Flood water level assessment, an educational use-case in uncertainty analysis using Open TURNS software

Alberto Pasanisi, Anne-Laure Popelin, Merlin Keller, Bertrand Iooss
EDF R&D Industrial Risk Management Dept.

Feb. 15, 2013

1 Summary

The problem under investigation is the assessment of the water level in the terminal section of a watercourse in case of flood. The water level is evaluated as a function of the discharge and other physical parameters (measured) which will be detailed hereinafter.

Even if the questions underlying this study are similar to ones engineers cope with in practice, the case presented hereby is a simplification of the actual technical problem. Nevertheless, it is of great value as it can be used to illustrate several possible answers which can be given by Uncertainty Quantification (UQ) methods [Pasanisi and Dutfoy, 2012]. Indeed, this case study is intensively used in professional training courses on uncertainty analysis and computer experiments, hosted by the French National Laboratory of Metrology and Testing (LNE) and Électricité de France (EDF). It has also been widely used (as is or under slight variants) in a number of recent papers for illustrating different methods and problems in UQ. Cf. [Limbourg and De Rocquigny, 2010, Baraldi et al., 2011, Barbillon et al., 2011, Iooss, 2011, Pasanisi et al., 2012, Fu et al., 2012, Bousquet, 2012, Chastaing et al., 2012, Lemaître et al., 2013] as non-exhaustive list of examples.

We will start by assessing the probability distribution of the inputs of the physical model, based on the available data. A particulate care will be taken in modelling the dependence between variables. Actually, in principle, neglecting the dependence (i.e. considering only the marginal distributions

variables taken one-by-one) can introduce an uncontrolled and potentially high error in the results of a study.

Then, three different kinds of evaluations are undertaken on this case study.

- First, a "central tendency" study (using the Monte Carlo sampling algorithm and the Taylor decomposition based perturbation method) will be performed to assess the mean and standard deviation of the output.
- Then, the probability for the output to exceed a given threshold will be evaluated, using different sampling algorithms as well as the popular FORM (First Order Reliability Method) approach.
- Finally, a sensitivity analysis will also be undertaken to find out the variables the output is most sensible to.

The calculations shown hereby have been realised by means of the *Open TURNS* software [Dutfoy et al., 2009], jointly developed by the three companies EDF, EADS and Phiméca. Specifically intended for uncertainty quantification in numerical simulation, Open TURNS is an open source C++ library, available as a Python module. Open TURNS includes several specific methods for uncertainty propagation, sensitivity analysis and structural reliability and has been designed to be easily coupled to an external numerical code, seen as a *black box* trough which uncertainties are propagated (non-intrusive approach). The software and its documentation can be downloaded from the URL: `http://www.openturns.org`.

2 Introduction

Let us consider a portion of watercourse of length L. The variable of interest of this study is the water level in the terminal section of the portion. The phenomenon this study is concerned with (cf. Figure 1) is governed by the *de St. Venant* shallow water equations (1871), relying the water level H (measured w.r.t. the riverbed) at the abscissa x and time t to the discharge Q, the water section S, the lateral inflows q_L, the slope I and the head losses J due to the friction between the water body and the riverbed:

$$\begin{aligned} \frac{\partial S}{\partial t} + \frac{\partial Q}{\partial x} &= q_L, \\ \frac{\partial Q}{\partial t} + \frac{\partial (Q^2/S)}{\partial x} + gS\frac{\partial H}{\partial x} &= gS(I - J). \end{aligned}$$

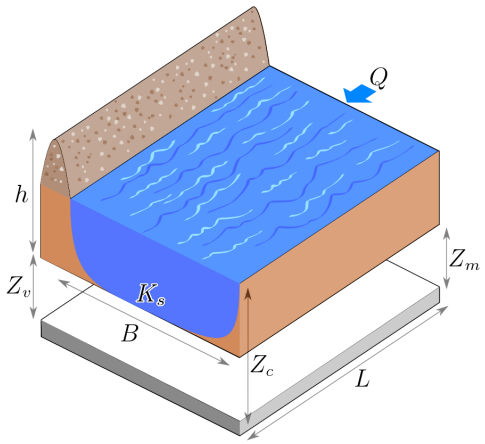

Figure 1: Schematic view of the physical problem and of the quantities involved in the study.

In the case of steady flow, with no inflows, and large rectangular section (i.e. width $B \gg H$), assuming that the classical Manning-Strickler formulation is used for the head losses, the equation above has the closed-form solution:

$$H = \left(\frac{Q}{K_s \cdot B \cdot \sqrt{(Z_m - Z_v)/L}} \right)^{3/5}, \tag{1}$$

where K_s is the *Strickler's* friction coefficient, Z_m and Z_v the riverbed levels (w.r.t. a fixed reference, e.g. the mean sea level) upstream and downstream the part of the watercourse under consideration (whose length is L) and B the width of the water section. The variable of interest, noted Y in the following, is the water level in the terminal section of the river portion, measured with respect to a fixed level (e.g. the sea mean level):

$$Y = Z_v + \left(\frac{Q}{K_s \cdot B \cdot \sqrt{(Z_m - Z_v)/L}} \right)^{3/5}. \tag{2}$$

3 Model

A preliminary and fundamental step in any UQ study is the probabilistic modelling of the input variables. Actually, the uncertainty tainting the inputs is transferred to the output Y trough the equation 2, that we will

rewrite:

$$Y = G(X, d). \tag{3}$$

In this expression we explicitly consider two kinds of input variables. The uncertain ones are gathered into a random vector X, while the deterministic ones constitutes the constant vector d.

The width of the cross section B and the length L of the portion of the watercourse are here considered as constant and they are given the values of 300 m and 5000 m respectively. The components of the random vector X are then (Q, K_s, Z_m, Z_v).

The probability distribution of Q, Z_m and Z_v are fitted on available data (cf. next section).

In particular, we will consider a Gumbel distribution as the probabilistic model for the discharges Q and triangular distributions for both Z_m and Z_v. Moreover, we will explicitly take into account the dependence between Z_m and Z_v, which sounds logical by an engineering viewpoint. Actually, the phenomena that makes the riverbed level be uncertain (erosion, accumulation of sediments, presence of vegetation etc.) act in the same way in the section located upstream and downstream the part of river considered here, as the distance between these two section is relatively small with respect to the scale of these phenomena.

From a mathematical viewpoint, the dependence between Z_m and Z_v will be modelled by means of a copula [Nelsen, 2006], fitted on the pairwise data (Z_m, Z_v) available, i.e. the joint cumulative distribution function (cdf) $F(Z_m, Z_m)$ is written:

$$F(z_m, z_v) = \mathcal{C}\left(F(z_m), F(z_v)\right), \tag{4}$$

$F(Z_m)$ and $F(Z_v)$, being the marginal univariate cdf's of Z_m and Z_v and $\mathcal{C}(\cdot, \cdot)$ being the copula of the joint distribution, assumed as Gaussian:

$$\mathcal{C}_\Xi(w_1, w_2) = \Phi_\Xi\left(\Phi^{-1}(w_1), \Phi^{-1}(w_2)\right), \tag{5}$$

where $\Phi^{-1}(\cdot)$ is the inverse cdf of the standard Gaussian distribution and Φ_Ξ the density of a bivariate Gaussian distribution with null mean and correlation matrix Ξ.

Concerning the Strickler's coefficient K_s, no measures are available. Actually, even if this parameter has a certain *physical* sense, i.e. an inverse quantification of the roughness of the riverbed (the smaller K_s, the higher

the friction between the water-body and the riverbed), in practice it is to be rather regarded as a parameter of the hydraulic model, than an actual physical quantity. The distribution of K_s can be, in practice, inversely inferred from a set of pairwise data (discharge vs. water level), as in Barbillon et al. [2011] or Fu et al. [2012], but often it is reasonable to expect that some expertise is available. Indeed, according to his/her prior knowledge of the watercourse an expert can often provide an interval containing likely values of K_s or his/her best guess with some error bounds.

In the case under investigation, the available expertise states that the Strickler's coefficient should have a value "around" $30 \pm 5\,\mathrm{m}^{1/3}\mathrm{s}^{-1}$. Based on this information, it seems reasonable to chose a Gaussian distribution with mean and standard deviation equal to 30 and 7.5 as appropriate for modeling the uncertainty tainting K_s: the reference value corresponds to the mean of the distribution and the probability for the K_s to be in interval 30 ± 15 is approximately 95%.

4 Data

For this problem, the analyst has at his/her disposal several data, shown in Table 1. First, a dataset of 149 historical records of annual maximum water discharges (in $\mathrm{m\,s}^{-3}$) is available. The history of the maxima is plotted in Figure 2 (left). It is worth noting that a look at the graph shows that the data distribution is not symmetric and, namely, positive-skewed. Very much higher values than the mean are found in the sample, as it is typical for extreme values problems.

The second piece of information is a set of 29 couples of records of the riverbed levels. Data are all measured with respect to a given fixed level, and are here expressed in m ASL (above sea level). Not surprisingly the data plot (in Figure 2, right) clearly shows a dependence between upstream and downstream levels. That is logical, as in practice the variation of these levels depends on local phenomena like erosion, sediment accumulation, vegetation grow which can be imagined to act in the same way upstream and downstream the portion of river under investigation (if the length L of it is reasonably small).

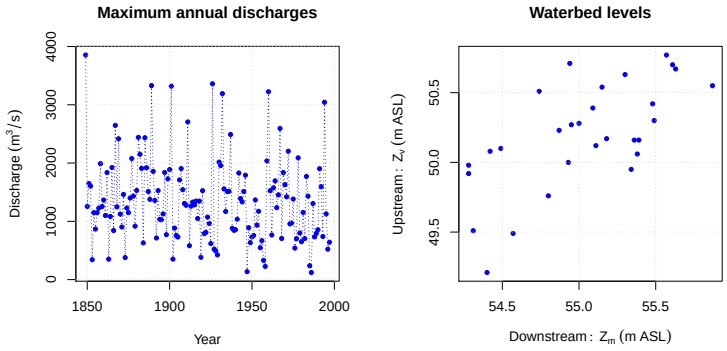

Figure 2: Plot of the input data: annual maximum discharges (left) and pairwise riverbed levels (right).

Discharge data (m^3/s)														
3854	1256	1649	1605	341	1149	868	1148	1227	1991	1255	1366	1100	1837	351
1084	1924	843	2647	1248	2417	1125	903	1462	378	1230	1149	1400	2078	1433
917	1530	2442	2151	1909	630	2435	1920	1512	1377	3330	1858	1359	714	1528
1035	1026	1127	1839	771	1730	1889	3320	352	885	759	731	1711	1906	1543
1307	1275	2706	582	1260	1331	1283	1348	1048	1348	383	1526	789	811	1073
965	619	3361	523	493	424	2017	1958	3192	1556	1169	1511	1515	2491	881
846	856	1036	1830	1391	1334	1512	1792	136	891	635	733	758	1368	935
1173	547	669	331	227	2037	3224	1525	766	1575	1695	1235	1454	2595	706
1837	1629	1421	2204	956	971	1383	541	703	2090	800	651	1153	704	1771
1433	238	122	1306	733	793	856	1903	1594	740	3044	1128	522	642	

Riverbed level data (m ASL)															
Z_m	55.1	55	54.9	54.3	54.7	55.5	55.4	55.4	54.8	55.2	54.9	54.4	55.3	55.3	54.3
Z_v	50.4	50.3	50.2	49.9	50.5	50.4	50.2	50.2	49.8	50.2	50.7	50.1	50	50.6	49.5

Z_m	55.6	54.3	55.5	54.5	55.1	55.2	54.4	55.9	55.6	54.9	55.6	55	55.4	54.6
Z_v	50.8	50	50.3	50.1	50.1	50.5	49.2	50.6	50.7	50	50.7	50.3	50.1	49.5

Table 1: Values of annual maximum discharges (up) and riverbed levels (down), available for the study.

5 Methods and results

5.1 Assessing the probability distribution of the input

First, let us focus on the discharge data (Q variable). A number of continuous probability distribution functions (pdf's) is automatically tested by Open TURNS and compared using the Kolmogorov Smirnov statistic and

the Bayesian Information Criterion (BIC). Finally a Gumbel distribution is retained:

$$\mathrm{Gu}(\alpha,\beta) = \alpha \exp\left[-\alpha\left(q-\beta\right) - \exp\left(-\alpha\left(q-\beta\right)\right)\right], \tag{6}$$

with $\alpha = 1.797 \cdot 10^{-3}$ and $\beta = 1014.14$. The QQ-plot in Figure 3 shows a fairly good adjustment between the data and the fitted distribution.

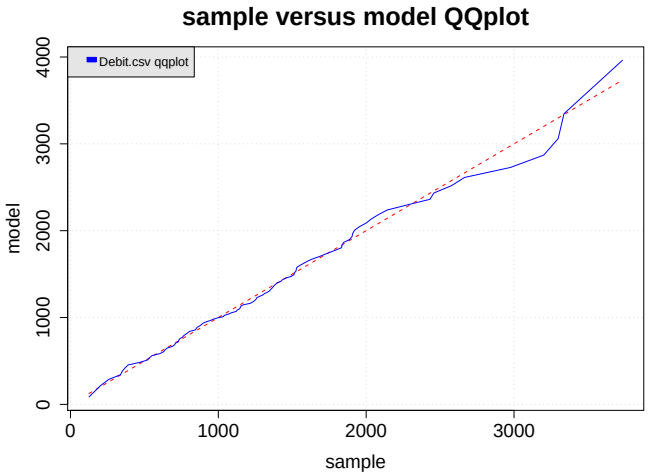

Figure 3: QQ plot of discharge data vs. fitted Gumbel distribution.

Concerning the joint pdf of the riverbed levels vector (Z_m, Z_v), first two triangular distributions $\mathcal{T}(a,m,b)$ (with a and b being the lower and upper bounds and m the mode of the distribution) are independently fitted on the available samples of Z_m and Z_v respectively. That leads to take as marginal distributions $\mathcal{T}(52.53, 54.89, 57.67)$ and $\mathcal{T}(47.62, 50.55, 52.41)$ for Z_m and Z_v respectively.

Then, a Gaussian copula is fitted on the pairwise ranked sample. The correlation matrix of the fitted copula is:

$$\Xi = \begin{bmatrix} 1 & 0.677 \\ 0.677 & 1 \end{bmatrix}.$$

The good fitting between pairwise data and the estimated copula is graphically tested by means of the Kendall plot [Genest and Boies, 2003], which can be interpreted as a multivariate version of the QQ-plot: the more the

points of the plot are close to the identity line, the more the data are adjusted on the proposed distribution. The Figure 4 shows the good quality of the proposed adjustment.

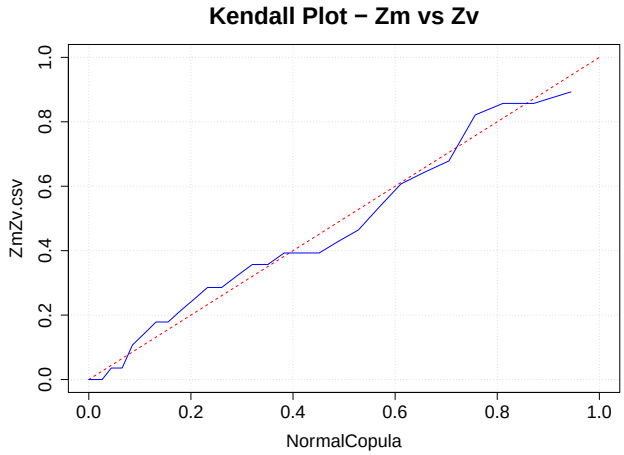

Figure 4: Kendall plot showing the fair adjustment of the vector (Z_m, Zv) to a Gaussian copula.

5.2 Central tendency evaluation

A central tendency evaluation aims at evaluating a reference value for the variable of interest, here the water level Y given by the expression 2, and an indicator of the dispersion of the variable around the reference. To address this problem, mean $\mu_Y = \mathbb{E}(Y)$, and the standard deviation $\sigma_Y = \sqrt{\mathbb{V}(Y)}$ of Y have been evaluated using two different methods.

First, following a popular method within the Measurement Science community [Joint Committee for Guides in Metrology, 2008], μ_Y and σ_Y have been computed under a Taylor first order approximation of the function $Y = G(X)$ (notice that the explicit dependence on the deterministic variable d is here omitted for simplifying notations):

$$\mu_Y \approx G\left(\mathbb{E}(X)\right) \tag{7}$$

$$\sigma_Y \approx \sum_{i=1}^{4}\sum_{j=1}^{4} \frac{\partial G}{\partial X_i}\Big|_{\mathbb{E}(X)} \frac{\partial G}{\partial X_j}\Big|_{\mathbb{E}(X)} \rho_{ij}\sigma_i\sigma_j, \tag{8}$$

σ_i and σ_j being the standard deviation of the ith and jth component (X_i and X_j) of the vector X and ρ_{ij} their correlation coefficient. Thanks to the formulas above, the mean and the standard deviation of Y are evaluated as 52.75 m and 1.15 m respectively.

Then, the same quantities have been evaluated by a Monte Carlo evaluation [Robert and Casella, 2004]: a set of 10000 samples of the vector X is generated and the function G(X) is evaluated, getting thus a sample of Y. The empirical mean and standard deviation of this sample are 52.75 m and 1.42 m respectively. The figure 5 shows the empirical histogram of the generated sample of Y.

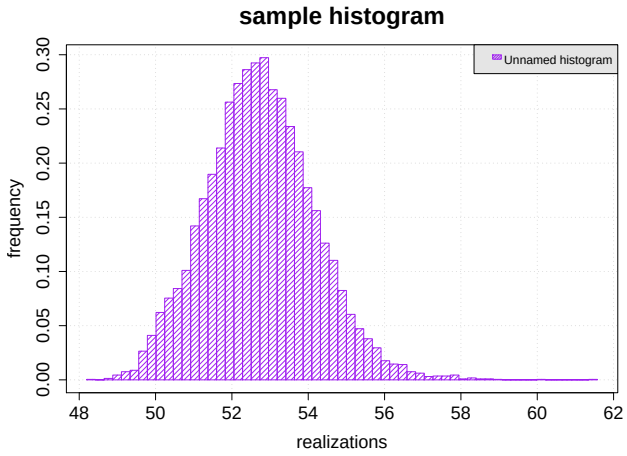

Figure 5: Empirical histogram of 10000 samples of Y.

5.3 Excess probability evaluation

We now turn to the estimation of the probability for the output Y to exceed a certain threshold s, which we note P_f in the following. If s is the altitude of a flood protection dyke, then the above excess probability, P_f can be interpreted as the probability of an overflow of the dyke, i.e. a failure probability.

Note that an equivalent way of formulating this reliability problem would be to estimate the $(1-p)$-th quantile of the output's distribution. This quantile can be interpreted as the flood height q_p which is attained with probability p each year. $T = 1/p$ is then seen to be a return period, i.e. a flood as high

than $q_{1/T}$ occurs on average every T years.

Hence, the probability of overflowing a dyke with height s is less than p (where p, for instance, could be set according to safety regulations) if and only if $s \geq q_p$, i.e. if the dyke's altitude is higher than the flood with return period equal to $T = 1/p$.

5.3.1 FORM

A popular way of evaluating such failure probabilities is through the so-called First Order Reliability Method (FORM) [Ditlevsen and Madsen, 1996, Lemaire, 2010]. This approach starts by a applying a transformation $\mathcal{H}$ to the input vector X such that the result $U = \mathcal{H}(X)$ has a spherical distribution (i.e. the density only depend on the norm of U). We consider here the Rosenblatt transformation, such that the transformed vector U is distributed according to the standard normal distribution, with zero-mean and identity covariance matrix [Rosenblatt, 1952, Lebrun and Dutfoy, 2009].

Next, the failure domain D_f, defined as the set of input values such that the output Y exceeds s, is considered. It is bounded by the limit-state surface, which is the set of input values such that $Y = s$. Using the FORM method, the limit-state-surface is assumed approximately linear after having applied the transformation $\mathcal{H}$. The algorithm then aims at finding the so-called "design point" u^*, which is the point on the limit-state surface whose distance β_{HL} to the origin is minimal (assuming this point is unique). Such a point can be found for instance through a gradient-descent algorithm which searches for zeros of the function $Y - s = G \circ \mathcal{H}^{-1}(U) - s$ using the origin $U = 0$ as a starting point.

The distance β_{HL} of u^* to the origin of the standard space, termed the Hasofer-Lind reliability index, then provides the following convenient approximation to the failure probability:

$$\hat{P}_{f,FORM} = \begin{cases} \Phi(-\beta_{HL}) & \text{If the origin lies in the failure domain} \\ \Phi(\beta_{HL}) & \text{otherwise,} \end{cases}$$

where Φ is the cdf of the standard normal distribution.

We evaluated the probability that the yearly maximal water height Y exceeds s=58 m using FORM. The Hasofer-Lind Reliability index was found to be equal to: $\beta_{HL} = 3.04$, yielding a final estimate of:

$$\hat{P}_{f,FORM} = 1.19 \times 10^{-3}.$$

5.3.2 Monte Carlo

Because the FORM approximation relies on many assumptions, it is usually recommended to complement it by a Monte Carlo estimate, which is always valid but in general much more computationally intensive. It consists in sampling many input values $(X^{(i)})_{1\leq i\leq N}$ from the input vector joint distribution, then computing the corresponding output values $Y^{(i)} = G(X^{(i)})$. The excess probability P_f is then estimated by the proportion of sampled values $Y^{(i)}$ that exceed t :

$$\hat{P}_{f,MC} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{Y^{(i)}>s\}}. \tag{9}$$

The sample average of the estimation error $\hat{P}_{f,MC} - P_f$ decreases as $1/\sqrt{N}$, and can be precisely quantified by the following confidence interval:

$$I_{P_{f,MC}} = \left[\hat{P}_{f,MC} - 1.96\,\hat{\sigma}_{\mathbf{1}_{\{Y>s\}}}/\sqrt{N}, \hat{P}_{f,MC} + 1.96\,\hat{\sigma}_{\mathbf{1}_{\{Y>s\}}}/\sqrt{N}\right],$$

where $\hat{\sigma}_{\mathbf{1}_{\{Y>s\}}} = \hat{P}_{f,MC} \times (1 - \hat{P}_{f,MC})$ is the estimation of the variance of the Bernoulli distributed variable $\mathbf{1}_{\{Y^{(i)}>s\}}$, which contains the true value approximately 95 times out of 100. In the present case we found:

$$\hat{P}_{f,MC} = 1.50 \times 10^{-3},$$

with the following 95% confidence interval:

$$I_{P_f,MC} = \left[1.20 \times 10^{-3}, 1.79 \times 10^{-3}\right].$$

These results are coherent with those of the FORM approximation, confirming that the assumptions underlying the latter are correct.

An even more precise estimate can be obtained through importance sampling [Robert and Casella, 2004], using the Gaussian distribution with identity covariance matrix and mean equal to the design point u^* as the proposal distribution. Thus many values $(U^{(i)})_{1\leq i\leq N}$ are sampled from this proposal. Then, we use the following identity:

$$\begin{aligned}\hat{P}_f &= \int \mathbf{1}_{\{G\circ T^{-1}(u)>s\}}\phi_4(u)du \\ &= \int \mathbf{1}_{\{G\circ T^{-1}(u)>s\}}\frac{\phi_4(u)}{\phi_n(u-u^*)}\phi_n(u-u^*)du,\end{aligned}$$

where ϕ_4 is the density of the standard normal distribution on $\mathbb{R}^4$, i.e. the distribution of the transformed input vector U. Because $\phi_n(u-u^*)$ is the

proposal density from which the $U^{(i)}$ have been sampled, the failure probability can estimated without bias by:

$$\hat{P}_{f,IS} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{G\circ T^{-1}U^{(i)}>s\}} \frac{\phi_n(U^{(i)})}{\phi_n(U^{(i)}-u^*)} \tag{10}$$

The rationale of this approach is that by sampling in the vicinity of the failure domain boundary, a larger proportion of values fall within the failure domain than by sampling around the origin, leading to a better evaluation of the failure probability, and a reduction in the estimation variance. Using this approach, we found:

$$\hat{P}_{f,IS} = 1.40 \times 10^{-3}$$

As in the simple Monte-Carlo approach, a 95%-level confidence interval can be derived from the output of the Importance Sampling algorithm. In the present case, this is equal to:

$$I_{P_f,IS} = \left[1.26 \times 10^{-3}, 1.53 \times 10^{-3}\right],$$

and indeed provides tighter confidence bounds for P_f.

5.4 Global sensitivity analysis

The sensitivity analysis aims to investigate how a given computational model responds to variations in its inputs. Such knowledge is useful for determining the degree of resemblance of a model and a real system, distinguishing the factors that mostly influence the output variability and those that are insignificant, revealing interactions among input parameters and correlations among output variables, etc. A detailed description of sensitivity analysis methods can be found in [Saltelli et al., 2000, Iooss, 2011]. In the global sensitivity analysis strategy, the emphasis is put on apportioning the output uncertainty to the uncertainty in the input factors, given by their uncertainty ranges and probability distributions. Most of the used methods are based on a model output variance analysis.

If the behaviour of Y compared to each parameter is overall linear, it is possible to obtain quantitative measurements of their influence from the regression coefficients α_i of the linear regression connecting Y to the $X = (X_1, \ldots, X_p)$:

$$\hat{Y} = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_i \tag{11}$$

where $\hat{Y}$ represents the estimation of Y by the regression model.
The Standard Regression Coefficients (SRC), defined by:

$$\mathrm{SRC}_i = \alpha_i \frac{\sigma_i}{\sigma_Y} \text{ (for } i = 1 \dots p), \tag{12}$$

measure the variation of the response for a given variation of the parameter X_i. In practice, we start by making the multiple linear regression between Y and all the parameters X_i (Equation 11). We then determine if their relation is approximately linear by making classical statistical tests on residuals and by calculating the coefficient of determination R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (Y^{(i)} - \hat{Y}^{(i)})^2}{\sum_{i=1}^{N} (\overline{Y} - Y^{(i)})^2} , \tag{13}$$

where $\{Y^{(i)}\}_{i=1\dots N}$ is a N-size sample of the output variable, $\overline{Y}$ is the average of Y. The coefficient R^2 represents the variance percentage of the output variable Y explained by the regression model $\hat{Y}$. Therefore, if R^2 is close to one, the relation connecting Y to all the parameters X_i is almost linear and we can use the SRC as sensitivity indices.

In our case study, we obtain from a Monte Carlo sample of size $N = 1000$ a linear regression model with $R^2 = 0.96$, which allows to consider that the linear relation hypothesis is valid. The regression coefficients and sensitivity indices (SRC) are given in Table 2.

	Q	K_s	Z_v	Z_m	α_0
α_i	0.00114413	-0.0599834	1.17645	-0.185676	4.21087
SRC_i	0.347504	0.108773	0.662248	0.020291	

Table 2: Regression coefficients and SRC of the flood model inputs.

If the relation between two variables X and Y is not linear, the correlation coefficients of the ranks (or Spearman coefficients) can be used. By replacing the values $X^{(1)}, \dots, X^{(N)}$ and $Y^{(1)}, \dots, Y^{(N)}$ by their rank, the assumption of linearity is replaced by the assumption of a monotonic relation. In the same way that previously, one can also compute the standard rank regression coefficients (SRRC) by carrying out the linear regressions on the ranks.

The SRC and SRRC are related respectively to linear and monotonic assumptions. More general variance-based sensitivity indices, called Sobol' indices [Sobol, 1993], have been defined for the first order and second order as:

$$S_i = \frac{\mathbb{V}(\mathbb{E}(Y|X_i))}{\mathbb{V}(Y)} \qquad S_{ij} = \frac{\mathbb{V}(\mathbb{E}(Y|X_i, X_j))}{\mathbb{V}(Y)} - S_i - S_j \tag{14}$$

and so on until the pth order. A first order Sobol' index is related to the sole contribution of an input, while a second order Sobol' index gives the contribution of the interaction between two inputs. A Sobol' index can be directly interpreted as the contribution percentage of an input in the variance of the model output. In practice, there are many methods to compute Sobol indices: Monte-Carlo method (quite expensive), non parametric functional estimation (for low-order indices), repose surface techniques, etc. [Sobol, 1993, Saltelli et al., 2000, Iooss, 2011].

In sensitivity analysis, graphical techniques can also be useful. For example, with all the scatterplots between each input variable and the model output, one can detect some trends in their functional relation. However scatterplots do not capture some interaction effects between the inputs. Cobweb plots [Kurowicka and Cooke, 2006], also called parallel coordinate plots, can then be used to visualize the simulations as a set of trajectories. In Figure 6, the simulations leading to the largest values of the model output H have been colored in red. This allows to immediately understand that these simulations correspond to large values of the flowrate Q and small values of the Strickler coefficient K_s.

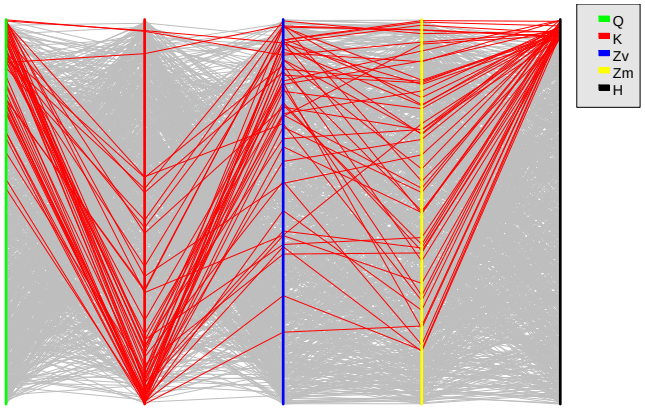

Figure 6: Cobweb plot of 10000 simulations of the flood model.

Finally, if the quantity of interest related to the model output is not the variance but a probability of rare events (as a probability of treshold ex-

ceedance), different sensitivity indices have to be considered. For example, we can compute some importance factors obtained with FORM (see section 5.3.1): these are the coordinates of the design point. The Figure 7 gives the importance factors of the inputs of the flood model resulting from FORM. Other sensitivity analysis methods related to rare events of a model output are described in [Lemaître et al., 2013]

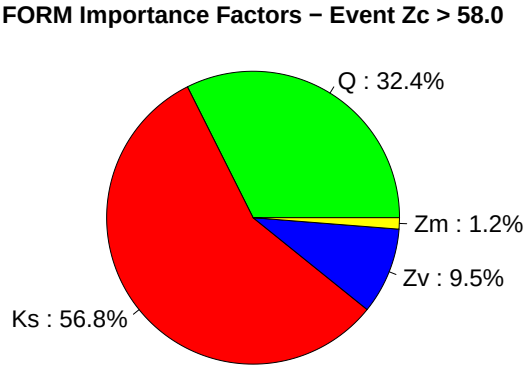

Figure 7: Importance factors obtained with FORM.

6 Conclusion

This educational example has shown a number of questions and problems that can be addressed by UQ methods: central tendency evaluation, excess probability assessment and sensitivity analysis.

Different numerical methods have been used for solving these three classes of problems, leading substantially to the same (or very similar) results. In industrial practice of UQ, the main issue (which actually motivates the choice of one mathematical method instead of another) is the computational budget, which is actually given by the number of allowed runs of the deterministic model $G(\cdot)$. When the computer code implementing $G(\cdot)$ is computationally expensive, one needs specifically designed mathematical and software tools.

Open TURNS is specially intended to cope with this problem : (i) it includes a set of efficient mathematical methods for UQ and (ii) it can be easily

connected to any external black box model $G(\cdot)$ under very mild condition on the computer code. In practice, once the model has been specified as a specific Open TURNS object, the *NumericalMathFunction* (cf. listing 3 below), it is seen by Open TURNS as a *simple* function, relying inputs and output, whatever its complexity.

7 Open TURNS code

The following listings show the implementation of the exercise presented hereinbefore in the Python environment by means of Open TURNS:

- listing 1 implements the statistical analysis of discharge data,
- listing 2 implements the statistical analysis of riverded level data, leading to the fit of two triangular marginal pdf's for Z_m and Z_v and a Gaussian copula,
- listing 3 shows the way the connection between the deterministic and the probabilistic models (i.e. the function $G(X, d)$ and the pdf of X respectively) is made within the Open TURNS framework: the main object to be defined is the *NumericalMathFunction* and its arguments are the random inputs, the output variable of interest and a formula relying them (notice that the NumericalMathFunction can also be defined from an external code and a *wrapper*, acting as an interface between Open TURNS and the external code [Open TURNS Consortium, 2012]),
- listing 4 implements the central tendency analysis (Taylor's approximation and Monte Carlo methods),
- listing 5 implements the calculation of excess probability (FORM, Monte Carlo, Importance Sampling): notice the particular syntax for specifying the "Event" one wants to evaluate the probability of,
- listing 6 implements the sensitivity analysis, using SRC's and FORM index, as well as the graphical method of the cobweb plot.

Listing 1: Flood water level: Fitting the pdf of the discharge Q

```
from openturns import *
from openturns.viewer import ViewImage
# Read the sample of data from a csv file
qSample =
   NumericalSample.ImportFromCSVFile("Debit.csv")
# Create a collection of "factories" of pdf's
CollContFactory = DistributionFactoryCollection(
   (NormalFactory(), WeibullFactory(),
   LogNormalFactory(), GumbelFactory() ) )
# Rank the pdf's w.r.t. the Kolmogorov p-values
bestDistributionKolmogorov =
   FittingTest.BestModelKolmogorov(qSample,
   CollContFactory)
# Get all information on that distribution
print bestDistributionKolmogorov
 # Get the test result associated to the best
    distribution
print FittingTest.GetLastResult()
# Rank the continuous models w.r.t.the BIC values
bestDistributionBIC =
   FittingTest.BestModelBIC(qSample,
   CollContFactory)
# Get all information on that distribution
print bestDistributionBIC
# Validate by a graphical test (QQ-plot)
sampleQQPlot =
   VisualTest.DrawQQplot(qSample,bestDistributionBIC,100)
sampleQQPlot.draw("SampleQQPlot")
ViewImage(sampleQQPlot.getBitmap())
```

Listing 2: Flood water level: Fitting the pdf of the waterbed levels (Z_m, Z_v).

```
# Read the 2-dimension sample from a csv file
initSample =
   NumericalSample.ImportFromCSVFile("ZmZv.csv")
# Create two 1-dimension samples
ZmValues = [[initSample[k][0]] for k in
   range(initSample.getSize())]
ZvValues = [[initSample[k][1]] for k in
   range(initSample.getSize())]
ZmSample = NumericalSample(ZmValues);
ZvSample = NumericalSample(ZvValues);
# Fit triangular pdf's on the samples Zm and Zv
ZmfittedTriangular =
   TriangularFactory().build(ZmSample)
print ZmfittedTriangular
ZvfittedTriangular =
   TriangularFactory().build(ZvSample)
print ZvfittedTriangular
# Create the operator which transforms the
   marginals into uniform pdf's
ranksTransf = MarginalTransformationEvaluation(
   DistributionCollection( [
   Triangular(52.529,54.8949,57.6723),
   Triangular(47.6226,50.5531,52.4077)] ),
   MarginalTransformationEvaluation.FROM )
# Transform the initial sample into a ranked one
transformedSample = NumericalSample(
   initSample.getSize(), initSample.getDimension() )
for i in range(initSample.getSize()) :
        transformedSample[i] =
           ranksTransf(initSample[i])
estimatedCopula =
   NormalCopulaFactory().build(transformedSample)
print estimatedCopula
# Draw a Kendall Plot
myValue = 1000; ResourceMap.SetAsUnsignedLong(
   'VisualTest-KendallPlot-MonteCarloSize', myValue
   )
#Run the Kendall test and show the graph
kendallPlot1 =
   VisualTest.DrawKendallPlot(initSample,
   estimatedCopula); Show(kendallPlot1)
```

Listing 3: Specification of the uncertainty analysis problem in Open TURNS

```
# Definition of the input vector [Q, Ks, Zv, Zm]
inputDim =  4; randomInput = Description(inputDim)
randomInput[0] = "Q"; randomInput[1] = "Ks";
   randomInput[2] = "Zv"; randomInput[3] = "Zm"
# Description of the output vector
outputVar = Description(1); outputVar[0] = "Zc"
# Description of the model
formula = Description(1); formula[0] = "Zv + (Q /
   (Ks * 300 * sqrt(abs(Zm-Zv) / 5000))) ^ 0.6"
# Link deterministic and probabilistic models
finalModelCrue = NumericalMathFunction(randomInput,
   outputVar, formula)
# Definition of the probabilistic problem
loi_Q =
   TruncatedDistribution(Distribution(Gumbel(0.00179654,
   1014.14)),0.0, TruncatedDistribution.LOWER)
loi_K =
   TruncatedDistribution(Distribution(Normal(30.,
   7.5)),0., TruncatedDistribution.LOWER)
loi_Zv = Triangular(47.6226,50.5531,52.4077)
loi_Zm = Triangular(52.529,54.8949,57.6723)
# Dependence structure: Gauss. copula for (Zm,Zv)
dim=2; R=CorrelationMatrix(dim); R[0,1]=0.677732
normalCopula = NormalCopula(R)
# Dependence structure: Final composed copula
copulaColl = CopulaCollection(2)
copulaColl[0] = Copula(IndependentCopula(2))
copulaColl[1] = Copula(normalCopula)
copula = ComposedCopula(copulaColl)
# Definition of the joint distribution
inputJointDistribution =
   ComposedDistribution(inputDistribCollection,
   Copula( copula ) )
inputJointDistribution.setDescription(randomInput)
# Input Random Vector
inputRandomVector =
   RandomVector(inputJointDistribution)
# Output Random Vector
outputVariable =
   RandomVector(finalModelCrue,inputRandomVector)
```

Listing 4: Central tendency analysis

```
# Taylor's approximation
myQuadCum = QuadraticCumul(outputVariable)
print "Mean - First order = ",
   myQuadCum.getMeanFirstOrder()[0]
print "Mean - Second order = ",
   myQuadCum.getMeanSecondOrder()[0]
print "Variance - First order = ",
   myQuadCum.getCovariance()[0,0]
# Monte Carlo
# Create a random sample of the output variabe of
   interest of size 10000
size = 10000; outputSample =
   outputVariable.getNumericalSample(size)
 # Get the empirical mean
empiricalMean = outputSample.computeMean()
print "Empirical Mean = ", empiricalMean
# Get the empirical covariance matrix
empiricalCovarianceMatrix =
   outputSample.computeCovariance()
print "Empirical Covariance Matrix = ",
   empiricalCovarianceMatrix
print "Standard deviation of output = ",
   sqrt(empiricalCovarianceMatrix[0,0])
# Histogram
H_Hist = VisualTest.DrawHistogram(outputSample);
   H_Hist.draw("Histogram_H");
```

Listing 5: Excess probability evaluation

```
# Event definition
threshold = 58.0; myEvent = Event(outputVariable,
   ComparisonOperator(Greater()), threshold)
# FORM algorithm
myCobyla = Cobyla() ; meanInputVector =
   inputRandomVector.getMean()
myFORM = FORM(NearestPointAlgorithm(myCobyla),
   myEvent, meanInputVector)
myFORM.run() ; FormResult = myFORM.getResult()
# FORM Importance factors
importanceFactorsGraph =
   FormResult.drawImportanceFactors()
# Importance Sampling
maximumOuterSampling_IS = 40000; StdPt =
   FormResult.getStandardSpaceDesignPoint()
# Define the importance distribution
mean = StdPt; sigma = NumericalPoint(4,1.0)
importanceDistrib =
   Normal(mean,sigma,CorrelationMatrix(4))
# Define the IS algorithm : event, distribution,
   criteria of convergence,...
myStandardEvent = StandardEvent(myEvent)
myAlgoImportanceSampling = ImportanceSampling
   (myStandardEvent,
   Distribution(importanceDistrib))
myAlgoImportanceSampling.setMaximumOuterSampling
   (maximumOuterSampling_IS)
myAlgoImportanceSampling.setMaximumCoefficientOfVariation
   (0.05)
myAlgoImportanceSampling.setConvergenceStrategy
   (HistoryStrategy(Full()))
myAlgoImportanceSampling.run()
# Monte Carlo algorithm
myMonteCarlo = MonteCarlo(myEvent)
numberSimulation = 100000
myMonteCarlo.setMaximumOuterSampling(numberSimulation)
myMonteCarlo.setBlockSize(1)
myMonteCarlo.setMaximumCoefficientOfVariation(0.1)
myMonteCarlo.run()
```

Listing 6: Sensitivity Analysis

```
# Compute the SRC's
from math import sqrt
inputSample =
   inputRandomVector.getNumericalSample(1000)
outputSample = finalModelCrue(inputSample)
SRCCoefficient =
   CorrelationAnalysis.SRC(inputSample,
   outputSample)
print "SRC Coefficients", SRCCoefficient
linearRegressionModel =
   LinearModelFactory().build(inputSample,
   outputSample, 0.90)
print "Coefficients of the linear regression model
   =" , linearRegressionModel.getRegression()
resultLinearModelRSquared =
   LinearModelTest.LinearModelRSquared(inputSample,
   outputSample, linearRegressionModel,0.90)
print "R-2 =", resultLinearModelRSquared
# Draw the Cobweb plot
descr_input =
   Description(inputSample.getDimension())
descr_output =
   Description(outputSample.getDimension())
descr_output[0]  ='H'; descr_input[0] = 'Q';
   descr_input[1] = 'K'; descr_input[2] = 'Zv';
   descr_input[3] = 'Zm'
inputSample.setDescription(descr_input)
outputSample.setDescription(descr_output)
sample_trie = outputSample.sort(); i=50
# Graph 1 : value based scale to describe the Y
   range
minValue = sample_trie[inputSample.getSize()-i][0]
maxValue = sample_trie[inputSample.getSize()-1][0]
myCobweb = VisualTest.DrawCobWeb(inputSample,
   outputSample, minValue, maxValue, 'red', False)
myCobweb.setTitle('')
myCobweb.draw('cobWeb', 640, 480,
   GraphImplementation.PDF)
```

References

P. Baraldi, N. Pedroni, E. Zio, E. Ferrario, A. Pasanisi, and M. Couplet. Monte Carlo and fuzzy interval propagation of hybrid uncertainties on a risk model for the design of a flood protection dike. In *Proceedings of the European Safety and RELiability Conference 2011*, pages 2167–2175, Troyes, France, 2011.

P. Barbillon, G. Celeux, A. Grimaud, Y. Lefebvre, and E. De Rocquigny. Nonlinear methods for inverse statistical problems. *Computational Statistics & Data Analysis*, 55(1):132 – 142, 2011.

N. Bousquet. Accelerated Monte Carlo estimation of exceedance probabilities under monotonicity constraints. *Annales de la faculté des sciences de Toulouse, Série 6*, 21(3):557 – 591, 2012.

G. Chastaing, F. Gamboa, and C. Prieur. Generalized Hoeffding-Sobol decomposition for dependent variables. Application to sensitivity analysis. *Electronic Journal of Statistics*, 6:2420–2448, 2012.

O. Ditlevsen and H.O. Madsen. *Structural Reliability Methods*. Wiley, 1996.

A. Dutfoy, I. Dutka-Malen, A. Pasanisi, R. Lebrun, F. Mangeant, J. Sen Gupta, M. Pendola, and T. Yalamas. OpenTURNS, an Open Source initiative to Treat Uncertainties, Risks'N Statistics in a structured industrial approach. In *41èmes Journées de Statistique, SFdS, Bordeaux, France*, 2009.

S. Fu, G. Celeux, N. Bousquet, and M. Couplet. Bayesian inference for inverse problems occurring in uncertainty analysis. Research Report RR-7995, INRIA, 2012.

C. Genest and J.C. Boies. Detecting dependence with Kendall plots. *The American statistician*, 57(4):275 – 284, 2003.

B. Iooss. Revue sur l'analyse de sensibilité globale de modèles numériques. *Journal de la Société Française de Statistique*, 152:1–23, 2011.

Joint Committee for Guides in Metrology. *Evaluation of measurement data — Guide to the expression of uncertainty in measurement*. International Bureau of Weights and Measures (BIPM), Sèvres, France, September 2008.

D. Kurowicka and R.M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling*. Wiley, 2006.

R Lebrun and A. Dutfoy. Do Rosenblatt and Nataf isoprobabilistic transformations really differ? *Probabilistic Engineering Mechanics*, 24(4):577 – 584, 2009.

M. Lemaire. *Structural reliability*. Wiley, 2010.

P. Lemaître, E. Sergienko, A. Arnaud, N. Bousquet, F. Gamboa, and B. Iooss. Density modification based reliability sensitivity analysis. *Journal of Statistical Computation and Simulation*, submitted, arXiv:1210.1074, 2013.

Ph. Limbourg and E. De Rocquigny. Uncertainty analysis using evidence theory: confronting level-1 and level-2 approaches with data availability and computational constraints. *Reliability Engineering & System Safety*, 95(5):550 – 564, 2010.

R.B. Nelsen. *An Introduction to Copulas*. Springer, 2006.

Open TURNS Consortium. *Wrappers Guide, Open TURNS version 1.0*, 2012. URL `http://www.openturns.org`.

A. Pasanisi and A. Dutfoy. An Industrial Viewpoint on Uncertainty Quantification in Simulation: Stakes, Methods, Tools, Examples. In A.M. Dienstfrey and R.F. Boisvert, editors, *Uncertainty Quantification in Scientific Computing*, pages 27–45. Springer, 2012.

A. Pasanisi, M. Keller, and E. Parent. Estimation of a quantity of interest in uncertainty analysis: Some help from Bayesian decision theory. *Reliability Engineering and System Safety*, 100(0):93 – 101, 2012.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

A. Saltelli, K. Chan, and E.M. Scott, editors. *Sensitivity analysis*. Wiley, 2000.

I.M. Sobol. Sensitivity estimates for non linear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407–414, 1993.