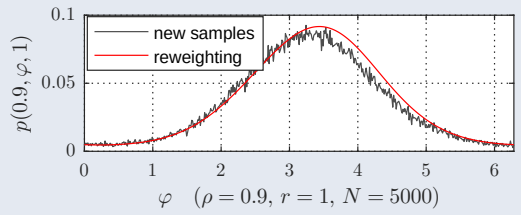
Efficient Computation of Upper Probabilities of Failure Using Monte Carlo Simulation & Reweighting Techniques

Thomas Fetz, Unit for Engineering Mathematics, University of Innsbruck, Austria

Problem statement

Given: Expensive limit state function $g : D \subseteq \mathbb{R}^d \to \mathbb{R} : x \to g(x)$ and family $\{X_\lambda\}_{\lambda\in\Lambda}$ of random variables.
Aim: Upper probability of failure $\overline{p}_f = \sup_{\lambda\in\Lambda} p(\lambda)$ with $p(\lambda) = P(g(X_\lambda) \le 0) = \int_D \mathbb{1}_{g(x)\le 0} f_{X_\lambda}(x)\,\mathrm{d}x$.
Method: Efficient Monte-Carlo simulation using importance sampling and reweighting for approximating $p(\lambda)$.



Classical importance sampling

Given: $X \sim \mathcal{N}(\mu,\Sigma)$ and limit state function g.
Goal: $p_f = P(g(X) \le 0) = \int_D \mathbb{1}_{g(x)\le 0} f_X(x)\,\mathrm{d}x$.
Problem: p_f is very small.
→ Large sample size for ordinary MC simulation.
→ High computational effort in case of expensive g.
Idea: Place sample points where it is "important" and reweight!

1 Transformation from standard normal space

$T : E \subseteq \mathbb{R}^d \to D \subseteq \mathbb{R}^d : \xi \to T(\xi) = C\cdot\xi + \mu$.
C is the Cholesky factor of Σ.

2 Design point

Find $\xi^\diamond \in E \subseteq \mathbb{R}^d$ such that

$$\|\xi^\diamond\| = \min \text{ subject to } g(T(\xi^\diamond)) \le 0.$$

$\xi^\diamond$ is the point in the transformed failure domain which is the closest to the origin.

3 Importance sample

Generate sample $\xi_1,\dots,\xi_N \sim \mathcal{N}(\xi^\diamond, I)$.
Sample points in original space: $x_k = T(\xi_k)$.
$x_1,\dots,x_N \sim \mathcal{N}(T(\xi^\diamond),\Sigma) \sim X^0$.

4 Evaluation of g

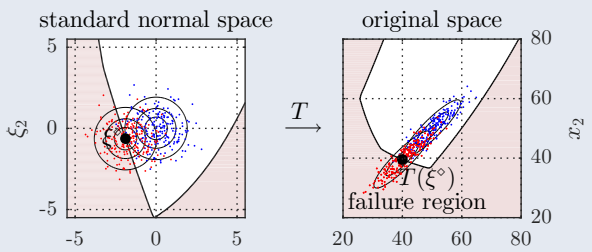
Evaluate limit state function g at sample $x_1,\dots,x_N$.

5 Approximation of p_f by reweighting

$$p_f = P(g(X) \le 0) = \int_D \mathbb{1}_{g(x)\le 0} f_X(x)\,\mathrm{d}x$$
$$= \int_D \mathbb{1}_{g(x)\le 0} \frac{f_X(x)}{f_{X^0}(x)} f_{X^0}(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{k=1}^N \mathbb{1}_{g(x_k)\le 0}\, w_k$$

with weights $w_k = \dfrac{f_X(x_k)}{f_{X^0}(x_k)}$.

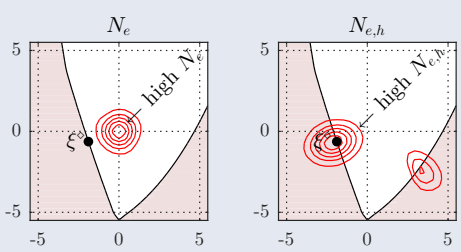
Design point / importance sample



Importance sample and original sample.

Importance sample diagnostics (Owen)

- Effective sample size: $N_e = \dfrac{\left(\sum_{k=1}^N w_k\right)^2}{\sum_{k=1}^N w_k^2}$.
  Best value for $X^0 = X$ (!).
- Taking $h := \mathbb{1}_{g(\cdot)\le 0}$ into account:
  $$N_{e,h} = \frac{1}{\sum_{k=1}^N \tilde{w}_k^2}, \quad \tilde{w}_k = \frac{|h(x_k)|\,w_k}{\sum_{k=1}^N |h(x_k)|\,w_k}.$$
  Only sample points in the failure domain involved.



Values of N_e and $N_{e,h}$ at ξ for $X^0 \sim \mathcal{N}(\xi, I)$.

Simulation of a family of random variables

Goal: Approximation of $p(\lambda) = P(g(X_\lambda) \le 0)$.
Naive approach: Compute $p(\lambda)$ using Monte-Carlo simulations with samples $x_1^\lambda,\dots,x_N^\lambda \sim X_\lambda$ for each λ occurring in the optimization process.
→ High computational cost, non-smooth function.

Better: **Reweighting and importance sampling**.

1 Basic sample $x_1,\dots,x_N$

- Generate a sample $x_1,\dots,x_N$ distributed as a **basic random variable** X^0 with density $f_{X^0} > 0$.
- The sample $x_1,\dots,x_N$ should be an **importance sample** for all X_λ, $\lambda\in\Lambda$, at least for optimal X_{λ^*}.

2 N function evaluations $g(x_k)$

- Evaluate the limit state function g at all sample points $x_1,\dots,x_N$.

3 Approximation of $p(\lambda)$ by reweighting

The probability $p(\lambda) = P(g(X_\lambda) \le 0)$ for $\lambda\in\Lambda$ is approximated by **reweighting** the basic sample:

$$p(\lambda) = P(g(X_\lambda) \le 0) = \int_D \mathbb{1}_{g(x)\le 0} f_{X_\lambda}(x)\,\mathrm{d}x$$
$$\approx \frac{1}{N}\sum_{k=1}^N \mathbb{1}_{g(x_k)\le 0}\, w_k(\lambda) =: p^{X^0}_{x_1,\dots,x_N}(\lambda),$$

with weights $w_k(\lambda) = f_{X_\lambda}(x_k)/f_{X^0}(x_k)$.

The approximation $p^{X^0}_{x_1,\dots,x_N}(\lambda)$ of $p(\lambda)$ depends on X^0 and on the sample points $x_1,\dots,x_N$.

Approximation of $p(\lambda)$ for different $\lambda\in\Lambda$ **without additional function evaluations** of g!

4 Approximation of $\overline{p}_f$

- Solve optimization problem $p^{X^0}_{x_1,\dots,x_N}(\lambda) = \max$ subject to $\lambda\in\Lambda$ using standard optimization procedures.
- Evaluation of $p^{X^0}_{x_1,\dots,x_N}$ is very cheap.
- $p^{X^0}_{x_1,\dots,x_N}$ is continuous if f_{X_λ} is continuous, too.

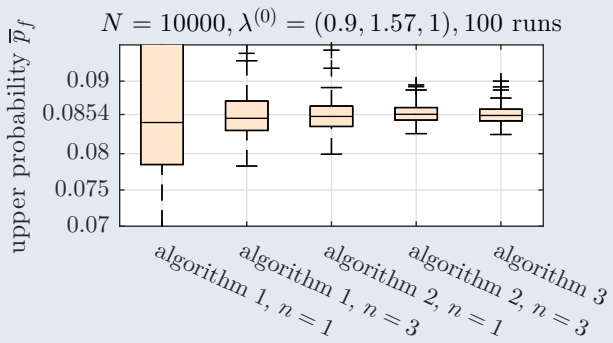
We present three algorithms with different strategies for the choice of the basic sample $x_1,\dots,x_N$.

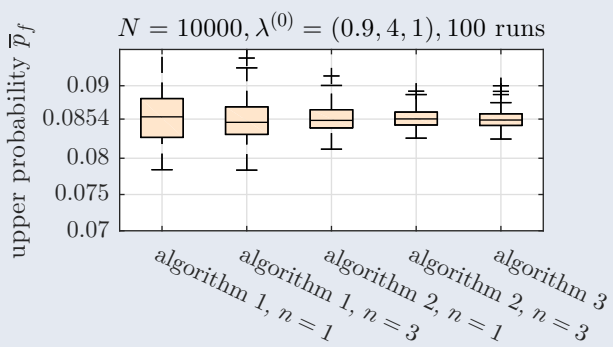
(1) $x_1,\dots,x_N \sim X_\lambda$, $\lambda\in\Lambda$, no importance sampling.
(2) Design point based on single X_λ, $\lambda\in\Lambda$.
(3) Global design point among all X_λ, $\lambda\in\Lambda$.

Testing algorithms

Approximations of upper probabilities obtained from 100 runs of the three algorithms plotted as box plots.

- Sample size: $N = 10000$.
- Starting points: $\lambda^{(0)} = (0.9, \frac{\pi}{2}, 1)$, $\lambda^{(0)} = (0.9, 4, 1)$.
- Number of iterations: $n = 1$ or $n = 3$.

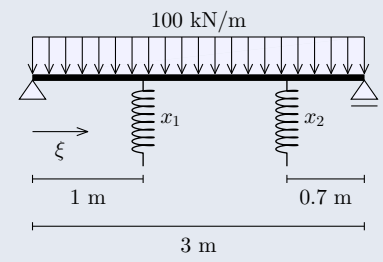


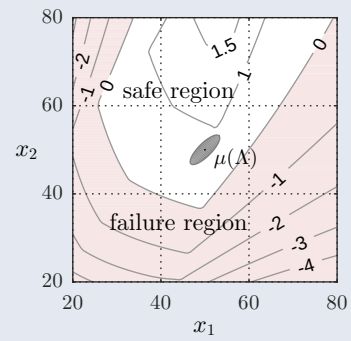


Numerical example

Given: Beam of length 3 m bedded on two springs with uncertain spring constants x_1 and x_2. The beam rigidity $EI = 1\,\mathrm{kNm}^2$, the elastic limit moment $M_{\text{yield}} = 12.3\,\mathrm{kNm}$, and the load $f(\xi) = 100\,\mathrm{kN/m}$ are deterministic.

Limit state function: $g(x_1,x_2) = M_{\text{yield}} - \max_{\xi\in[0,3]} |M(\xi,x_1,x_2)|$. The beam will fail in cases where the moment M exceeds the elastic limit moment M_{yield}.





Uncertain spring constants:

- The spring constants x_1 and x_2 are Gaussian distributed.
- The expectation μ_i of each spring constant x_i is "approximately" $50\,\mathrm{kNm}^{-1}$.
- The values of x_1 and x_2 are strongly correlated, $\rho\in[0.8,0.9]$.
- The uncertainty of x_1 and x_2 is modelled by a family $\{X_\lambda\}_{\lambda\in\Lambda}$ of two-dimensional Gaussian random variables $X_\lambda \sim \mathcal{N}(\mu(\lambda),\Sigma(\lambda))$.
- Expectation μ and covariance Σ are parametrised by
  $\lambda = (\lambda_1,\lambda_2,\lambda_3) = (\rho,\varphi,r) \in \Lambda$ with set $\Lambda = [0.8,0.9]\times[0,2\pi]\times[0,1]$.
- $\mu(\lambda) = \begin{bmatrix}\mu_1(\rho,\varphi,r)\\ \mu_2(\rho,\varphi,r)\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\begin{bmatrix}4.5 & 0\\ 0 & 1.5\end{bmatrix}\begin{bmatrix}r\cos\varphi\\ r\sin\varphi\end{bmatrix} + \begin{bmatrix}50\\ 50\end{bmatrix}$
  Transformation μ maps the unit disc onto an ellipse around $(50,50)\,\mathrm{kNm}^{-1}$.
- $\Sigma(\lambda) = \Sigma(\rho,\varphi,r) = \sigma^2\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}$ with $\sigma^2 = 28$.
- Exact $\overline{p}_f = 0.0854$.

Algorithms

Algorithm 1: Simple reweighting

Choose a starting point $\lambda^{(0)}\in\Lambda$, $i = 0$.
Repeat
  Generate N sample points $x_1,\dots,x_N \sim \mathcal{N}(\mu(\lambda^{(i)}),\Sigma(\lambda^{(i)}))$.
  Density function: $f_{\lambda^{(i)}}$ for distribution $\mathcal{N}(\mu(\lambda^{(i)}),\Sigma(\lambda^{(i)}))$.
  Weighting functions: $w_k^{\lambda^{(i)}}(\lambda) = f_\lambda(x_k)/f_{\lambda^{(i)}}(x_k)$ with density f_λ for $\mathcal{N}(\mu(\lambda),\Sigma(\lambda))$.
  Probability function: $p^{\lambda^{(i)}}_{x_1,\dots,x_k}(\lambda) = \frac{1}{N}\sum_{k=1}^N \mathbb{1}_{g(x_k)\le 0}\cdot w_k^{\lambda^{(i)}}(\lambda)$.
  Find $\lambda^*\in\Lambda$ with $p^{\lambda^{(i)}}_{x_1,\dots,x_k}(\lambda^*) = \max$. ($\lambda^{(i)}$ is starting point.)
  $\lambda^{(i+1)} = \lambda^*$. (New optimal λ^* is the starting point in the next step.)
  $i = i+1$.
Until a stopping criterion is satisfied.
Then $\overline{p}_f \approx p^{\lambda^{(i-1)}}_{x_1,\dots,x_k}(\lambda^*)$.

Algorithm 2: Importance sampling using design points

Choose starting points $\lambda^{(0)}\in\Lambda$ and $\xi^{(0)}\in E\subseteq\mathbb{R}^d$, $i = 0$.
Repeat
  Transformation from standard normal space: $T_{\lambda^{(i)}}(\xi) = C(\lambda^{(i)})\,\xi + \mu(\lambda^{(i)})$.
  Find $\xi^\diamond$ such that $\|\xi^\diamond\| = \min$ subject to $g(T_{\lambda^{(i)}}(\xi^\diamond)) \le 0$. ($\xi^{(i)}$ is starting point.)
  Generate N sample points $x_1,\dots,x_N \sim \mathcal{N}(T_{\lambda^{(i)}}(\xi^\diamond),\Sigma(\lambda^{(i)}))$.
  Density function: $f_{\lambda^{(i)}}$ for distribution $\mathcal{N}(T_{\lambda^{(i)}}(\xi^\diamond),\Sigma(\lambda^{(i)}))$.
  Weighting functions: $w_k^{\lambda^{(i)}}(\lambda) = f_\lambda(x_k)/f_{\lambda^{(i)}}(x_k)$ with density f_λ for $\mathcal{N}(\mu(\lambda),\Sigma(\lambda))$.
  Probability function: $p^{\lambda^{(i)}}_{x_1,\dots,x_k}(\lambda) = \frac{1}{N}\sum_{k=1}^N \mathbb{1}_{g(x_k)\le 0}\cdot w_k^{\lambda^{(i)}}(\lambda)$.
  Find $\lambda^*\in\Lambda$ with $p^{\lambda^{(i)}}_{x_1,\dots,x_k}(\lambda^*) = \max$. ($\lambda^{(i)}$ is starting point.)
  $\lambda^{(i+1)} = \lambda^*$, $\xi^{(i+1)} = \xi^\diamond$. ($\xi^\diamond$ and λ^* are starting points in the next step.)
  $i = i+1$.
Until a stopping criterion is satisfied.
Then $\overline{p}_f \approx p^{\lambda^{(i-1)}}_{x_1,\dots,x_k}(\lambda^*)$.

Algorithm 3: Global design point

Choose starting points $\lambda^{(0)}\in\Lambda$ and $\xi^{(0)}\in E\subseteq\mathbb{R}^d$.
Find $(\xi^\diamond,\lambda^\diamond)\in E\times\Lambda$ such that $\|\xi^\diamond\| = \min$ subject to $g(T_{\lambda^\diamond}(\xi^\diamond)) \le 0$.
(Starting points $\lambda^{(0)}$ and $\xi^{(0)}$.)
Generate N sample points $x_1,\dots,x_N \sim \mathcal{N}(T_{\lambda^\diamond}(\xi^\diamond),\Sigma(\lambda^\diamond))$.
Density function: $f_{\lambda^\diamond}$ for distribution $\mathcal{N}(T_{\lambda^\diamond}(\xi^\diamond),\Sigma(\lambda^\diamond))$.
Weighting functions: $w_k^{\lambda^\diamond}(\lambda) = f_\lambda(x_k)/f_{\lambda^\diamond}(x_k)$ with density f_λ for $\mathcal{N}(\mu(\lambda),\Sigma(\lambda))$.
Probability function: $p^{\lambda^\diamond}_{x_1,\dots,x_k}(\lambda) = \frac{1}{N}\sum_{k=1}^N \mathbb{1}_{g(x_k)\le 0}\cdot w_k^{\lambda^\diamond}(\lambda)$.
Find $\lambda^*\in\Lambda$ with $p^{\lambda^\diamond}_{x_1,\dots,x_k}(\lambda^*) = \max$. ($\lambda^\diamond$ is also starting point!)
Upper probability of failure: $\overline{p}_f \approx p^{\lambda^\diamond}_{x_1,\dots,x_k}(\lambda^*)$.

Conclusion

- Algorithm 3 (global design point) is the best.
  Effort: N (sample) + 20–40 (finding global design point) evaluations of g. Not much more than for a single random variable X instead of a family.
- **Iterating:** Has only an advantage for "bad" Algorithm 1. No convergence in general because of different samples. High computational cost (factor n).
- $p^{\lambda^\diamond}_{x_1,\dots,x_k}$ is a smooth function → use of fast (global) optimization procedures.