Introduction to Uncertainty Quantification

Olivier Le Maître[1]
(with Omar Knio KAUST)

[1] Centre de Mathématiques Appliquées, CNRS
Ecole Polytechnique, Palaiseau, France
https://perso.limsi.fr/olm/
olivier.le-maitre@polytechnique.edu

Ecole d'été de Mécanique Théorique, Quiberon

Objectives of the lecture

- Introduce and discuss **Uncertainty Quantification** problems
- Context, motivations, objectives of UQ analysis
- Monte-Carlo approach to serve as reference (and hands-on)

Some material of this lecture is taken from the book:
Spectral Methods for Uncertainty Quantification with applications in computational fluid dynamics with Omar Knio, Springer (2010).

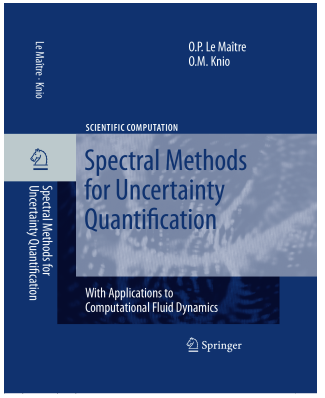


Table of content

Simulations & models

Historically

1. **Experiments & Observations**
2. Theories & Models
3. Use of models to produce predictions

   Model predictions compared to new observations in order to validate the theory

Simulations & models

Development of computational tools

1. Experiments & Observations
2. Theories & Models
3. **Exploit generalization capability** of (physical) models to produce new predictions

   Now needs new observations in oder to validate the computational model

Nowadays

- Experiments & Observations
- Theories & models
- Computational approaches & Simulations

  All the three are complementary and strongly coupled
- **Cross validation** of **simulations & experiments**
- Different information and knowledge gained

Some important concepts

VVUQ in simulation

- **Verification:** make sure that the simulation code is actually computing what we aim for (solving the targeted model)
- **Validation:** make sure that the simulation code can reproduce the targeted physics
- **Uncertainty Quantification:** make sure that the prediction fairly accounts for *all sources of uncertainty* in the definition of the numerical model.

UQ becomes more and more crucial as simulations gain in fidelity and resolution.

Other important concepts

- **Calibration:** procedure using observations to decide of the likely values of some model parameters
- **Model error:** remaining inadequacy between the (calibrated) model prediction and the physical truth.

Recall: `[All models are wrong, some are useful (G. Box)]`

Simulation framework

Basic ingredients

- Selection of a **mathematical model**:
  retain the essential physical processes
- Selection of a **numerical method**:
  to solve the model equations
- Define all **input-data** needed:
  select a specific system in the whole model class

Simulation framework

Basic ingredients

- Selection of a **mathematical model**:
  retain the essential physical processes
- Selection of a **numerical method**:
  to solve the model equations
- Define all **input-data** needed:
  select a specific system in the whole model class

Simulation errors

- **Model error**: physical approximations, simplifications, idealization, ...
- **Numerical errors**: discretization, approximate solvers, finite arithmetics, ...
- **Input-data error: boundary/initial conditions, model constants and parameters, external forcings, ...**

Input Uncertainty

Sources of input-data uncertainty

- Inherent **variability** (*e.g.* industrial processes)
- **Epistemic** uncertainty (*e.g.* model constants)
- May not be fully reducible, even theoretically

Input Uncertainty

Sources of input-data uncertainty

- Inherent **variability** (*e.g.* industrial processes)
- **Epistemic** uncertainty (*e.g.* model constants)
- May not be fully reducible, even theoretically

Probabilistic framework

- Define an abstract probability space $(\Omega, \mathcal{A}, d\mu)$
- Consider **input-data** D as a **random** quantity: $D(\omega),\ \omega \in \Omega$
- **Simulation output** S is **random** and on $(\Omega, \mathcal{A}, d\mu)$

Input Uncertainty

Sources of input-data uncertainty

- Inherent **variability** (*e.g.* industrial processes)
- **Epistemic** uncertainty (*e.g.* model constants)
- May not be fully reducible, even theoretically

Probabilistic framework

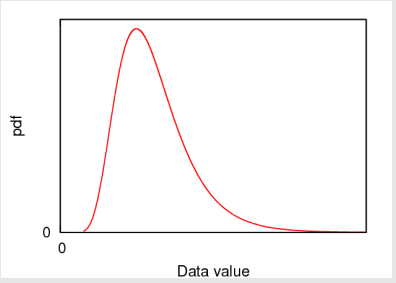
- Define an abstract probability space $(\Omega, \mathcal{A}, d\mu)$
- Consider **input-data** D as a **random** quantity: $D(\omega),\ \omega \in \Omega$
- **Simulation output** S is **random** and on $(\Omega, \mathcal{A}, d\mu)$
- **Data D and simulation output S are dependent random quantities (through the mathematical model $\mathcal{M}$):**

$$\mathcal{M}(S(\omega), D(\omega)) = 0, \quad \forall \omega \in \Omega.$$

Uncertainty Propagation

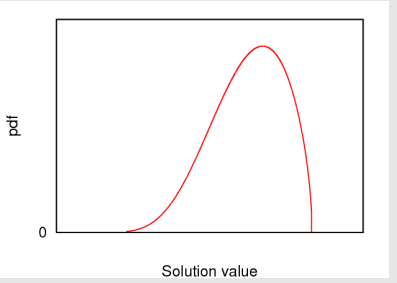
Propagation of data uncertainty

Data density



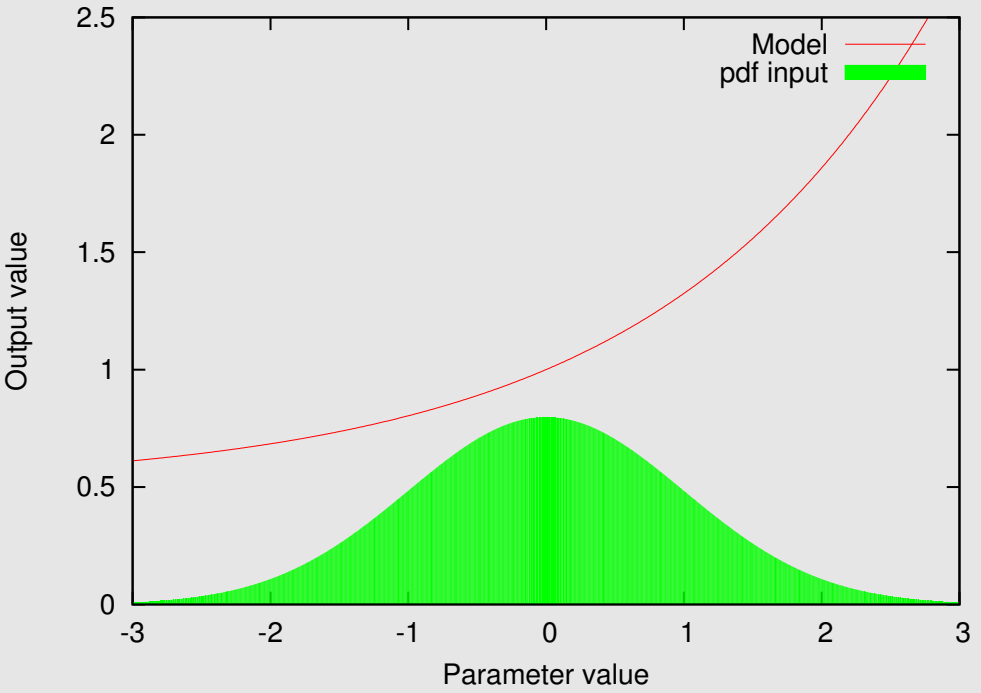
$$\mathcal{M}(S, D) = 0$$

Solution density



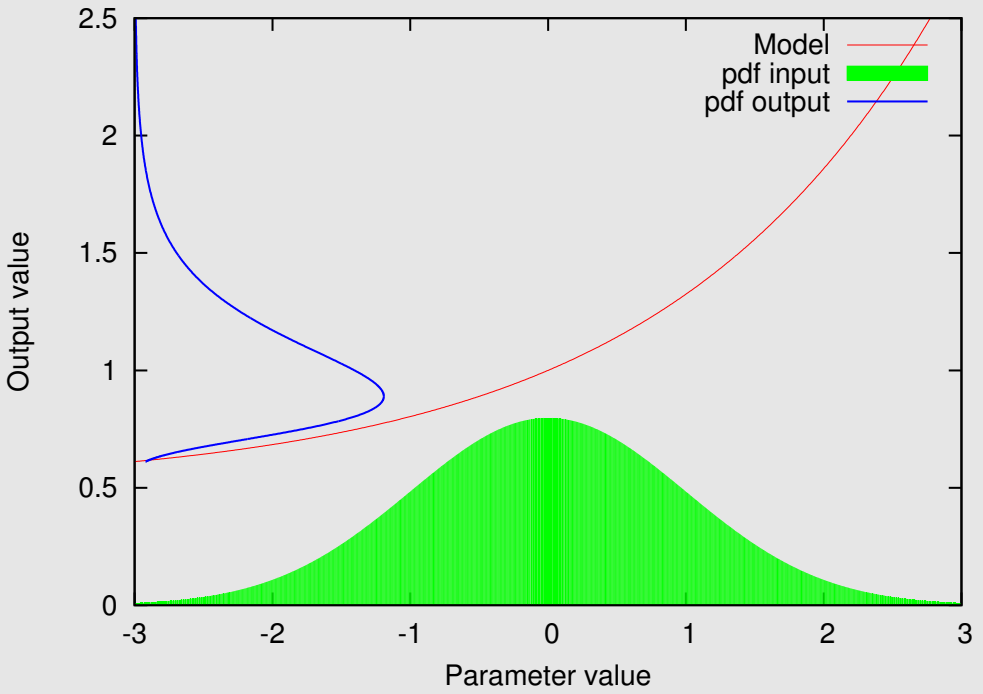
Uncertainty Propagation

Propagation of data uncertainty



Uncertainty Propagation

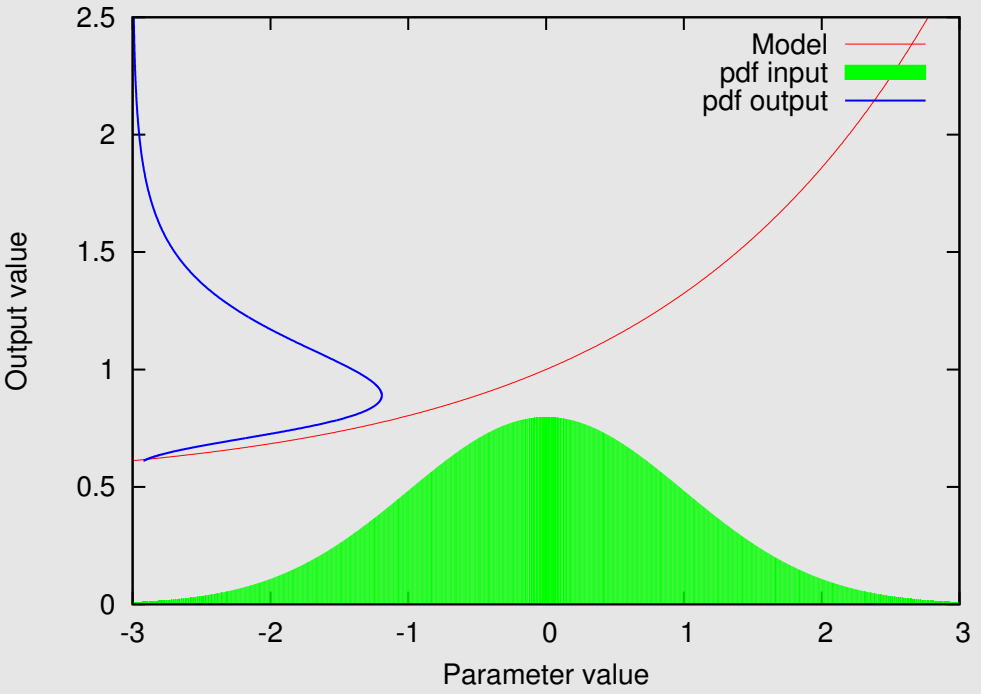
Propagation of data uncertainty



- Large number of parameters
- Costly model evaluation (PDE)
- Estimation of $p(S)$ is not the end of the story!

Uncertainty Propagation

Propagation of data uncertainty



- **Variability** in model output: numerical error bars.
- Assessment of **predictability**.
- Support **decision making process**.
- What type of information (abstract quantities, confidence intervals, density estimations, structure of dependencies, . . .) one needs?

UQ Methods

Deterministic methods

- **Sensitivity analysis** (adjoint based, AD, . . .): local.
- **Perturbation techniques**: limited to low order and simple data uncertainty.
- **Neumann expansions**: limited to low expansion order.
- **Moments** method: closure problem (non-Gaussian / non-linear problems).

Simulation techniques — Monte-Carlo

Spectral Methods

UQ Methods

Deterministic methods

Simulation techniques — Monte-Carlo

- Generate a **sample set of data** realizations and compute the corresponding **sample set of model ouput**.
- Use sample set based **random estimates** of abstract characterizations (moments, correlations, . . .).
- Plus: Very robust and re-use deterministic codes: (parallelization, complex data uncertainty).
- Minus: **slow convergence of the random estimates** with the sample set dimension.

Spectral Methods

UQ Methods

Deterministic methods

Simulation techniques — Monte-Carlo

Spectral Methods

- Parameterization of the data with random variables (RVs).
- $\perp$ projection of solution on the (L^2) space spanned by the RVs.
- Plus: arbitrary level of uncertainty, deterministic approach, convergence rate, information contained.
- Minus: **parameterizations** (limited # of RVs), **adaptation of simulation tools** (legacy codes), **robustness** (non-linear problems, non-smooth output, ...).
- **Not suited for model uncertainty**

Table of content

Sample estimate

Let $\Xi \ni \boldsymbol{\xi} \sim \mu(\boldsymbol{\xi})$ be a random quantify (variable, vector, field) and consider $F : \Xi \mapsto \mathbb{R}$, a **real-valued random variable**. We shall assume that $F \in L_2(\Xi, \mu)$:

$$\int_{\Xi} |F(\boldsymbol{\xi})|^2 d\mu(\boldsymbol{\xi}) = \mathbb{E}\left[F^2\right] < \infty.$$

In our context:

- $\boldsymbol{\xi}$ represents the random input/parameters of the model
- $F(\boldsymbol{\xi})$ is a function of the model output
- We want to compute a statistic of the model output

$$\mathbb{E}[F] \doteq \int_{\Xi} F(\boldsymbol{\xi}) d\mu(\boldsymbol{\xi}).$$

To approximate $\mathbb{E}[F]$, the arithmetic mean is typically used:

$$\bar{F}_N = \frac{1}{N} \sum_{i=1}^{N} F(\boldsymbol{\xi}^{(i)}),$$

where the $\boldsymbol{\xi}^{(i)}$ are **independent realizations** of $\boldsymbol{\xi}$ (drawn from μ).
The estimate $\bar{F}_N$ is random

Estimate convergence

If the expectation ($\mathbb{E}[F]$) exists, the law of large numbers implies that the arithmetic means converges to the mathematical expectations

$$\bar{F}_N \xrightarrow{p} \mathbb{E}[F] \quad \text{as } N \to \infty$$

Consequently, $\bar{F}_N \simeq \mathbb{E}[F]$ for large N.

If F has a finite variance $\mathbb{V}[F]$, **the central limit theorem** implies

$$\lim_{N\to\infty} Pr\left\{ x_1 < \frac{1}{\sqrt{N\mathbb{V}[F]}} \sum_{i=1}^{N} (F(\xi^{(i)}) - \mathbb{E}[F]) < x_2 \right\} = \frac{1}{\sqrt{2\pi}} \int_{x_1}^{x_2} \exp\left(-\frac{t^2}{2}\right) dt.$$

Monte Carlo error – I

Let $-x_1 = x_2 = x > 0$, it follows that

$$\lim_{N\to\infty} Pr\left\{\left|\mathbb{E}[F] - \frac{1}{N}\sum_{i=1}^{N} F(\boldsymbol{\xi}^{(i)}))\right| < x\sqrt{\frac{\mathbb{V}[F]}{N}}\right\} = \Phi(x)$$

where

$$\Phi(x) = \frac{2}{\sqrt{2\pi}}\int_0^x \exp\left(-\frac{t^2}{2}\right) dt \quad \text{is the probability integral}$$

When N is sufficiently large, we have

$$Pr\left\{\left|\bar{F}_N - \mathbb{E}[F]\right| < x\sqrt{\frac{\mathbb{V}[F]}{N}}\right\} \approx \Phi(x)$$

This gives a whole variety of estimates, depending on x

Monte Carlo error – II

If a probability $0 < \beta < 1$ is given, then the root x_β of the equation $\Phi(x) = \beta$ can be found (e.g. from tables)
It follows that

$$Pr\left\{\left|\bar{F}_N - \mathbb{E}[F]\right| < x_\beta \sqrt{\frac{\mathbb{V}[F]}{N}}\right\} \approx \beta$$

The probable error, r_N, is the value for which

$$Pr\left\{\left|\bar{F}_N - \mathbb{E}[F]\right| \leq r_N\right\} = \frac{1}{2} = Pr\left\{\left|\bar{F}_N - \mathbb{E}[F]\right| \geq r_N\right\}$$

Thus,

$$r_N = x_{0.5}\sigma(F)N^{-1/2}$$

where

$$\sigma(F) \equiv \sqrt{\mathbb{V}[F]} \qquad x_{0.5} \approx 0.6745$$

Remarks

When no information about the smoothness of the function is available, the probable error in plain MC is:

$$r_N = c\sqrt{\frac{\mathbb{V}[F]}{N}}$$

MC algorithms with reduced variance compared to plain MC are called efficient MC algorithms. Techniques used to achieve such reduction are called variance reduction techniques. Selected approaches are outlined in the following.

Separation of Principal Part

Consider again $\xi \in \Xi$ with distribution μ and $F \in L_2(\Xi, \mu)$.
Suppose we can find $H \in L_2(\Xi, \mu)$ close to F, such that

$$\|F - H\|_2 \equiv \mathbb{E}\left[|F - H|^2\right] = \int_\Xi |F(\boldsymbol{\xi}) - H(\boldsymbol{\xi})|^2 d\mu(\boldsymbol{\xi}) \leq \epsilon^2,$$

and $\mathbb{E}[H]$ **is known**.
Consider $\theta(\boldsymbol{\xi}) \doteq F(\boldsymbol{\xi}) - H(\boldsymbol{\xi}) + \mathbb{E}[H]$. Plain MC on θ' gives:

$$\overline{\theta}_N = \mathbb{E}[H] + \frac{1}{N}\sum_{i=1}^{N}\left[F(\boldsymbol{\xi}^{(i)}) - H(\boldsymbol{\xi}^{(i)})\right]$$

Clearly, $\mathbb{E}[\theta] = \mathbb{E}[F]$ and

$$\mathbb{V}[\theta] = \mathbb{E}\left[|F - H|^2\right] - (\mathbb{E}[F] - \mathbb{E}[H])^2 \leq \epsilon^2$$

$\Rightarrow$ probable error will be quite small if H predict well F. This idea is exploited in

- surrogate variance reduction
- reduce model variance reduction
- multi-fidelity and multilevel MC methods

Integration on Subdomains

Suppose we can integrate F analytically for $\boldsymbol{\xi} \in \Xi' \subset \Xi$

$$\int_{\Xi'} F(\boldsymbol{\xi}) d\mu(\boldsymbol{\xi}) = I' \qquad \text{and} \qquad \int_{\Xi'} d\mu(\boldsymbol{\xi}) = c, \quad 0 < c < 1.$$

Then

$$\mathbb{E}[F] = \int_{\Xi \setminus \Xi'} F(\boldsymbol{\xi}) d\mu(\boldsymbol{\xi}) + I'.$$

Define a random point $\boldsymbol{\xi}_* \in \boldsymbol{\xi} \setminus \boldsymbol{\xi}'$ with distribution $\mu_*(x) = \mu(x)/(1-c)$ and let

$$H(\boldsymbol{\xi}_*) \doteq I' + (1-c)F(\boldsymbol{\xi}_*)$$

Clearly, $\mathbb{E}[H] = \mathbb{E}[F]$. Thus, we define the MC estimator:

$$\overline{H}_N = I' + \frac{1-c}{N} \sum_{i=1}^{N} F(\boldsymbol{\xi}_*^{(i)})$$

where $\boldsymbol{\xi}_*^{(i)}$ are independent realizations of $\boldsymbol{\xi}_*$.
It can be shown that if $\mathbb{V}[F]$ exists, then

$$\mathbb{V}[H] \leq (1-c)\mathbb{V}[F]$$

Symmetrization of Integrand

Consider the case of $\Xi = [a, b]$ with **uniform distribution**.
The plain MC approximation of $\mathbb{E}[F]$ would yield:

$$\bar{F}_N = \frac{1}{N} \sum_{i=1}^{N} F(\boldsymbol{\xi}^{(i)})$$

where $\boldsymbol{\xi}^{(i)}$ are independent realizations of $\boldsymbol{\xi} \in [a, b]$.
Consider now the symmetric function

$$F_s(\boldsymbol{\xi}) = \frac{1}{2} [F(\boldsymbol{\xi}) + F(a + b - \boldsymbol{\xi})]$$

Clearly, $\mathbb{E}[F_s] = 2\mathbb{E}[F]$.
Then, we consider the MC estimate

$$\tilde{F}_{sN} = \frac{1}{2N} \sum_{i=1}^{N} \left[F(\boldsymbol{\xi}^{(i)}) + F(a + b - \boldsymbol{\xi}^{(i)}) \right]$$

One can show that **if** $F : [a, b] \mapsto \mathbb{R}$ **is monotonic**, then $\mathbb{V}[F_s] \leq \mathbb{V}[F]/2$. The result generalizes to non-monotonic functions under mild conditions.

Importance Sampling – I

Suppose that he distribution μ has a density ρ:

$$\mathbb{E}_\rho[F] = \int_\Xi F(\xi)\rho(\xi)d\xi.$$

Using the estimator

$$\bar{F}_N = \frac{1}{N}\sum_{i=1}^{N} F(\xi^{(i)}),$$

we have

$$v_N \equiv \mathbb{V}\left[\bar{F}_N\right] = \frac{1}{N^2}\sum_{i=1}^{N}\left[F(\xi^{(i)}) - \bar{F}_N\right]^2$$

and for N large

$$\frac{\bar{F}_N - \mathbb{E}_\rho[F]}{\sqrt{v_m}} \approx N(0,1).$$

The importance sampling uses samples $\xi^{(1)},\ldots,\xi^{(N)}$ from **an alternative distribution** ρ_* and the formula

$$\mathbb{E}_\rho[F] = \int_\Xi \frac{F(\xi)\rho(\xi)}{\rho_*(\xi)} g(\xi)d\xi = \mathbb{E}_{\rho_*}\left[\frac{F\times\rho}{\rho_*}\right].$$

Importance Sampling – II

The choice of ρ_* that minimizes the variance of the corresponding estimator is

$$\rho_*(\boldsymbol{\xi}) = \frac{|F(\boldsymbol{\xi})|\,\rho(\boldsymbol{\xi})}{\left|\int_\Xi F(\eta)\rho(\eta)d\eta\right|}.$$

This density is hard to evaluate because the denominator is essentially the unknown quantity.
A practical alternative is to use the result as follows:

$$\frac{\sum_{i=1}^N F(\boldsymbol{\xi}^{(i)})\rho(\boldsymbol{\xi}^{(i)})/\rho_*(\boldsymbol{\xi}^{(i)})}{\sum_{i=1}^N \rho(\boldsymbol{\xi}^{(i)})/\rho_*(\boldsymbol{\xi}^{(i)})} = \frac{\sum_{i=1}^N F(\boldsymbol{\xi}^{(i)})\left|F(\boldsymbol{\xi}^{(i)})\right|^{-1}}{\sum_{i=1}^N \left|F(\boldsymbol{\xi}^{(i)})\right|^{-1}}$$

where the $\boldsymbol{\xi}^{(i)} \sim \rho_* \propto |F|\rho$.
Unfortunately, this estimator is biased and may exhibit severe instability.

To avoid this, we look for distribution ρ_* for which $|F|\rho/\rho_*$ **is almost constant**, with finite variance.

Although finite variance is not necessary for convergence of the last estimator, importance sampling performs poorly when $\mathbb{E}_\rho[\rho/\rho_*] = +\infty$, whether in terms of behavior (high amplitude jumps, instability of path of the average, slow convergence) or in comparison with direct MC.

Iso-probabilistic transformations

- Most computer languages propose **pseudo-random number generators** for the classical distributions
- Main characteristic : appearance of independence for the sequence and length
- Dedicated libraries (std-lib in C++) propose high quality generators for standard distributions (uniform, normal, exponential, Poisson, Bernoulli, ...)

Well characterized distributions over $\mathbb{R}$ can be achieved by mean of **iso-probabilistic transformations**

For higher dimensional sampling space, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d)$, fast sampling can be achieved **if the components ξ_i are independent**, otherwise more involved techniques (*e.g.* MCMC sampler) must be considered.

Latin Hypercube Sampling – I

Alternative ways to reduce the MC estimate error are based on **controlled sampling** to improve the covering of Ξ

Latin hypercube selects n different values from each of the d **iid rv's** $\xi_1, \ldots, \xi_d$ according to the following procedure:

- The range of each variable is divided into n non-overlapping **intervals having equal probability**
- One value from each interval is selected at random with respect to the probability density in the interval
- The n values thus obtained for ξ_1 are paired in a random manner (**equally likely combinations**) with the n values of ξ_2
- These n pairs are combined in a random manner with the n values of ξ_3 to form n triplets, and so on, until n d-tuplets are formed.

Latin Hypercube Sampling – II

$$\rho(\boldsymbol{\xi}) = \rho_1(\xi_1) \dots \rho_d(\xi_d).$$

It is convenient to think of the LH sample (or any random sample of size n) as forming an $(n \times d)$ matrix of input where the j-th row contains the specific values of each of the d input variables ξ_i to be used on the j-th realization of the input (j-th simulation)

The pairing is typically done by associating a random permutation of the first n integers with each input variable. The permutations ensure that **no variable is taken twice from the same interval for every dimension**.

Consider a 2D example, where we are interested in generating a LH with 5 samples
The ranges of ξ_1 and ξ_2 are first divided into 5 equal probability intervals
We next consider two random permutations of the integers $\{1, 2, 3, 4, 5\}$, say:

- Permutation 1: (3, 1, 5, 2, 4)
- Permutation 2: (2, 4, 1, 3, 5)

Then the resulting LHS would be

Sample	Interval used for ξ_1	Interval used for ξ_2
1	3	2
2	1	4
3	5	1
4	2	3
5	4	5

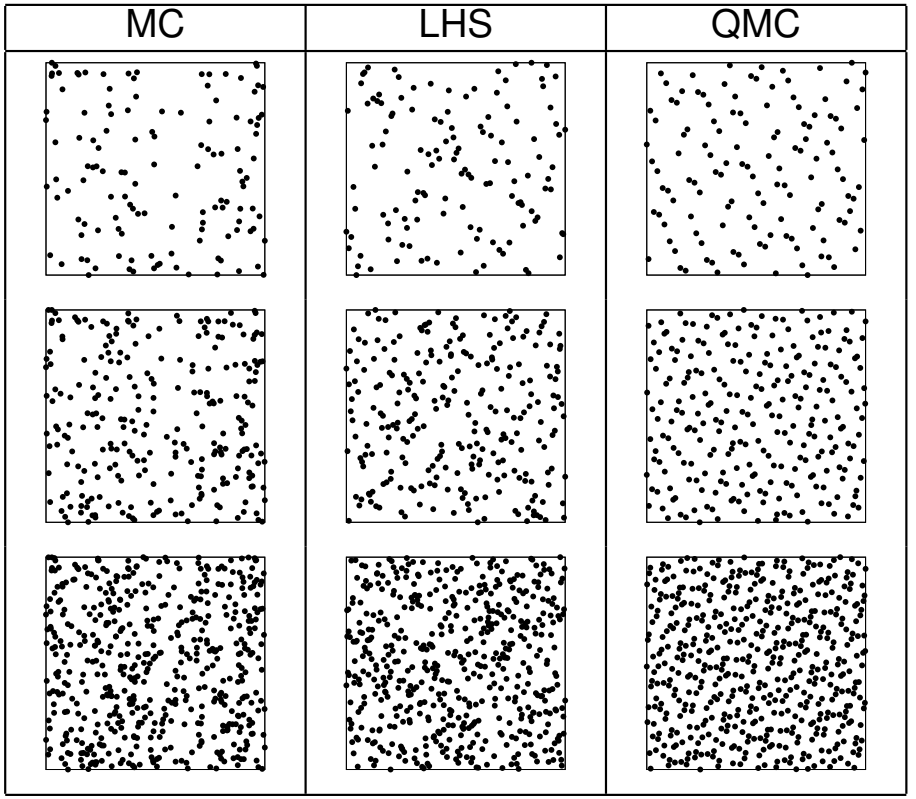
Latin Hypercube Sampling – III

Elaborate LHS algorithms have been developed, particularly to incorporate key ingredients to ensure proper performance. These include:

- uniform random permutations – a uniform random permutation of 1 to n is one in which all $n!$ possible orderings have the same probability
- imposing pairing restrictions to avoid or minimize correlations between statistically independent quantities

Sophisticated software tools implementing powerful LHS algorithms are widely available, including MATLAB.

Comparison of samplin methods



- Quasi-Monte Carlo (deterministic sequence)
- **Convergence rate**
- **Error estimate**
- Optimal **sampling strategy**

Table of content

Background

- Markov chain Monte Carlo (MCMC) methods: class of algorithms aimed at simulating direct draws from some complex distribution of interest.
- Origin of the name: one uses the previous sample values to randomly (Monte Carlo) generate the next sample value, thus creating a Markov chain.
- A Markov chain is indeed defined as a process where the transition probability between the current and following state is only a function of the current state.
- Many different flavors of generating a Markov Chain. Focus on Metropolis-Hastings algorithm: a random walk using a proposal density and a method for accepting/rejecting proposed moves.

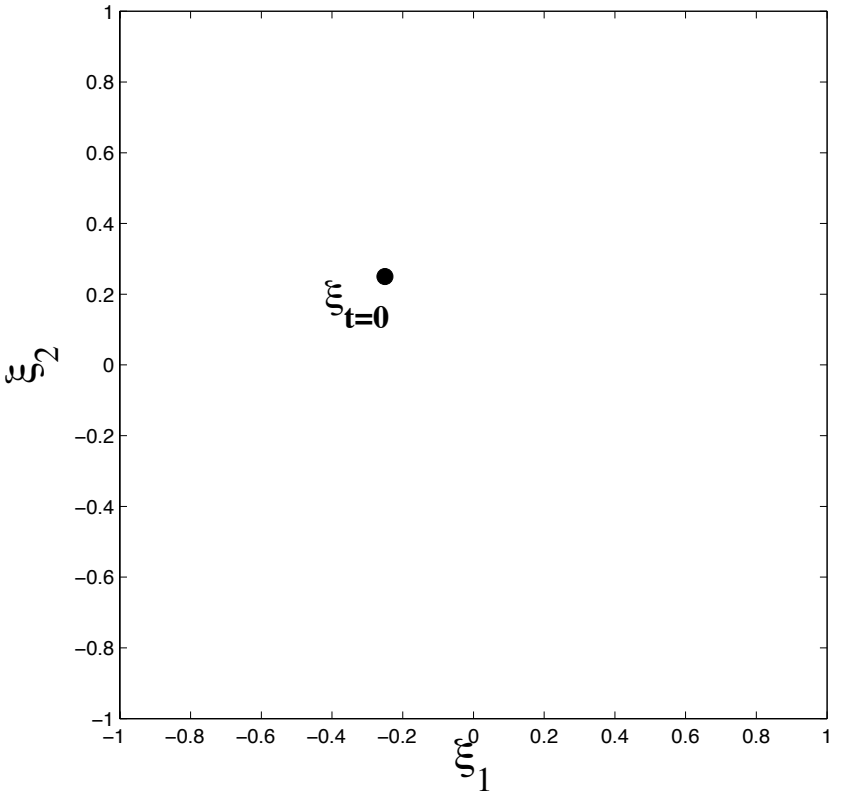
MCMC: features

- The states of the chain after a large number of steps are used as samples of the desired distribution.
- The quality of the sample improves as a function of the number of steps.
- The more difficult problem is to determine how many steps are needed to converge to the stationary distribution within an acceptable error: usually one needs at least $\sim$ 10000 samples.
- A good chain is one that has rapid mixing, i.e. the stationary distribution is reached quickly starting from an arbitrary position and the target probability is explored well and efficiently.
- A common application of these algorithms is for numerically calculating multi-dimensional integrals.
- Let's look in detail at the Metropolis algorithm and how to generate samples from a certain distribution.

Metropolis (MH)

- MH algorithm can draw samples from a target probability distribution, π, requiring only the knowledge of a function proportional to the target PDF.
- It uses a proposal distribution, P, to generate (Markov chain) candidates that are accepted or rejected according to a certain rule. Let P be a Gaussian for simplicity.

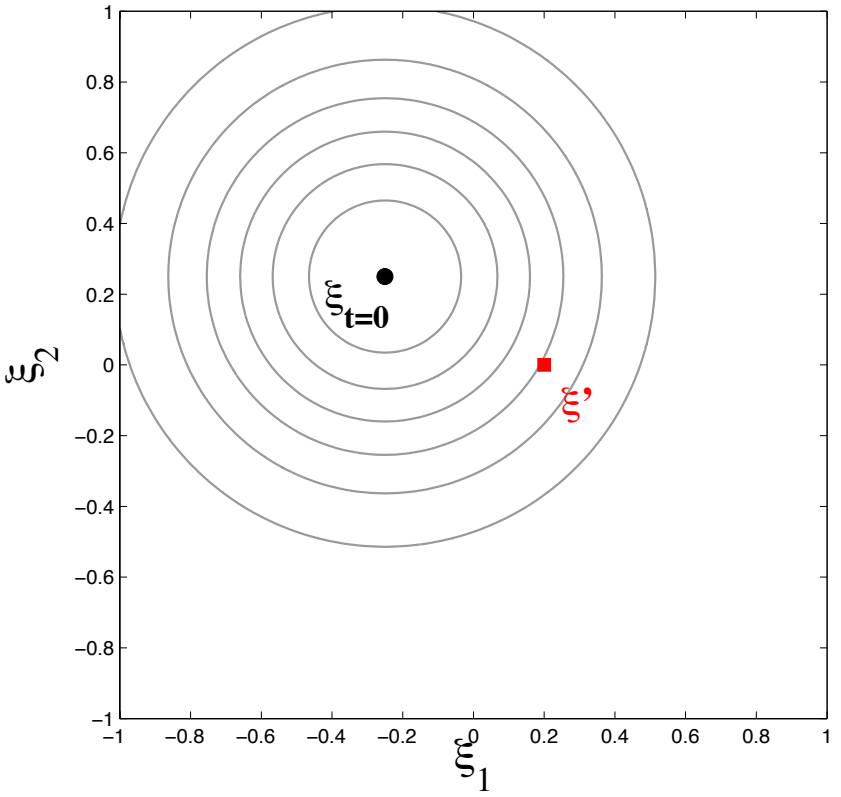
1 Let $\boldsymbol{\xi}_{t=0}$ be an initial guess for a 2D problem.



Metropolis (MH)

- MH algorithm can draw samples from a target probability distribution, π, requiring only the knowledge of a function proportional to the target PDF.
- It uses a proposal distribution, P, to generate (Markov chain) candidates that are accepted or rejected according to a certain rule. Let P be a Gaussian for simplicity.

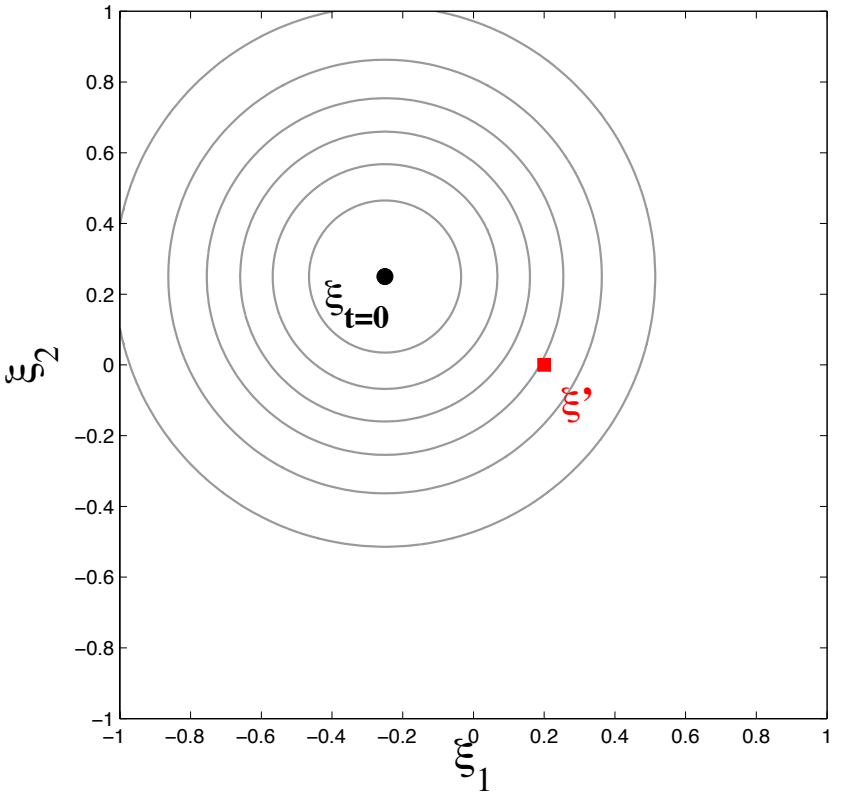
2 Draw a candidate $\boldsymbol{\xi}'$ from a Gaussian centered on the currer state: $\boldsymbol{\xi}' \sim \mathcal{N}(\boldsymbol{\xi}_0, Cov)$ where Cov is chosen a priori.



Metropolis (MH)

- MH algorithm can draw samples from a target probability distribution, π, requiring only the knowledge of a function proportional to the target PDF.
- It uses a proposal distribution, P, to generate (Markov chain) candidates that are accepted or rejected according to a certain rule. Let P be a Gaussian for simplicity.

3 Calculate the ratio: $$r = \frac{\pi(\xi')}{\pi(\xi_0)},$$

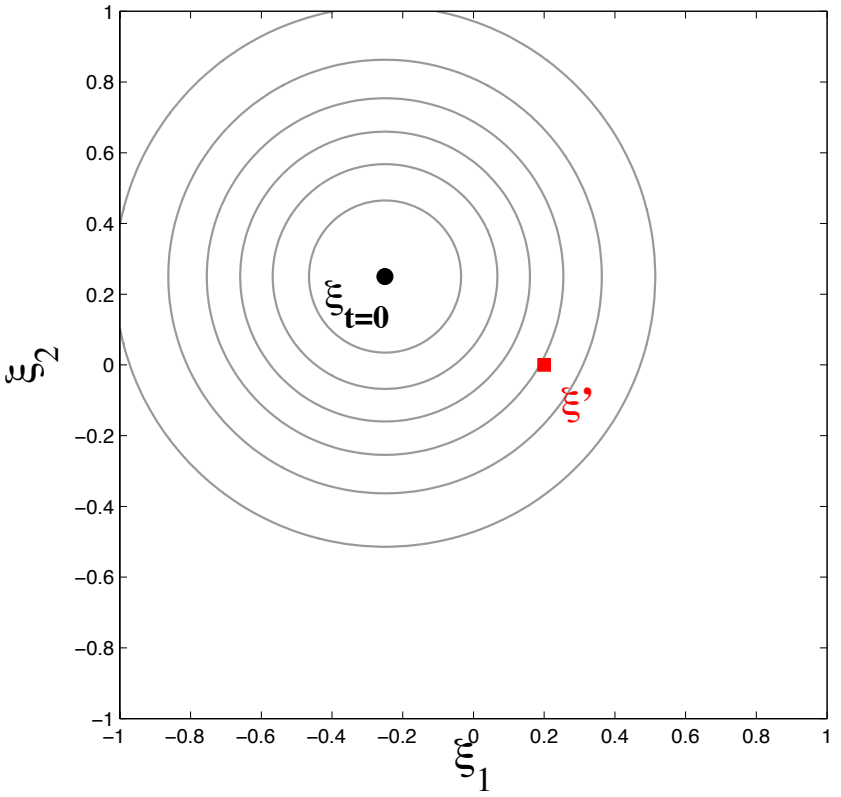


Metropolis (MH)

- MH algorithm can draw samples from a target probability distribution, π, requiring only the knowledge of a function proportional to the target PDF.
- It uses a proposal distribution, P, to generate (Markov chain) candidates that are accepted or rejected according to a certain rule. Let P be a Gaussian for simplicity.

3 Calculate the ratio:
$$r = \frac{\pi(\xi')}{\pi(\xi_0)},$$

4 Draw a random number $\alpha \sim U(0,1)$.



Metropolis (MH)

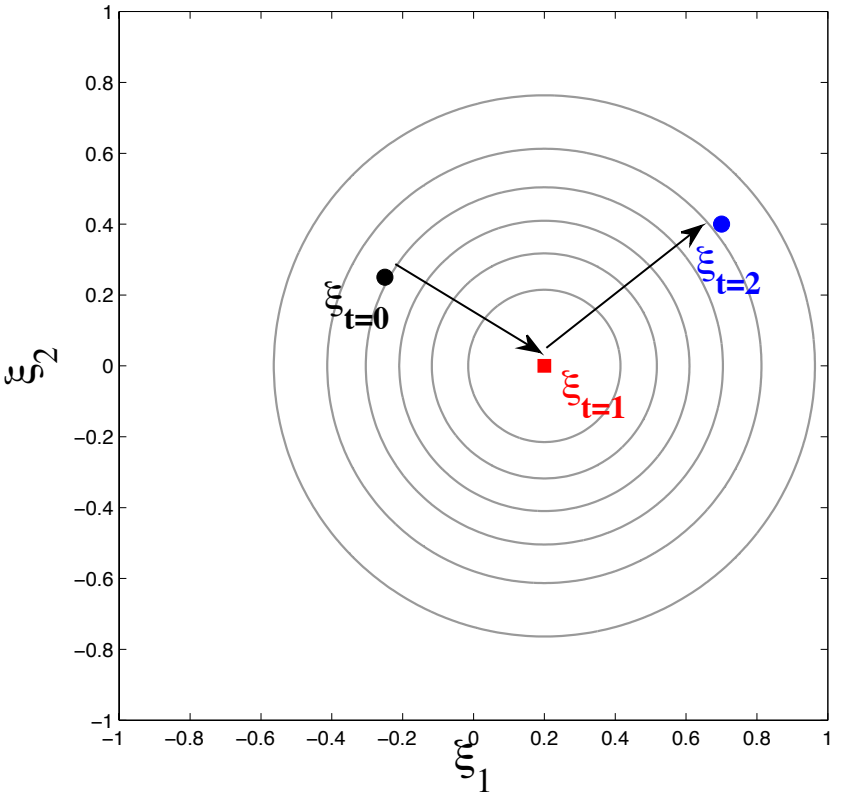
- MH algorithm can draw samples from a target probability distribution, π, requiring only the knowledge of a function proportional to the target PDF.
- It uses a proposal distribution, P, to generate (Markov chain) candidates that are accepted or rejected according to a certain rule. Let P be a Gaussian for simplicity.

3 Calculate the ratio:
$$r = \frac{\pi(\boldsymbol{\xi}')}{\pi(\boldsymbol{\xi}_0)},$$

4 Draw a random number $\alpha \sim U(0, 1)$.

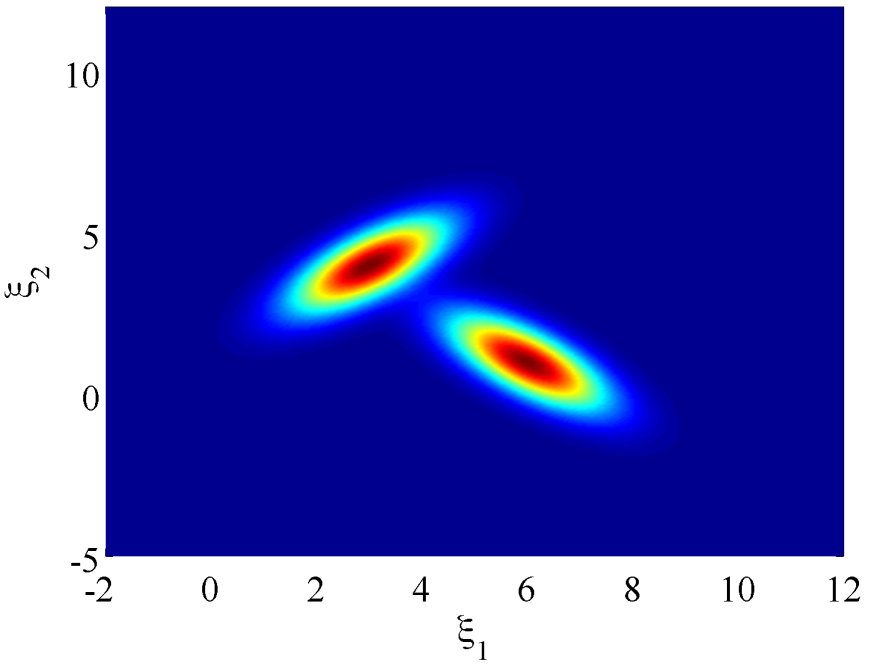
5 Chain moves (i.e. candidate is accepted/rejected) according to:
$$\boldsymbol{\xi}_1 = \begin{cases} \boldsymbol{\xi}' & \text{if} \quad \alpha < r, \\ \boldsymbol{\xi}_0 & \textit{otherwise}. \end{cases}$$

6 Repeat the loop.



Example

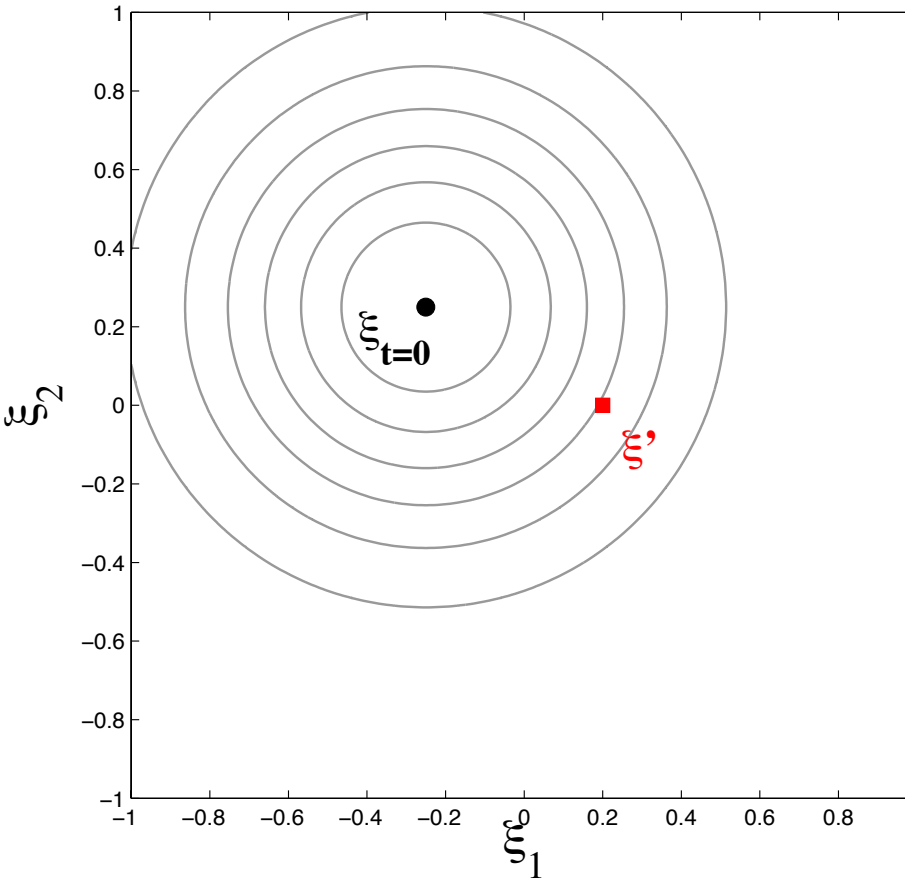
- Suppose that you want to test MCMC to sample a certain bimodal PDF, π, which is proportional to a mixture of two bivariate gaussians:
  $\pi \propto 0.5 * N(\mu_1, \Sigma_1) + 0.5 * N(\mu_2, \Sigma_2)$.
- Method: Metropolis algorithm.



What about sensitivity to *n* and proposal amplitude?

Example

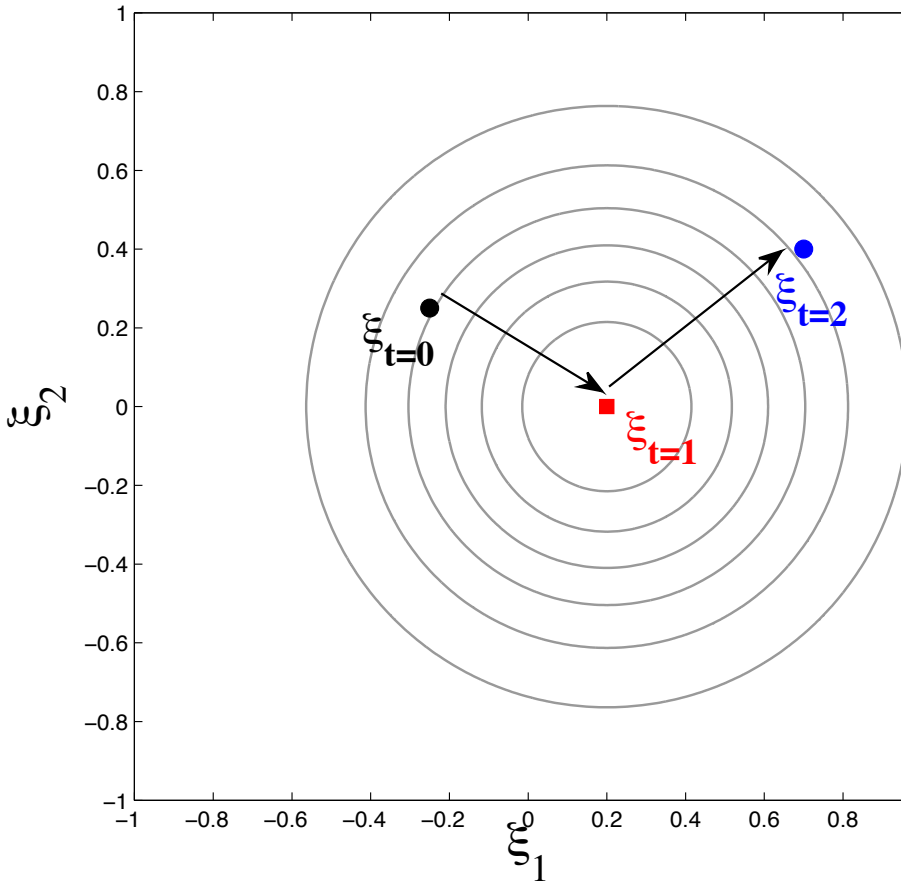
1. Choose the proposal distribution: e.g. a gaussian with covariance: $\Sigma_{prop} = 0.1 * \mathbf{I}_2$.
2. Choose a starting point: $\boldsymbol{\xi}_0 = \{10, 10\}$.
3. Run the machinery for n steps: draw a candidate, accept/reject, repeat loop.



What about sensitivity to n and proposal amplitude?

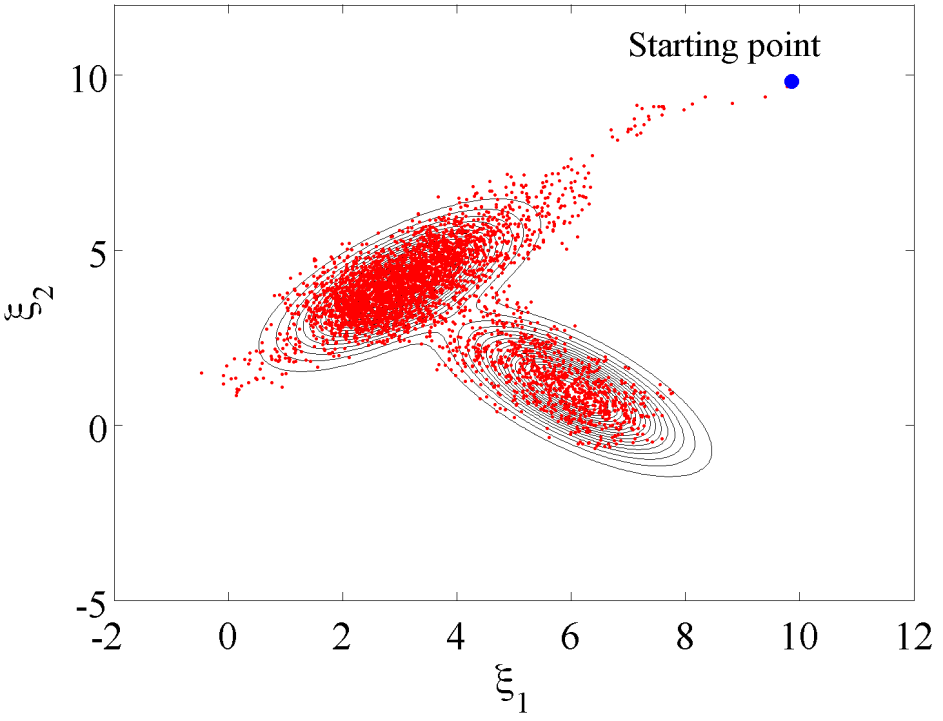
Example

1. Choose the proposal distribution: e.g. a gaussian with covariance: $\Sigma_{prop} = 0.1 * \mathbf{I}_2$.
2. Choose a starting point: $\boldsymbol{\xi}_0 = \{10, 10\}$.
3. Run the machinery for n steps: draw a candidate, accept/reject, repeat loop.



What about sensitivity to n and proposal amplitude?

Example



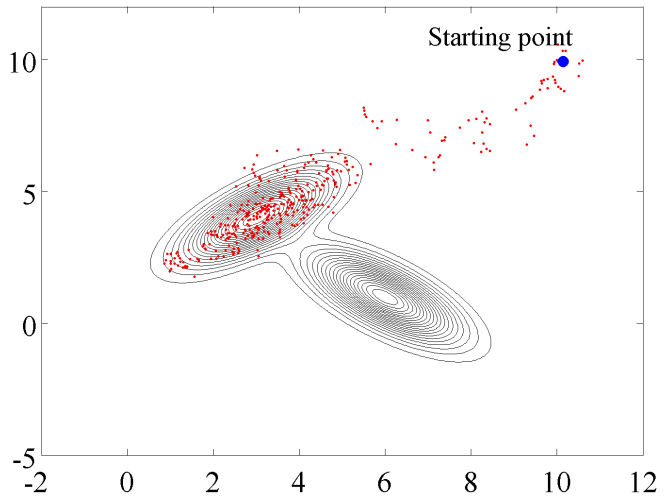
4 Plot the n samples (this case $n = 5000$)

- The proposal amplitude, 0.1, must be varied to obtain mixing and fast convergence.
- The number of samples, n, must be as large as possibile to have a reliable statistics.

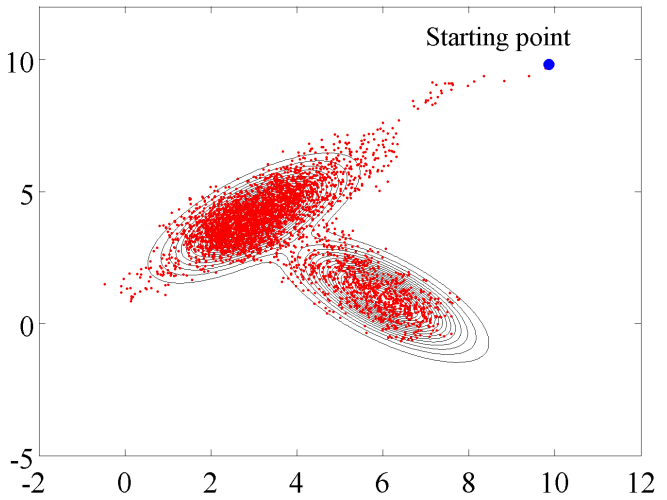
What about sensitivity to n and proposal amplitude?

Sensitivity to number of steps

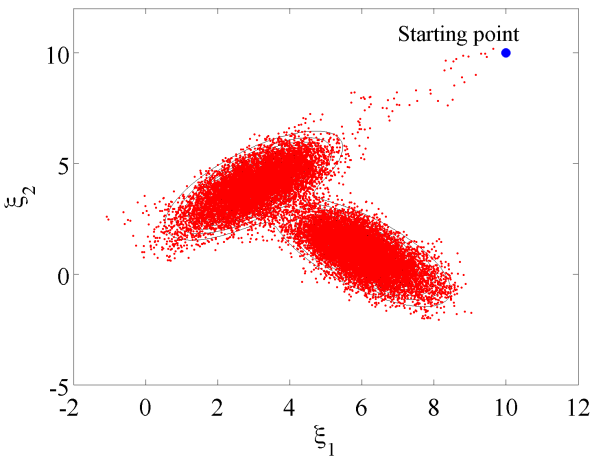
- The proposal distribution has covariance: $\Sigma_{prop} = 0.1 * \mathbf{I}_2$.
- Results for 3 different values of total steps $n = 500$, 5000 and 25000.
- The larger n, the better the approximation.



$n = 500$



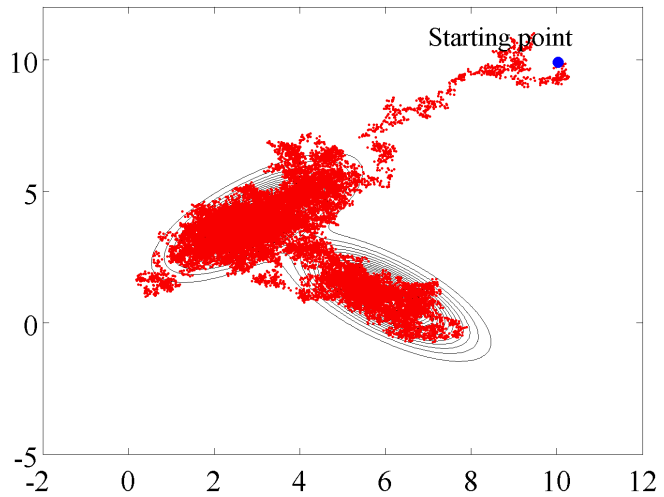
$n = 5000$



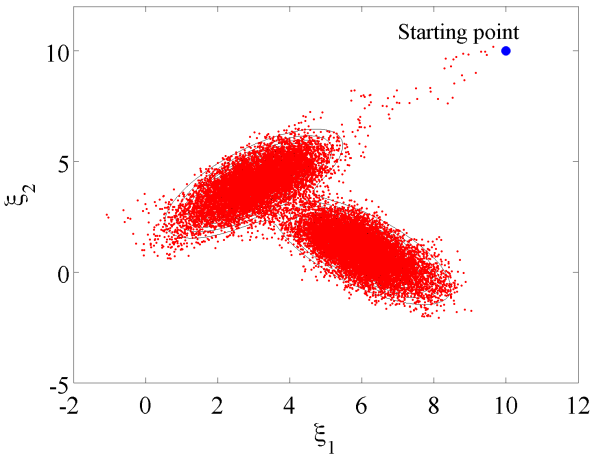
$n = 25000$

Sensitivity to proposal amplitude

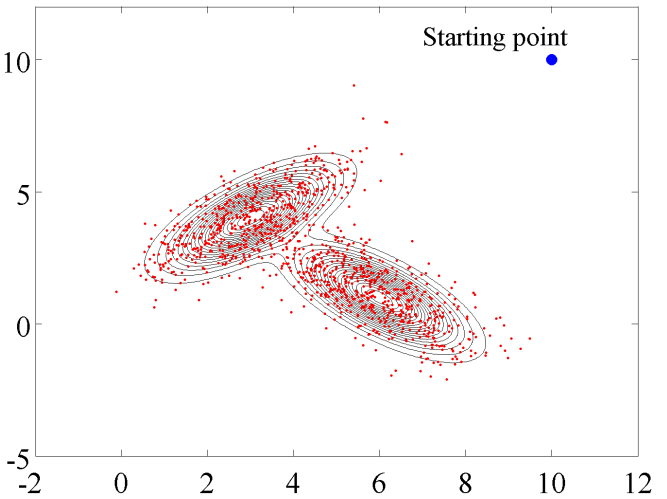
- The proposal amplitude must be tuned to obtain good exploration of the space and fast convergence of the chain toward the high-probability regions.
- Results shown for $0.005 * \mathbf{I}_2$, $0.1 * \mathbf{I}_2$ and $50 * \mathbf{I}_2$.
- The smaller the proposal amplitude, the larger the number of the accepted moves.
- Large proposals lead to small acceptance and slow exploration of the space.
- Ideally, the acceptance rate should be between 30 to 60%.



amplitude $= 0.005$
acceptance rate $\sim$ 95 %.



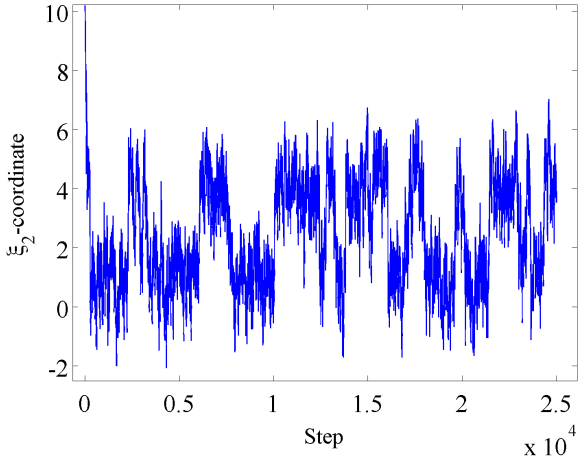
amplitude $= 0.1$
acceptance rate $\sim$ 50 %.



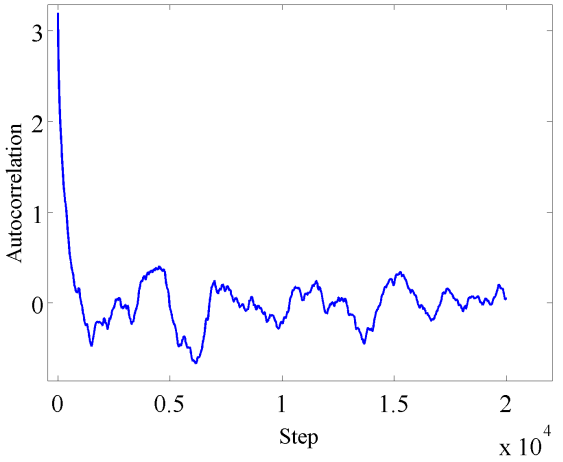
amplitude $= 50$
acceptance rate $\sim$ 4 %.

Sensitivity to proposal amplitude

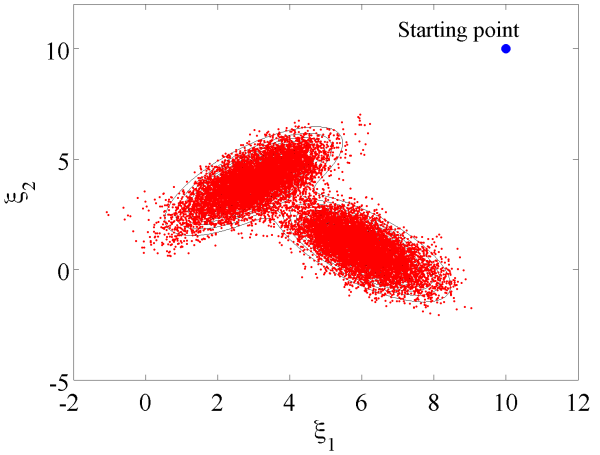
- To evaluate the mixing properties of a chain:
  - visually should look like a white noise.
  - the autocovariance should be rapidly decaying.
  - the acceptance rate should be 30 to 60%.
- Before computing statistics, the initial steps before convergence should be dropped: these steps are referred to as “burn-in” period.
- The burn-in period is estimated from the autocorrelation as the step at which it drops to and becomes oscillatory around zero: in this case it is about 3000 steps.



Chain for ξ_2 showing the bimodality.



Autocovariance for chain of ξ_2.



Chain samples after omitting 3000 steps.

Table of content

Sampling strategies

- Given a vector of QoI (quantity of interest) $S \in \mathbb{R}^n$, we would like to obtain **probabilistic information** regarding S.
- We have seen that the mean and covariance of S are given by

$$\mathbb{E}[S] = S_0, \quad \mathbb{E}\left[SS^T\right] \in \mathbb{R}^{n\times n}.$$

- Higher moments can also be derived, in particular **coefficient of variation, skewness & flatness factors,...**
- No direct mean to assess probabilities of events, for instance

$$\mathrm{P}(\mathrm{a} \leq \mathrm{S_i} < \mathrm{b}), \quad \mathrm{P}(\mathrm{S} > \mathrm{s}), \quad \mathrm{P}(\{\mathrm{S_i} < \mathrm{a}\} \cap \{\mathrm{S_j} > \mathrm{b}\}), \quad \ldots$$

- Such probabilities must be estimated by **sampling strategies**.

Sampling strategies

- Assume we want to estimate the probability of the event $S \in \mathcal{R}$:

$$\mathrm{P}(S \in \mathcal{R}).$$

We can apply the generic recipe:

1. generate a sample set $\{\boldsymbol{\xi}^{(i)}, i = 1, \ldots, M\}$ of $\boldsymbol{\xi}$ where $\boldsymbol{\xi}^{(i)} \sim p_{\boldsymbol{\xi}}$,
2. construct the sample set $\{S^{(i)} \doteq S(\boldsymbol{\xi}^{(i)}), i = 1, \ldots, M\}$ **solving the model**,
3. estimate the probability using the empirical estimator by

$$\mathrm{P}(S \in \mathcal{R}) = \lim_{M \to \infty} \frac{\#\text{of samples } S^{(i)} \in \mathcal{R}}{M}.$$

In other words, we make use of the **relative frequency** to estimate probabilities.

- This approach raises several concerns, regarding convergence, estimation of low probability events, ...
- **Observe:** the empirical estimates are random variables, since they are based on a random sample set.

Density estimation

- We denote $\mathbb{S}$ the sample space, that is $\mathrm{P}(S \in \mathbb{S}) = 1$, and assume that S has a (smooth) density denoted π

$$\pi : \mathbb{S} \mapsto \mathbb{R}_+, \qquad \int_{\mathbb{S}} \pi(s)ds = 1.$$

- The question becomes:

how to approximate π from a sample set $\{S^{(i)}\}$?

The simplest density estimation: histogram method

- Consider the **1-d** case first, that is $n = 1$ and $\mathbb{S} \subset \mathbb{R}$.
- We partition $\mathbb{S}$ into uniform **bins** of size h.
- Let $b_i = [x_i - h/2, x_i + h/2)$ be the bin centered on x_i, and define

$$p_i = \frac{\#\text{of samples } S^{(i)} \in b_i}{M},$$

  the relative frequency of the i-th bin.
- **Observe:** $\sum_i p_i = 1$.
- The density is then estimated by

$$\pi(s) \approx \pi_h(s) = \frac{1}{h} \sum_i p_i \mathbb{I}_{b_i}(s).$$

- π_h **is a piecewise constant approximation of** π, satisfying

$$\forall s \in \mathbb{S}, \pi_h(s) \geq 0, \quad \int \pi_h(s) ds = \sum_i \int_{b_i} \pi_h(s) ds = \sum_i \frac{1}{h} p_i h = 1.$$

Histogram method

- The histogram method is simple and intuitive.
- It can be easily extended to $S \subset \mathbb{R}^n$, using for instance hyper-rectangular bins or any other partition of $\mathbb{S}$.

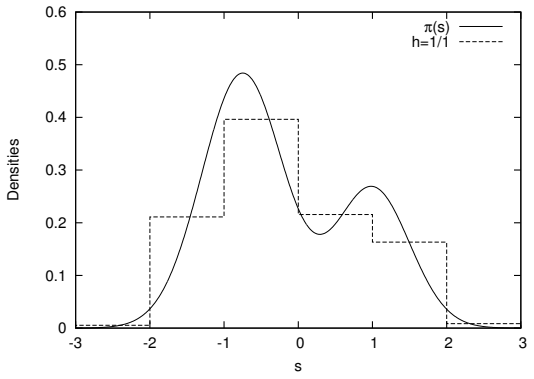
But:

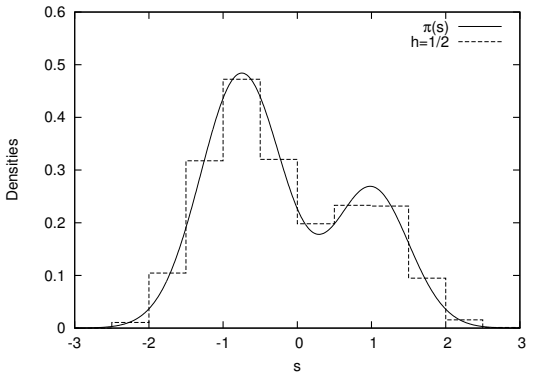
- the approximation depends on the position (centroids x_i) of the bins (orientation too),
- it is susceptible to artifacts (choice of the bins size, outliers,...),
- control of accuracy is difficult in high dimension.

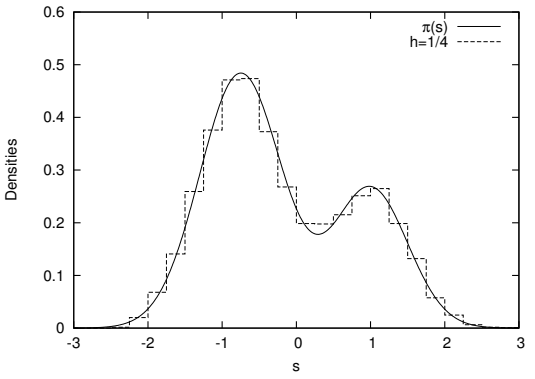
A less arbitrary and more robust approach is needed.

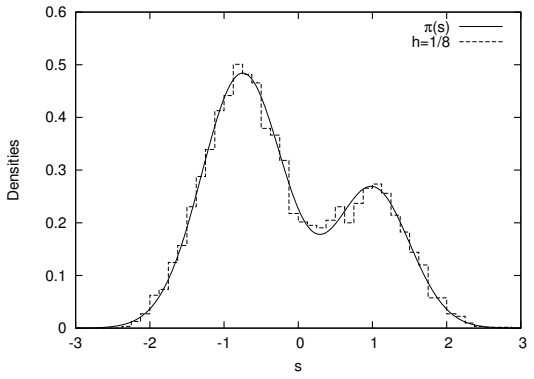
Histogram method

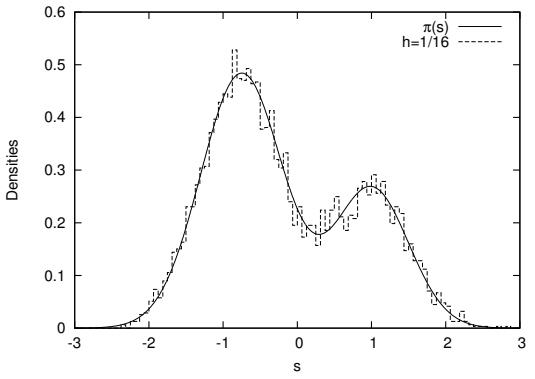
Example : Gaussian mixture for 5000 samples

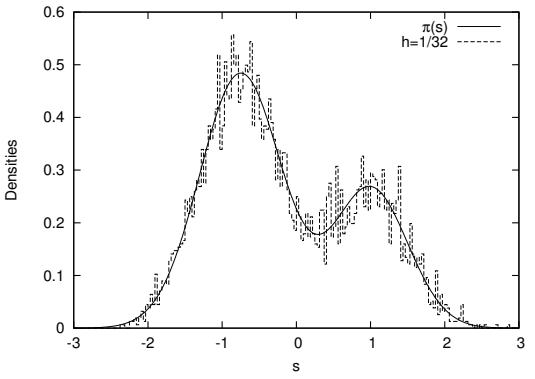












Kernel density estimation (KDE)

- Recall the definition of the probability that S falls in some region $\mathcal{R}$:

$$\mathrm{P}(\mathrm{S} \in \mathcal{R}) = \int_{\mathcal{R}} \pi(s)ds \doteq \mathrm{P}_{\mathcal{R}}.$$

- Then, if we have M vectors independently drawn at random from π, the probability that **k** of these vectors fall in $\mathcal{R}$ is

$$\mathrm{P}(\mathrm{k}|\mathrm{M}) = \begin{pmatrix} \mathrm{M} \\ \mathrm{k} \end{pmatrix} \mathrm{P}_{\mathcal{R}}^{k}(1 - \mathrm{P}_{\mathcal{R}})^{M-k}.$$

- It can be shown that the mean and variance of the ratio k/M are

$$\mathbb{E}\left[k/M\right] = \mathrm{P}_{\mathcal{R}}, \quad \mathbb{V}\left[k/M\right] = \mathbb{E}\left[\left(\frac{k}{M} - \mathrm{P}_{\mathcal{R}}\right)^2\right] = \frac{\mathrm{P}_{\mathcal{R}}(1 - \mathrm{P}_{\mathcal{R}})}{M}.$$

Therefore, as $M \to \infty$ becomes large the mean fraction of the points falling within $\mathcal{R}$ is (as one would expect)

$$\frac{k}{M} = \mathrm{P}_{\mathcal{R}}.$$

Non-parametric density estimation

- If $\mathcal{R}$ is small enough, π should not vary much over $\mathcal{R}$ and one would expect

$$\mathrm{P}_{\mathcal{R}} = \int_{\mathcal{R}} \pi(s')ds' \approx \pi(s)V,$$

  $\forall s \in \mathcal{R}$ and where V **is the volume of** $\mathcal{R}$.

- Consequently, we shall consider the approximation

$$\pi(s \in \mathcal{R}) \approx \frac{k}{MV}.$$

Of course, we want V small enough for the constant approximation to be valid, and we need M large enough for the limit k/M to make sense. Clearly, M should increase as V is decreased. **This calls for a compromise.**

Parzen window

Consider

$$K(u) = \begin{cases} 1 & \text{if } |u| < 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

(if n dimensions, $K(u) = 1$ if $|u_j| < 1/2$ for $j = 1, \ldots, n$).
In words, $K = 1$ for u in the **unit hypercube centered at the origin**.
This function is known as the **Parzen window.**

- The quantity

$$K\left(\frac{s - S^{(i)}}{h}\right)$$

  is equal to 1 if the random sample $S^{(i)}$ is inside the hypercube of side h centered at s.

- **The total number** of samples inside this hypercube is

$$k = \sum_{i=1}^{M} K\left(\frac{s - S^{(i)}}{h}\right).$$

Parzen window

Consider

$$K(u) = \begin{cases} 1 & \text{if } |u| < 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

(if n dimensions, $K(u) = 1$ if $|u_j| < 1/2$ for $j = 1, \ldots, n$).
In words, $K = 1$ for u in the **unit hypercube centered at the origin**.
This function is known as the **Parzen window.**

- The quantity

$$K\left(\frac{s - S^{(i)}}{h}\right)$$

  is equal to 1 if the random sample $S^{(i)}$ is inside the hypercube of side h centered at s.

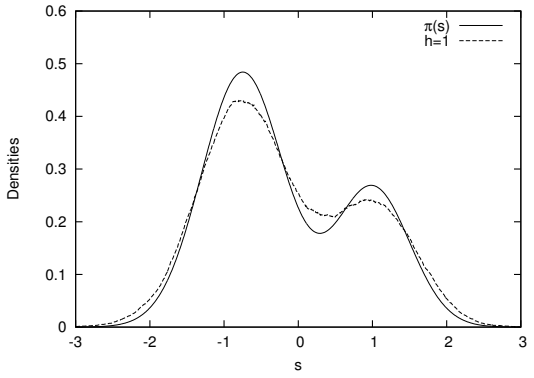
- Consequently, we shall consider the approximation

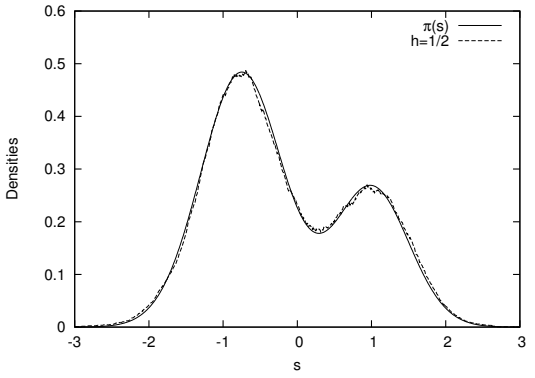
$$\pi_{KDE}(s) = \frac{1}{Mh^n} \sum_{i=1}^{M} K\left(\frac{s - S^{(i)}}{h}\right).$$

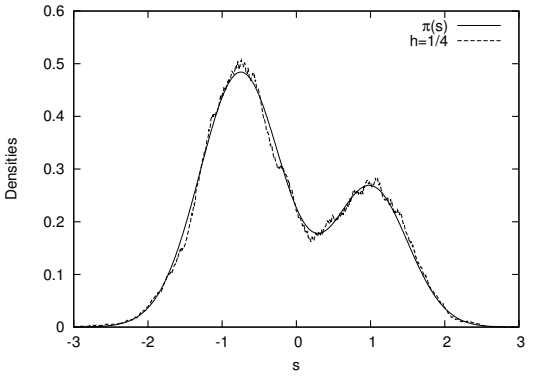
  Parzen Window resembles histogram, except bin are centered on s.

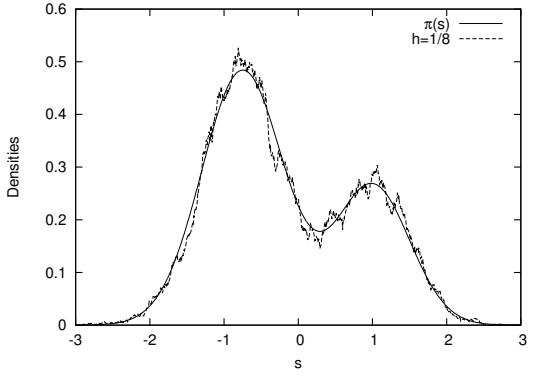
KDE using Parzen window

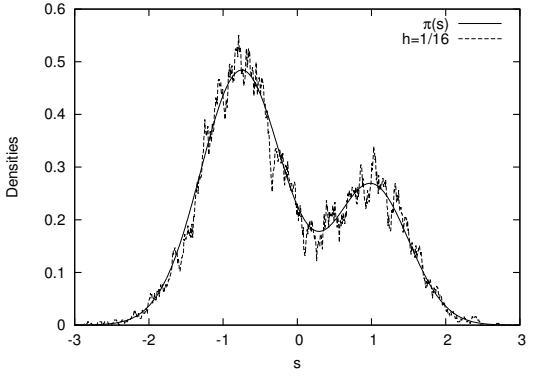
Example : Gaussian mixture for 5000 samples

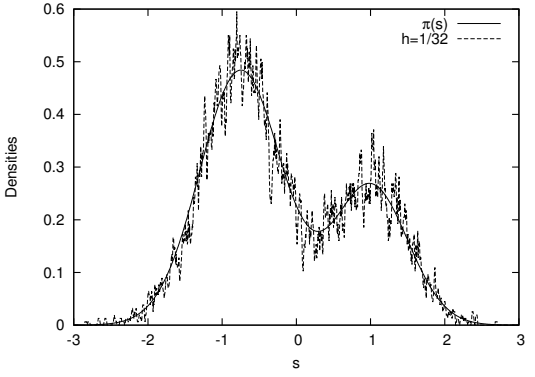












Role of the kernel

To understand the role of K, compute the **expectation** of π_{KDE}

$$\mathbb{E}\left[\pi_{KDE}(s)\right] = \frac{1}{h^n}\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}\left[K\left(\frac{s-S^{(i)}}{h}\right)\right] = \frac{1}{h^n}\int_{\mathbb{S}} K\left(\frac{s-s'}{h}\right)\pi(s')ds'.$$

$\mathbb{E}\left[\pi_{KDE}(s)\right]$ is equal to the convolution of the true density (π) with the kernel K.
As $h \to 0$, the kernel goes to δ (Dirac delta-function), in the sense of distribution:

$$\pi_{KDE}(s) \to \pi(s).$$

- This observation paves the way to **better choice of kernel**, in particular using **smooth** functions $K(s)$, with the property
$$\int K(s)ds = 1.$$
- Typically $K : \mathbb{S} \mapsto \mathbb{R}$ are chosen as radially symmetric, positive and unimodal.

Role of the kernel

To understand the role of K, compute the **expectation** of π_{KDE}

$$\mathbb{E}\left[\pi_{KDE}(s)\right] = \frac{1}{h^n}\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}\left[K\left(\frac{s-S^{(i)}}{h}\right)\right] = \frac{1}{h^n}\int_{\mathbb{S}} K\left(\frac{s-s'}{h}\right)\pi(s')ds'.$$

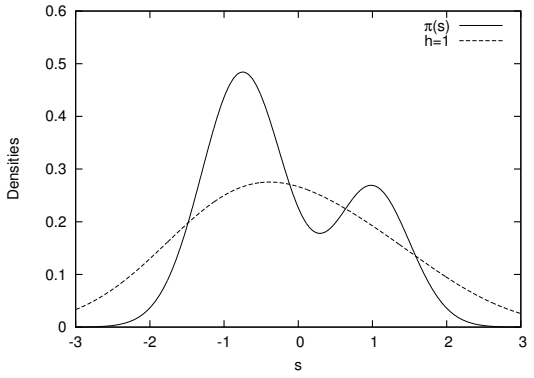
$\mathbb{E}\left[\pi_{KDE}(s)\right]$ **is equal to the convolution of the true density (π) with the kernel** K.
As $h \to 0$, the kernel goes to δ (Dirac delta-function), in the sense of distribution:

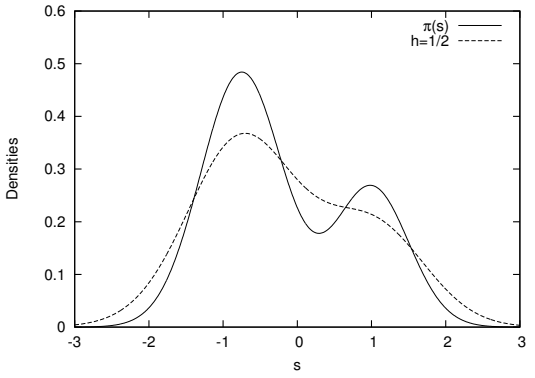
$$\pi_{KDE}(s) \to \pi(s).$$

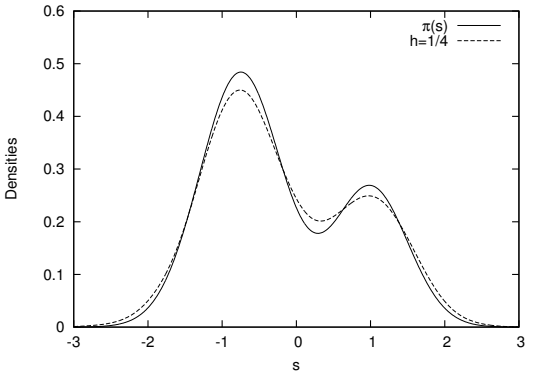
- This observation paves the way to **better choice of kernel**, in particular using **smooth** functions $K(s)$, with the property
$$\int K(s)ds = 1.$$
- A common choice is **Gaussian Kernel**:
$$K(s) = \frac{1}{(2\pi)^{n/2}}\exp\left[-(s^T s)/2\right].$$

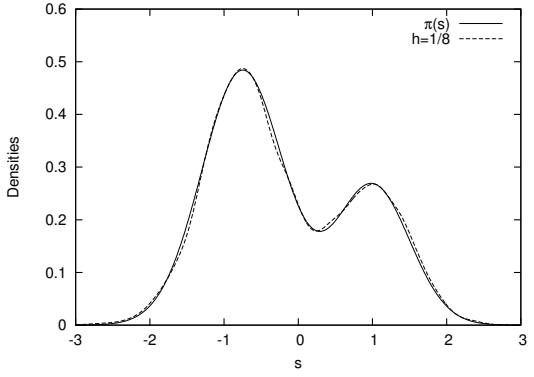
KDE with Gaussian kernel

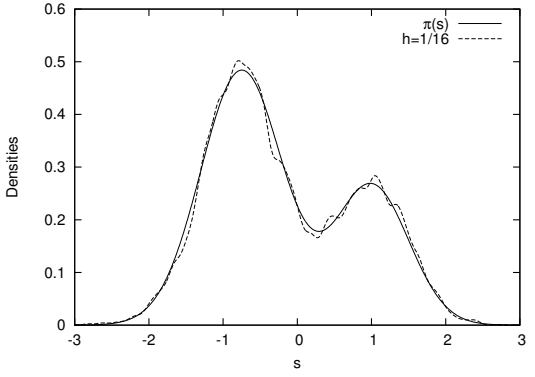
Example : Gaussian mixture for 5000 samples

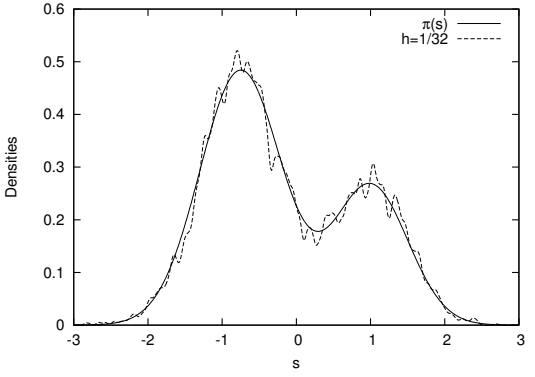












Smoothing parameter

$$\pi_{KDE}(s) = \frac{1}{Mh^n} \sum_{i=1}^{M} K\left(\frac{s - S^{(i)}}{h}\right).$$

It remains to fix h (bandwidth, core-radius), which is a **critical parameter**:

- large h: too much smoothing,
- small h: too many spikes.

The best h minimizes the error between the estimated and true densities.
Using **mean square error measure**, we get

$$\epsilon^2 \doteq \mathbb{E}\left[(\pi_{KDE}(s) - \pi(s))^2\right] = \mathbb{E}\left[\pi_{KDE}(s) - \pi(s)\right]^2 + \mathbb{V}\left[\pi_{KDE}(s)\right].$$

Bias-variance tradeoff:

- large h: reduces variance **but increase bias**
- small h: reduces bias **but increase variance**

How to pick h?

Setting of the smoothing parameter

- **A priori selection.** Assumes that the true distribution π is Gaussian, and K is the Gaussian kernel, one can get an explicit minimizer for ϵ:

$$h_{opt} = 1.06\sigma M^{-1/5},$$

  where σ is the standard deviation of the distribution π.

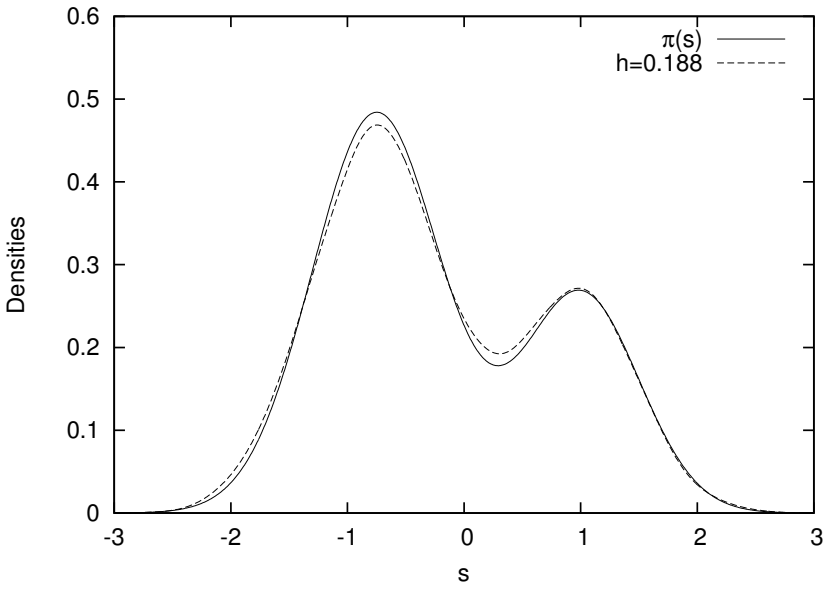
- **Data based selection.** For general distributions, better results are obtained using

$$h_{data} = 0.9AM^{-1/5}, \quad A = \min\left(\sigma, \frac{\text{IQR}}{1.34}\right),$$

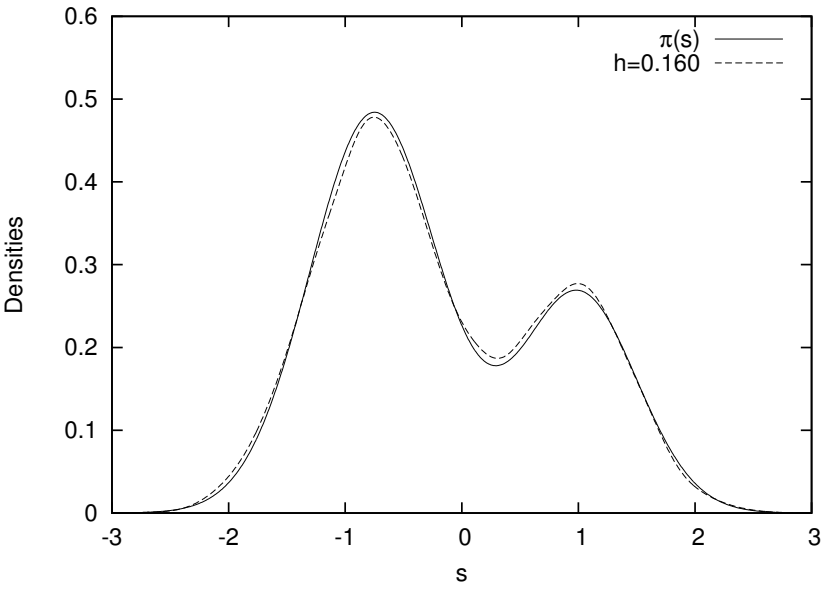
  where IQR is the inter-quantile range, defined as the difference between the 75% and 25% percentiles.

KDE with Gaussian kernel

Example : Gaussian mixture for 5000 samples



Gaussian assumption



IQR based

Closing remarks:

- This ideas can be extended to the multivariate case (isotropic kernels)
- Use of variable *h* to adapt local concentration of observations
- Many implementations available (MATLAB).

Table of content

Assumptions

L_2 functions over unit-hypercubes

Let $L_2(\mathcal{U}^d)$ be the space of real-valued **squared-integrable functions** over the d-dimensional hypercube $\mathcal{U}$:

$$f : \boldsymbol{x} \in \mathcal{U}^d \mapsto f(\boldsymbol{x}) \in \mathbb{R}, \quad f \in L_2(\mathcal{U}^d) \Leftrightarrow \int_{\mathcal{U}^d} f(\boldsymbol{x})^2 d\boldsymbol{x} < \infty.$$

$L_2(\mathcal{U}^d)$ is equipped with the inner product $\langle \cdot, \cdot \rangle$,

$$\forall f, g \in L_2(\mathcal{U}^d), \quad \langle u, v \rangle := \int_{\mathcal{U}^d} f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x},$$

and norm $\| \cdot \|_2$,

$$\forall f \in L_2(\mathcal{U}^d), \quad \|f\|_2 := \langle f, f \rangle^{1/2}.$$

Assumptions

L_2 functions over unit-hypercubes

Let $L_2(\mathcal{U}^d)$ be the space of real-valued **squared-integrable functions** over the d-dimensional hypercube $\mathcal{U}$:

$$f : \boldsymbol{x} \in \mathcal{U}^d \mapsto f(\boldsymbol{x}) \in \mathbb{R}, \quad f \in L_2(\mathcal{U}^d) \Leftrightarrow \int_{\mathcal{U}^d} f(\boldsymbol{x})^2 d\boldsymbol{x} < \infty.$$

NB: all subsequent developments immediately extend to product-type situations, where

$$\boldsymbol{x} \in A = A_1 \times \cdots \times A_d \subseteq \mathbb{R}^d,$$

and weighted spaces $L_2(A, \rho)$,

$$\rho : \boldsymbol{x} \in A \mapsto \rho(\boldsymbol{x}) \geq 0, \quad \rho(\boldsymbol{x}) = \rho_1(x_1) \times \cdots \times \rho_d(x_d).$$

(*e.g.*: ρ is a pdf of a random vector $\boldsymbol{x}$ with mutually independent components.)

Ensemble notations

Let $\mathfrak{D} = \{1, 2, \ldots, d\}$.
Given $\mathfrak{i} \subseteq \mathfrak{D}$, we denote $\mathfrak{i}_\sim := \mathfrak{D} \setminus \mathfrak{i}$ **its complement set in** $\mathfrak{D}$, such that

$$\mathfrak{i} \cup \mathfrak{i}_\sim = \mathfrak{D}, \quad \mathfrak{i} \cap \mathfrak{i}_\sim = \emptyset.$$

For instance

- $\mathfrak{i} = \{1, 2\}$ and $\mathfrak{i}_\sim = \{3, \ldots, d\}$,
- $\mathfrak{i} = \mathfrak{D}$ and $\mathfrak{i}_\sim = \emptyset$.

Ensemble notations

Let $\mathfrak{D} = \{1, 2, \ldots, d\}$.
Given $\mathfrak{i} \subseteq \mathfrak{D}$, we denote $\mathfrak{i}_\sim := \mathfrak{D} \setminus \mathfrak{i}$ **its complement set in** $\mathfrak{D}$, such that

$$\mathfrak{i} \cup \mathfrak{i}_\sim = \mathfrak{D}, \quad \mathfrak{i} \cap \mathfrak{i}_\sim = \emptyset.$$

Given $\boldsymbol{x} = (x_1, \ldots, x_d)$, we denote $\boldsymbol{x}_\mathfrak{i}$ the vector having for components the $x_{i \in \mathfrak{i}}$, that is

$$\mathfrak{D} \supseteq \mathfrak{i} = \{i_1, \ldots, i_{|\mathfrak{i}|}\} \Rightarrow \boldsymbol{x}_\mathfrak{i} = (x_{i_1}, \ldots, x_{i_{|\mathfrak{i}|}}),$$

where $|\mathfrak{i}| := \mathrm{Card}(\mathfrak{i})$. For instance

$$\int_{\mathcal{U}^{|\mathfrak{i}|}} f(\boldsymbol{x}) d\boldsymbol{x}_\mathfrak{i} = \int_{\mathcal{U}^{|\mathfrak{i}|}} f(x_1, \ldots, x_d) \prod_{i \in \mathfrak{i}} dx_i,$$

and

$$\int_{\mathcal{U}^{d-|\mathfrak{i}|}} f(\boldsymbol{x}) d\boldsymbol{x}_{\mathfrak{i}_\sim} = \int_{\mathcal{U}^{d-|\mathfrak{i}|}} f(x_1, \ldots, x_d) \prod_{i \in \mathfrak{D}}^{i \notin \mathfrak{i}} dx_i,$$

Sobol-Hoeffding decomposition

Any $f \in L_2(\mathcal{U}^d)$ has a **unique hierarchical orthogonal decomposition** of the form

$$f(\boldsymbol{x}) = f(x_1, \ldots, x_d) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i=1}^{d} \sum_{j=i+1}^{d} f_{i,j}(x_i, x_j) + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \sum_{k=j+1}^{d} f_{i,j,k}(x_i, x_j, x_k) + \cdots + f_{1,\ldots,d}(x_1, \ldots, x_d).$$

Hierarchical: 1st order functionals $(f_i) \rightarrow$ 2nd order functionals $(f_{i,j}) \rightarrow$ 3rd order functionals $(f_{i,j,l}) \rightarrow \cdots \rightarrow d$-th order functional $(f_{1,\ldots,d})$.

Decomposition in a sum of 2^k functionals

Using ensemble notations:

$$f(\boldsymbol{x}) = \sum_{\mathfrak{i} \subseteq \mathfrak{D}} f_{\mathfrak{i}}(\boldsymbol{x}_{\mathfrak{i}}).$$

Sobol-Hoeffding decomposition

Any $f \in L_2(\mathcal{U}^d)$ has a **unique hierarchical orthogonal decomposition** of the form

$$f(\boldsymbol{x}) = \sum_{\mathfrak{i} \subseteq \mathfrak{D}} f_{\mathfrak{i}}(\boldsymbol{x}_{\mathfrak{i}}).$$

Orthogonal: the functionals if the S-H decomposition verify the following orthogonality relations:

$$\int_{\mathcal{U}} f_{\mathfrak{i}}(\boldsymbol{x}_{\mathfrak{i}}) dx_j = 0, \qquad \forall \mathfrak{i} \subseteq \mathfrak{D},\ j \in \mathfrak{i},$$

$$\int_{\mathcal{U}^d} f_{\mathfrak{i}}(\boldsymbol{x}_{\mathfrak{i}}) f_{\mathfrak{j}}(\boldsymbol{x}_{\mathfrak{j}}) d\boldsymbol{x} = \langle f_{\mathfrak{i}}, f_{\mathfrak{j}} \rangle = 0, \qquad \forall \mathfrak{i}, \mathfrak{j} \subseteq \mathfrak{D},\ \mathfrak{i} \neq \mathfrak{j}.$$

It follows the hierarchical construction

$$f_{\emptyset} = \int_{\mathcal{U}^d} f(\boldsymbol{x}) d\boldsymbol{x} = \langle f \rangle_{\emptyset_\sim = \mathfrak{D}}$$

$$f_{\{i\}} = \int_{\mathcal{U}^{d-1}} f(\boldsymbol{x}) d\boldsymbol{x}_{\{i\}_\sim} - f_{\emptyset} = \langle f \rangle_{\mathfrak{D} \setminus \{i\}} - f_{\emptyset} \qquad i \in \mathfrak{D}$$

$$f_{\mathfrak{i}} = \int_{\mathcal{U}^{|\mathfrak{i}_\sim|}} f(\boldsymbol{x}) d\boldsymbol{x}_{\mathfrak{i}_\sim} - \sum_{\mathfrak{j} \subsetneq \mathfrak{i}} f_{\mathfrak{j}} = \langle f \rangle_{\mathfrak{i}_\sim} - \sum_{\mathfrak{j} \subsetneq \mathfrak{i}} f_{\mathfrak{j}} \qquad \mathfrak{i} \in \mathfrak{D},\ |\mathfrak{i}| \geq 2.$$

Parametric sensitivity analysis

Consider $\boldsymbol{x}$ as a set of d independent random parameters uniformly distributed on $\mathcal{U}^d$, and $f(\boldsymbol{x})$ a model-output depending on these random parameters. It is assumes that f is a 2nd order random variable: $f \in L_2(\mathcal{U}^d)$. Thus, f has a unique S-H decomposition

$$f(\boldsymbol{x}) = \sum_{\mathfrak{i} \subseteq \mathfrak{D}} f_{\mathfrak{i}}(\boldsymbol{x}_{\mathfrak{i}}).$$

Further, the integrals of f with respect to $\mathfrak{i}_\sim$ are in this context conditional expectations,

$$\mathbb{E}\left[f|\boldsymbol{x}_{\mathfrak{i}}\right] = \int_{\mathcal{U}^{|\mathfrak{i}_\sim|}} f(\boldsymbol{x}) d\boldsymbol{x}_{\mathfrak{i}_\sim} = g(\boldsymbol{x}_{\mathfrak{i}}) \quad \forall \mathfrak{i} \subseteq \mathfrak{D},$$

so the S-H decomposition follows the hierarchical structure

$$\begin{aligned} f_\emptyset &= \mathbb{E}\left[f\right] \\ f_{\{i\}} &= \mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right] - \mathbb{E}\left[f\right] && i \in \mathfrak{D} \\ f_{\mathfrak{i}} &= \mathbb{E}\left[f|\boldsymbol{x}_{\mathfrak{i}}\right] - \sum_{\mathfrak{j} \subsetneq \mathfrak{i}} f_{\mathfrak{j}} && \mathfrak{i} \subseteq \mathfrak{D},\ |\mathfrak{i}| \geq 2. \end{aligned}$$

Variance decomposition

Because of the orthogonality of the S-H decomposition the variance $\mathbb{V}[f]$ of the model-output can be decomposed as

$$\mathbb{V}[f] = \sum_{\mathfrak{i} \subseteq \mathfrak{D}}^{\mathfrak{i} \neq \emptyset} \mathbb{V}[f_{\mathfrak{i}}], \quad \mathbb{V}[f_{\mathfrak{i}}] = \langle f_{\mathfrak{i}}, f_{\mathfrak{i}} \rangle .$$

$\mathbb{V}[f_{\mathfrak{i}}]$ is interpreted as the contribution to the total variance $\mathbb{V}[f]$ of the interaction between parameters $x_{i \in \mathfrak{i}}$.
The S-H decomposition thus provide a rich mean of analyzing the respective contributions of individual or sets of parameters to model-output variability.
However, as there are $2^d - 1$ contributions, so one needs more "abstract" characterizations.

Sensitivity indices

To facilitate the hierarchization of the respective influence of each parameter x_i, the partial variances $\mathbb{V}[f_{\mathfrak{i}}]$ are normalized by $\mathbb{V}[f]$ to obtain the **sensitivity indices**:

$$S_{\mathfrak{i}}(f) = \frac{\mathbb{V}[f_{\mathfrak{i}}]}{\mathbb{V}[f]} \leq 1, \quad \sum_{\mathfrak{i} \subseteq \mathfrak{D}}^{\mathfrak{i} \neq \emptyset} S_{\mathfrak{i}}(f) = 1.$$

The **order of the sensitivity indices** $S_{\mathfrak{i}}$ is equal to $|\mathfrak{i}| = \mathrm{Card}(\mathfrak{i})$.
1st order sensitivity indices. The d **first order indices** $S_{\{i\} \in \mathfrak{D}}$ characterize the **fraction of the variance due the parameter** x_i **only**, *i.e.* **without any interaction with others**. Therefore,

$$1 - \sum_{i=1}^{d} S_{\{i\}}(f) \geq 0,$$

measures globally the effect on the variability of all interactions between parameters. **If** $\sum_{i=1}^{d} S_{\{i\}} = 1$**, the model is said additive**, because its S-H decomposition is

$$f(x_i, \ldots, x_d) = f_0 + \sum_{i=1}^{d} f_i(x_i),$$

and the impact of the parameters can be studied separately.

Sensitivity indices

To facilitate the hierarchization of the respective influence of each parameter x_i, the partial variances $\mathbb{V}[f_{\mathfrak{i}}]$ are normalized by $\mathbb{V}[f]$ to obtain the **sensitivity indices**:

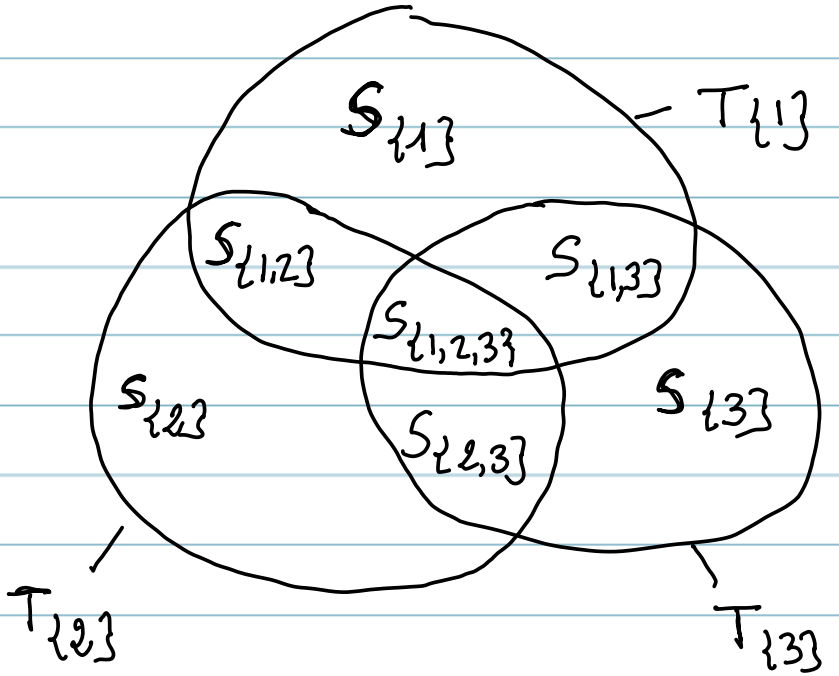
$$S_{\mathfrak{i}}(f) = \frac{\mathbb{V}[f_{\mathfrak{i}}]}{\mathbb{V}[f]} \leq 1, \quad \sum_{\mathfrak{i} \subseteq \mathfrak{D}}^{\mathfrak{i} \neq \emptyset} S_{\mathfrak{i}}(f) = 1.$$

The **order of the sensitivity indices** $S_{\mathfrak{i}}$ is equal to $|\mathfrak{i}| = \mathrm{Card}(\mathfrak{i})$.
Total sensitivity indices. The first order SI $S_{\{i\}}$ measures the variability due to parameter x_i alone.
The **total SI $T_{\{i\}}$ measures the variability due to the parameter x_i, including all its interactions** with other parameters:

$$T_{\{i\}} := \sum_{\mathfrak{i} \ni i} S_{\mathfrak{i}} \geq S_{\{i\}}.$$

Important point: for x_i to be deemed non-important or non-influent on the model-output, $S_{\{i\}}$ **and** $T_{\{i\}}$ have to be negligible.
Observe that $\sum_{i \in \mathfrak{D}} T_{\{i\}} \geq 1$, the excess from 1 **characterizes the presence of interactions** in the model-output.

Illustration



Case of $d=3$.

$$\sum S_i = 1$$

$$\sum S_{\{i\}} \leq 1$$

$$\sum T_{\{i\}} \geqslant 1$$

Sensitivity indices

In many uncertainty problem, the **set of uncertain parameters can be naturally grouped into subsets** depending on the process each parameter accounts for. For instance, boundary conditions BC, material property φ, external forcing F, and $\mathfrak{D}$ is the union of these distinct subsets:

$$\mathfrak{D} = \mathfrak{D}_{BC} \cup \mathfrak{D}_{\varphi} \cup \mathfrak{D}_{F}.$$

The notion of **first order and total sensitivity indices can be extended to characterize the influence of the subsets of parameters**. For instance,

$$S_{\mathfrak{D}_{\varphi}} = \sum_{\mathfrak{i} \subseteq \mathfrak{D}_{\varphi}} S_{\mathfrak{i}},$$

measures the fraction of variance induced by the material uncertainty alone, while

$$T_{\mathfrak{D}_{F}} = \sum_{\mathfrak{i} \cap \mathfrak{D}_{F} \neq \emptyset} S_{\mathfrak{i}}.$$

measures the fraction of variance due to the external forcing uncertainty and all its interactions.

Example

Let (ξ_1, ξ_2) be two independent centered, normalized random variables

$$\xi_i \sim \mathrm{N}(0,1), \quad i = 1, 2.$$

Consider the model-output $f : (\xi_1, \xi_2) \in \mathbb{R}^2 \mapsto \mathbb{R}$ given by

$$f(\xi_1, \xi_2) = (\mu_1 + \sigma_1 \xi_1) + (\mu_2 + \sigma_2 \xi_2)\,.$$

1. Determine the S-H decomposition of f
2. Compute the 1st order and total sensitivity indices of f
3. Comment
4. Repeat for $f(\xi_1, \xi_2) = (\mu_1 + \sigma_1 \xi_1)\,(\mu_2 + \sigma_2 \xi_2)$.

Example

$$f(\xi_1, \xi_2) = \mu_1 + \mu_2 + \sigma_1\xi_1 + \sigma_2\xi_2$$

- $\mathbb{E}[f] = (\mu_1 + \mu_2)$
- $\mathbb{E}[f|\xi_1] = (\mu_1 + \mu_2) + \sigma_1\xi_1$

$$\Rightarrow f_1(\xi_1) = \mathbb{E}[f|\xi_1] - \mathbb{E}[f] = \sigma_1\xi_1$$

- $\mathbb{E}[f|\xi_2] = (\mu_1 + \mu_2) + \sigma_2\xi_2$

$$\Rightarrow f_2(\xi_2) = \mathbb{E}[f|\xi_2] - \mathbb{E}[f] = \sigma_2\xi_2$$

- $\mathbb{E}[f|\xi_1, \xi_2] = f(\xi_1, \xi_2)$

$$\Rightarrow f_{1,2}(\xi_1, \xi_1) = \mathbb{E}[f|\xi_1, \xi_2] - \mathbb{E}[f] - f_1(\xi_1) - f_2(\xi_2) = 0$$

- Then, $\mathbb{V}[f] = \sigma_1^2 + \sigma_2^2$, so

$$S_1 = T_1 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \text{ and } S_2 = T_2 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

- Comment: obvious case, as f is a linear (additive) model.

Example

$$f(\xi_1, \xi_2) = \mu_1\mu_2 + \mu_2\sigma_1\xi_1 + \mu_1\sigma_2\xi_2 + \sigma_1\sigma_2\xi_1\xi_2.$$

- $\mathbb{E}[f] = \mu_1\mu_2$
- $\mathbb{E}[f|\xi_1] = \mu_1\mu_2 + \mu_2\sigma_1\xi_1$

  $\Rightarrow f_1(\xi_1) = \mathbb{E}[f|\xi_1] - \mathbb{E}[f] = \mu_2\sigma_1\xi_1$
- $\mathbb{E}[f|\xi_2] = \mu_1\mu_2 + \mu_1\sigma_2\xi_2$

  $\Rightarrow f_2(\xi_2) = \mathbb{E}[f|\xi_2] - \mathbb{E}[f] = \mu_1\sigma_2\xi_2$
- $\mathbb{E}[f|\xi_1, \xi_2] = f(\xi_1, \xi_2)$

  $\Rightarrow f_{1,2}(\xi_1, \xi_1) = \mathbb{E}[f|\xi_1, \xi_2] - \mathbb{E}[f] - f_1(\xi_1) - f_2(\xi_2) = \sigma_1\sigma_2\xi_1\xi_2$
- Then, $\mathbb{V}[f] = \mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2$, so

$$S_1 = \frac{\mu_2^2\sigma_1^2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2} \text{ and } S_2 = \frac{\mu_1^2\sigma_2^2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2},$$

$$T_1 = \frac{\mu_2^2\sigma_1^2 + \sigma_1\sigma_2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2} \text{ and } T_2 = \frac{\mu_1^2\sigma_2^2 + \sigma_1\sigma_2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2},$$

- Comment: fraction of variance due to interactions is

$$\sigma_1^2\sigma_2^2/(\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2)$$

Example

$$f(\xi_1, \xi_2) = \mu_1\mu_2 + \mu_2\sigma_1\xi_1 + \mu_1\sigma_2\xi_2 + \sigma_1\sigma_2\xi_1\xi_2.$$

- Then, $\mathbb{V}[f] = \mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2$, so

$$S_1 = \frac{\mu_2^2\sigma_1^2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2} \text{ and } S_2 = \frac{\mu_1^2\sigma_2^2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2},$$

$$T_1 = \frac{\mu_2^2\sigma_1^2 + \sigma_1\sigma_2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2} \text{ and } T_2 = \frac{\mu_1^2\sigma_2^2 + \sigma_1\sigma_2}{\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2},$$

- Comment: fraction of variance due to interactions is

$$\sigma_1^2\sigma_2^2/(\mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2)$$

- Example: $(\mu_1, \sigma_1) = (1, 3)$ and $(\mu_2, \sigma_2) = (2, 2)$, so

$$S_1 = 9/19, \quad S_2 = 1/19, \quad S_{1,2} = 9/19.$$

One can draw the conclusions:

- ξ_1 is the most influential variable as $S_1 > S_2$ and $T_1 > T_2$.
- interactions are important as $1 - S_1 - S_2 = 9/19 \approx 0.5$, especially for ξ_2 for which $(T_2 - S_2)/T_2 = 9/10 \approx 1$.

1-st order sensitivity indices by Monte-Carlo sampling

The S_i can be computed by MC sampling as follow. Recall that

$$S_{\{i\}}(f) = \frac{\mathbb{V}\left[f_{\{i\}}\right]}{\mathbb{V}[f]} = \frac{\mathbb{V}\left[\mathbb{E}\left[f|x_i\right]\right)]}{\mathbb{V}[f]} = \frac{\mathbb{E}\left[\mathbb{E}\left[f|x_i\right]^2\right] - \mathbb{E}\left[\mathbb{E}\left[f|x_i\right]\right]^2}{\mathbb{V}[f]}.$$

Observe: $\mathbb{E}\left[\mathbb{E}\left[f|x_i\right]\right] = \mathbb{E}[f]$.
$\mathbb{E}[f]$ and $\mathbb{V}[f]$ can be **estimated using MC sampling**.
Let $\chi_M = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(M)}\}$ be a set of **independent samples drawn uniformly** in $\mathcal{U}^d$, the **mean and variance estimators** are:

$$\widehat{\mathbb{E}[f]} = \frac{1}{M}\sum_{l=1}^{M} f(\boldsymbol{x}^{(l)}), \quad \widehat{\mathbb{V}[f]} = \frac{1}{M-1}\sum_{l=1}^{M} f(\boldsymbol{x}^{(l)})^2 - \widehat{\mathbb{E}[f]}^2.$$

It now remains to compute the variance of conditional expectations, $\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right]$.

Monte-Carlo estimator: variance of conditional expectation

Observe:

$$\int_{\mathcal{U}^{d+|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}'_{\mathrm{i}_\sim}) \mathrm{d}\boldsymbol{x}_\mathrm{i} d\boldsymbol{x}_{\mathrm{i}_\sim} d\boldsymbol{x}_{\mathrm{i}_\sim}{}'$$
$$= \int_{\mathcal{U}^{|\mathrm{i}|}} d\boldsymbol{x}_\mathrm{i} \int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) d\boldsymbol{x}_{\mathrm{i}_\sim} \int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}'_{\mathrm{i}_\sim}) d\boldsymbol{x}'_{\mathrm{i}_\sim}$$
$$= \int_{\mathcal{U}^{|\mathrm{i}|}} d\boldsymbol{x}_\mathrm{i} \left[\int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) d\boldsymbol{x}_{\mathrm{i}_\sim} \right]^2 .$$

Monte-Carlo estimator: variance of conditional expectation

Observe:

$$\begin{aligned}
&\int_{\mathcal{U}^{d+|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}'_{\mathrm{i}_\sim}) \mathrm{d}\boldsymbol{x}_\mathrm{i} d\boldsymbol{x}_{\mathrm{i}_\sim} d\boldsymbol{x}_{\mathrm{i}_\sim}{}' \\
&= \int_{\mathcal{U}^{|\mathrm{i}|}} d\boldsymbol{x}_\mathrm{i} \int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) d\boldsymbol{x}_{\mathrm{i}_\sim} \int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}'_{\mathrm{i}_\sim}) d\boldsymbol{x}'_{\mathrm{i}_\sim} \\
&= \int_{\mathcal{U}^{|\mathrm{i}|}} d\boldsymbol{x}_\mathrm{i} \left[\int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) d\boldsymbol{x}_{\mathrm{i}_\sim} \right]^2 .
\end{aligned}$$

$$\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right] = \mathbb{E}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]^2\right] - \mathbb{E}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right]^2 = \mathbb{E}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]^2\right] - \mathbb{E}\left[f\right]^2$$

Monte-Carlo estimator: variance of conditional expectation

Observe:

$$\int_{\mathcal{U}^{d+|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}'_{\mathrm{i}_\sim}) \mathrm{d}\boldsymbol{x}_\mathrm{i} d\boldsymbol{x}_{\mathrm{i}_\sim} d\boldsymbol{x}_{\mathrm{i}_\sim}{}'$$
$$= \int_{\mathcal{U}^{|\mathrm{i}|}} d\boldsymbol{x}_\mathrm{i} \int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) d\boldsymbol{x}_{\mathrm{i}_\sim} \int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}'_{\mathrm{i}_\sim}) d\boldsymbol{x}'_{\mathrm{i}_\sim}$$
$$= \int_{\mathcal{U}^{|\mathrm{i}|}} d\boldsymbol{x}_\mathrm{i} \left[\int_{\mathcal{U}^{|\mathrm{i}_\sim|}} f(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_{\mathrm{i}_\sim}) d\boldsymbol{x}_{\mathrm{i}_\sim} \right]^2 .$$

$$\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right] = \mathbb{E}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]^2\right] - \mathbb{E}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right]^2 = \mathbb{E}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]^2\right] - \mathbb{E}\left[f\right]^2$$
$$= \lim_{M\to\infty} \frac{1}{M} \sum_{l=1}^{M} f\left(\boldsymbol{x}^{(l)}_{\{i\}_\sim}, \boldsymbol{x}^{(l)}_{\{i\}}\right) f\left(\tilde{\boldsymbol{x}}^{(l)}_{\{i\}_\sim}, \boldsymbol{x}^{(l)}_{\{i\}}\right) - \mathbb{E}\left[f\right]^2$$
$$\left(\text{independent samples } \boldsymbol{x}^{(l)}_{\{i\}_\sim}, \tilde{\boldsymbol{x}}^{(l)}_{\{i\}_\sim} \text{ and } \boldsymbol{x}^{(k)}_{\{i\}}\right)$$

Monte-Carlo estimators for 1st order SI $S_{\{i\}}$

1. Draw 2 independent sample sets, with size M, χ_M and $\tilde{\chi}_M$
2. Compute estimators $\widehat{\mathbb{E}[f]}$ and $\widehat{\mathbb{V}[f]}$ from χ_M (or $\tilde{\chi}_M$) [M model evaluations]
3. For $i = 1, 2, \ldots, d$:
   - Estimate variance of conditional expectation through

$$\mathbb{V}\left[\widehat{\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]}\right] = \frac{1}{M}\sum_{l=1}^{M} f\left(\boldsymbol{x}^{(l)}_{\{i\}\sim}, \boldsymbol{x}^{(l)}_{\{i\}}\right) f\left(\tilde{\boldsymbol{x}}^{(l)}_{\{i\}\sim}, \boldsymbol{x}^{(l)}_{\{i\}}\right) - \widehat{\mathbb{E}[f]}^2$$

[M new model evaluations at $(\tilde{\boldsymbol{x}}^{(l)}_{\{i\}\sim}, \boldsymbol{x}^{(l)}_{\{i\}})$]

   - Estimator of the 1-st order SI:

$$\widehat{S_{\{i\}}(f)} = \frac{\mathbb{V}\left[\widehat{\mathbb{E}\left[f|\xi_{\{i\}}\right]}\right]}{\widehat{\mathbb{V}[f]}}.$$

Requires $(d+1) \times M$ model evaluations.

$$\mathcal{X}_M : \begin{bmatrix} x_1^{(1)} & \dots & x_i^{(1)} & \dots & x_d^{(1)} \\ \vdots & & & & \vdots \\ x_1^{(l)} & \dots & x_i^{(l)} & \dots & x_d^{(l)} \\ \vdots & & & & \vdots \\ x_1^{(M)} & \dots & x_i^{(M)} & \dots & x_d^{(M)} \end{bmatrix} \quad \tilde{\mathcal{X}}_M : \begin{bmatrix} \tilde{x}_1^{(1)} & \dots & \tilde{x}_i^{(1)} & \dots & \tilde{x}_d^{(1)} \\ \vdots & & & & \vdots \\ \tilde{x}_1^{(l)} & \dots & \tilde{x}_i^{(l)} & \dots & \tilde{x}_d^{(l)} \\ \vdots & & & & \vdots \\ \tilde{x}_1^{(M)} & \dots & \tilde{x}_i^{(M)} & \dots & \tilde{x}_d^{(M)} \end{bmatrix}.$$

$$\tilde{\mathcal{X}}_M^{\{i\}} : \begin{bmatrix} \tilde{x}_1^{(1)} & \dots & x_i^{(1)} & \dots & \tilde{x}_d^{(1)} \\ \vdots & & \vdots & & \vdots \\ \tilde{x}_1^{(l)} & \dots & x_i^{(l)} & \dots & \tilde{x}_d^{(l)} \\ \vdots & & \vdots & & \vdots \\ \tilde{x}_1^{(M)} & \dots & x_i^{(M)} & \dots & \tilde{x}_d^{(M)} \end{bmatrix}.$$

$$\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right] \approx \frac{1}{M}\sum_{l=1}^{M} f(\boldsymbol{x}_{\{i\}\sim}, \boldsymbol{x}_{\{i\}\sim}) f(\tilde{\boldsymbol{x}}_{\{i\}\sim}, \boldsymbol{x}_{\{i\}\sim}) - \left(\frac{1}{M}\sum_{l=1}^{M} f(\boldsymbol{x}_{\{i\}\sim}, \boldsymbol{x}_{\{i\}\sim})\right)^2.$$

Total sensitivity indices by Monte-Carlo sampling

The $T_{\{i\}}$ can be computed by MC sampling as follow. Recall that

$$T_{\{i\}}(f) = \sum_{\mathfrak{i} \ni \{i\}} S_{\mathfrak{i}}(f) = 1 - \sum_{\mathfrak{i} \subseteq \mathfrak{D} \setminus \{i\}} S_{\mathfrak{i}}(f) = 1 - \frac{\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}\sim}\right]\right]}{\mathbb{V}[f]}.$$

As for $\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right]$, we can derive the following Monte-Carlo estimator for $\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}\sim}\right]\right]$, using the two independent sample sets χ_M and $\tilde{\chi}_M$

$$\widehat{\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}\sim}\right]\right]} = \frac{1}{M}\sum_{l=1}^{M} f\left(\boldsymbol{x}^{(l)}_{\{i\}\sim}, \boldsymbol{x}^{(l)}_{\{i\}}\right) f\left(\boldsymbol{x}^{(l)}_{\{i\}\sim}, \tilde{\boldsymbol{x}}^{(l)}_{\{i\}}\right) - \mathbb{E}[f]^2,$$

so finally

$$\widehat{T_{\{i\}}(f)} = 1 - \frac{1}{\widehat{\mathbb{V}[f]}}\left(\frac{1}{M}\sum_{l=1}^{M} f\left(\boldsymbol{x}^{(l)}_{\{i\}\sim}, \boldsymbol{x}^{(l)}_{\{i\}}\right) f\left(\boldsymbol{x}^{(l)}_{\{i\}\sim}, \tilde{\boldsymbol{x}}^{(l)}_{\{i\}}\right) - \mathbb{E}[f]^2\right)$$

The MC estimation of the $T_{\{i\}}$ needs $d \times M$ additional model evaluations.

$$\chi_M : \begin{bmatrix} x_1^{(1)} & \dots & x_i^{(1)} & \dots & x_d^{(1)} \\ \vdots & & & & \vdots \\ x_1^{(l)} & \dots & x_i^{(l)} & \dots & x_d^{(l)} \\ \vdots & & & & \vdots \\ x_1^{(M)} & \dots & x_i^{(M)} & \dots & x_d^{(M)} \end{bmatrix} \qquad \tilde{\chi}_M : \begin{bmatrix} \tilde{x}_1^{(1)} & \dots & \tilde{x}_i^{(1)} & \dots & \tilde{x}_d^{(1)} \\ \vdots & & & & \vdots \\ \tilde{x}_1^{(l)} & \dots & \tilde{x}_i^{(l)} & \dots & \tilde{x}_d^{(l)} \\ \vdots & & & & \vdots \\ \tilde{x}_1^{(M)} & \dots & \tilde{x}_i^{(M)} & \dots & \tilde{x}_d^{(M)} \end{bmatrix}.$$

$$\tilde{\chi}_M^{\{i\}} : \begin{bmatrix} x_1^{(1)} & \dots & \tilde{x}_i^{(1)} & \dots & x_d^{(1)} \\ \vdots & & \vdots & & \vdots \\ x_1^{(l)} & \dots & \tilde{x}_i^{(l)} & \dots & x_d^{(l)} \\ \vdots & & \vdots & & \vdots \\ x_1^{(M)} & \dots & \tilde{x}_i^{(M)} & \dots & x_d^{(M)} \end{bmatrix}.$$

$$\mathbb{V}\left[\mathbb{E}\left[f|\boldsymbol{x}_{\{i\}}\right]\right] \approx \frac{1}{M}\sum_{l=1}^{M} f(\boldsymbol{x}_{\{i\}\sim}, \boldsymbol{x}_{\{i\}\sim}) f(\boldsymbol{x}_{\{i\}\sim}, \tilde{\boldsymbol{x}}_{\{i\}\sim}) - \left(\frac{1}{M}\sum_{l=1}^{M} f(\boldsymbol{x}_{\{i\}\sim}, \boldsymbol{x}_{\{i\}\sim})\right)^2.$$

Epidemic model

The system is governed by the set of ODEs*:

$$\dot{S}(t) = -\tau(t)S(t)\left[I(t) + U(t)\right], \tag{1}$$

$$\dot{I}(t) = \tau(t)S(t)\left[I(t) + U(t)\right] - \nu I(t), \tag{2}$$

$$\dot{R}(t) = \nu_1 I(t) - \eta U(t), \tag{3}$$

$$\dot{U}(t) = \nu_2 I(t) - \eta R(t), \tag{4}$$

$$\dot{D}(t) = \nu_1 I(t) - D(t). \tag{5}$$

The parameters are the **transmission rate** $\tau(t) > 0$, the average **infectious times** for the asymptomatic individuals ($\nu^{-1} > 0$) and unreported symptomatic ($\eta^{-1} > 0$), and the fraction $f \in (0, 1)$ that fixes the rates of asymptomatic individuals that become symptomatic and reported ($\nu_1 = f\nu$) or symptomatic and unreported ($\nu_2 = (1 - f)\nu$). The transmission rate τ is expressed as

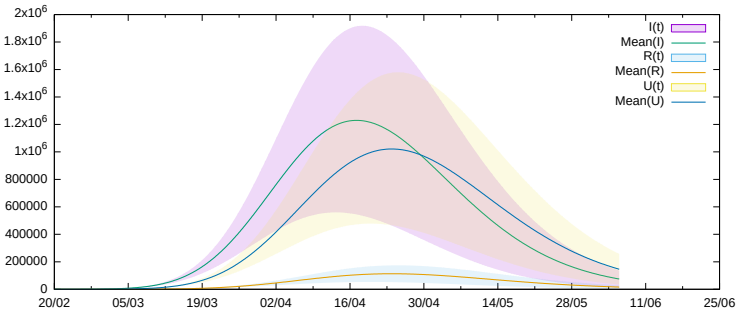
$$\tau(t) = \begin{cases} \tau_0, & t < N_c, \\ \tau_0 \exp[-\mu(t - N_c)], & t \geq N_c, \end{cases} \tag{6}$$

where μ and N_c are uncertain.
A priori range (other parameters are not free):

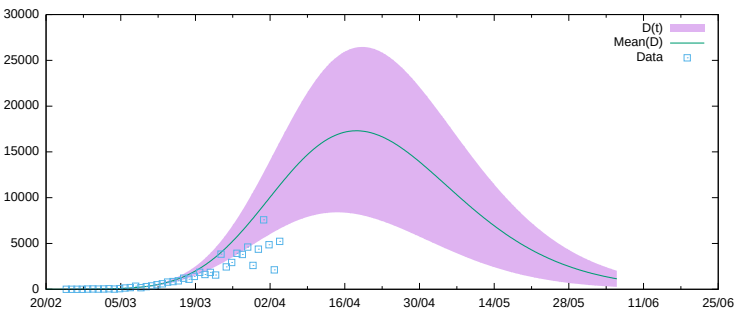
	f	ν^{-1}	η^{-1}	μ	N_c
Nominal values	0.1 or 0.6	7 [days]	7 [day]	0.032	7 [day]
Uncertainty ranges	$\pm$10%	$\pm$1 [day]	$\pm$1 [day]	$\pm$10%	$\pm$1 [day]

* Webb, Predicting the number of reported and unreported cases for the covid-19 epidemic in South France and Germany, MedRXiv, (2020)

Low f ($0.1 \pm 10\%$) - PSP Method, 351 simulations



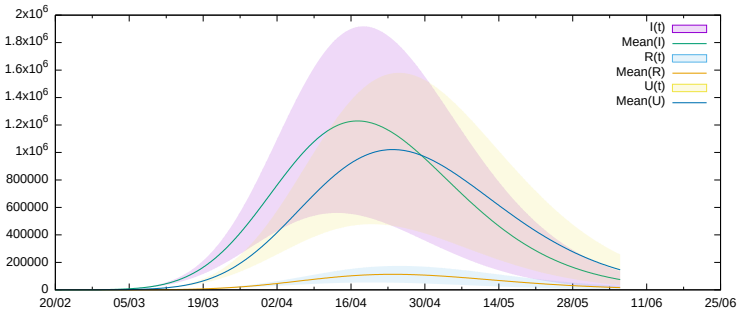
$I(t)$, $R(t)$ and $U(t)$



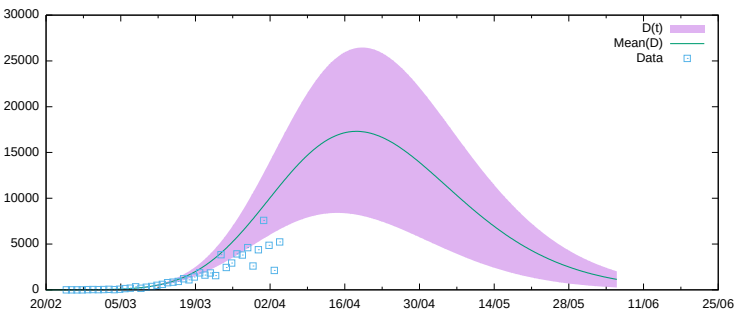
$D(t)$

Mean and standard deviation bounds of the solution.

Low f ($0.1 \pm 10\%$) - PSP Method, 351 simulations



$I(t)$, $R(t)$ and $U(t)$



$D(t)$

Mean and standard deviation bounds of the solution.

Sobol-Hoeffding's decomposition:

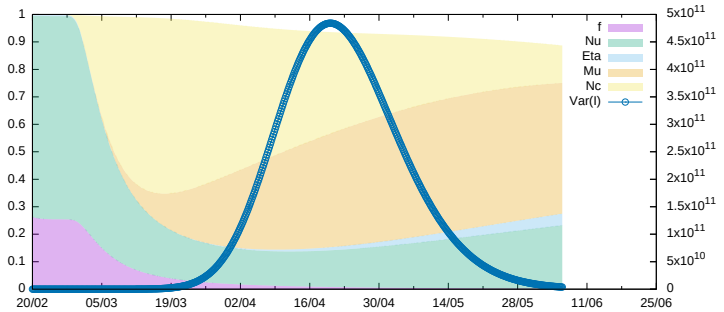
$$U(\boldsymbol{\xi}) = U(\xi_1, \ldots, \xi_N) = \sum_i U_{\{i\}}(\xi_i) + \sum_{i,j>i} U_{\{i,j\}}(\xi_i, \xi_j) + \sum_{i,j>i,k>j} U_{\{i,j,k\}}(\xi_i, \xi_j, \xi_k) \ldots$$

Therefore, the decomposition of the variance

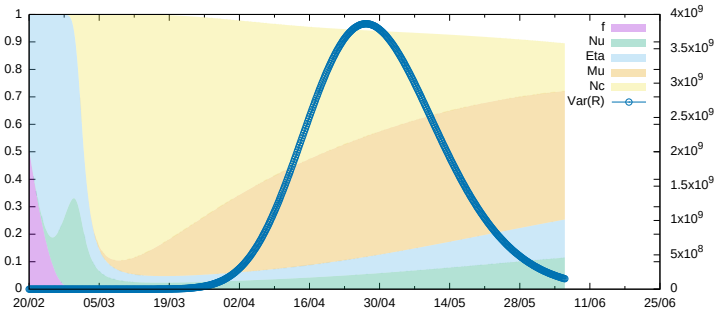
$$\mathbb{V}[U] = \sum_i \mathbb{V}\left[U_{\{i\}}\right] + \sum_{i,j>i} \mathbb{V}\left[U_{\{i,j\}}\right] + \sum_{i,j>i,k>j} \mathbb{V}\left[U_{\{i,j,k\}}\right] \ldots$$

Low f ($0.1 \pm 10\%$)

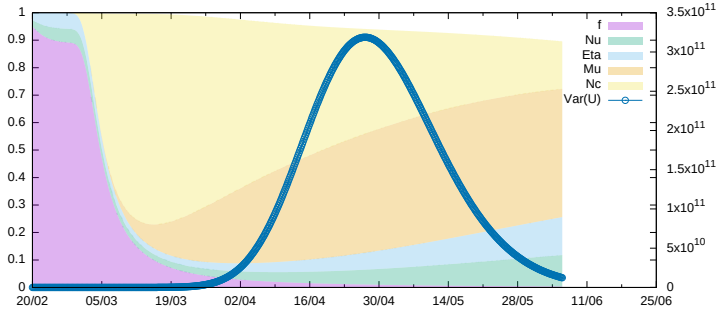
$I(t)$



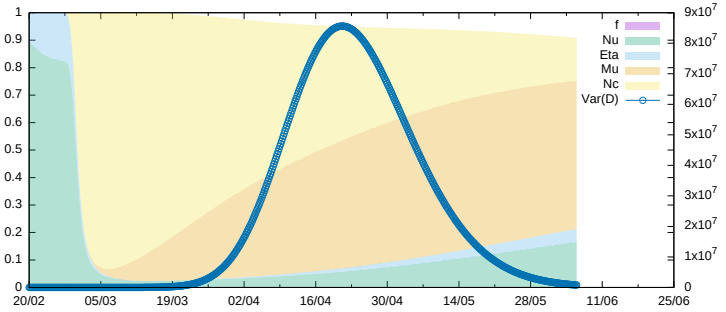
$R(t)$



$U(t)$



$D(t)$



First order sensitivity indices: $\mathbb{V}\left[U_{\{i\}}\right]/\mathbb{V}\left[U\right]$.

MC estimation of 1st order and total sensitivity indices

- requires $M \times (2d + 1)$ **simulations**
- convergence of estimators in $\mathcal{O}(1/\sqrt{M})$
- slow convergence, but d-independent
- convergence not related to smoothness of f