

Chapitre I

Emulation par processus gaussiens

On présente ici les mathématiques qui fondent les choix et l'utilisation des processus gaussiens dans la méthode d'history matching. Connaître ces mathématiques n'est pas indispensable pour utiliser l'outil mais aide à mieux comprendre le fonctionnement de la méthode et à gagner en confiance sur la robustesse de son utilisation.

Le choix des processus gaussiens comme émulateurs est double. Les processus gaussiens permettent de travailler avec une base d'apprentissage petite, typiquement les valeurs d'une métrique calculée sur une centaine de simulations, et ils fournissent l'incertitude de la prévision statistique faite par l'émulateur.

Cette annexe ne représente pas à proprement parler un résultat de la thèse puisque qu'il ne fait que décrire une approche développée dans la communauté de la quantification des incertitudes (uncertainty quantification ou UQ en anglais), approche utilisée dans le cadre de `htexplo`. En revanche, s'appropriier, dans l'équipe de modélisation du climat, les principes de la construction des émulateurs a été travaillé au cours de ma thèse dans le cadre d'ateliers avec une poignée de collaborateurices.

I.1 Cadre théorique des processus gaussiens

Un processus gaussien est un type de processus stochastique particulier.

I.1.1 Définition d'un processus stochastique

Un processus stochastique est une variable aléatoire de type fonction, appelé aussi fonction aléatoire ou processus aléatoire. Cela signifie qu'une réalisation d'un processus stochastique est une fonction.

Les processus stochastiques F se définissent de manière équivalente par leur famille ou collection $\{F(x_i)\}_{i \in \mathcal{I}}$, où chaque $F(x_i)$ est une variable aléatoire et \mathcal{I} un ensemble d'indices qui peut être fini ou infini (par exemple l'ensemble des entiers naturels \mathbb{N}). Le processus stochastique F est ainsi défini sur la collection de points $\{x_i\}_{i \in \mathcal{I}}$. On note que les variables aléatoires $F(x_i)$ de la collection ne sont pas indépendantes les unes des autres en toute généralité. La loi de probabilité d'une collection $\{F(x_i)\}_{i \in \mathcal{I}}$ est une loi multivariée, qui décrit la distribution de toute la collection (et non pas des variables aléatoire $F(x_i)$ une à une). *[est-ce qu'on peut définir une loi multivariée sur une collection infinie ?]*

[Je trouve bizarre ce début de paragraphe :] L'expérience permettant de construire une réalisation d'un processus stochastique consiste à échantillonner une collection finie selon la loi suivie par le processus stochastique, sur une collection finie de points $\{x_i\}_{i \in \mathcal{I}_f}$. Le résultat est une collection finie notée $\{f(x_i)\}_{i \in \mathcal{I}_f}$, [Je trouve bizarre cette précision : où les $f(x_i)$ ne sont plus des

variables aléatoires.] et \mathcal{I}_f un ensemble d'indice fini inclus dans \mathcal{I} . La fonction réalisée f s'appelle aussi la trajectoire. Celle-ci ne peut être en pratique connue que partiellement, sur une collection finie de points.

Par exemple, on considère une fourmi dont la trajectoire, c'est-à-dire sa position en fonction du temps, est la réalisation d'un processus stochastique noté F . La collection de points $\{x_i\}_{i \in \mathcal{I}}$ correspond alors à un ensemble de temps classés dans l'ordre croissant. $F(x_i)$ est une variable aléatoire de la position de la fourmi au temps x_i . $f(x_i)$ est la réalisation du processus stochastique au temps x_i , c'est-à-dire la position de la fourmi au temps x_i . On voit bien dans ce processus que la position de la fourmi au temps x_{i+1} dépend de la position de la fourmi au temps précédent, donc que les différentes $F(x_i)$ de la collection $\{F(x_i)\}_{i \in \mathcal{I}}$ ne sont pas indépendantes.

I.1.2 Définition d'un processus gaussien

Un processus stochastique G est dit gaussien si toute collection finie de ce processus suit une loi normale multivariée. Un processus gaussien est à valeur dans \mathbb{R} .

La loi normale multivariée est la généralisation de la loi normale à une collection de variables aléatoires.

Soit $Z = \{G(x_i)\}_{i \in [1;N]}$ une collection de taille N suivant une loi normale multivariée. Soit z une réalisation de Z . La densité de probabilité de la collection Z suivant une loi normale multivariée est :

$$p_Z(z) = \frac{\exp^{-\frac{1}{2}(z-\mu)^T \Sigma^{-1}(z-\mu)}}{\sqrt{(2\pi)^N \det(\Sigma)}} \quad (\text{I.1})$$

où μ , l'espérance de la loi normale multivariée, est un vecteur de taille N , et Σ , la matrice de variance covariance de la loi normale multivariée, est inversible et de taille $N \times N$. On note la loi normale multivariée $\mathcal{N}_{ND}(\mu, \Sigma)$, par analogie avec la loi normale classique $\mathcal{N}(\mu, \sigma)$, avec pour différence que σ est l'écart type de la loi normale et Σ la généralisation de la variance pour la loi normale multivariée.

Si Z suit une loi normale multivariée $\mathcal{N}_{ND}(\mu, \Sigma)$, alors chacun de ces éléments $Z_i = G(x_i)$ suit une loi normale $\mathcal{N}(\mu_i, \sigma_i)$. Avec μ_i la $i^{\text{ème}}$ composante du vecteur moyenne μ et σ_i la racine carré du $i^{\text{ème}}$ élément diagonal de la matrice de variance covariance Σ .

Pour que le processus gaussien soit bien défini pour n'importe quelle collection de points où réaliser la fonction, la matrice de covariance doit être définie à partir d'un noyau ou fonction ou opérateur de covariance (*kernel* ou *kernel covariance operator* en anglais). Le noyau, pour une paire de points (x, x') de la collection, est $k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$. Le processus gaussien est donc défini par

- Une fonction moyenne $m(x)$
- Un noyau $k(x, x')$

La loi suivie par un tel processus gaussien est fréquemment notée $\mathcal{GP}(m(\cdot), k(\cdot, \cdot))$ dans la littérature. Toute collection finie de ce processus suit une loi normale multivariée $\mathcal{N}_{ND}(\mu, \Sigma)$ où $\mu_i = m(x_i)$ et $\Sigma_{i,j} = k(x_i, x_j)$.

I.1.3 Apprentissage

Dans l'utilisation des processus gaussiens pour construire des émulateurs, le modèle est "sondé" en une série de points $\{x_i\}_{i \in \mathcal{I}}$ auxquels une observable ou métrique y_i est calculée. Notons : Aucune observation n'entre dans cette phase. Il s'agit d'utiliser ces métriques simulées pour estimer la métrique à un point non sondé.

L'apprentissage va consister en deux choses :

- apprendre la fonction moyenne $m(\cdot)$ et le noyau $k(\cdot, \cdot)$ du processus gaussien.
- utiliser un émulateur qui “passe” (éventuellement pas parfaitement) par les points sondés.

Pour calculer la fonction moyenne, on ne dispose que des valeurs aux points sondés. On pourrait prendre directement y_i . Cependant, en présence de bruit ou de sources d’incertitude, on introduirait à coup sûr un sur ajustement hors de contrôle. On aurait surtout plus rien à apprendre du champ puisqu’on aurait un modèle passant parfaitement par les points sondés.

Deux solutions sont classiquement pratiquées dans la communauté UQ, suscitant de vifs débats entre les tenants de l’une et de l’autre (Daniel Williamson, communication personnelles). Soit on suppose qu’on ne connaît que la moyenne des y_i qu’on utilise pour spécifier $m(\cdot)$ comme une fonction constante. Ou alors, on commence par effectuer une régression classique des valeurs de y_i en fonction de x_i . C’est cette seconde approche qui est utilisée dans `htexplo` comme on l’explique un peu plus loin.

On choisit ensuite une forme de la fonction noyau, qui doit typiquement décroître quand les points auxquels on calcule la métrique s’éloignent. On choisit dans `htexplo` la forme la plus classique à savoir un noyau exponentiel, où la corrélation entre deux points de la collection décroît exponentiellement en fonction de la distance entre les deux points. Apprendre l’émulateur consiste alors en apprendre les distances de décorrélation dans chacune des dimensions de x (chacun des paramètres du modèle constituant le vecteur x pour éviter les confusions).

Un exemple classique d’utilisation de cette approche est le krigeage, du nom du statisticien sud-africain D.-G. Krige. Cette approche, développée à l’origine pour la prospection minière, est également utilisée pour reconstituer des champs météorologiques, notamment des champs de pluies. Dans ces applications, l’espace des paramètres est en général un espace physique horizontale sur lequel on dispose de sondages du sol en un certain nombre de points, ou d’un réseau de pluviomètres sur une surface de quelques dizaines de kilomètres par exemples. Les collections sont alors constituées de point x_i à deux dimensions (x, y) , les deux dimensions d’espace jouant donc ici le rôle des P paramètres d’un modèle de climat. Cette méthode utilise, pour reconstituer le champ de pluie, des processus gaussiens de moyenne nulle et d’opérateur de covariance k défini par l’Equation I.2

$$k(x, x') = \sigma^2 \exp^{-\frac{(x-x')^2 + (y-y')^2}{2l^2}} \quad (\text{I.2})$$

où σ^2 et l sont des hyper paramètres à déterminer. l correspond dans cette formule à une longueur de décorrélation : plus l est grand, plus $k(x, x')$ tend vers 0 pour $x \neq x'$ et moins x et x' sont corrélés. De la même manière dans cette formule, plus x et x' sont éloignés, plus $k(x, x')$ est petit donc moins les points x et x' sont corrélés. Lorsque $x = x'$, $k(x, x) = \sigma^2$, ce qui correspond à la variance de la loi normale suivie par $G(x)$.

I.1.4 Prédiction et incertitude de prédiction

Une fois ce cadre posé, on peut estimer la prédiction et l’incertitude de la prédiction, à savoir l’espérance et la variance de l’émulateur à un point quelconque de l’espace des paramètres, x^* .

Supposons qu’on souhaite prédire, en toute généralité, la réalisation d’un processus gaussien suivant la loi $GP(m(\cdot), k(\cdot, \cdot))$ sur une collection de points $\mathbf{x}_2 = \{x_j\}_{j \in \mathcal{I}_2}$, à laquelle est associé la collection $\mathbf{Z}_2 = \{G(x_i)\}_{i \in \mathcal{I}_2}$. Nous connaissons la réalisation du processus gaussien sur une collection de points $\mathbf{x}_1 = \{x_i\}_{i \in \mathcal{I}_1}$, à laquelle est associée la collection $\mathbf{Z}_1 = \{G(x_i)\}_{i \in \mathcal{I}_1}$ dont les valeurs réalisées sont notées $S = \{S_i\}_{i \in \mathcal{I}_1}$. Cet ensemble S nous a servi de base d’apprentissage des hyperparamètres des opérateurs de moyenne m et de covariance k , qui sont donc entièrement connu à ce stade. Faire cette prédiction signifie calculer l’espérance μ^* et la variance Σ^* de la loi

$G(x|\{G(x_i)\}_{i \in \mathcal{I}_1} = \{S_i\}_{i \in \mathcal{I}_1})$ sur la collection de points \mathbf{x}_2 à laquelle est associée la collection $\mathbf{Z}^* = \{G(x|\{G(x_i)\}_{i \in \mathcal{I}_1} = \{S_i\}_{i \in \mathcal{I}_1})\}$

On note \mathbf{x} la concaténation des collections \mathbf{x}_1 et \mathbf{x}_2 , à laquelle on associe la collection $\mathbf{Z} = \{G(x_i)\}_{i \in \mathcal{I}}$, qui est la concaténation des collections \mathbf{Z}_1 et \mathbf{Z}_2 . \mathbf{Z} est une autre collection du même processus gaussien. Elle suit donc également une loi normale multivariée de moyenne notée μ et de matrice de variance covariance notée Σ .

Si μ_1 et Σ_1 (respectivement μ_2 et Σ_2) sont les moyenne et matrice de covariance de \mathbf{Z}_1 (respectivement \mathbf{Z}_2), alors la moyenne et la matrice de covariance de \mathbf{Z} s'écrivent :

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad (\text{I.3})$$

et

$$\Sigma = \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{pmatrix} \quad (\text{I.4})$$

On cherche donc à déterminer la probabilité de la variable aléatoire “ \mathbf{Z}_2 sachant \mathbf{Z}_1 ”, $\mathbf{Z}^* = \mathbf{Z}_2 | (\mathbf{Z}_1 = S)$. Pour ce faire, on utilise le théorème de Bayes, qui dit que pour deux variables aléatoires A et B , la probabilité de A sachant B , ou probabilité de A sachant que $B = b$, notée $P(A|B = b)$ répond à l'Equation I.5.

$$P(A|B = b) = \frac{P(A \cap B)}{P(B)} \quad (\text{I.5})$$

Ce théorème se généralise aux fonctions de densité de probabilité. On applique l'Equation I.5 sur les densités de probabilités avec $A = \mathbf{Z}_2$, $B = \mathbf{Z}_1$, b la réalisation de \mathbf{Z}_1 égale S et $A \cap B = \mathbf{Z}$.

$$p_{\mathbf{Z}^*}(z) = p(\mathbf{Z}_2 | \mathbf{Z}_1 = S)(z) = \frac{p_{\mathbf{Z}}(z)}{p_{\mathbf{Z}_1}(z)} \quad (\text{I.6})$$

Les densités de probabilités des lois normales multivariées. \mathbf{Z}_1 et \mathbf{Z} sont données par l'Equation I.1, soit plus précisément :

$$p_{\mathbf{Z}}(z) = \frac{\exp^{-\frac{1}{2}(z-\mu)^T \Sigma^{-1}(z-\mu)}}{\sqrt{(2\pi)^N \det(\Sigma)}} \text{ et } p_{\mathbf{Z}_1}(z_1) = \frac{\exp^{-\frac{1}{2}(z_1-\mu_1)^T \Sigma_1^{-1}(z_1-\mu_1)}}{\sqrt{(2\pi)^{N_1} \det(\Sigma_1)}} \quad (\text{I.7})$$

On substitue $p_{\mathbf{Z}}(z)$ et $p_{\mathbf{Z}_1}(z_1)$ par leurs expressions de l'Equation I.7 dans l'Equation I.6 [Pourquoi : et sachant que $z_1 = S$?] et on aboutit à

$$p_{\mathbf{Z}^*}(z) = \frac{\exp^{-\frac{1}{2}[(z-\mu)^T \Sigma^{-1}(z-\mu) - (z_1-\mu_1)^T \Sigma_1^{-1}(z_1-\mu_1)]}}{\sqrt{(2\pi)^{N_2} \det(\Sigma) / \det(\Sigma_1)}} \quad (\text{I.8})$$

Une dérivation mathématique relativement simple permet de réécrire cette distribution de probabilité comme

$$p_{\mathbf{Z}^*}(z) = \frac{\exp^{-\frac{1}{2}(z-\mu^*)^T \Sigma^{*-1}(z-\mu^*)}}{\sqrt{(2\pi)^{N_2} \det(\Sigma^*)}} \quad (\text{I.9})$$

avec :

$$\mu^* = \mu_2 + \Sigma_{21} \Sigma_1^{-1} (S - \mu_1) \quad (\text{I.10})$$

et

$$\Sigma^* = \Sigma_2 - \Sigma_{21} \Sigma_1^{-1} \Sigma_{12} \quad (\text{I.11})$$

ce qui montre que \mathbf{Z}^* suit également une loi normale multivariée d'espérance μ^* et de matrice de variance covariance Σ^* définies respectivement par l'Equation I.10 et l'Equation I.11. μ^* est ainsi la prédiction ou l'émulation sur la collection de point \mathbf{x}_2 et Σ^* la matrice de variance covariance associée à cette prédiction.

I.2 Mise en œuvre dans htexplo

Dans la méthode d'history matching, on travaille métrique par métrique. On note g cette métrique définie sur un espace Ω de vecteur de paramètres λ . Pour chaque métrique, on connaît un ensemble de N réalisations (N nombres réels) calculées sur les résultats de N simulations réalisées avec le modèle de climat sur N vecteurs λ tirés aléatoirement dans Ω . On prend en général $N = 10 \times P$ où P est le nombre de paramètres. Dans le processus d'HMIR, Ω est l'hyper-cube initial Λ ou l'espace NROY de la vague précédente.

Pour chaque métrique, on construit indépendamment un émulateur par processus gaussien sur l'espace hypercubique des paramètres de tuning Λ . On apprend pour cette métrique les hyper paramètres d'un processus gaussien $G = GP(m(\cdot), k(\cdot, \cdot))$ sur l'ensemble des N réalisations de la métrique vues comme une réalisation (une collection) du processus gaussien, appelée S (étape 2). Toute la subtilité dans la définition des processus gaussien réside dans la définition des opérateurs de moyenne et de covariance et dans leur méthode d'apprentissage. Les choix faits pour la méthode d'history matching utilisée dans l'outil **htexplo** sont résumés dans Couvreur et al. (2021) et détaillés entre autre dans Williamson et al. (2017), Williamson et al. (2013) et Volodina and Challenor (2021).

L'opérateur moyenne est vu comme un terme de régression sur l'ensemble d'apprentissage, qui rend compte des relations entre les paramètres et la métrique sur tout l'espace des paramètres. Il est défini comme une combinaison linéaire de fonctions simple des vecteurs de paramètres, notées $\mathbf{h}(\lambda)$, ayant comme hyper-paramètres les poids de la combinaison linéaire notés β . L'opérateur moyenne s'exprime selon l'Equation I.12.

$$m(\lambda, \beta) = \beta^T \mathbf{h}(\lambda) \quad (\text{I.12})$$

Les fonctions $\mathbf{h}(\lambda)$ sont par exemple des monômes, des fonctions de Fourier et des fonctions d'interactions entre les paramètres [*je ne sais pas exactement ce qu'on utilise dans htexplo*]. Pour déterminer ces fonctions, une "stepwise selection method" est utilisé, avec une "forwards selection and backwards elimination" des fonctions, basée sur leur capacité à réduire le résidu de la régression et à expliquer la variabilité [*je crois que c'est mal dit*]. La méthode de sélection des fonctions est détaillée dans Williamson et al. (2013).

On utilise un kernel k exponentiel ou opérateur de covariance k dont les hyper paramètres appris sur les résidus par rapport à la moyenne m . Autrement dit, ce noyau quantifie la corrélation des résidus entre deux points λ et λ' de l'espace des paramètres :

$$\tilde{k}(\lambda, \lambda') = k(\lambda, \lambda', \delta, \sigma, \nu) = \sigma^2 \left[\nu \mathbb{1}_{\lambda=\lambda'} + (1 - \nu) \exp \left\{ - \sum_{i=1}^N \left(\frac{\lambda_i - \lambda'_i}{\delta_i} \right)^2 \right\} \right] \quad (\text{I.13})$$

Le premier terme ν est le terme pépité ou *nugget* en anglais. Ce terme rend compte d'une part d'incertitude irréductible qui peut faire que deux simulations entre des points infiniment proches puissent prendre des valeurs différentes. Dans les applications climatiques, on peut l'utiliser par exemple pour tenir compte de la proportion de variabilité interne qui fait que deux simulations peuvent aboutir à des valeurs de métriques différentes pour les mêmes valeurs de paramètres si on change l'état initial de la simulation.

Le deuxième terme est exprimé comme fonction exponentielle puissance (ici puissance 2) de la distance entre les deux paramètres λ et λ' , dépendant du paramètre δ . δ_i est la longueur de décorrélation dans la direction i , c'est-à-dire la distance à partir de laquelle les variabilités résiduelles des points λ et λ' ne sont plus corrélés pour le $i^{\text{ème}}$ paramètres sondé [*à la distant δ_i la correlation est de $\exp(-1) = 0.36$: me débrouiller avec cette infor pour mettre une image*]. [Je n'en ressens pas l'utilité là. Mais c'est toi qui vois Maëlle] On fait l'hypothèse dans cette formulation

que la décorrélation des variabilité résiduelles ne dépend que de la distance entre les vecteurs de paramètres, et non pas de la valeur des vecteurs eux mêmes. On dit dans ce cas que l'opérateur de covariance est stationnaire.

La construction des émulateurs consiste à apprendre les fonctions $h(\boldsymbol{\lambda})$ et les hyperparamètres β , δ , σ et ν . Ces hyperparamètres sont appris en utilisant l'algorithme de Monte Carlo hamiltonien implémenté dans le programme Stan Carpenter et al. (2017).

Une fois les émulateurs construits, on souhaite prédire la valeur de la métrique en un nouveau point $\boldsymbol{\lambda}^*$. On utilise alors les équations I.10 et I.11 avec les modèles de moyenne et de covariance définis par les équations I.12 et I.13. On considère une collection de $N+1$ vecteurs de paramètres, où les N premiers vecteurs sont les points sondés et le dernier est le vecteur de paramètres auquel on souhaite estimer la métrique.

La matrice Σ_1 est donc de dimension $N \times N$ dont les éléments sont fixés après la phase d'apprentissage des hyper-paramètres :

$$\Sigma_1 = \begin{pmatrix} \tilde{k}(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_1) & \tilde{k}(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) & \cdot & \cdot & \cdot & \tilde{k}(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_N) \\ \tilde{k}(\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_1) & \tilde{k}(\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_2) & & & & \tilde{k}(\boldsymbol{\lambda}_2, \boldsymbol{\lambda}_N) \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \tilde{k}(\boldsymbol{\lambda}_N, \boldsymbol{\lambda}_1) & \tilde{k}(\boldsymbol{\lambda}_N, \boldsymbol{\lambda}_2) & \cdot & \cdot & \cdot & \tilde{k}(\boldsymbol{\lambda}_N, \boldsymbol{\lambda}_N) \end{pmatrix} \quad (\text{I.14})$$

Les termes non diagonaux de la matrice Σ sont ici des vecteurs

$$\Sigma_{12}(\boldsymbol{\lambda}^*) = \Sigma_{21}(\boldsymbol{\lambda}^*)^T = \begin{pmatrix} \tilde{k}(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}^*) \\ \tilde{k}(\boldsymbol{\lambda}_2, \boldsymbol{\lambda}^*) \\ \cdot \\ \cdot \\ \cdot \\ \tilde{k}(\boldsymbol{\lambda}_N, \boldsymbol{\lambda}^*) \end{pmatrix} \quad (\text{I.15})$$

et le dernier élément de la matrice est

$$\Sigma_2 = \tilde{k}(\boldsymbol{\lambda}^*, \boldsymbol{\lambda}^*) = \sigma^2 \quad (\text{I.16})$$

La prédiction (espérance de l'émulateur) au point $\boldsymbol{\lambda}^*$ est notée μ^* et l'incertitude σ^* avec :

$$\mu^*(\boldsymbol{\lambda}^*) = m^*(\boldsymbol{\lambda}^*, \boldsymbol{\beta}) = m(\boldsymbol{\lambda}^*, \boldsymbol{\beta}) + \Sigma_{21}(\boldsymbol{\lambda}^*) \Sigma_1^{-1} \begin{pmatrix} S_1 - m(\boldsymbol{\lambda}_1, \boldsymbol{\beta}) \\ \dots \\ S_{10N} - m(\boldsymbol{\lambda}_{10N}, \boldsymbol{\beta}) \end{pmatrix} \quad (\text{I.17})$$

et

$$\sigma^{*2}(\boldsymbol{\lambda}^*) = \sigma^2 - \Sigma_{21}(\boldsymbol{\lambda}^*) \Sigma_1^{-1} \Sigma_{12}(\boldsymbol{\lambda}^*) \quad (\text{I.18})$$

On peut regarder ces formules pour deux cas extrêmes.

Si le point $\boldsymbol{\lambda}^*$ auquel on estime l'émulateur est un des points sondés, c'est à dire si $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_j$, alors Σ_{12} est la colonne j de Σ_1 , c'est à dire que $\Sigma_{12}(\boldsymbol{\lambda}_j) = \Sigma_1 \mathbf{e}_j$ où \mathbf{e}_j est le vecteur unitaire, de composantes nulles ou nulle sur la ligne j . On a donc $\Sigma_1^{-1} \Sigma_{12}(\boldsymbol{\lambda}_j) = \Sigma_1^{-1} \Sigma_1 \mathbf{e}_j = \mathbf{e}_j$. Donc $\Sigma_{21}(\boldsymbol{\lambda}_j) \Sigma_1^{-1} \Sigma_{12}(\boldsymbol{\lambda}_j) = \tilde{k}(\boldsymbol{\lambda}_j, \boldsymbol{\lambda}_j)$. De la même façon, on a $\Sigma_{12}(\boldsymbol{\lambda}_j) \Sigma_1^{-1} = \Sigma_{21}(\boldsymbol{\lambda}_j)^T \Sigma_1^{-1} = \mathbf{e}_j^T \Sigma_1^T \Sigma_1^{-1} = \mathbf{e}_j^T$ par symétrie de la matrice Σ_1 .

En l'absence de nugget, i. e. si en plus $\nu = 0$ (ou si λ^* est strictement égal à λ_k), on a $\tilde{k}(\lambda_j, \lambda_j) = \sigma^2$, l'émulateur passe par les points sondés

$$\mu^*(\lambda^* = \lambda_j) = S_j \quad (\text{I.19})$$

et

$$\sigma^{*2}(\lambda^* = \lambda_j) = 0 \quad (\text{I.20})$$

En présence de nugget, i. e. si en plus $\nu \neq 0$ et si λ^* est infiniment près de λ_j mais pas strictement, de sorte que $\mathbb{1}_{\lambda=\lambda'} = 0$ et donc $\tilde{k}(\lambda_j, \lambda_j) \simeq (1 - \nu)\sigma^2$, l'incertitude de l'émulateur est

$$\sigma^{*2}(\lambda^* \simeq \lambda_j) = \nu\sigma^2 \quad (\text{I.21})$$

Sachant que la probabilité de tirer $\lambda^* = \lambda_j$ strictement est nulle, c'est de fait aussi l'incertitude de l'émulateur en λ_j en présence de nugget.

Avec ou sans nugget, si le point λ^* est très loin de tous les points sondés (comparativement aux hyper-paramètres δ_i), alors l'incertitude de l'émulateur est

$$\sigma^{*2}(\lambda^* = \lambda_j) = \sigma^2 \quad (\text{I.22})$$

A la fin de l'étape de construction des émulateurs, le test standard Leave One Out (LOO) est réalisé afin de vérifier les bonnes capacités de prédiction des émulateurs (voir par exemple Rougier et al. (2009)). Ce test consiste à enlever un point λ_j de l'ensemble d'apprentissage et à évaluer la prédiction $\mu^*(\lambda_j)$ et l'incertitude autour de sa prédiction $\sigma^*(\lambda_j)$.

En pratique, le test LOO utilisé en standard dans l'outil `htexplo` n'est que partiel en ce sens qu'on ne réapprend pas à chaque fois le modèle linéaire ni les hyper-paramètres du processus gaussien. Seules les équations I.17 et I.18 sont évaluées sans le point λ_j .

La distribution de l'émulateur suivant une loi normale on a donc 95% de chance que notre prédiction soit dans l'intervalle $[\mu^*(\lambda_j) - 2\sigma^*(\lambda_j), \mu^*(\lambda_j) + 2\sigma^*(\lambda_j)]$. Pour tester le bon comportement du modèle, on calcule les statistiques pour chaque point de l'ensemble d'apprentissage, en enlevant à chaque fois ce seul point de l'ensemble. Si les émulateurs ont bien été construits et ont des bonnes capacités prédictives, alors environ 5% des points vont se retrouver en dehors de l'intervalle $[\mu^*(\lambda_j) - 2\sigma^*(\lambda_j), \mu^*(\lambda_j) + 2\sigma^*(\lambda_j)]$. Des exemples de figures LOO sorties automatiquement par l'outil `htexplo` sont montrées à plusieurs endroits dans le document. [\[Pointer sur les figures du texte principal\]](#)

Bibliographie

- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan : A Probabilistic Programming Language. *Journal of Statistical Software*, 76 :1–32.
- Couvreur, F., Hourdin, F., Williamson, D., Roehrig, R., Volodina, V., Villefranque, N., Rio, C., Audouin, O., Salter, J., Bazile, E., Brient, F., Favot, F., Honnert, R., Lefebvre, M.-P., Madeleine, J.-B., Rodier, Q., and Xu, W. (2021). Process-Based Climate Model Development Harnessing Machine Learning : I. A Calibration Tool for Parameterization Improvement. *Journal of Advances in Modeling Earth Systems*, 13(3) :e2020MS002217.
- Rougier, J., Sexton, D. M. H., Murphy, J. M., and Stainforth, D. (2009). Analyzing the Climate Sensitivity of the HadSM3 Climate Model Using Ensembles from Different but Related Experiments. *Journal of Climate*, 22(13) :3540–3557.
- Volodina, V. and Challenor, P. (2021). The importance of uncertainty quantification in model reproducibility. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 379(2197) :20200071.
- Williamson, D., Blaker, A. T., and Sinha, B. (2017). Tuning without over-tuning : Parametric uncertainty quantification for the NEMO ocean model. *Geoscientific Model Development*, 10(4) :1789–1816.
- Williamson, D., Goldstein, M., Allison, L., Blaker, A., Challenor, P., Jackson, L., and Yamazaki, K. (2013). History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Climate Dynamics*, 41(7) :1703–1729.