases},$$

As before $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(x)\boldsymbol{\beta}_\delta, C_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_\delta^2 R_\delta(x, \tilde{x})\right)$. But, differently than before $\tilde{Z}_1(x)$ is a GP with distribution $[Z_1(x)|\mathbf{Z}_1 = \mathbf{z}_1, \boldsymbol{\beta}_1, \sigma_1^2, \theta_1]$.

Recursive AR(1) model with 2 levels of code

Multi-fidelity co-kriging model : recursive formulation

- Reminders : We have 2 levels of code output $z_1(\boldsymbol{x})$) and ($z_2(\boldsymbol{x})$ modeled by Gaussian processes $Z_1(\boldsymbol{x})$) and ($Z_2(\boldsymbol{x})$. We want approximate $z_2(\boldsymbol{x})$, the most accurate and costly code, using observations from both code outputs.
- The recursive AR(1) model is done by : Le Gratiet L. & Garnier J. (2014), Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, Int. J. Uncertain. Quantif.

$$\begin{cases} Z_2(\boldsymbol{x}) = \rho \tilde{Z}_1(\boldsymbol{x}) + \delta(\boldsymbol{x}) \\ \tilde{Z}_1(\boldsymbol{x}) \perp \delta_t(\boldsymbol{x}) \end{cases},$$

As before $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(x)\boldsymbol{\beta}_\delta, C_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_\delta^2 R_\delta(x, \tilde{x})\right)$. But, differently than before $\tilde{Z}_1(x)$ is a GP with distribution $[Z_1(x)|\mathbf{Z}_1 = \mathbf{z}_1, \boldsymbol{\beta}_1, \sigma_1^2, \theta_1]$.

→ We know its mean and its covariance (see GPR for univariate functions) :

$$\mu_{Z_1}(\boldsymbol{x}) = \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 + \boldsymbol{r_1}'(\boldsymbol{x})\boldsymbol{R}_{1,\mathbf{D}_1}^{-1}(\mathbf{z}_1 - \mathbf{f}_1'(\mathbf{D}_1)\boldsymbol{\beta}_1$$

$$\sigma_{Z_1}^2(\boldsymbol{x}) = \sigma_1^2(1 - \boldsymbol{r_1}'(\boldsymbol{x})\boldsymbol{R}_{1,\mathbf{D}_1}^{-1}\boldsymbol{r_1}(\boldsymbol{x}))$$

Recursive AR(1) model with 2 levels of code

Multi-fidelity co-kriging model : recursive formulation

- Reminders : We have 2 levels of code output $z_1(\boldsymbol{x})$) and ($z_2(\boldsymbol{x})$ modeled by Gaussian processes $Z_1(\boldsymbol{x})$) and ($Z_2(\boldsymbol{x})$. We want approximate $z_2(\boldsymbol{x})$, the most accurate and costly code, using observations from both code outputs.
- The recursive AR(1) model is done by : Le Gratiet L. & Garnier J. (2014), Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, Int. J. Uncertain. Quantif.

$$\begin{cases} Z_2(\boldsymbol{x}) = \rho \tilde{Z}_1(\boldsymbol{x}) + \delta(\boldsymbol{x}) \\ \tilde{Z}_1(\boldsymbol{x}) \perp \delta_t(\boldsymbol{x}) \end{cases},$$

As before $\delta(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_\delta'(x)\boldsymbol{\beta}_\delta, C_\delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \sigma_\delta^2 R_\delta(x, \tilde{x})\right)$. But, differently than before $\tilde{Z}_1(x)$ is a GP with distribution $[Z_1(x)|\mathbf{Z}_1 = \mathbf{z}_1, \boldsymbol{\beta}_1, \sigma_1^2, \theta_1]$.

$\rightarrow$ We know its mean and its covariance (see GPR for univariate functions) :

$$\mu_{Z_1}(\boldsymbol{x}) = \mathbf{f}_1'(\boldsymbol{x})\boldsymbol{\beta}_1 + \boldsymbol{r_1}'(\boldsymbol{x})\boldsymbol{R}_{1,\mathbf{D}_1}^{-1}(\mathbf{z}_1 - \mathbf{f}_1'(\mathbf{D}_1)\boldsymbol{\beta}_1$$

$$\sigma_{Z_1}^2(\boldsymbol{x}) = \sigma_1^2(1 - \boldsymbol{r_1}'(\boldsymbol{x})\boldsymbol{R}_{1,\mathbf{D}_1}^{-1}\boldsymbol{r_1}(\boldsymbol{x}))$$

- We can now deduce the distribution $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$ (with mean and covariance functions known)

Recursive AR(1) model with 2 levels of code

Multi-fidelity co-kriging model : recursive formulation

- Once again we suppose that all means and covariances parameters (i.e. β_1, β_δ, ρ, σ_1, σ_δ, $R_1(\cdot,\cdot)$ and $R_\delta(\cdot,\cdot)$ are known.
- The conditioned distribution $\left[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2\right]$ is Gaussian with mean $\mu_{Z_2}(\boldsymbol{x})$ and variance $\sigma_{Z_2}(\boldsymbol{x})$ done by :

$$\mu_{Z_2}(\boldsymbol{x}) = \rho\mu_{Z_1}(\boldsymbol{x}) + \mu_\delta(\boldsymbol{x})$$

$$\sigma^2_{Z_2}(\boldsymbol{x}) = \rho\sigma^2_{Z_1}(\boldsymbol{x}) + \sigma^2_\delta(\boldsymbol{x})$$

where $\mu_\delta(\boldsymbol{x})$ and $\sigma^2_\delta(\boldsymbol{x})$ are the mean and the variance function of the conditioned distribution $\left[\delta(\boldsymbol{x})|\mathbf{Z}_2 = \mathbf{z}_2\right]$:

$$\mu_\delta(\boldsymbol{x}) = f'_\delta(\boldsymbol{x})\boldsymbol{\beta}_\delta + \boldsymbol{r_\delta}'(\boldsymbol{x})\boldsymbol{R}^{-1}_{\delta,\mathbf{D}_2}(\mathbf{z}_2 - \rho z_1(\mathbf{D}_2) - \mathbf{f}'_\delta(\mathbf{D}_2)\boldsymbol{\beta}_\delta)$$

$$\sigma^2_\delta(\boldsymbol{x}) = \sigma^2_\delta(1 - \boldsymbol{r_\delta}'(\boldsymbol{x})\boldsymbol{R}^{-1}_{\delta,\mathbf{D}_2}\boldsymbol{r_\delta}(\boldsymbol{x}))$$

Recursive AR(1) model with 2 levels of code

Multi-fidelity co-kriging model : recursive formulation

- Once again we suppose that all means and covariances parameters (i.e. β_1, β_δ, ρ, σ_1, σ_δ, $R_1(\cdot,\cdot)$ and $R_\delta(\cdot,\cdot)$ are known.
- The conditioned distribution $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$ is Gaussian with mean $\mu_{Z_2}(\boldsymbol{x})$ and variance $\sigma_{Z_2}(\boldsymbol{x})$ done by :

$$\mu_{Z_2}(\boldsymbol{x}) = \rho\mu_{Z_1}(\boldsymbol{x}) + \mu_\delta(\boldsymbol{x})$$

$$\sigma^2_{Z_2}(\boldsymbol{x}) = \rho\sigma^2_{Z_1}(\boldsymbol{x}) + \sigma^2_\delta(\boldsymbol{x})$$

where $\mu_\delta(\boldsymbol{x})$ and $\sigma^2_\delta(\boldsymbol{x})$ are the mean and the variance function of the conditioned distribution $[\delta(\boldsymbol{x})|\mathbf{Z}_2 = \mathbf{z}_2]$:

$$\mu_\delta(\boldsymbol{x}) = f'_\delta(\boldsymbol{x})\boldsymbol{\beta}_\delta + \boldsymbol{r_\delta}'(\boldsymbol{x})\boldsymbol{R}^{-1}_{\delta,\mathbf{D}_2}(\mathbf{z}_2 - \rho z_1(\mathbf{D}_2) - \mathbf{f}'_\delta(\mathbf{D}_2)\boldsymbol{\beta}_\delta)$$

$$\sigma^2_\delta(\boldsymbol{x}) = \sigma^2_\delta(1 - \boldsymbol{r_\delta}'(\boldsymbol{x})\boldsymbol{R}^{-1}_{\delta,\mathbf{D}_2}\boldsymbol{r_\delta}(\boldsymbol{x}))$$

- It can be shown that the original AR(1) model and its recursive formulation have the **same conditioned distribution** $[Z_2(\boldsymbol{x})|\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2]$.

Recursive AR(1) model with s levels of code

The recursive AR(1) is easily generalised

- Easily generalised, for $t = 2, \dots, s$, the model become :

$$\begin{cases} Z_t(\boldsymbol{x}) = \rho_{t-1}(\boldsymbol{x})\tilde{Z}_{t-1}(\boldsymbol{x}) + \delta_t(\boldsymbol{x}) \\ \tilde{Z}_{t-1}(\boldsymbol{x}) \perp \delta_t(\boldsymbol{x}) \end{cases},$$

where $\tilde{Z}_{t-1}(x)$ is a Gaussian process with distribution
$[Z_{t-1}(x)|\mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}, \boldsymbol{\beta}_{t-1}, \rho_{t-2}, \sigma^2_{t-1}]$,
$\delta_t(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_t'(\boldsymbol{x})\boldsymbol{\beta}_t, \sigma_t^2 R_t(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$ is defined in the same way as before
$\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$ are nested designs

The conditioned distribution $[Z_s(x)|\mathbf{Z}_1 = \mathbf{z}_1, \dots \mathbf{Z}_s = \mathbf{z}_s]$ (with all parameters known) is Gaussian with :

$$\mu_{Z_t}(\boldsymbol{x}) = \rho_{t-1}\mu_{Z_{t-1}}(\boldsymbol{x}) + \mu_{\delta_t}(\boldsymbol{x})$$
$$\sigma^2_{Z_t}(\boldsymbol{x}) = \rho_{t-1}\sigma^2_{Z_{t-1}}(\boldsymbol{x}) + \sigma^2_{\delta_t}(\boldsymbol{x})$$

Recursive AR(1) model with s levels of code

The recursive AR(1) is easily generalised

- Easily generalised, for $t = 2, \ldots, s$, the model become :

$$\begin{cases} Z_t(\boldsymbol{x}) = \rho_{t-1}(\boldsymbol{x})\tilde{Z}_{t-1}(\boldsymbol{x}) + \delta_t(\boldsymbol{x}) \\ \tilde{Z}_{t-1}(\boldsymbol{x}) \perp \delta_t(\boldsymbol{x}) \end{cases},$$

where $\tilde{Z}_{t-1}(x)$ is a Gaussian process with distribution
$[Z_{t-1}(x)|\mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}, \boldsymbol{\beta}_{t-1}, \rho_{t-2}, \sigma^2_{t-1}]$,
$\delta_t(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{f}_t'(\boldsymbol{x})\boldsymbol{\beta}_t, \sigma_t^2 R_t(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right)$ is defined in the same way as before
$\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$ are nested designs

The conditioned distribution $[Z_s(x)|\mathbf{Z}_1 = \mathbf{z}_1, \ldots \mathbf{Z}_s = \mathbf{z}_s]$ (with all parameters known) is Gaussian with :

$$\mu_{Z_t}(\boldsymbol{x}) = \rho_{t-1}\mu_{Z_{t-1}}(\boldsymbol{x}) + \mu_{\delta_t}(\boldsymbol{x})$$
$$\sigma^2_{Z_t}(\boldsymbol{x}) = \rho_{t-1}\sigma^2_{Z_{t-1}}(\boldsymbol{x}) + \sigma^2_{\delta_t}(\boldsymbol{x})$$

- The model can also be generalised to the case where the scale parameter ρ depends on $\boldsymbol{x}$. Then it is assumed that $\rho_{t-1}(\boldsymbol{x}) = \mathbf{g}_{t-1}'(x)\boldsymbol{\beta}_{\rho_{t-1}}$.

AR(1) model with 2 levels of code

Advantages of the recursive AR(1) model

- It provides the surrogate models of all the responses $(z_t(x))_{t=1\ldots,s}$.

AR(1) model with 2 levels of code

Advantages of the recursive AR(1) model

- It provides the surrogate models of all the responses $(z_t(x))_{t=1,\dots,s}$.
- Building a s-level co-kriging is equivalent in terms of numerical complexity to build s independent krigings

AR(1) model with 2 levels of code

Advantages of the recursive AR(1) model

- It provides the surrogate models of all the responses $(z_t(x))_{t=1,\ldots,s}$.
- Building a s-level co-kriging is equivalent in terms of numerical complexity to build s independent krigings

  The computational cost is dominated by the inversion of the covariance matrices.

  In the original approach, we have to invert one matrix of size $(n_1+n_2)\times(n_1+n_2)$, in the recursive approach, we have to invert s matrices of size $(n_l \times n_l)_{l=1,\ldots,s}$

AR(1) model with 2 levels of code

Advantages of the recursive AR(1) model

- It provides the surrogate models of all the responses $(z_t(x))_{t=1,\dots,s}$.
- Building a s-level co-kriging is equivalent in terms of numerical complexity to build s independent krigings

  The computational cost is dominated by the inversion of the covariance matrices.

  In the original approach, we have to invert one matrix of size $(n_1+n_2)\times(n_1+n_2)$, in the recursive approach, we have to invert s matrices of size $(n_t \times n_t)_{t=1,\dots,s}$

- The model provides the contribution of the code level $t-1$ to the total conditioned variance at level t, $t=2,\dots,s$ $\rightarrow$ useful for active learning.

AR(1) model with 2 levels of code

Advantages of the recursive AR(1) model

- It provides the surrogate models of all the responses $(z_t(x))_{t=1,\dots,s}$.
- Building a s-level co-kriging is equivalent in terms of numerical complexity to build s independent krigings

  The computational cost is dominated by the inversion of the covariance matrices.

  In the original approach, we have to invert one matrix of size $(n_1+n_2)\times(n_1+n_2)$, in the recursive approach, we have to invert s matrices of size $(n_t \times n_t)_{t=1,\dots,s}$

- The model provides the contribution of the code level $t-1$ to the total conditioned variance at level t, $t=2,\dots,s$ → useful for active learning.
- This formulation makes it possible to obtain Leave-One-Out formulas for the mean and the variance without having to build many models (as in the univariate case).

AR(1) model with 2 levels of code

About the nested design assumption

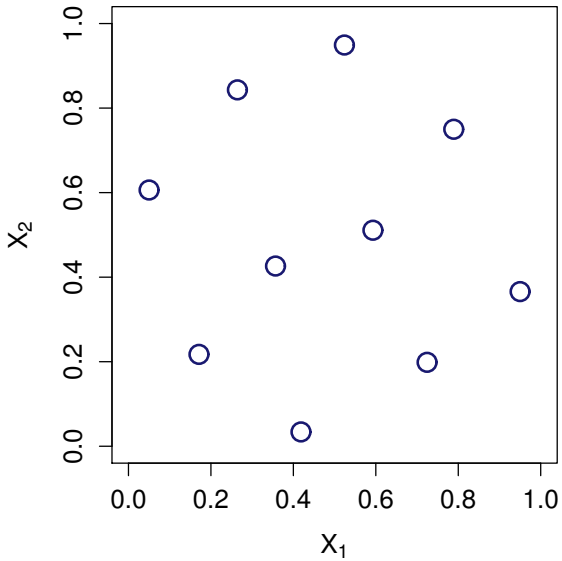
- The **nested property** of the design sets is **not necessary** to build the model, but it allows for a **simple estimation** of the model parameters.
- Exemple of easy construction
  See also : Forrester & al (2007), Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. A
  Qian & al (2009), Construction of nested space-filling designs, The Annals of Statistics, ...

AR(1) model with 2 levels of code

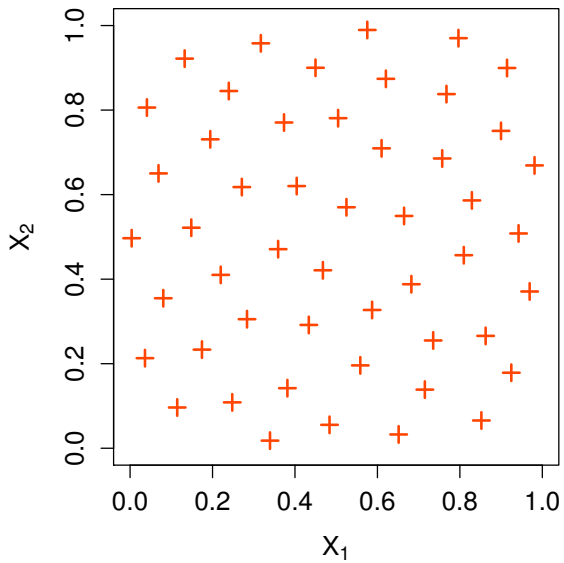
About the nested design assumption

- The **nested property** of the design sets is **not necessary** to build the model, but it allows for a **simple estimation** of the model parameters.
- Exemple of easy construction
  See also : Forrester & al (2007), Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. A
  Qian & al (2009), Construction of nested space-filling designs, The Annals of Statistics, ...

First we build the experimental design set $\mathbf{D}_2$ for the most accurate code $z_2(x)$.



Then we build $\tilde{\mathbf{D}}_1$ the experimental design set from which we will build $\mathbf{D}_1$.

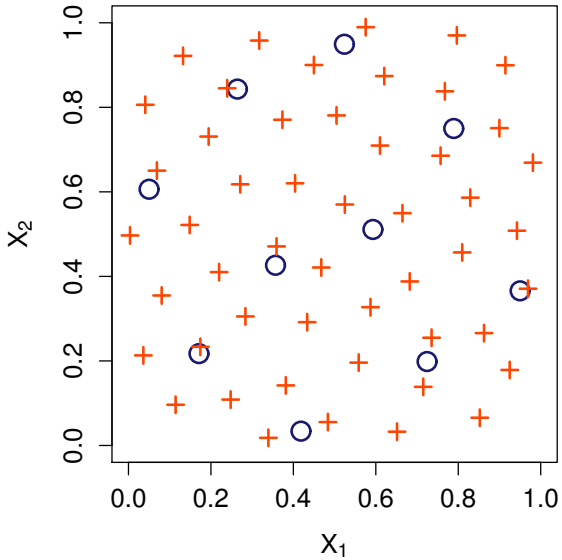


AR(1) model with 2 levels of code

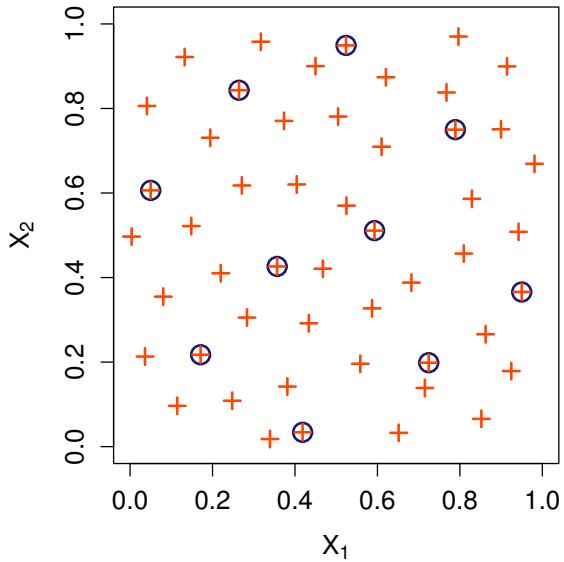
About the nested design assumption

- The **nested property** of the design sets is **not necessary** to build the model, but it allows for a **simple estimation** of the model parameters.
- Exemple of easy construction
  See also : Forrester & al (2007), Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. A
  Qian & al (2009), Construction of nested space-filling designs, The Annals of Statistics, ...

We find the points of $\tilde{\mathbf{D}}_1$ the closest to those of $\mathbf{D}_2$ and we remove them.

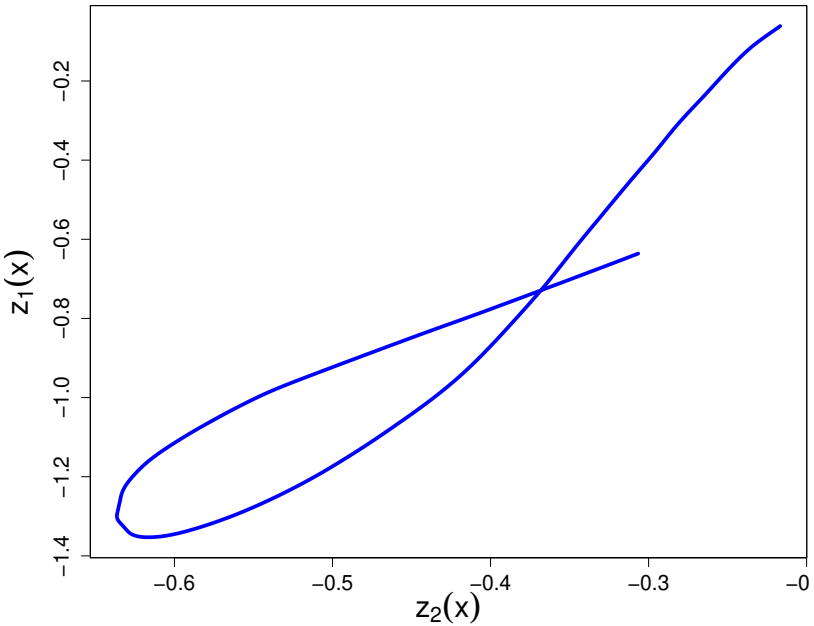


$\mathbf{D}_1$ is built by concatenating $\mathbf{D}_2$ and $\tilde{\mathbf{D}}_1$ without its removed points.



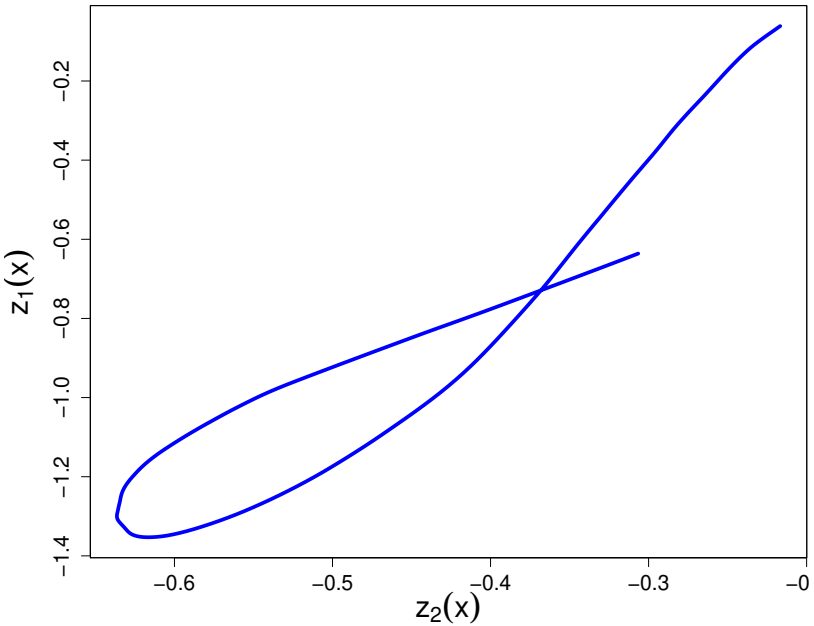
AR(1) model with 2 levels of code

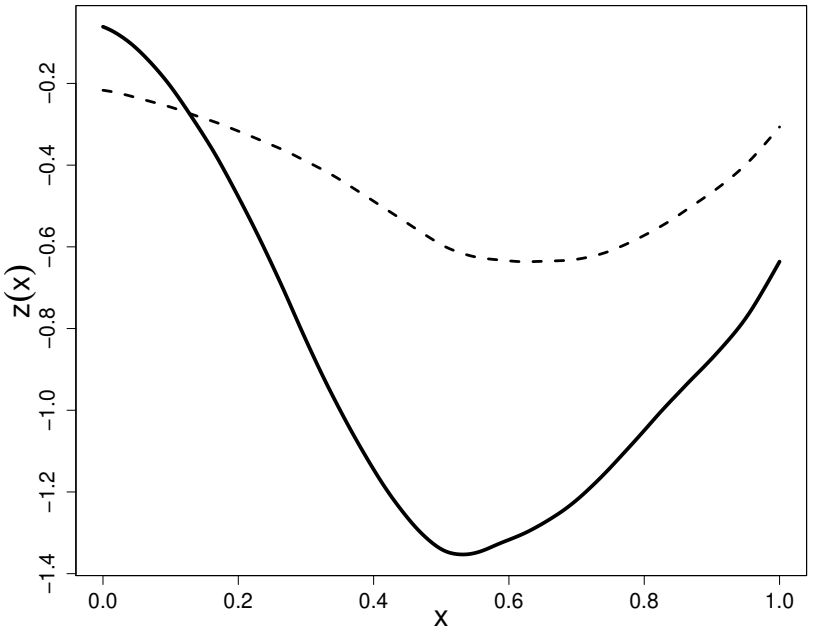
Are the assumptions of the model verified ?



AR(1) model with 2 levels of code

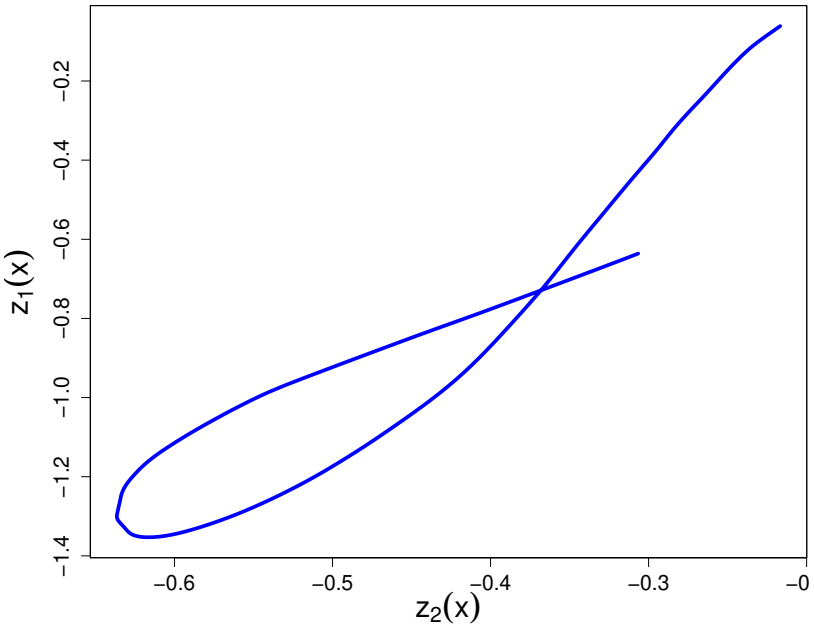
Are the assumptions of the model verified ?

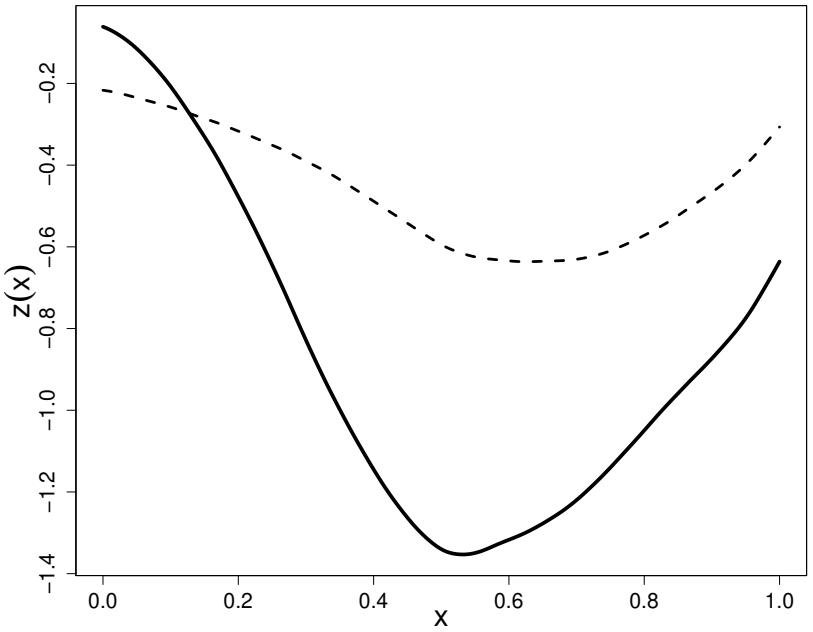




AR(1) model with 2 levels of code

Are the assumptions of the model verified ?

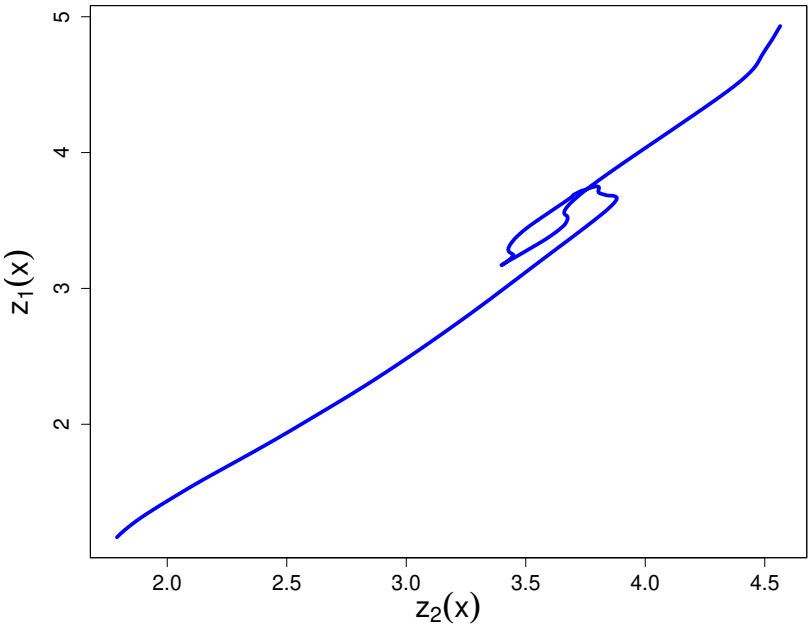




Yes : $\rho = 1$,
$z_1(x)$ is a realization of $\mathcal{GP}$ of mean zero and of covariance Matern $5/2$
$\delta(x)$ is a realization of $\mathcal{GP}$ of mean zero and of covariance Matern $5/2$.

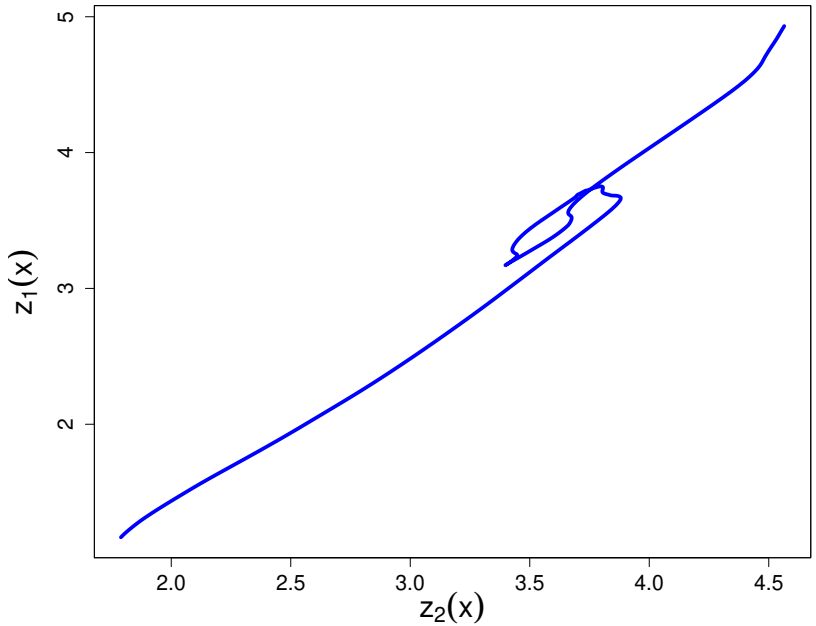
AR(1) model with 2 levels of code

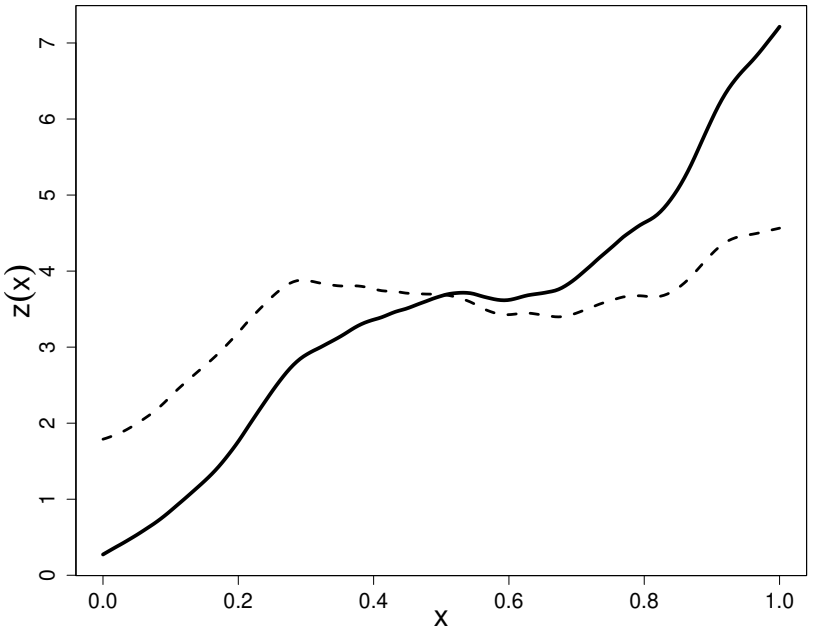
Are the assumptions of the model verified ?



AR(1) model with 2 levels of code

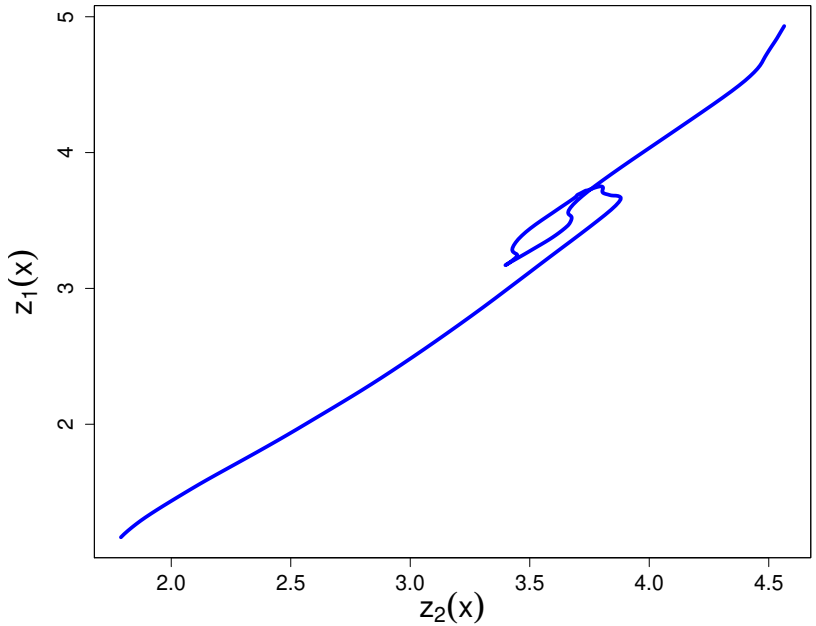
Are the assumptions of the model verified ?

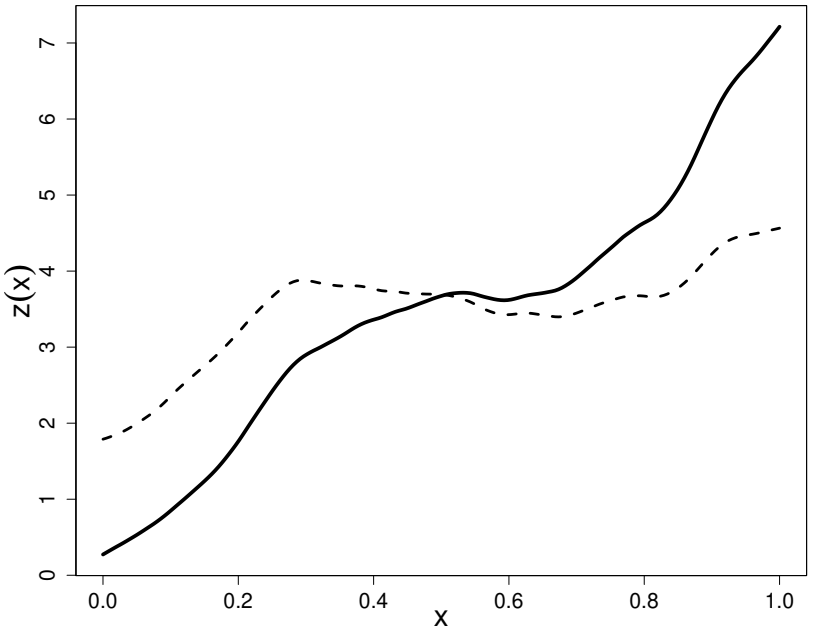




AR(1) model with 2 levels of code

Are the assumptions of the model verified ?

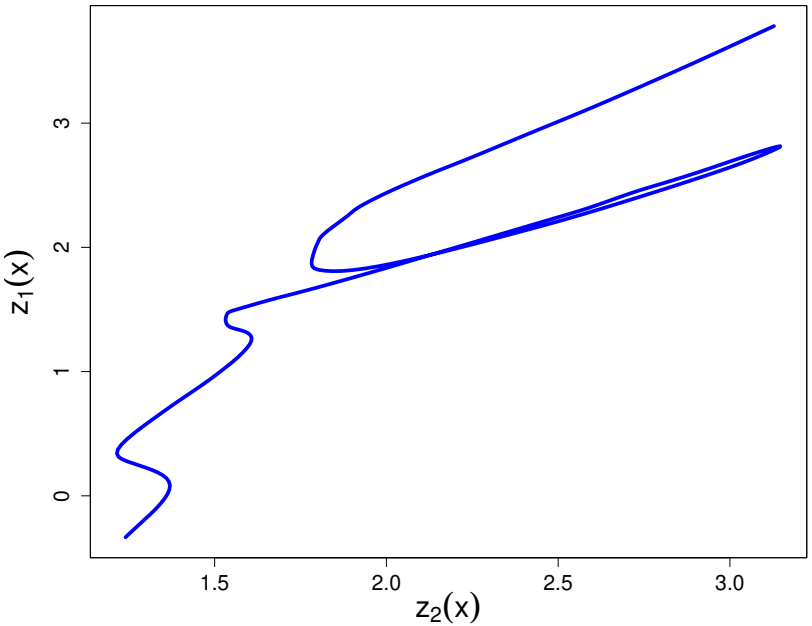




Yes : $\rho = \rho_0 + \rho_1 x$,
$z_1(x)$ is a realization of $\mathcal{GP}$ of mean $\beta_0 + \beta_1 x$ and of covariance Matern $3/2$
$\delta(x)$ is a realization of $\mathcal{GP}$ of mean $\gamma_0 + \gamma_1 x$ and of covariance Matern $3/2$.

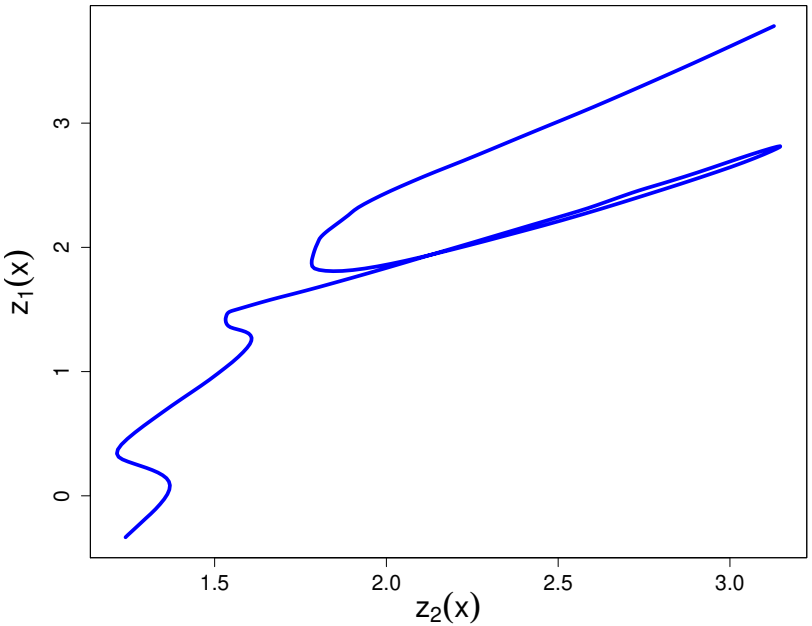
AR(1) model with 2 levels of code

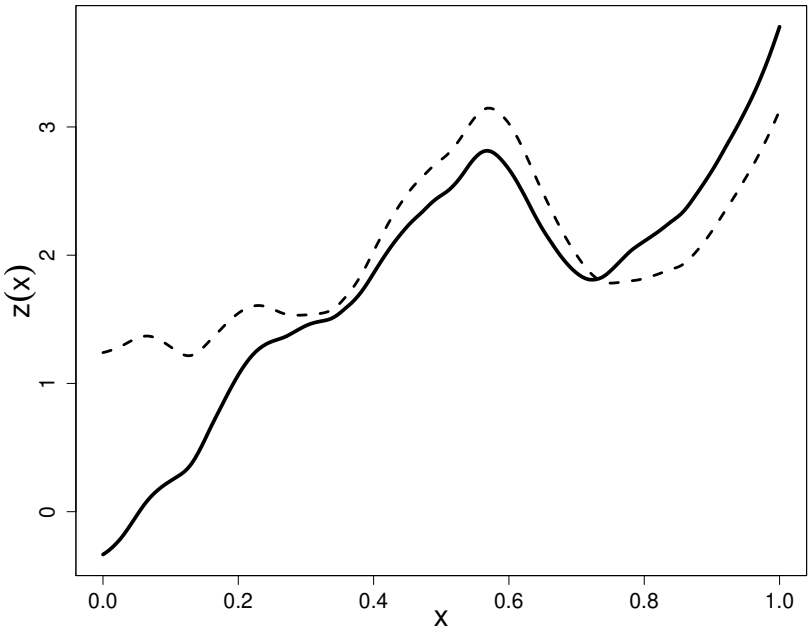
Are the assumptions of the model verified ?



AR(1) model with 2 levels of code

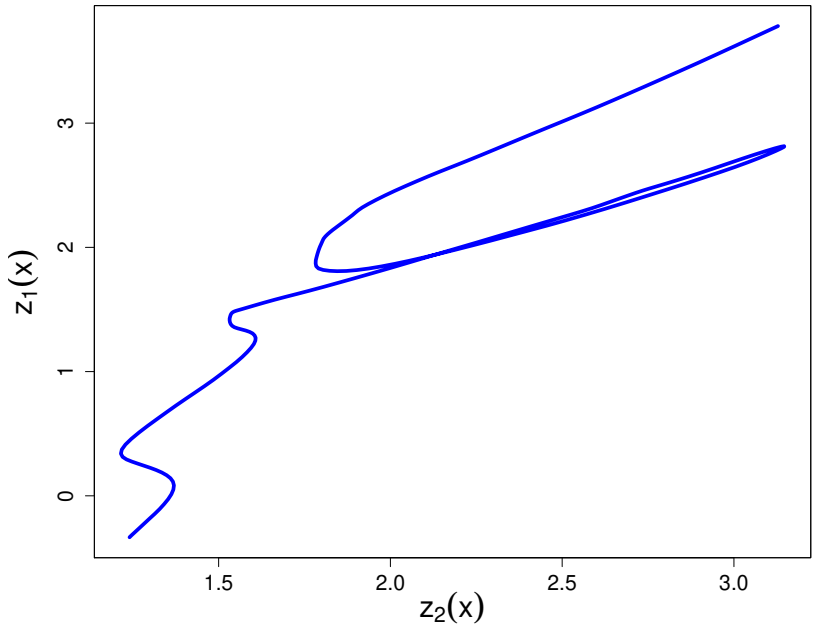
Are the assumptions of the model verified ?

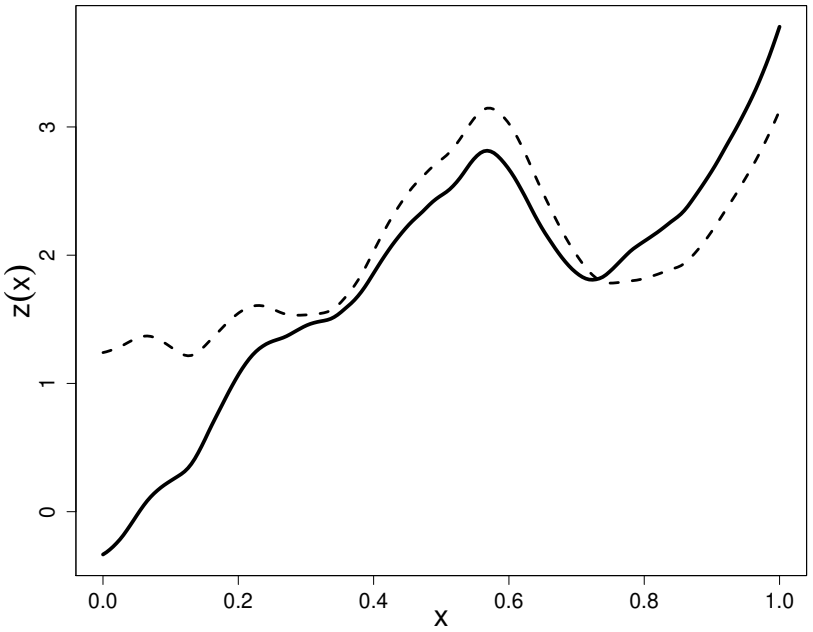




AR(1) model with 2 levels of code

Are the assumptions of the model verified ?

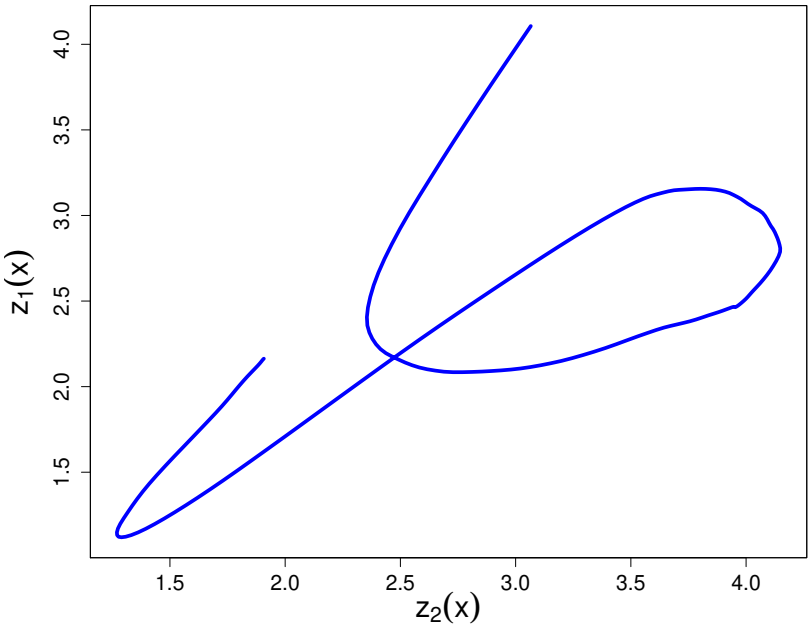




Yes : $\rho = 0.5$,
$z_1(x)$ is a realization of $\mathcal{GP}$ of mean $\beta_0 + \beta_1 x$ and of covariance Matern $5/2$
$\delta(x)$ is a realization of $\mathcal{GP}$ of mean $\gamma_0 + \gamma_1 x$ and of covariance Matern $5/2$.

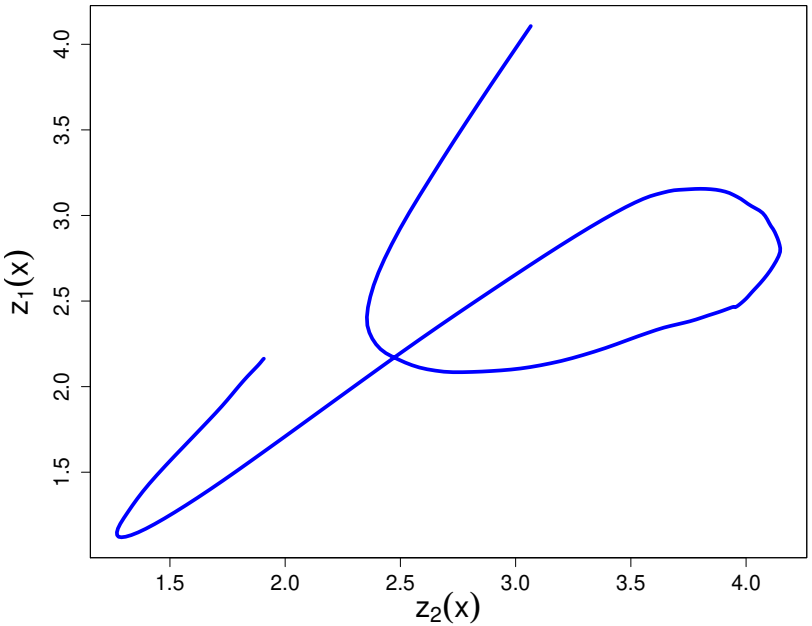
AR(1) model with 2 levels of code

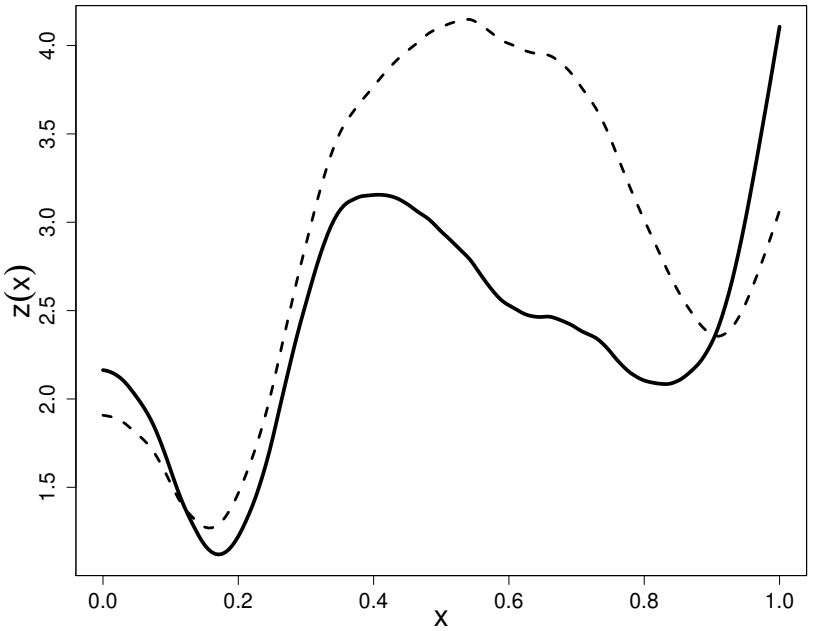
Are the assumptions of the model verified ?



AR(1) model with 2 levels of code

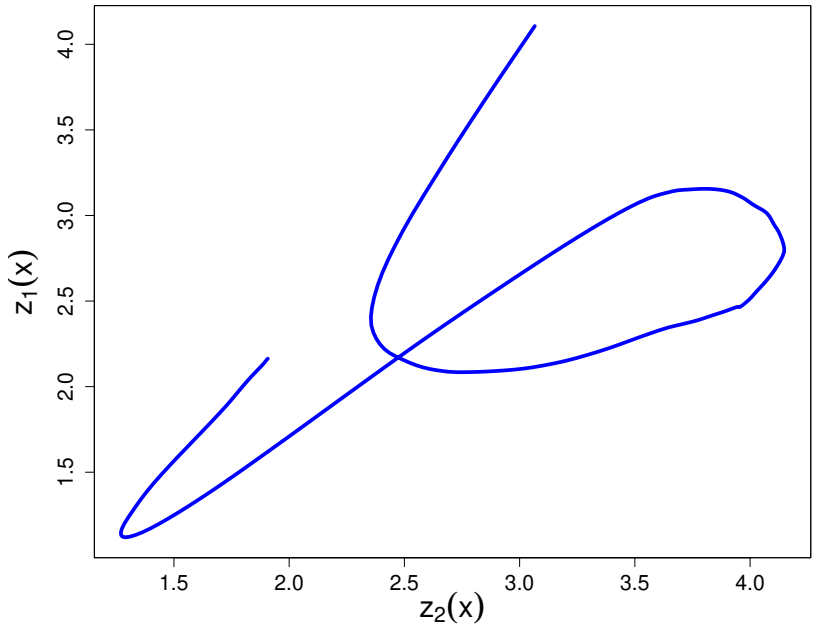
Are the assumptions of the model verified ?

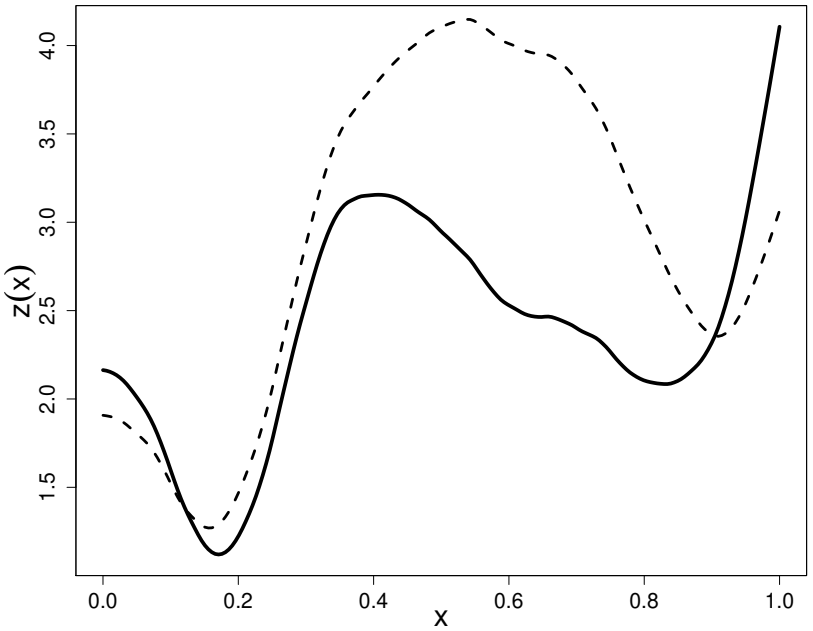




AR(1) model with 2 levels of code

Are the assumptions of the model verified ?





No : $Z_2(x) = \sqrt{Z_1(x)} + \delta(x)$,
$z_1(x)$ is a realization of $\mathcal{GP}$ of mean $\beta_0 + \beta_1 x$ and of covariance Matern $5/2$
$\delta(x)$ is a realization of $\mathcal{GP}$ of mean $\gamma_0 + \gamma_1 x$ and of covariance Matern $5/2$.

AR(1) model with 2 levels of code

Some remarks on AR(1) model

- Despite the assumption of linearity which may seem restrictive, the model is really adapted to many multi-fidelity applications (different models of the same phenomenon)
- However, there are cases where this assumption is not valid and where other methods should be considered (non linear relationship)
- The recursive formulation has allowed it to become numerically lighter and has made it quite popular in recent years.
- It is not designed to be used on a very large number of levels.

  In this case we speak of tunable fidelity, and approaches based on PGR exist.

Picheny, V. and Ginsbourger, D. (2013), A nonstationary space-time Gaussian process model for partially converged simulations, SIAM/ASA Journal on Uncertainty Quantification.

Tuo R. Wu C. F. J. and Yu, D (2014), Surrogate modeling of computer experiments with different mesh densities, Technometrics.

Outline