

# OpenTURNS Developer Training

## Probabilistic uncertainty propagation

Trainer : Régis LEBRUN  
Airbus  
regis.lebrun@airbus.com

Developers training



# Probabilistic uncertainty propagation

- 1 What is uncertainty propagation?
- 2 Numerical models: the differential calculus potion
- 3 A cloud of probabilities in my code, please

# Main objective

## Probabilistic uncertainty propagation

OpenTURNS = Open Source Treatment of Uncertainty, Risk'N Statistics

- **Uncertainty** = unknown quantities, lack of exact knowledge, non predictable fluctuations
- **Risk** = dangerous state, critical conditions and their impact (cost, consequences)
- **Statistics** = observation and modelling of random quantities, partial knowledge
- **Treatment** = algorithmic tools to analyse, model and quantify the previous points

# Main objective

## UQ in a nutshell

Given:

- a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$  called a **model**
- a random vector  $X$  of known joint distribution  $\mathcal{L}$  of dimension  $n$  called the **sources of uncertainties**

Uncertainty quantification is focused on getting information about the **push-forward** distribution of the random vector  $Y = f(X)$ :

- Moments  $E[h(Y)]$  where  $h : \mathbb{R}^p \rightarrow \mathbb{R}^q$
- Probability of events  $\mathbb{P}(Y \in B)$  as a special case ( $h(y) = 1_{y \in B}$ )
- Sensitivity of  $Y$  to part of  $X$

# Numerical models

## Function, gradient, hessian

- **Function:** the notion of **numerical model** is a function  $f$  maps  $\mathbb{R}^n$  to  $\mathbb{R}^p$ . A shortcut is to say that:

$$f \in \mathcal{F}(\mathbb{R}^n, \mathbb{R}^p) \quad (1)$$

The integers  $n \in \mathbb{N}$ ,  $p \in \mathbb{N}^*$  are the **input** and **output** dimensions. The set  $\{\underline{x} \in \mathbb{R}^n | f(\underline{x}) \text{ is well defined}\}$  is the **domain of definition** of  $f$ . For all  $i \in \{1, \dots, p\}$ , the function  $f_i \in \mathcal{F}(\mathbb{R}^n, \mathbb{R})$  defined by  $f_i(\underline{x}) = \pi_i(f(\underline{x}))$ , where  $\pi_i$  is the projection on the  $i^{\text{th}}$  coordinate in  $\mathbb{R}^p$ , is called the  **$i^{\text{th}}$  component** of  $f$ .

- **Gradient:** a function  $f \in \mathcal{F}(\mathbb{R}^n, \mathbb{R}^p)$  is said to be **differentiable** at  $\underline{x} \in \mathbb{R}^n$  if one has:

$$\forall \underline{h} \in \mathbb{R}^n, f(\underline{x} + \underline{h}) = f(\underline{x}) + D(f)(\underline{x})(\underline{h}) + o(\|\underline{h}\|) \quad (2)$$

where  $D(f)(\underline{x})$  is a (continuous) linear application from  $\mathbb{R}^n$  to  $\mathbb{R}^p$ :

$D(f)(\underline{x}) \in L_c(\mathbb{R}^n, \mathbb{R}^p)$ . The application  $D(f)$  that maps  $\underline{x} \in \mathbb{R}^n$  into  $L_c(\mathbb{R}^n, \mathbb{R}^p)$  is the **differential** of  $f$ . The linear application  $D(f)(\underline{x})$  is always continuous in the setting of  $\mathbb{R}^n$  and has an associated matrix  $\underline{\underline{M}}(\underline{x}) \in \mathcal{M}_{n,p}(\mathbb{R})$  with  $M_{ij} = \frac{\partial f_i}{\partial x_j}$ : it is the **jacobian matrix** of  $f$  at  $\underline{x}$ . The application that maps  $\underline{x}$  into  $\underline{\underline{M}}^t(\underline{x}) \in \mathcal{M}_{p,n}(\mathbb{R})$  is the **gradient** of  $f$  at  $\underline{x}$ .

# Numerical models

## Function, gradient, hessian

- **Hessian**: a function  $f \in \mathcal{F}(\mathbb{R}^n, \mathbb{R}^p)$  is said to be **twice differentiable** at  $\underline{x} \in \mathbb{R}^n$  if the application  $D(f)$  that maps  $\mathbb{R}^n$  into  $L_c(\mathbb{R}^n, \mathbb{R}^p)$  is differentiable at  $\underline{x}$ . It means that:

$$\forall \underline{h} \in \mathbb{R}^n, D(f)(\underline{x} + \underline{h}) = D(f)(\underline{x}) + D^2(f)(\underline{x})(\underline{h}) + o(\|\underline{h}\|) \quad (3)$$

where  $D^2(f)(\underline{x})$  is a (continuous) linear application from  $\mathbb{R}^n$  to  $L_c(\mathbb{R}^n, \mathbb{R}^p)$ :  $D^2(f)(\underline{x}) \in L_c(\mathbb{R}^n, L_c(\mathbb{R}^n, \mathbb{R}^p))$ . The application  $D^2(f)$  that maps  $\underline{x} \in \mathbb{R}^n$  into  $L_c(\mathbb{R}^n, L_c(\mathbb{R}^n, \mathbb{R}^p))$  is the **second differential** of  $f$ . The linear application  $D^2(f)(\underline{x})$  is always continuous in the setting of  $\mathbb{R}^n$  and has a tensor representation

$\underline{\underline{T}}(\underline{x}) \in \mathcal{T}_{n,n,p}(\mathbb{R})$  with  $M_{ijk} = \frac{\partial^2 f_i}{\partial x_j \partial x_k}$ : it is the **second jacobian tensor** of  $f$  at  $\underline{x}$ . The application that maps  $\underline{x}$  into  $\underline{\underline{T}}^t(\underline{x}) \in \mathcal{T}_{p,n,n}(\mathbb{R})$  is the **hessian tensor** of  $f$  at  $\underline{x}$ . This tensor is made of sheets  $T_{k,..} \in \mathcal{M}_{n,n}(\mathbb{R})$  that are **symmetric**.

## Random vector, distribution

### Definition

- A **random vector**  $\underline{X}$  is a measurable function from a probability space  $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$  into the probability space  $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n), \mu_{\underline{X}})$ .
- The associated **distribution** is the probability measure  $\mu_{\underline{X}}$  defined by:

$$\forall B \in \mathcal{B}(\mathbb{R}), \mu_{\underline{X}}(B) = \mathbb{P}(\underline{X}^{-1}(B)) \quad (4)$$

- The **main advantage** of a random vector is that we are now working on a **numeric space**  $\mathbb{R}^n$  instead of a general set  $\Omega$ .
- The different values  $\forall \omega \in \Omega, \underline{X}(\omega)$  taken by a random vector  $\underline{X}$  are called the **realizations** of the random vector.
- A distribution is completely defined by its **cumulative distribution function or CDF**  $F_{\underline{X}}$  which maps  $\mathbb{R}^n$  into  $[0, 1]$  and is defined by:

$$F_{\underline{X}}(\underline{x}) = \mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n) \quad (5)$$

# Random vector, distribution

## Discrete random vectors, continuous random vectors

There are two distinguished classes of random vectors:

- Those that take their values in  $\mathbb{N}^n$  and are called **discrete integral** random vectors. The distribution of this kind of random vectors is equivalently described by the function  $p_{\underline{X}}$  that maps  $\mathbb{N}^n$  into  $[0, 1]$  such that:

$$p_{\underline{X}}(\underline{x}) = \mathbb{P}(\underline{X} = \underline{x}) \quad (6)$$

The function  $p_{\underline{X}}$  is called its **probability distribution function or PDF**.

- Those such that there exist a function  $p_{\underline{X}}$  that maps  $\mathbb{R}^n$  into  $R^+$  such that:

$$F_{\underline{X}}(\underline{x}) = \int_{\mathbb{R}^n} p_{\underline{X}}(\underline{x}) d\underline{x} \quad (7)$$

These random vectors are called **absolutely continuous** random vectors with respect to the Lebesgue measure  $d\underline{x}$ . The function  $p_{\underline{X}}$  is called its **probability density function or PDF**.

**WARNING 1:** The PDF acronym is used for two distinct functions, but the context makes it clear which kind of PDF is relevant in practical applications.

**WARNING 2:** A random vector can be neither discrete nor continuous!



# PDF and CDF, 1D case

Continuous



Discrete

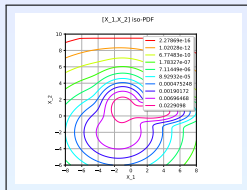


General



## PDF and CDF, 2D case

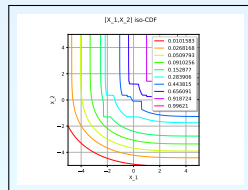
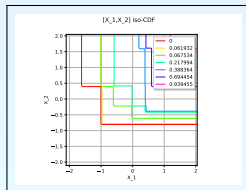
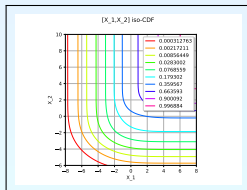
Continuous



Discrete



General



## Expectation, mean, covariance

### Definition

- The **expectation**  $\mathbb{E}[\underline{X}]$  of a real random vector  $\underline{X}$  of dimension  $n$  is the deterministic vector of  $\mathbb{R}^n$  defined by:

$$\mathbb{E}[\underline{X}]_i = \int_{\mathbb{R}} x_i d\mu_{X_i}(x_i) \quad (8)$$

where  $\mu_{X_i}$  is the distribution of the 1D random vector corresponding to the projection of  $\underline{X}$  on its  $i^{th}$  coordinate. It is thus a quantity that is defined component by component. For bell shaped distributions, it is an indication of the location of most of the realizations of  $\underline{X}$ .

- The **covariance**  $\text{Cov}[\underline{X}]$  of a real random vector  $\underline{X}$  of dimension  $n$  is the deterministic symmetric square matrix of  $\mathcal{M}_{n,n}(\mathbb{R})$  defined by:

$$\text{Cov}[\underline{X}]_{ij} = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] \quad (9)$$

where  $X_i$  and  $X_j$  are the  $i^{th}$  and  $j^{th}$  components of  $\underline{X}$ . This matrix is semidefinite positive. For bell shaped distributions, this quantity express the dispersion of the distribution around its mean value.

# Convergence

## Definition

- A sequence of random vectors  $(\underline{X})_{n \in \mathbb{N}}$  defined over the **same** probability space  $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$  is said to **converge almost surely** to the random vector  $\underline{X}$  if and only if:

$$\mathbb{P}[\{\omega \in \Omega \mid \underline{X}_n(\omega) \not\rightarrow \underline{X}(\omega) \text{ as } n \rightarrow \infty\}] = 0 \quad (10)$$

- A sequence of random vectors  $(\underline{X})_{n \in \mathbb{N}}$  defined over the probability spaces  $(\Omega_n, \mathcal{B}(\Omega_n), \mathbb{P}_n)$  is said to **converge in law** to the random vector  $\underline{X}$  defined over the probability space  $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$  if and only if:

$$\forall \phi \in \mathcal{C}^b(\mathbb{R}^n, \mathbb{R}), \lim_{n \rightarrow \infty} \mathbb{E}[\phi(\underline{X}_n)] = \mathbb{E}[\phi(\underline{X})] \quad (11)$$

where  $\mathcal{C}^b(\mathbb{R}^n, \mathbb{R})$  is the set of bounded continuous functions defined on  $\mathbb{R}^n$  and taking value into  $\mathbb{R}$ .

# Simulation

## Strong number law and Central Limit Theorem

- **(Strong law of large numbers)** For all sequence of random vectors  $(\underline{X})_{n \in \mathbb{N}}$  defined over the **same** probability space  $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$ , **independent** and **sharing the same distribution**  $\mu_{\underline{X}}$ , for all measurable function  $f \in \{(\mathbb{R}^n, \mathbb{R}^p)$  such that  $\mathbb{E}[f(\underline{X}_1)]$  exists, **the sequence of random vectors**  $(\frac{1}{n} \sum_{k=1}^n f(\underline{X}))_{n \in \mathbb{N}}$  **converges almost surely to the constant random vector**  $\mathbb{E}[f(\underline{X}_1)]$
- **(Central Limit Theorem)** Moreover, if  $\text{Cov}[\underline{X}_1]$  is well-defined and finite, the sequence of random vectors defined by  $(\sqrt{n} (\frac{1}{n} \sum_{k=1}^n f(\underline{X}) - \mathbb{E}[f(\underline{X}_1)]))_{n \in \mathbb{N}}$  **converges in law to a multivariate Normal distribution with covariance**  $\text{Cov}[\underline{X}_1]$

The first theorem gives a mean to compute any quantity related to a random vector  $\underline{Y}$  defined as the image of a random vector  $\underline{X}$  through a numerical model  $f$ , for exemple its CDF: generate many independent realizations of  $\underline{Y}$ , take  $\phi(\underline{y}) = 1_{\{\tilde{\underline{y}} \in \mathbb{R}^p \mid \tilde{y}_i \leq y_i\}}$  and the quantity  $\frac{1}{n} \sum_{k=1}^n \phi(\underline{Y})$  will almost surely converge towards  $\mathbb{E}[\phi(\underline{Y})] = \mathbb{P}[Y_1 \leq y_1, \dots, Y_n \leq y_n] = F_{\underline{Y}}(\underline{y})$ . It is the so-called **Monte Carlo method**. The second theorem gives a mean to quantify the ecision of a Monte Carlo approximation for large but finite values of  $n$ . As we know the asymptotic behaviour of the fluctuations of  $(\frac{1}{n} \sum_{k=1}^n f(\underline{X}))_{n \in \mathbb{N}}$  we can determine a region  $R$  for which the needed quantity has a large probability to be, and we see that this region shrinks with a speed proportional to  $\frac{1}{\sqrt{n}}$ .

# Simulation

## Monte Carlo algorithm



## The end

### No more maths!

- The probabilistic approach to uncertainty propagation involves some high level maths,
- The following presentations will show that an OpenTURNS developer must have a basic knowledge of these maths (at least the basic vocabulary) in order to be comfortable with the platform and its objects...
- ... but he/she has not to be an expert in order to be efficient!