

Emulator-based global sensitivity analysis for flow-like landslide run-out models

Abstract Landslide run-out modeling involves various uncertainties originating from model input data. It is therefore desirable to assess the model's sensitivity to these uncertain inputs. A global sensitivity analysis that is capable of exploring the entire input space and accounts for all interactions often remains limited due to computational challenges resulting from a large number of necessary model runs. We address this research gap by integrating Gaussian process emulation into landslide run-out modeling and apply it to the open-source simulation tool *r.avaflow*. The feasibility and efficiency of our approach is illustrated based on the 2017 Bondo landslide event. The sensitivity of aggregated model outputs, such as the angle of reach, impact area, and spatially resolved maximum flow height and velocity, to the dry-Coulomb friction coefficient, turbulent friction coefficient, and the release volume is studied. The results of first-order effects are consistent with previous results of common one-at-a-time sensitivity analyses. In addition to that, our approach allows us to rigorously investigate interactions. Strong interactions are detected on the margins of the flow path where the expectation and variation of maximum flow height and velocity are small. The interactions generally become weak with an increasing variation of maximum flow height and velocity. Besides, there are stronger interactions between the two friction coefficients than between the release volume and each friction coefficient. In the future, it is promising to extend the approach for other computationally expensive tasks like uncertainty quantification, model calibration, and smart early warning.

Keywords Landslide run-out modeling · Global sensitivity analysis · Gaussian process emulation · Emulator uncertainty

Introduction

Flow-like landslides, e.g., rock avalanches and debris flows, pose an ongoing threat to life, property, and environment in mountainous regions around the world. In order to assess their hazard and design mitigation strategies, many research efforts have been devoted to developing computational landslide run-out models which are capable of simulating the dynamics of the flow over complex topographies. The majority of these models employ depth-averaged shallow flow equations derived from mass and momentum balance. Examples are TITAN2D (Pitman et al., 2003), Volcflow (Kelfoun & Druitt, 2005), SHALTOP (Mangeney et al., 2007), DAN3D (Hung & McDougall, 2009), RAMMS (Christen et al., 2010), *r.avaflow* (Mergili et al., 2017), and *faSavageHutterFOAM* (Rauter et al., 2018) (see McDougall (2017) for a review).

Such models generally require a variety of input data, including release area and volume (a release polygon given as a shape file or a raster map of release heights), flow resistance parameters (dry-Coulomb friction and turbulent friction parameters for the Voellmy rheology), and topographic data (a digital elevation model). If the input data are accurate, the models can be deterministically run to predict characteristics of the landslide's bulk behavior, such as run-out distance, impact area, spatio-temporally resolved flow height, and

velocity. In practice, however, the input data usually involve large uncertainties (Dalbey et al., 2008). For example, release areas and volumes of landslides are challenging to predict due to the complexity of geological pre-conditioning factors and often a lack of subsurface information. They may be approximated by heavily tailed probability density functions based on the statistical properties of landslide inventories (Quan Luna et al., 2013). The flow resistance parameters are more conceptual than physical (Fischer et al., 2015). For a past landslide event, deterministic trial-and-error calibration (Hung & McDougall, 2009; Lucas et al., 2014; Moretti et al., 2015; Schraml et al., 2015) or probabilistic Bayesian calibration (Aaron et al., 2019; Heredia et al., 2020; Moretti et al., 2020) is commonly conducted to obtain the flow resistance parameters that reproduce field observations well. For landslide run-out forecasting, however, values of the flow resistance parameters usually cannot be deterministically determined. In that case, error bounds or probability density functions of the flow resistance parameters based on group calibration of similar events can be used in a probabilistic framework for reliable landslide run-out forecasting (McDougall, 2017). Topographic data may also be subject to uncertainties due to error introduced during source data acquisition or data processing (Zhao & Kowalski, 2020). Therefore, it is essential to study the model's sensitivity to uncertain inputs, which could improve our understanding of the computational landslide run-out models and provide guidelines for their future usage.

Sensitivity analyses on landslide run-out models are commonly based upon local one-at-a-time approaches, i.e., changing one input variable at a time while keeping others at their baseline values in order to explore its isolated effect on model outputs. For example, Borstad and McClung (2009) and Moretti et al. (2015) studied the sensitivity of the run-out model employing the Coulomb-type friction law to the Coulomb friction coefficient and initial condition of the release mass, based on a hypothetical parabolic slope and a real rockslide-debris flow event respectively. Both found model outputs are more sensitive to the Coulomb friction coefficient than the initial condition of release mass. In terms of the Voellmy friction law, Barbolini et al. (2000) and Schraml et al. (2015) studied the sensitivity of model outputs to the two Voellmy friction coefficients and initial condition of release mass, while Hussin et al. (2012) studied the sensitivity of model outputs to the two Voellmy friction coefficients and the entrainment coefficient. A common finding is that the run-out distance is mainly influenced by the Coulomb friction coefficient; Barbolini et al. (2000) reported that the release area generally has a lower influence than the other parameters and Schraml et al. (2015) found the release volume causes little variation of the output of RAMMS-DF; Hussin et al. (2012) found the turbulent friction coefficient has the strongest impact on the maximum flow velocity at control points. Similar one-at-a-time sensitivity analyses of run-out model employing other friction laws, such as the Pouliquen or Mohr-Coulomb law, can be found in Fathani et al. (2017), Pirulli and Mangeney (2008). While straightforward to implement, these types of local sensitivity analysis methods cannot assess potential interactions between input variables. Their result may highly depend on the chosen

baseline values (Girard et al., 2016). In contrast, variance-based global sensitivity analyses can fully explore the input space, quantify the contribution of each variable to the output variation, and identify interactions between different variables. The Sobol' method, one typical variance-based method, has been developed and widely used since the 1990s (Saltelli, 2002; Saltelli et al., 2010; Sobol, 1993; Sobol, 2001). The key idea of a Sobol' sensitivity analysis is that the variance of model output can be quantitatively decomposed into contributions due to the independent effect of every single input factor and combined effects of input factors. These are represented by first-order and higher-order Sobol' indices respectively. The Sobol' indices can therefore be interpreted as measures of relative sensitivity. They allow identifying coupled effects between the various model inputs. The calculation of Sobol' sensitivity indices usually requires Monte Carlo-based methods, leading to a large number of necessary model evaluations. For computationally demanding models, the calculation may be prohibitively expensive. In that case, it is rather promising to employ emulation techniques to overcome the computational challenge.

An emulator is a statistical representation of a computationally demanding model, also referred to as a simulator. While it comes at the price of an additional statistical error, it is typically evaluated several orders of magnitude faster than the simulator. Different emulation techniques have been used in run-out analyses. For example, Bayes linear method (Stefanescu et al., 2012), separable scalar Gaussian process (GP) emulators (Bayarri et al., 2009; Bayarri et al., 2015; Rutarindwa et al., 2019; Spiller et al., 2014), a physics-based emulator using the Ornstein-Uhlenbeck process (Mahmood et al., 2015), and multi-output GP emulator (Gu & Berger, 2016) have been used for probabilistic risk assessment and hazard mapping of pyroclastic flows. Navarro et al. (2018) employed polynomial chaos expansion for Bayesian inference of parameters of a one-dimensional run-out model based on debris flow flume experiment data, and conducted a priori global sensitivity analysis for the flow height at specific locations. Sun et al. (2021) employed scalar GP emulator for Bayesian inference of run-out model parameters and probabilistic prediction of landslide run-out distance. A detailed review of various emulation techniques can be found in (Asher et al. (2015), Razavi et al. (2012)). In this study, we employ GP emulation due to its rich theoretical background and its ability to take emulator uncertainty into account in any following emulator-based analyses.

GP emulation has been developed since 1980s (Currin et al., 1991; O'Hagan, 2006; Sacks et al., 1989). It has been utilized for the purpose of global sensitivity analyses in different fields (Aleksankina et al., 2019; Bounceur et al., 2015; Girard et al., 2016; Lee et al., 2011; Lee et al., 2012; Rohmer & Foerster, 2011). These studies either focus on emulating the evaluation of a few scalar outputs (Girard et al., 2016; Lee et al., 2011; Rohmer & Foerster, 2011), or build separate emulators for each of the many outputs (Aleksankina et al., 2019; Lee et al., 2012). One exception among them is Bounceur et al. (2015), which combines emulation techniques with the principal component analysis leading to the emulation of a reduced-order model. For a simulator with massive outputs like a landslide run-out model, building separate emulators for each output can be computationally intensive (Gu & Berger, 2016). In recent years, great improvement has been made to enable simultaneous emulation for multi-output models (see for instance Gu and Berger (2016), Rougier (2008)).

The goal of this study is twofold: The first is a methodological goal, namely to combine the recent development of emulation techniques (Gu & Berger, 2016; Gu et al., 2018; Gu et al., 2019),

landslide run-out models (Mergili et al., 2017), and global sensitivity analyses (Le Gratiot et al., 2014) to enable global sensitivity analyses of computationally demanding landslide run-out models for the first time. The second goal is application-oriented and aims at employing the methodology to assess the relative importance of different uncertain inputs, specifically flow resistance parameters and the release volume, and their interactions in landslide run-out models based on the 2017 Bondo landslide event as a test case.

This paper is set out as follows. In the "Methodology" section, the methodology is described, including the computational landslide run-out model based on the Voellmy rheology, Sobol' sensitivity analysis, GP emulation, and an algorithm to take emulator uncertainty into account. The "Implementation" section presents our Python-based implementation. The "Case study" section describes the case study. The "Results and discussions" section is devoted to a discussion of our results. In the "conclusions" section, important conclusions are drawn.

Methodology

Computational landslide run-out model based on the Voellmy rheology

Depth-averaged shallow flow type process models have gained popularity in practice and in academia, owing to their good compromise between accuracy and computing time (Rauter et al., 2018). A variety of flow resistance laws can be used with the models depending on landslide types and characteristics of flow material (Hung & McDougall, 2009; Naef et al., 2006; Pirulli & Mangeny, 2008). In the case of flow-like landslides, the Voellmy rheology is one of the most widely used flow resistance laws (Bevilacqua et al., 2019; Frank et al., 2015; Hussin et al., 2012; Schraml et al., 2015). The governing system of the depth-averaged model employing the Voellmy rheology can be expressed in a surface-induced coordinate system as (Christen et al., 2010; Fischer et al., 2012).

$$\begin{aligned} \frac{\partial}{\partial t} \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} hu_x \\ hu_x^2 + g_z k_{a/p} \frac{h^2}{2} \\ hu_x u_y \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + g_z k_{a/p} \frac{h^2}{2} \end{pmatrix} \\ = \begin{pmatrix} g_x h - \frac{u_x}{\|u\|} \left(\mu g_z h + \frac{g}{\xi} \|u\|^2 \right) \\ g_y h - \frac{u_y}{\|u\|} \left(\mu g_z h + \frac{g}{\xi} \|u\|^2 \right) \end{pmatrix} \end{aligned} \quad (1)$$

where X , Y , and Z denote coordinates in the down-slope, cross-slope, and normal directions; t denotes time; h represents flow height; u_x and u_y represent components of the depth-averaged surface tangent flow velocity \mathbf{u} along X and Y directions; g_x , g_y and g_z are components of the gravitational acceleration which are calculated using a finite central differencing scheme (Mergili et al., 2017); μ and ξ are the dry-Coulomb friction coefficient and turbulent friction coefficient, which describe the flow resistance law known as the Voellmy rheology. (For comprehensive details of the model including a schematic plot of the flow model in the surface-induced coordinate system, please refer to Christen et al. (2010), Fischer et al. (2012)).

The process model is solved forward in time; hence, an initial condition $h(X, Y, t_0)$ and $\mathbf{u}(X, Y, t_0)$ is needed. Typically, $\mathbf{u}(X, Y, t_0)$ is zero and $h(X, Y, t_0)$ denotes the release volume and release area. Other essential inputs include the flow resistance parameters and a digital elevation map of the topography. As stated in the

introduction, these input data usually involve uncertainties. The uncertainty of topographic data may be reduced by using high-accuracy remote sensing data. The uncertainty of the release volume and release area of a potential landslide may be more difficult to predict due to the complexity of geological preconditioning factors and often a lack of subsurface information. It is often based on expert judgment. The flow resistance parameters depend on back-analyzing past events. It is still a great challenge to select them for quantitative risk assessment in practice (McDougall, 2017). In this study, we focus on the sensitivity of selected model outputs to the release volume v_o (denoting the landslide magnitude) and the two flow resistance parameters μ and ξ of the Voellmy rheology.

The process model produces numerous outputs, essentially given by flow height h and flow velocity u at every space-time grid point. Other quantities of interest can be calculated based on the spatio-temporally resolved flow height and velocity data and have been used for the purpose of sensitivity analyses, including run-out distance, impact area, deposit area and volume, impact pressure at specific locations, maximum flow height, and velocity at specific locations (Barbolini et al., 2000; Borstad & McClung, 2009; Fathani et al., 2017; Hussin et al., 2012; Pirulli & Mangeney, 2008). In this study, we focus on the spatially resolved maximum flow height and velocity which provide detailed information for hazard assessment and mitigation, as well as the angle of reach and impact area which indicate the overall landslide impact.

- Angle of reach, the tangent of which equals to the ratio of the landslide fall height and projected run-out distance, namely the Heim's ratio (Lucas et al., 2014). The angle of reach generally decreases as the run-out distance increases.
- Impact area, defined as the area of the region where maximum flow height values exceed a threshold value, here 0.1 m.
- Maximum flow height over time at k locations $\{(X_j, Y_j)\}_{j=1, \dots, k}$ denoted as $(h_{i_1}^{\max}, \dots, h_{i_k}^{\max})^T$.
- Maximum flow velocity over time at k locations $\{(X_j, Y_j)\}_{j=1, \dots, k}$ denoted as $(\|u_{i_1}\|^{\max}, \dots, \|u_{i_k}\|^{\max})^T$.

The model defined in Eq. (1) does not include entrainment processes (Christen et al., 2010; Moretti et al., 2012) and topographic curvature effects (Favreau et al., 2010; Fischer et al., 2012). Both can have an impact on landslide run-out simulation and therefore may influence the results of a sensitivity analysis. We do not take them into account in our case study for simplicity. Our approach, however, can be easily extended.

Sobol' sensitivity analysis

Assume that a simulator is denoted by $f(x)$ with a p -dimensional input $x = (x_1, \dots, x_p)^T \in \mathbb{R}^p$ and a scalar output $y \in \mathbb{R}$. For the process model described in the "Computational landslide run-out model based on the Voellmy rheology" section, x is a three-dimensional vector consisting of the two friction coefficients and the release volume, namely $x = (\mu, \xi, v_o)^T$; y could be an aggregated scalar output like the angle of reach or the impact area or an element of a vector output like maximum flow height or velocity at a specific location. Input uncertainties of x induce output uncertainty of y . The essential idea of a Sobol' sensitivity analysis is to decompose the variance of y into contributions caused by

each x_i and their interactions. In practice, p first-order indices $\{S_i\}_{i=1, \dots, p}$ and p total-effect indices $\{S_{Ti}\}_{i=1, \dots, p}$ are usually computed. They are defined as (Saltelli et al., 2010)

$$S_i = \frac{V_{x_i}(E_{x_{-i}}(y|x_i))}{V(y)} \quad (2a)$$

$$S_{Ti} = 1 - \frac{V_{x_{-i}}(E_{x_i}(y|x_{-i}))}{V(y)} \quad (2b)$$

where V and E represent the variance and expectation operator respectively, and x_{-i} denotes the vector consisting of all input factors except x_i . A first-order index S_i accounts for the contribution of the input factor x_i to the variance of the output, independent from other input factors x_{-i} ; a total-effect index S_{Ti} indicates the total contribution of x_i to the output variation, i.e., the sum of its first-order contribution and all high-order effects owing to interactions (Saltelli et al., 2008). The difference of $S_{Ti} - S_i$ thus indicates any interaction between x_i and x_{-i} . Employing this concept to landslide run-out models will hence allow us to investigate the combined effects of the two friction coefficients and the release volume on simulation outputs.

Computing the conditional variances in Eqs. (2a)–(2b) involves nested integrals (Girard et al., 2016). This is analytically impractical for complex simulators like landslide run-out models. Instead, Monte Carlo-based methods are commonly used to estimate the Sobol' indices. The uncertainty introduced by Monte Carlo-based integration can be taken into account using a bootstrap strategy (Archer et al., 1997).

In this study, we employ the numerical procedure presented in Saltelli et al. (2010). The computational cost is $N \cdot (p + 2)$ evaluations of a simulator, where N is the base sample size. More specifically, the denominator $V(y)$ in Eqs. (2a)–(2b) can be estimated using $2 \cdot N$ simulation runs based on two independent sets of input samples. Each set consists of N input samples for the simulator. Moreover, each pair of numerators in Eqs. (2a)–(2b) requires additional N simulation runs corresponding to a new set of N input samples, which is constructed from the two independent sets. It leads to additional $p \cdot N$ simulation runs. (For the detailed procedure, please refer to Saltelli et al. (2010)).

As pointed out in Saltelli et al. (2010), N should be sufficiently large, e.g., 500 or higher, which is critical in our case as the landslide run-out model itself is computationally intensive. If a single run of the simulator described in the "Computational landslide run-out model based on the Voellmy rheology" section costs 32 min, which corresponds to the average run time of the 200 simulation runs in the "Emulator design and validation" section, the sensitivity analysis for three input variables will cost at least $32 \times 500 \times (3 + 2) = 80000$ min, roughly 56 days on a single core. Therefore, it is necessary to employ emulation techniques to improve computational efficiency in order to carry out this type of global sensitivity analysis.

Gaussian process emulation

A simulator, such as a landslide run-out model, represents a deterministic input-output mapping. It is usually computationally impractical to directly use such a simulator for analysis requiring a large number of simulation runs, e.g., a global sensitivity analysis described in the previous section, or an uncertainty quantification,

or a model calibration. In that case, GP emulators have been widely employed owing to their robustness and rich theoretical background (Girard et al., 2016). GP emulation views a simulator as an unknown function from a Bayesian perspective; the prior belief of the simulator behavior, namely a Gaussian process, is updated based on a modest number of simulation runs, leading to a posterior which can be evaluated much faster than the simulator and can then be used for computationally demanding analyses. The fundamental assumption of GP emulation is that the simulator is a smooth continuous function of its inputs (O'Hagan, 2006). Here, we recap the principal ideas of GP emulators used in this study (for detailed information, please refer to Bastos and O'Hagan (2009), Gu and Berger (2016), Gu et al. (2018), O'Hagan (1994)).

Gaussian process emulator for a scalar output

Let $f(\mathbf{x})$ denote a simulator with a p -dimensional input $\mathbf{x} = (x_1, \dots, x_p)^T \in \mathbb{R}^p$ and a scalar output $y \in \mathbb{R}$. For example, if $f(\mathbf{x})$ is the landslide run-out model, \mathbf{x} is the triplet consisting of the release volume and the two friction coefficients, and y is the angle of reach or impact area. $f(\mathbf{x})$ is regarded as an unknown function and will be modeled as a Gaussian process. The Gaussian process is defined by a mean function $m(\cdot)$ and a covariance function $\sigma^2 c(\cdot, \cdot)$ with variance σ^2 and correlation function $c(\cdot, \cdot)$, hence:

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), \sigma^2 c(\cdot, \cdot)) \quad (3)$$

The mean function for any input \mathbf{x} is given by the regression as follows:

$$m(\mathbf{x}) = \mathbf{h}^T(\mathbf{x})\boldsymbol{\theta} \quad (4)$$

where $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_q(\mathbf{x}))^T$ is a q -dimensional vector specifying basis functions, e.g., $\mathbf{h}(\mathbf{x}) = (1, x_1, \dots, x_p)^T$ for a simple linear regression, and $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_q)^T$ is the corresponding q -dimensional vector consisting of q unknown regression parameters. There are a variety of choices for the correlation functions like power exponentials, sphericals, and Matérn. The Matérn correlation function is chosen here following Gu et al. (2018). For any $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ and $\mathbf{x}_j = (x_{j1}, \dots, x_{jp})^T$, their correlation is described by the following:

$$c(\mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^p \left(1 + \frac{\sqrt{5}d_l}{\gamma_l} + \frac{5d_l^2}{3\gamma_l^2} \right) \exp \left(-\frac{\sqrt{5}d_l}{\gamma_l} \right) \quad (5)$$

where $d_l = |x_{il} - x_{jl}|$ represents the distance between the two inputs in the l th dimension, and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)^T$ is a p -dimensional vector consisting of p unknown range parameters.

Equations (3)–(5) represent the prior belief of the simulator's behavior. The fundamental idea now is to update the prior belief following a Bayesian methodology based on evaluations of the simulator at N_{sim} selected inputs $\mathbf{x}^D = \{\mathbf{x}_i\}_{i=1, \dots, N_{sim}}$. Owing to the property of the Gaussian process, the outputs corresponding to \mathbf{x}^D , denoted as $\mathbf{y}^D = \{f(\mathbf{x}_i)\}_{i=1, \dots, N_{sim}}$, follow a multivariate Gaussian distribution:

$$\mathbf{y}^D | \boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma} \sim \mathcal{N}_{N_{sim}}(\mathbf{H}\boldsymbol{\theta}, \sigma^2 \mathbf{R}) \quad (6)$$

where $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \dots, \mathbf{h}(\mathbf{x}_{N_{sim}})]^T$ is the $N_{sim} \times q$ basis design matrix and \mathbf{R} is the $N_{sim} \times N_{sim}$ correlation matrix with (i, j)

element $c(\mathbf{x}_i, \mathbf{x}_j)$. Again, owing to the property of the Gaussian process, the output y^* at any new input \mathbf{x}^* follows a Gaussian distribution conditioned on \mathbf{y}^D , given by the following:

$$y^* | \mathbf{y}^D, \boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma} \sim \mathcal{N}(m', \sigma^2 c') \quad (7a)$$

$$m' = \mathbf{h}^T(\mathbf{x}^*)\boldsymbol{\theta} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}(\mathbf{y}^D - \mathbf{H}\boldsymbol{\theta}) \quad (7b)$$

$$c' = c(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*) \quad (7c)$$

where $\mathbf{r}(\mathbf{x}^*) = (c(\mathbf{x}^*, \mathbf{x}_1), \dots, c(\mathbf{x}^*, \mathbf{x}_{N_{sim}}))^T$.

The parameters $\boldsymbol{\theta}$, σ^2 , and $\boldsymbol{\gamma}$ in Eq. (7a) are the unknowns that need to be updated. Of these, regression parameters $\boldsymbol{\theta}$ and the variance σ^2 can be integrated out using a conjugate analysis and Bayes' theorem. More specifically, a weak prior for $(\boldsymbol{\theta}, \sigma^2)$ is assumed to have the form $p(\boldsymbol{\theta}, \sigma^2) \propto (\sigma^2)^{-1}$, which is within the conjugate family as the likelihood, i.e., Eq. (6). Combining the weak prior and the likelihood gives the posterior $p(\boldsymbol{\theta}, \sigma^2 | \mathbf{y}^D, \boldsymbol{\gamma})$. Then, $\boldsymbol{\theta}$ and σ^2 are successively integrated out from Eq. (7a) by applying the Bayesian chain rule to $p(\boldsymbol{\theta}, \sigma^2 | \mathbf{y}^D, \boldsymbol{\gamma})$ and Eq. (7a). This yields Student's t-distribution with $N_{sim} - q$ degrees of freedom, which describes the distribution of y^* conditioned on \mathbf{y}^D and $\boldsymbol{\gamma}$:

$$y^* | \mathbf{y}^D, \boldsymbol{\gamma} \sim St(m'', \hat{\sigma}^2 c'', N_{sim} - q) \quad (8a)$$

$$m' = \mathbf{h}^T(\mathbf{x}^*)\hat{\boldsymbol{\theta}} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}(\mathbf{y}^D - \mathbf{H}\hat{\boldsymbol{\theta}}) \quad (8b)$$

$$\hat{\sigma}^2 = (N_{sim} - q)^{-1} (\mathbf{y}^D - \mathbf{H}\hat{\boldsymbol{\theta}})^T \mathbf{R}^{-1} (\mathbf{y}^D - \mathbf{H}\hat{\boldsymbol{\theta}}) \quad (8c)$$

$$c' = c(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*) + (\mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{H} - \mathbf{h}^T(\mathbf{x}^*)) \times (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (\mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{H} - \mathbf{h}^T(\mathbf{x}^*))^T \quad (8d)$$

where $\hat{\boldsymbol{\theta}} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{y}^D$. From a Bayesian viewpoint, the remaining unknown $\boldsymbol{\gamma}$ in Eq. (8a) should also be integrated out by employing a certain prior for $\boldsymbol{\gamma}$. The integral, however, is highly intractable and would require computationally intensive methods like Markov Chain Monte Carlo sampling strategies. Instead, $\boldsymbol{\gamma}$ is often estimated by solving an optimization problem, e.g., maximizing its marginal likelihood or finding its marginal posterior mode. In this study, we use the marginal posterior mode estimation, recommended by Gu et al. (2018) due to its robustness. Substituting the marginal posterior mode estimation of $\boldsymbol{\gamma}$ into Eqs. (8a)–(8d), finally, gives the GP emulator, denoted as $\hat{f}(\mathbf{x})$. It provides a prediction of the simulator output at any new input \mathbf{x}^* in the form of Eq. (8b), as well as an assessment of the prediction uncertainty, like a 95% credible interval (CI(95%)) of the prediction. To give a direct impression on the emulation technique, we present an example of how a GP emulator is constructed to approximate a simple one-dimensional function (O'Hagan, 2006), as shown in Fig. 1.

Gaussian process emulator for a vector output

Let $f(\mathbf{x})$ denote a simulator with a p -dimensional input $\mathbf{x} = (x_1, \dots, x_p)^T \in \mathbb{R}^p$ and a k -dimensional output $\mathbf{y} = (y_1, \dots, y_k)^T \in \mathbb{R}^k$. For example, $f(\mathbf{x})$ is the landslide run-out model, \mathbf{x} is the triplet consisting of the release volume and the two flow resistance parameters, and \mathbf{y} is maximum flow height or velocity over time at k locations. In a straightforward Many Single emulator approach (Gu & Berger, 2016), each component of the simulator, i.e., $\{y_j = f_j(\mathbf{x})\}_{j=1, \dots, k}$ is assumed to follow an independent Gaussian process having the form of Eq. (3), with independent parameters $\{\theta_j\}_{j=1, \dots, k}$ $\{\sigma_j^2\}_{j=1, \dots, k}$, and $\{\gamma_j\}_{j=1, \dots, k}$.

For each independent emulator, the range parameters $\gamma_j = (\gamma_{j1}, \dots, \gamma_{jp})^T$ need to be estimated by solving an optimization problem as described in the ‘‘Gaussian process emulator for a scalar output’’ section. As a consequence, the training of the k emulators may take a lot of time when k is large, since k optimization problems need to be solved.

In this study, we use however an alternative approach, namely the parallel partial GP emulator developed by Gu and Berger (2016) to simultaneously emulate the relation between the p -dimensional input and k -dimensional output. Similar to the Many Single emulator approach, each element of the simulator is assumed to follow an independent Gaussian process of the form Eq. (3). The main difference is that all of the k Gaussian processes are assumed to share common range parameters γ , which are then estimated from the overall likelihood (Gu & Berger, 2016). The q -dimensional basis functions $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_q(\mathbf{x}))^T$ are also assumed to be the same. These modifications greatly reduce the emulator training time. Once the estimation of the common γ is obtained, the parallel partial GP emulator is determined, which is now a collection of k Student’s t -distributions. Here, it is denoted as $\{\hat{f}_j(\mathbf{x})\}_{j=1, \dots, k}$. The exact form of the emulator can be found in Gu and Berger (2016).

Algorithm 1 Emulator-based Sobol’ index evaluation

- 1: Choose input configurations $\mathbf{x}^D = \{\mathbf{x}_i\}_{i=1, \dots, N_{sim}}$, usually using a Latin hypercube design.
- 2: Run the simulator $f(\mathbf{x})$, e.g. the landslide run-out model, at each of N_{sim} chosen input configurations to obtain outputs \mathbf{y}^D (see section 2.1).
- 3: Build the emulator $\hat{f}(\mathbf{x})$ based on \mathbf{x}^D - \mathbf{y}^D (see section 2.3.1).
- 4: Set the base sample size N , realization sample size N_r , and bootstrap sample size N_b . Sample inputs $\{\mathbf{x}_i\}_{i=1, \dots, N \cdot (p+2)}$ from the input domain according to Saltelli et al. (2010) (see section 2.2).
- 5: **for** $n_r = 1, \dots, N_r$ **do**
- 6: Sample a set of $N \cdot (p+2)$ realizations of $\hat{f}(\mathbf{x})$ with $\{\mathbf{x}_i\}_{i=1, \dots, N \cdot (p+2)}$, denoted as $\{\hat{f}^{n_r}(\mathbf{x}_i)\}_{i=1, \dots, N \cdot (p+2)}$.
- 7: Compute $\{\hat{S}_i^{n_r, 1}\}_{i=1, \dots, p}$ and $\{\hat{S}_{T_i}^{n_r, 1}\}_{i=1, \dots, p}$ based on the realizations $\{\hat{f}^{n_r}(\mathbf{x}_i)\}_{i=1, \dots, N \cdot (p+2)}$.
- 8: **for** $n_b = 2, \dots, N_b$ **do**
- 9: Sample with replacements $\{\tilde{\mathbf{x}}_i\}_{i=1, \dots, N \cdot (p+2)}$ from $\{\mathbf{x}_i\}_{i=1, \dots, N \cdot (p+2)}$ and record realizations $\{\hat{f}^{n_r}(\tilde{\mathbf{x}}_i)\}_{i=1, \dots, N \cdot (p+2)}$.
- 10: Compute $\{\hat{S}_i^{n_r, n_b}\}_{i=1, \dots, p}$ and $\{\hat{S}_{T_i}^{n_r, n_b}\}_{i=1, \dots, p}$ based on the realizations $\{\hat{f}^{n_r}(\tilde{\mathbf{x}}_i)\}_{i=1, \dots, N \cdot (p+2)}$.
- 11: **end for**
- 12: **end for**
- 13: **return** $\{\hat{S}_i^{n_r, n_b}\}_{i=1, \dots, p}^{n_r=1, \dots, N_r; n_b=1, \dots, N_b}$ and $\{\hat{S}_{T_i}^{n_r, n_b}\}_{i=1, \dots, p}^{n_r=1, \dots, N_r; n_b=1, \dots, N_b}$.
- 14: Estimate S_i and S_{T_i} defined in eqs. (2a)-(2b) using $\hat{S}_i = \frac{1}{N_r \cdot N_b} \sum \hat{S}_i^{n_r, n_b}$ and $\hat{S}_{T_i} = \frac{1}{N_r \cdot N_b} \sum \hat{S}_{T_i}^{n_r, n_b}$, with $i = 1, \dots, p$. Quantify the overall uncertainty (i.e. Monte Carlo-based sampling uncertainty and emulator uncertainty) of an estimated Sobol’ index using its standard deviation or CI(95%).

Emulator uncertainty in Sobol’ sensitivity analysis

The efficiency improvement by using GP emulators comes at a cost, i.e., additional emulator uncertainty. We can quantify this type of uncertainty as it can be evaluated from the emulator directly. Yet, we need to find a way to account for this uncertainty in the subsequent analysis. Alongside the development of emulation techniques and global sensitivity analysis methods, a number of approaches have been developed in recent years to address this issue in global sensitivity analyses, e.g., Janon et al. (2014), Le Gratiet et al. (2014), Marrel et al. (2009), Oakley and O’Hagan (2004).

For this study, we choose to integrate the method proposed by Le Gratiet et al. (2014), which combines the work of Janon et al. (2014), Oakley and O’Hagan (2004). It can simultaneously take the Monte Carlo-based sampling uncertainty (Sobol’ sensitivity analysis) and emulator uncertainty into account when calculating the Sobol’ indices. We adapt the method to combine the sampling scheme presented in Saltelli et al. (2010) and the GP emulators developed by Gu and Berger (2016, Gu et al. (2018).

The adapted method for a simulator with a scalar output, namely $f(\mathbf{x})$, is shown in Algorithm 1. For a simulator with a k -dimensional output, i.e., $f(\mathbf{x})$, the method is essentially similar. Minor modifications are as follows.

- In steps 1–3, a parallel partial GP emulator $\{\hat{f}_j(\mathbf{x})\}_{j=1, \dots, k}$ is built (the ‘‘Gaussian process emulator for a vector output’’ section) instead of $\hat{f}(\mathbf{x})$.
- Steps 5–14 are repeated for each $\hat{f}_j(\mathbf{x})$ to evaluate the Sobol’ indices at the j th element of the k -dimensional output, where $j = 1, \dots, k$.

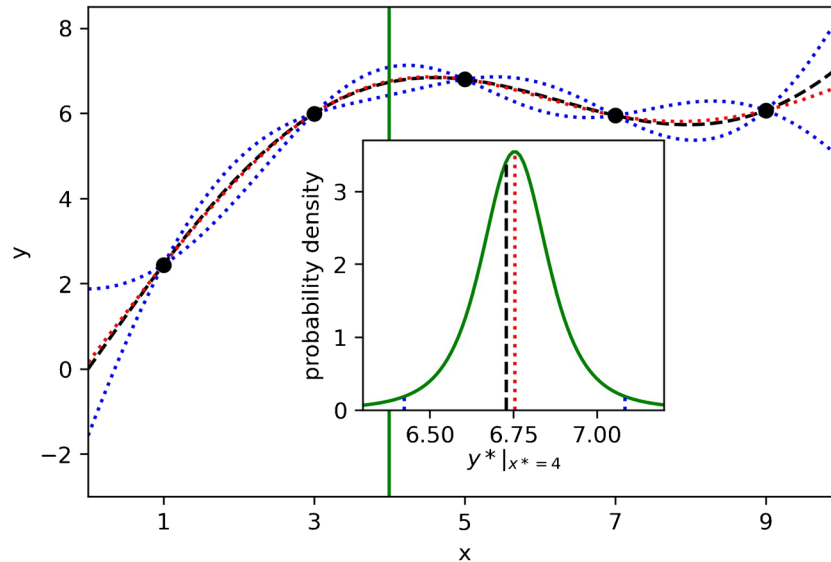


Fig. 1 One-dimensional example of GP emulation. The dashed black line represents the true function $y = x + 3 \sin(x/2)$ that we want to approximate (O’Hagan, 2006); the black dots denote the training data; the dotted red line represents the mean of the trained GP emulator corresponding to Eq. (8b); the dotted blue lines represent the 95% credible intervals; note how the credible interval reduces to zero at the given training data. The embedded plot shows the GP output for input $x^* = 4$ (location indicated by the green vertical line), which results in a student-t distribution of y^* at the input $x^* = 4$

Implementation

The methodology presented in “Methodology” consists of several components, including the Voellmy-type landslide run-out model, multi-output GP emulation (Gu & Berger, 2016), Sobol’ sensitivity analysis (Saltelli et al., 2010), and an algorithm that deals with emulator uncertainty in Sobol’ sensitivity analysis (Le Gratiet et al., 2014). To implement it, we rely on open-source software and packages that have been recently developed for each component. It should be noted that although these individual building blocks exist to date, they do not interact seamlessly as of now. A software framework that allows us to efficiently couple and leverage these building blocks together does not exist. Our Python-based implementation provides such a framework. Its benefit is that only one controlling Python script is required to automatically run simulations at design points based on a Latin hypercube design (see the “Emulator design and validation” section), construct GP emulators, and conduct Sobol’ sensitivity analysis. It coordinates the individual building blocks which involve different programming languages and dependencies, from within a single Python environment. It therefore automatizes the workflow, reduces the redundant manual and potentially error-prone data format transformation between different software and packages, and minimizes the requirement of users’ knowledge on the dependent software and packages. The principle components of the implementation are as follows:

- Simulator: Mergili et al. (2017) presented the open-source software `r.avaflow` for simulation of a variety of mass flows, which relies on GRASS GIS 7. It employs a Voellmy-type model (“Computational landslide run-out model based on the Voellmy rheology” section) and a multi-phase mass flow model (Pudasaini & Mergili, 2019). Here, the former is the simulator under investigation. We implemented a Python-based wrapper

to automatically prepare a batch job, run simulations, and extract outputs given the selected values of input variables x^D , without explicitly starting GRASS and `r.avaflow`.

- Emulator: Gu et al. (2019) presented the R package RobustGaSP (Robust Gaussian Stochastic Process Emulation), in which they implemented the marginal posterior mode estimator for the range parameters γ (see the “Gaussian process emulator for a scalar output” section) and the parallel partial GP emulator (see the “Gaussian process emulator for a vector output” section). We implemented a Python-based wrapper based on `rpy2` (the Python interface to the R language) to utilize RobustGaSP within the unified Python-based framework.
- Emulator-based Sobol’ analysis: Herman and Usher (2017) presented the Python package SALib (Sensitivity Analysis Library in Python), in which the numerical procedure of calculating the Sobol’ indices for a simulator is implemented. We extended their codes to realize Algorithm 1 which enables emulator-based Sobol’ analysis for multi-output simulators.

It should be noted that our Python-based framework is implemented in a modular way. The sensitivity of any other landslide run-out model can therefore be studied using our workflow by simply replacing the simulator.

Case study

Case background

Pizzo Cengalo (see Fig. 2), located in the Swiss Alps, is subjected to rock fall and landslide events for decades due to its geological preconditioning factors (Walter et al., 2020). Two recent landslide events in that area are well-documented and widely studied. The first event occurred on December 27, 2011. Around 1.5 million m^3

of rock detached from the northeastern face of Pizzo Cengalo and evolved into a rock avalanche traveling 2.7 km down the Bondasca valley. The second event occurred on August 23, 2017. Approximately 3 million m³ of rock was released from the northeastern face of Pizzo Cengalo, leading to a rock avalanche traveling 3.2 km down the Bondasca valley. A part of the rock avalanche turned into an initial debris flow, followed by a series of additional debris flows within 48 h, which reached the village Bondo (Walter et al., 2020).

Our case study is based on the topography and release area of the 2017 landslide event. A pre-event digital elevation model (DEM) and a post-event DEM are available, both with 1-m resolution. They are based on airborne laser scans after the 2011 and after the 2017 events, as well as aerial images acquired by the Swiss topographic services Swisstopo (Walter et al., 2020). Release area and initial mass distribution of the event can be obtained from the height difference map of the two DEMs. As the topographic input, we use a merged DEM based on the pre-event and post-event DEMs. The merged DEM reflects the post-event topography in the release area and pre-event topography in other areas. In addition, we use the same release area as the 2017 landslide event, as shown in Fig. 2. The grid size of the computational mesh for the simulator is set to be 10 m.

It should be noted that the intention of the case study is not to back-analyze the 2017 landslide event. Other publications are devoted to that research question (Mergili et al., 2020; Walter et al., 2020). Our focus is to apply the novel emulator-based global sensitivity analysis to the Bondo event in order to assess the model's sensitivity to flow resistance parameters μ and ξ , as well as the release volume v_o (see "Computational landslide run-out model based on the Voellmy rheology" section).

Ranges of uncertain inputs

Sosio et al. (2008) summarized typical ranges for μ and ξ based on a variety of literature. For rock avalanches and debris flows, the range for μ is 0.05–0.25 and that for ξ is 200–1000 m/s². Schraml et al. (2015) presented many back-analyzed $\mu - \xi$ sets, consisting of published values in the literature and their own case study. For most of the rock avalanche and debris flow events, μ lies within the range 0.02–0.25 and ξ varies between 100 and 2000 m/s². Aaron and McDougall (2019) presented back-analyses results of a rock avalanche dataset consisting of 45 past rock avalanche events. Their calibrated values of μ vary between 0.025 and 0.29, except in 4 cases in which the path material is bedrock. The calibrated values of ξ are in the range 200–2100 m/s².

Based on the reference studies, we set the ranges 0.02–0.3 and 100–2200 m/s² for μ and ξ respectively. As regards the release volume v_o , we assume it varies between 1.5 and 4.5 million m³, namely $\pm 50\%$ based on the 3-million m³ release volume of the 2017 landslide event. This is achieved by multiplying the distribution of the initial mass of the 2017 landslide event with a value between 0.5 and 1.5. To sum up, the three uncertain inputs result in a three-dimensional input space, where μ , ξ , and v_o vary independently within 0.02–0.3, 100–2200 m/s², and 1.5–4.5 million m³.

Emulator design and validation

To prepare the emulator training data, $N_{sim} = 200$ samples are drawn from the three-dimensional input space using the

maximin Latin hypercube design which maximizes the minimum distance between design points to achieve optimum space-filling properties (Aleksankina et al., 2019) (see Fig. 3). This results in $\mathbf{x}^D = \{(\mu_i, \xi_i, v_{oi})^T\}_{i=1, \dots, 200}$. One run-out simulation takes 32 min on average on a laptop with an Intel Core i7-9750H CPU. For each simulation run, we extract the angle of reach and impact area, as well as $(h_{i_1}^{max}, \dots, h_{i_k}^{max})^T$ and $(\|u_{i_1}\|^{max}, \dots, \|u_{i_k}\|^{max})^T$ at $k = 47958$ chosen locations. This corresponds to the two aggregated scalar outputs and the two vector outputs in the "Computational landslide run-out model based on the Voellmy rheology" section. At each of the 47,958 locations, at least one of the 200 simulation runs has a maximum flow height value larger than 0.1 m. Correspondingly, two scalar GP emulators (the "Gaussian process emulator for a scalar output" section) and two parallel partial GP emulators (the "Gaussian process emulator for a vector output" section) are built based on \mathbf{x}^D and its respective simulation outputs. Each parallel partial GP emulator takes about 0.05 s to determine maximum flow height or velocity at all 47,958 locations for a new input configuration.

Before using the emulators for our further sensitivity analysis, we validate their performance. The proportion of validation outputs that lie in emulator-based 95% credible intervals is chosen as the diagnostic, denoted as $P_{CI(95\%)}$. This is commonly used in the literature (Bounceur et al., 2015; Gu & Berger, 2016; Lee et al., 2011; Spiller et al., 2014). It is defined as follows:

$$P_{CI(95\%)} = \frac{1}{n} \sum_{i=1}^n \mathbf{1} \left\{ f(\mathbf{x}_i^*) \in \hat{f}(\mathbf{x}_i^*)_{CI(95\%)} \right\} \quad (9)$$

where n is the number of input configurations for validation, $f(\mathbf{x}_i^*)$ and $\hat{f}(\mathbf{x}_i^*)_{CI(95\%)}$ denote the simulation output and the CI(95%) of the emulator prediction at the input \mathbf{x}_i^* respectively. $P_{CI(95\%)}$ would be close to 0.95 for an ideal emulator.

The two scalar emulators are validated using the leave-one-out cross-validation method as implemented in the RobustGaSP package (meaning $n = 200$) (see Fig. 4). Both emulators perform well with emulator prediction values being close to simulator outputs and $P_{CI(95\%)}$ close to 0.95. As no cross-validation scheme is implemented in the RobustGaSP package for a parallel partial GP emulator, we validate the two parallel partial GP emulators for $(h_{i_1}^{max}, \dots, h_{i_k}^{max})^T$ and $(\|u_{i_1}\|^{max}, \dots, \|u_{i_k}\|^{max})^T$ using additional 20 simulation runs based on an independent maximin Latin hypercube design (see Fig. 3). Figure 5(a) shows $P_{CI(95\%)}$ values at each location and their distribution in the form of a box plot based on the maximum flow height emulator. Figure 5(b) shows the same evaluation based on the maximum flow velocity emulator. The lowest $P_{CI(95\%)}$ value of the maximum flow height/velocity emulator is 0.6/0.65, and 95% of the $P_{CI(95\%)}$ values of both emulators are within 0.8–1. Both emulators show good performance with mean values of $P_{CI(95\%)}$ over all locations being 0.93 and 0.94 respectively.

Preliminary convergence analysis

The base sample size N , realization sample size N_r , and bootstrap sample size N_b need to be determined before using the validated

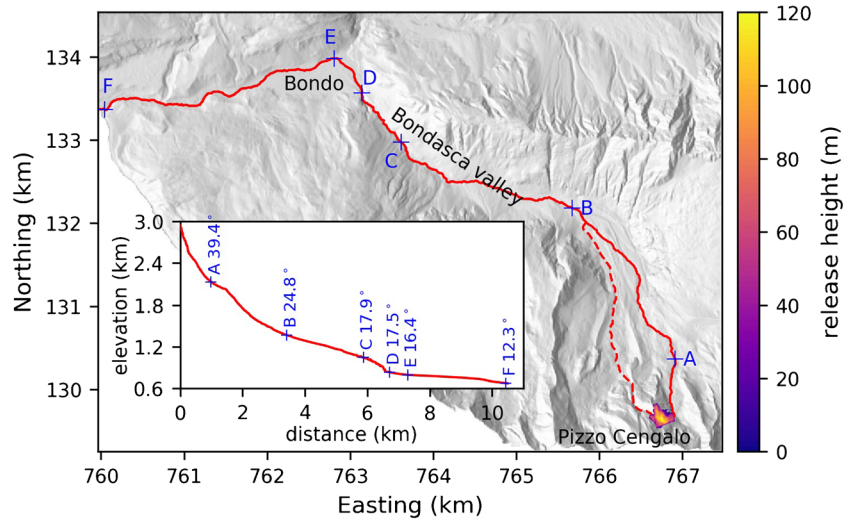


Fig. 2 Pizzo Cengalo-Bondo topography. The colormap shows the distribution of the released mass of the 2017 landslide event (shown in the 10-m resolution of computational mesh grid used for the simulations). The solid line and dashed line denote the major and minor flow paths. The embedded plot in the bottom-left corner shows the profile of the major flow path, on top of which locations A–F with respective angle of reach are noted for our later discussion in the “Maximum flow height and velocity” section.

emulators for Sobol’ sensitivity analysis (see Algorithm 1). Here, we present the results of a convergence analysis based on the validated emulator for the angle of reach in order to determine values for these sample sizes. Figure 6 shows how the estimated Sobol’ indices and their CI(95%) values change with N increasing from 200 to 10,000 with a step size of 200, while keeping $N_r = N_b = 50$. It can be seen that the estimated Sobol’ indices tend to converge when N is large than 4000, and their CI(95%) lengths almost do not decrease for $N \geq 6000$. We conducted the same analysis with $N_r = N_b = 100$ and $N_r = N_b = 200$. The results are similar to our findings with $N_r = N_b = 50$, indicating little impact of N_r and N_b . Therefore, we set $N = 6000$ and $N_r = N_b = 50$ for the following sensitivity study. It leads to $N(p+2) = 6000(3+2) = 30000$ samples from the three-dimensional input space to estimate the Sobol’ indices, namely $\{(\mu_i, \xi_i, \nu_{0i})^T\}_{i=1, \dots, 30000}$. Among them, 2000 samples are used to estimate the overall variance term $V(y)$ in Eqs. (2a)–(2b) (see section Sobol’ sensitivity analysis).

Results and discussions

Angle of reach and impact area

The box plot in Fig. 7(a) shows the distribution of emulator-predicted angle of reach values corresponding to the 12,000 samples used to estimate the variance of the angle of reach (see the “Preliminary convergence analysis” section). Due to input uncertainties, the angle of reach could vary in a wide range, around 11.8–25.7°. The mean is 17.9°. The standard deviation is 3.1° which corresponds to the square root value of $V(y)$ in Eqs. (2a)–(2b). The bar plots in Fig. 7(a) display the estimated first-order and total-effect Sobol’ indices, with CI(95%) denoting the Monte Carlo-based sampling uncertainty and emulator uncertainty. Each pair of bar plots corresponds to the first-order and total-effect Sobol’ indices of one input variable. It is evident that angle of reach is dominated by the dry-Coulomb friction coefficient μ of which the first-order index is over 0.9, whereas both the turbulent

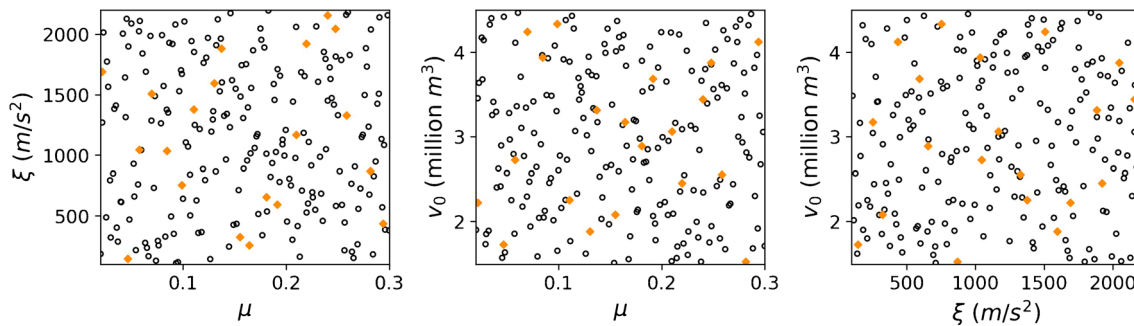


Fig. 3 Two-dimensional projection of the 200 training samples (void circles) and 20 validation samples (solid diamonds) from two independent maximin Latin hypercube designs. Left ξ – μ , middle ν_0 – μ , right ν_0 – ξ . The 200 samples are used to build the emulators (see the “Gaussian process emulation” section). The 20 samples are used to validate the parallel partial GP emulators

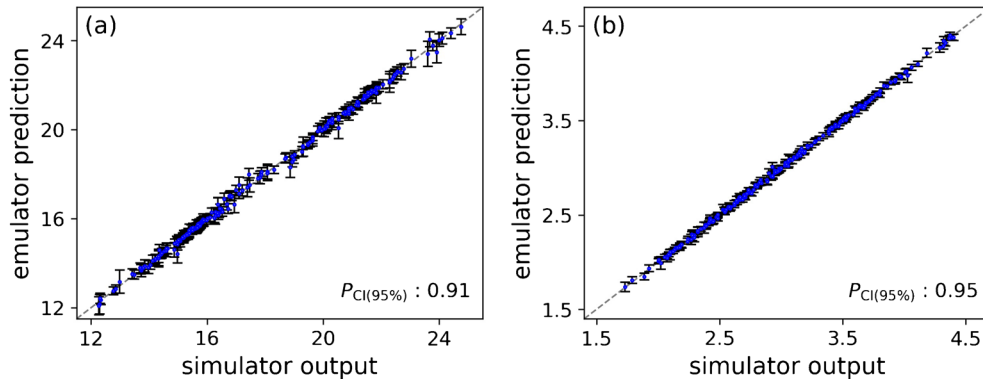


Fig. 4 Leave-one-out cross-validation of the GP emulators for scalar outputs (a) angle of reach (in degree) and (b) impact area (in million m²). The error bars denote 95% credible intervals of the emulator predictions

friction coefficient ξ and the release volume v_o show little influence on the angle of reach, with both first-order indices being smaller than 0.05. This result is expected since μ governs the slope angle on which flow mass begins to deposit (McDougall, 2017). It is also consistent with the common finding in former one-at-a-time sensitivity analyses on landslide run-out models employing the Voemly rheology, such as Barbolini et al. (2000), Frey et al. (2016), Hussin et al. (2012), Schraml et al. (2015). All of them found that the run-out distance (indicated by the angle of reach) is predominantly affected by the dry-Coulomb friction coefficient μ . In particular, Barbolini et al. (2000) found that there is a difference of about half an order of magnitude between the sensitivity of run-out distance to μ and to other parameters like ξ , release height, and release area. Furthermore, it is noteworthy that the difference between the first-order and total-effect indices is small, indicating weak interactions among the three input variables regarding the angle of reach.

Similarly, the box plot in Fig. 7(b) shows the distribution of emulator-predicted impact area values. Owing to input uncertainties, the impact area could vary between 1.5 and 4.5 million m² with a standard deviation 0.6 million m². From the bar plots, it

can be seen that estimated first-order indices of μ , ξ , and v_o are around 0.67, 0.15, 0.18 respectively. It indicates that μ contributes the most to the variance of the impact area, followed by v_o and ξ . Similar to the results on the angle of reach, the small difference between the first-order and total-effect indices implies that the three input variables barely interact with each other concerning the impact area. Compared to the results of the angle of reach, the importance of μ on the impact area decreases and that of ξ and v_o increases. A plausible explanation is that the angle of reach only depends on the deposit (assuming that the release area remains the same) where μ plays the dominant role, whereas the impact area depends on all inundated regions where all three input variables may have an impact.

Maximum flow height and velocity

Before discussing global sensitivity analysis results on maximum flow height and velocity, we summarize the statistics that are needed to interpret the results. Figure 8(a)–(c) show the mean, standard deviation, and coefficient of variation of emulator-predicted maximum flow height values at each location. Figure 8(d)–(f) show the counterparts of emulator-predicted

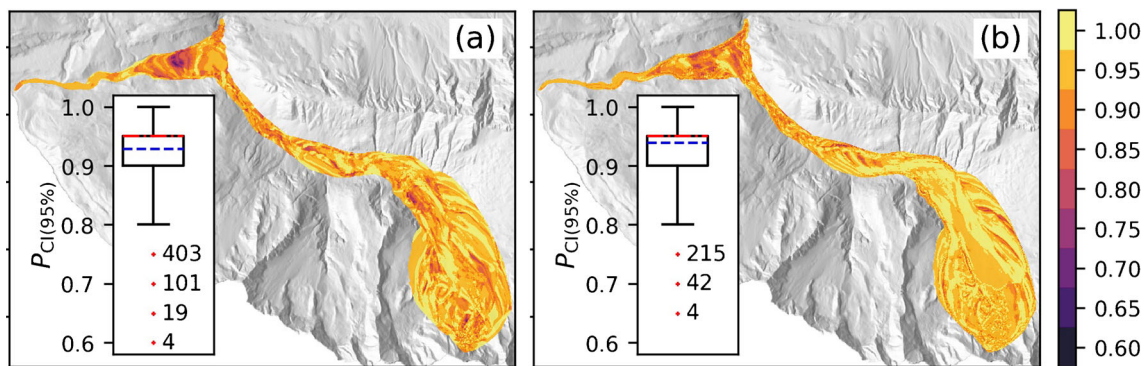


Fig. 5 Validation of the parallel partial GP emulators for vector outputs (a) $(h_i^{\max}, \dots, h_k^{\max})^T$ and (b) $(\|u_i\|^{\max}, \dots, \|u_k\|^{\max})^T$ with $k = 47958$, using 20 validation runs based on an independent maximin Latin hypercube design. In each panel, the colormap shows the $P_{CI(95\%)}$ values at each location; the box plot presents the distribution of $P_{CI(95\%)}$ values. In the box plot, the whiskers denote the 2.5th and 97.5th percentiles; the blue dashed line denotes the mean; the number of outliers for each outlier value is noted due to overlapping. The mean of $P_{CI(95\%)}$ over all locations for maximum flow height/velocity is 0.93/0.94

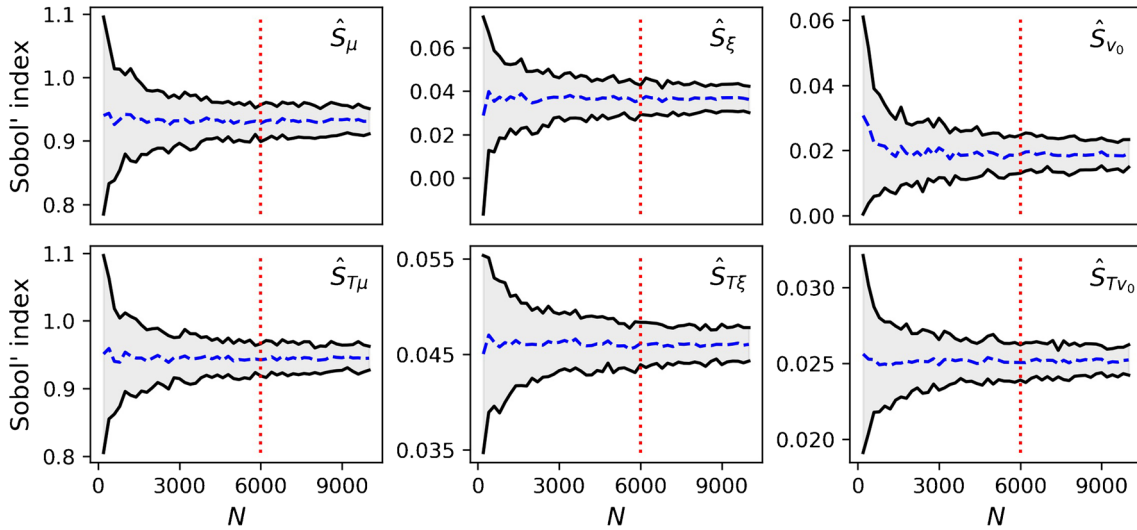


Fig. 6 First-order (first row) and total-effect Sobol' indices (second row) based on the GP emulator for the angle of reach, with $N_r = N_b = 50$ and N increasing from 200 to 10,000 with a step size of 200. \hat{S}_μ , \hat{S}_ξ , and \hat{S}_{v_0} denote estimated first-order Sobol' indices of μ , ξ , and v_0 . $\hat{S}_{T\mu}$, $\hat{S}_{T\xi}$, and \hat{S}_{Tv_0} denote estimated total-effect Sobol' indices of μ , ξ , and v_0 (see step 14 in Algorithm 1 and Eq. (2a)–(2b)). In each panel, the dashed line and solid line show the change of the estimated Sobol' index and its 95% credible interval respectively; the estimated Sobol' index tends to converge for $N \geq 4000$ and the length of its 95% credible interval hardly decreases for $N \geq 6000$

maximum flow velocity values. The major and minor flow paths as well as locations A–F along the major flow path are noted to facilitate the description of results. The profile of the major flow path and the angle of reach values corresponding to locations A–F are shown in Fig. 2. Location A sits near the release area, where the slope is steep. From location B to location D is the Bondasca valley. Location C corresponds to the mean location of 12,000 angle of reach values (17.9°), denoting the average run-out distance. From location D to location E is the debris flow retention basin (Walter et al., 2020). Location F is near the west boundary of the DEM.

It can be seen from Fig. 8(a) and (d) that in general, the mean of maximum flow height gradually decreases along the flow path

whereas the mean of maximum flow velocity first increases then decreases reflecting the acceleration and deceleration process. Along the path cross-section direction, both the mean of maximum flow height and that of maximum flow velocity generally decrease from the center to the sides. In addition, the mean values in the upstream area of location B are on average much larger than the mean values in the downstream area of location B, possibly because the average slope from the release zone to location B is larger than that beyond location B (see Fig. 2) and the corner around location B decelerates the flow mass.

The standard deviation shown in Fig. 8(b) and (e) reflects the variation of maximum flow height and velocity at each location

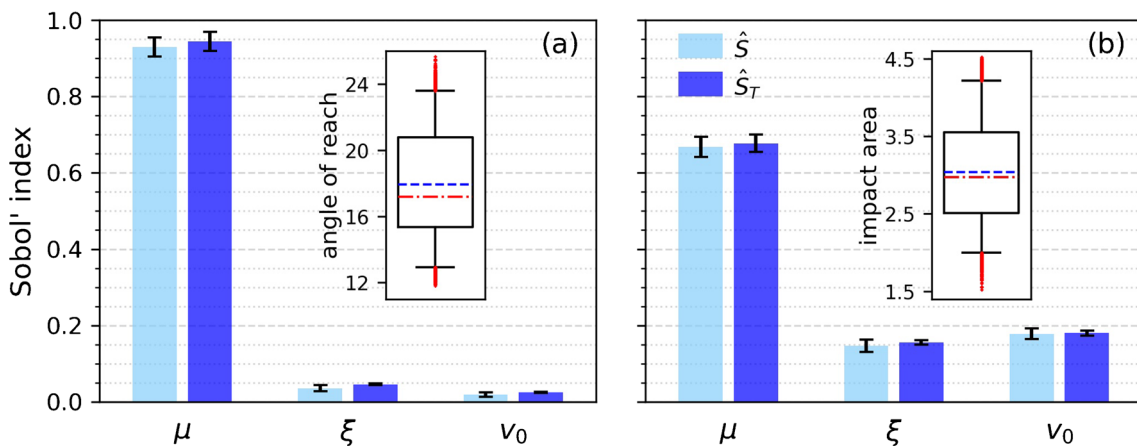


Fig. 7 Sobol' indices for aggregated scalar outputs (a) angle of reach and (b) impact area. The error bars of the bar plots indicate 95% credible intervals of estimated Sobol' indices, which account for Monte Carlo–based sampling uncertainty and emulator uncertainty. The box plots show the distribution of emulator-predicted angle of reach values (in degree) and that of emulator-predicted impact area values (in million m^2). They visualize the variation of the angle of reach and impact area resulting from the uncertain input variables respectively. In each box plot, the whiskers denote the 2.5th and 97.5th percentiles; the blue dashed line denotes the mean; the red dotted dashed line denotes the median; the red crosses denote the outliers

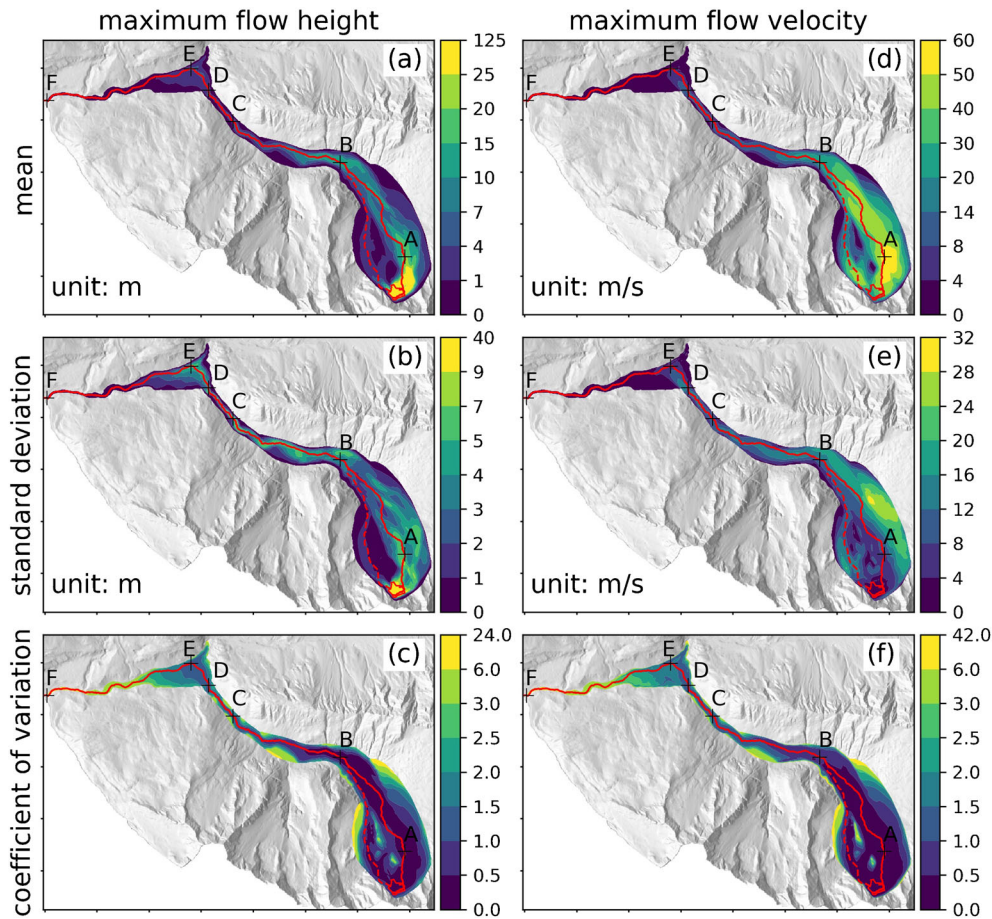


Fig. 8 Statistics of emulator-predicted maximum flow height (left column) and velocity (right column) at $k = 47958$ locations. For each location, the mean (first row), standard deviation (second row), and coefficient of variation (third row) are calculated from 12,000 emulator-predicted maximum flow height and velocity values at that location (see the “Preliminary convergence analysis” section). The polygon at the bottom-right corner of each panel denotes the release area. The local low/high values on the left side of location A in each panel result from the local ridges (see Fig. 2)

resulting from uncertainties of the three input variables. It corresponds to the square root of $V(y)$ in Eqs. (2a)–(2b). In the Bondasca valley between location B and location D, where the channel is well defined, the standard deviation generally decreases from the center to the sides in lateral direction, similar to the trend observed in Fig. 8(a) and (d).

Figure 8(c) and (f) present the coefficient of variation defined as the ratio of the standard deviation to the mean, representing the relative variation. Comparing Fig. 8(c) and (f) with Fig. 8(a) and (d), we find strong negative correlation between the coefficient of variation and the mean. The coefficient of variation generally increases both along the longitudinal direction and from the center to the sides in the lateral direction. A noteworthy feature is that Fig. 8(b) shows large differences to Fig. 8(e), whereas Fig. 8(c) and (f) greatly resemble each other. It indicates that for maximum flow height and velocity, their absolute variation represented by the standard deviation differs from each other, whereas their relative variation represented by the coefficient of variation shows great similarities.

Figures 9 and 10 present results of the Sobol’ sensitivity analysis on maximum flow height and velocity at each location. The

uncertainties of estimated Sobol’ indices are found to be negligible and have little impact on the discussion (see Fig. 7). The CI(95%) is therefore omitted here to avoid redundancy. In addition, values smaller than 0.1 are not shown in the color maps to highlight the trends that we will shortly discuss.

Figure 9(a)–(c) show the first-order contributions of μ , ξ , and v_o to the variation of maximum flow height at each location. The mean values of \widehat{S}_μ , \widehat{S}_ξ , and \widehat{S}_{v_o} over the 47,958 locations are 0.3, 0.17, and 0.27 respectively. A closer look shows that the dry-Coulomb friction coefficient μ dominates in the downstream area beyond location B, whereas its impact in the upstream area of location B is limited; the turbulent friction coefficient ξ is an influential factor in the upstream area of location B especially in areas around the major flow path, whereas it has a negligible impact in the downstream area of location B; the release volume v_o contributes the most in areas surrounding the release zone and has a significant impact in areas near the minor flow path as well as areas surrounding location B, whereas it shows little influence in the downstream area similar as ξ .

Figure 9(d)–(f) present the first-order contributions of μ , ξ , and v_o to the variation of maximum flow velocity at each location. The

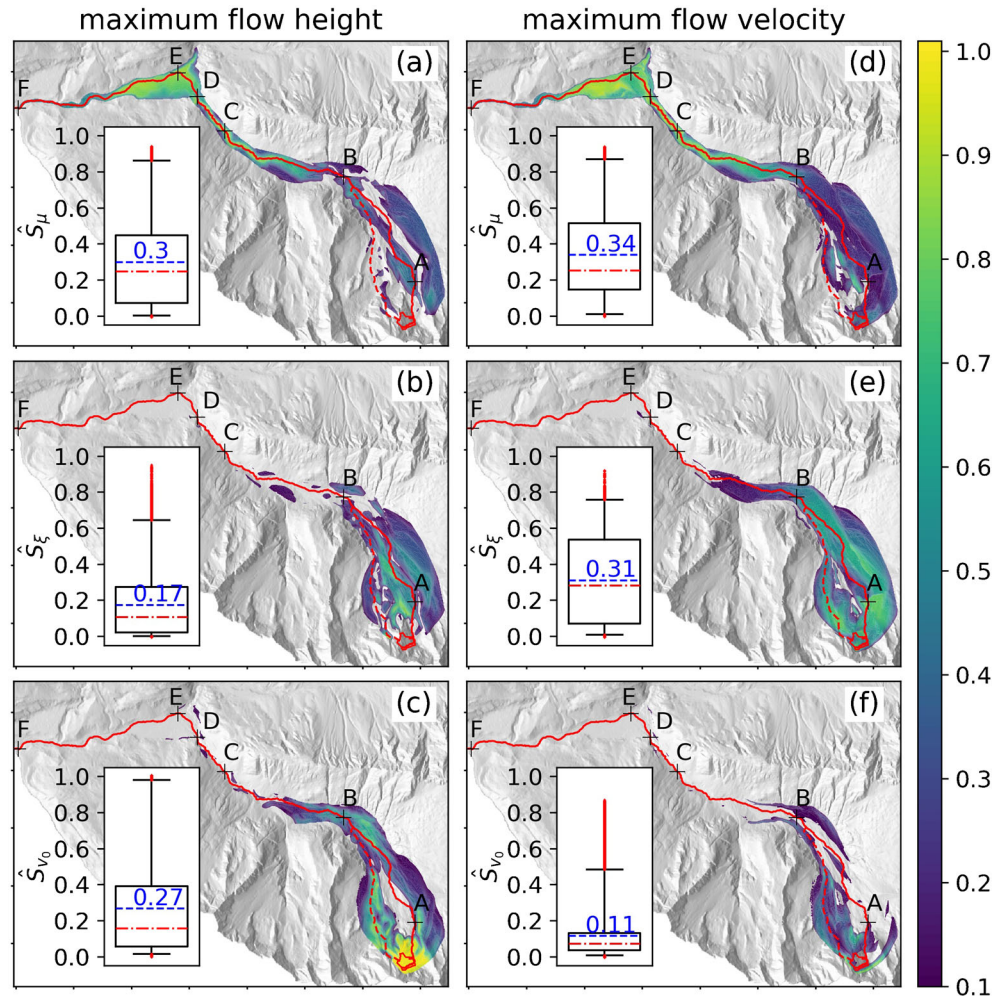


Fig. 9 First-order Sobol' indices for $(h_{l_i}^{\max}, \dots, h_{l_k}^{\max})^T$ (left column) and for $(\|u_{l_i}\|^{\max}, \dots, \|u_{l_k}\|^{\max})^T$ (right column). In each panel, values smaller than 0.1 are not shown in the colormap; the box plot presents the distribution of respective first-order indices at all locations (including values smaller than 0.1); the mean over all locations is notated in the box plot

mean values of \hat{S}_μ , \hat{S}_ξ , and \hat{S}_{v_0} over all the locations are 0.34, 0.31, and 0.11 respectively. A closer inspection shows that the variation of maximum flow velocity in the downstream area beyond location B is predominantly driven by μ , while it has mild impact in the upstream area; ξ contributes the most to the variation of maximum flow velocity in the upstream area of location B, where the mean values of maximum flow velocity are large (comparing Fig. 9(e) with Fig. 8(d)); v_0 only has a mild impact in areas near the release zone and near the minor flow path.

Comparing Fig. 9(a)–(c) with Fig. 9(d)–(f), we find that the first-order contribution of μ to the variation of maximum flow height only slightly differs from its contribution to the variation of maximum flow velocity, with the mean over all locations increasing from 0.3 to 0.34. ξ has more impact on maximum flow velocity than on maximum flow height, with a difference of 0.14 on average. The influence of v_0 on maximum flow height is more important than its influence on maximum flow velocity, with a difference of 0.16 on average. The dominant role of μ in the downstream area agrees with the finding in the “Angle of reach and impact area” section that μ predominantly affects the angle of reach. The

observation can be well explained based on Mangeny-Castelnaud et al. (2003). More specifically, Mangeny-Castelnaud et al. (2003) studied the forces involved in the momentum equation for the Coulomb friction law and found that the force caused by the dry-Coulomb friction is negligible in the early stage of the flow event (corresponds to the upstream area) while it becomes dominant in the later stage (corresponds to the downstream area). The importance of ξ in the upstream area with large mean values of maximum flow velocity is therefore expected since the turbulent friction term in Eq. (1) is proportional to the square of flow velocity and the role of the dry-Coulomb friction term is not important in this area. It should be noted that the turbulent term artificially limits the overestimated early-stage velocity which results from the hydrostatic hypothesis used in depth-averaged shallow flow models, and therefore leads to more realistic early-stage velocity (Garres-Díaz et al., 2021).

Figure 10(a)–(c) show the difference between total-effect and first-order Sobol' indices for maximum flow height at each location, which indicates the interactions between different input variables. Taking $\hat{S}_{T\mu} - \hat{S}_\mu$ as an example, it accounts for all high-

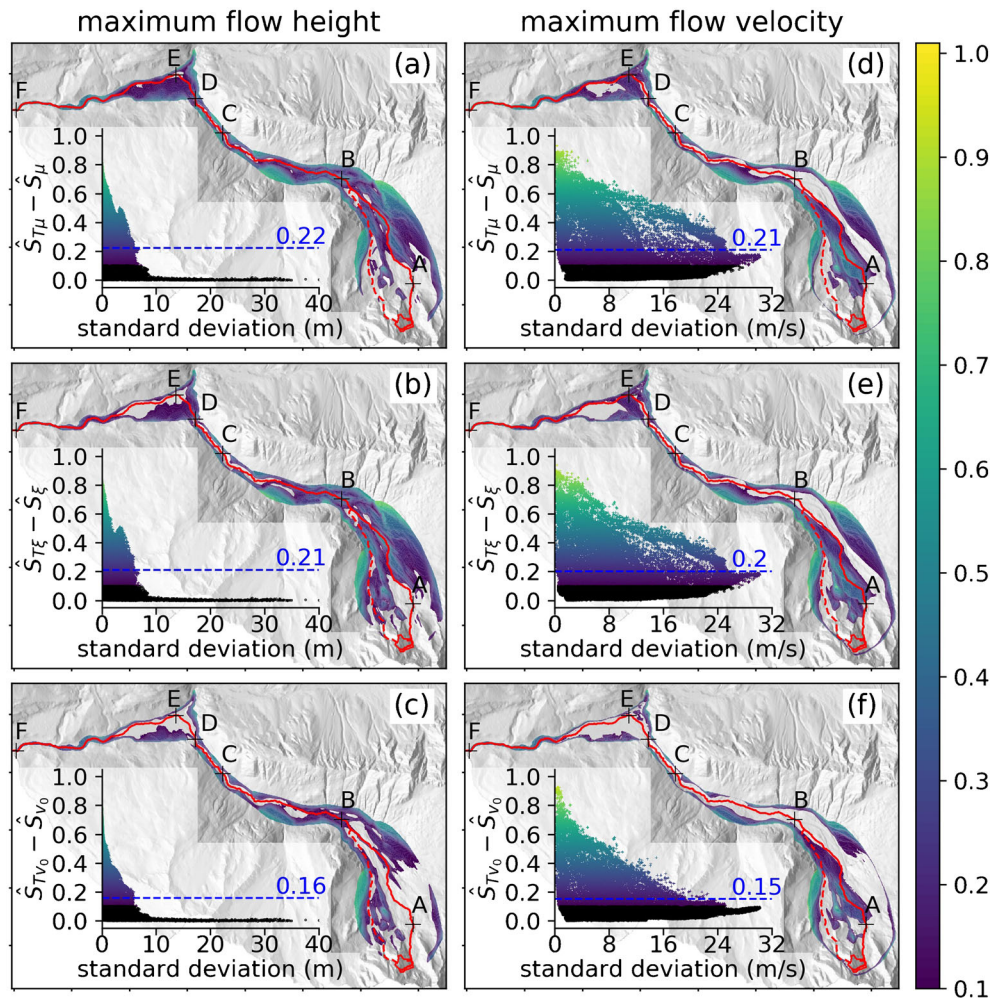


Fig. 10 Difference between total-effect and first-order Sobol' indices for $(h_{i_1}^{\max}, \dots, h_{i_k}^{\max})^T$ (left column) and for $(\|u_{i_1}\|^{\max}, \dots, \|u_{i_k}\|^{\max})^T$ (right column). In each panel, values smaller than 0.1 are not shown in the colormap; the scatter plot shows the difference versus the standard deviation shown in Fig. 8(b) and (e), where difference values larger than 0.1 are plotted using the same color bar as that used for the colormap, and difference values smaller than 0.1 are plotted in black; the mean over all locations is notated in the scatter plot

order effects related to μ , including the second-order interaction between μ and ξ , the second-order interaction between μ and v_o , and the third-order interaction among μ , ξ , and v_o . The mean values of $\hat{S}_{T\mu} - \hat{S}_\mu$, $\hat{S}_{T\xi} - \hat{S}_\xi$, and $\hat{S}_{Tv_o} - \hat{S}_{v_o}$ over all locations are 0.22, 0.21, and 0.16 respectively. The areas where $\hat{S}_{T\mu} - \hat{S}_\mu$, $\hat{S}_{T\xi} - \hat{S}_\xi$, and $\hat{S}_{Tv_o} - \hat{S}_{v_o}$ have large values (see Fig. 10(a)–(c)) are generally in accord with the areas where the mean and standard deviation of maximum flow height have small values (see Fig. 8(a)–(b)), and the coefficient of variation of maximum flow height has large values (see Fig. 8(c)). One exception is the area around the major flow path between location A and location B. The value of $\hat{S}_{Tv_o} - \hat{S}_{v_o}$ in this exception area is very small (see Fig. 10(c)). It means that all high-order effects related to v_o in this area are negligible, including the second-order $v_o - \mu$ interaction, the second-order $v_o - \xi$ interaction, and the third-order $v_o - \mu - \xi$ interaction. The large values of $\hat{S}_{T\mu} - \hat{S}_\mu$ and $\hat{S}_{T\xi} - \hat{S}_\xi$ in this area as shown in Fig. 10(a)–(b) are therefore mainly due to the second-order $\mu - \xi$ interaction since

contributions from $v_o - \mu$, $v_o - \xi$, and $v_o - \mu - \xi$ are negligible. From the inserted scatter plots in Fig. 10(a)–(c) which show respective difference versus the standard deviation, it is evident that the interactions generally decrease with increasing standard deviation. It means that the larger the variation of maximum flow height, the less the interactions between the three parameters.

Figure 10(d)–(f) show the difference between total-effect and first-order Sobol' indices for maximum flow velocity at each location. The mean values of $\hat{S}_{T\mu} - \hat{S}_\mu$, $\hat{S}_{T\xi} - \hat{S}_\xi$, and $\hat{S}_{Tv_o} - \hat{S}_{v_o}$ over all locations are 0.21, 0.2, and 0.15 respectively. Similar to the results on maximum flow height, the areas showing significant differences greatly resemble the areas with low mean values, low standard deviation values, and a high coefficient of variation values of maximum flow velocity (see Fig. 8(d)–(f)). Again, the area around the major flow path between location A and location B is an exception. It can be clearly seen from the scatter plots of respective difference versus the standard deviation that the interactions generally decrease with increasing standard deviation.

Comparing Fig. 10(a)–(c) with Fig. 10(d)–(f), the following trends can be observed for both maximum flow height and maximum flow velocity. First, most of the significant interactions occur on the margins of the flow paths where mean values and standard deviation values are relatively small, whereas values of coefficient of variation are relatively large (see Fig. 8). This may be due to the fact that a location on the margins is only reached by some of the forward simulations (hence some of the three-parameter combinations). Second, the interactions generally decrease with increasing standard deviation. Third, there are stronger interactions between the two friction coefficients μ and ξ than between the release volume v_0 and each friction coefficient.

Conclusions

In this study, we have presented a computationally efficient approach which enables variance-based global sensitivity analyses of computationally demanding landslide run-out models. The methodology couples the novel open-source mass flow simulation tool *r.avaflow* (Mergili et al., 2017), robust Gaussian process emulation for multi-output models (Gu & Berger, 2016; Gu et al., 2018; Gu et al., 2019), and a recent algorithm addressing the emulator uncertainty (Le Gratiet et al., 2014). We have implemented a unified Python-based framework to seamlessly integrate *r.avaflow*, RobustGaSP, and SALib. Based on the 2017 Bondo landslide event, we have employed the approach to study the global sensitivity of selected run-out model outputs to three input variables, namely the release volume and the two friction coefficients. Our main findings are as follows.

- The proposed approach can be successfully used to study the relative importance and interactions of input variables in landslide run-out models, when the trained Gaussian process emulators are validated and the base sample size of Sobol' analysis is properly chosen.
- The first-order effects of each input variable are broadly in line with the results of common one-at-a-time sensitivity analyses in the literature. The dry-Coulomb friction coefficient dominates the angle of reach, and maximum flow height and velocity in the downstream area. The turbulent friction coefficient contributes the most to the variation of maximum flow velocity in the area where maximum flow velocity values are expected to be large. The release volume is found to have a significant impact on maximum flow height in the area surrounding the release zone whereas it shows little impact on maximum flow velocity.
- Interactions between the input variables could be analyzed for the full flow path, which cannot be assessed by commonly used one-at-a-time approaches. Significant interactions between the input variables generally happen on the margins of the flow path. The mean values and standard deviation values of maximum flow height and velocity are small in those areas. The interactions generally decrease with an increasing variation of maximum flow height and velocity. Furthermore, there are stronger interactions between the two friction coefficients than between the release volume and each friction coefficient.

Our study does not consider entrainment processes and topographic curvature effects, as mentioned in the “Computational landslide run-out model based on the Voellmy rheology” section. Studies

have shown that they can have an impact on simulation results and therefore may influence the results of our sensitivity analysis. Work towards this direction should be conducted in the future. Moreover, traditional one-at-a-time sensitivity analyses based on multiple sites have shown that the results of sensitivity analyses can be strongly affected by the topography (Barbolini et al. 2000). To what extent our conclusions based on the Bondo site can be used elsewhere therefore requires further study.

It should be noted that the proposed methodology can be easily extended for variance-based global sensitivity analysis on landslide run-out models that take entrainment processes and topographic effects into account, or on landslide run-out models employing other basal rheologies, or potentially on any computationally demanding models, when the assumption of Gaussian process emulation is fulfilled as stated in the “Gaussian process emulation” section.

In addition, other computationally expensive tasks can also benefit from the significant speed-up owing to emulation techniques. While the run-out simulation takes 32 min on average to determine maximum flow height at the 47,958 locations for a given parameter setting, this time reduces to 0.05 s for evaluating the emulator. Hence, whenever an application requires a large number of model evaluations, like uncertainty quantification and model calibration of landslide run-out models, computational costs for training the emulator will be compensated. In our study, this threshold is determined by the 200 training simulation runs, around 107 h. The emulation techniques likewise have a great potential whenever a splitting between off-line computation (e.g., emulator training) and on-line computation (e.g., urgent computing for early warning systems) is feasible.

Acknowledgements

The authors gratefully acknowledge the support to Hu Zhao by the China Scholarship Council (grant number: 201706260262) and by the Helmholtz Graduate School for Data Science in Life, Earth and Energy.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aaron J, McDougall S (2019) Rock avalanche mobility: the role of path material. Eng Geol 257:105126. <https://doi.org/10.1016/j.enggeo.2019.05.003>

- Aaron J, McDougall S, Nolde N (2019) Two methodologies to calibrate landslide runoff models. *Landslides* 16:907–920. <https://doi.org/10.1007/s10346-018-1116-8>
- Aleksankina K, Reis S, Vieno M, Heal MR (2019) Advanced methods for uncertainty assessment and global sensitivity analysis of an Eulerian atmospheric chemistry transport model. *Atmos Chem Phys* 19(5):2881–2898. <https://doi.org/10.5194/acp-19-2881-2019>
- Archer GEB, Saltelli A, Sobol IM (1997) Sensitivity measures, ANOVA-like techniques and the use of bootstrap. *J Stat Comput Simul* 58(2):99–120. <https://doi.org/10.1080/00949659708811825>
- Asher MJ, Croke BFW, Jakeman AJ, Peeters LJM (2015) A review of surrogate models and their application to groundwater modeling. *Water Resour Res* 51(8):5957–5973. <https://doi.org/10.1002/2015WR016967>
- Barbolini M, Gruber U, Keylock C, Naaim M, Savi F (2000) Application of statistical and hydraulic-continuum dense-snow avalanche models to five real European sites. *Cold Reg Sci Technol* 31(2):133–149. [https://doi.org/10.1016/S0165-232X\(00\)00008-2](https://doi.org/10.1016/S0165-232X(00)00008-2)
- Bastos LS, O'Hagan A (2009) Diagnostics for Gaussian process emulators. *Technometrics* 51(4):425–438. <https://doi.org/10.1198/TECH.2009.08019>
- Bayarri MJ, Berger JO, Calder ES, Dalbey K, Lunagomez S, Patra AK, Pitman EB, Spiller ET, Wolpert RL (2009) Using statistical and computer models to quantify volcanic hazards. *Technometrics* 51(4):402–413. <https://doi.org/10.1198/TECH.2009.08018>
- Bayarri MJ, Berger JO, Calder ES, Patra A, Pitman EB, Spiller ET, Wolpert RL (2015) Probabilistic quantification of hazards: a methodology using small ensembles of physics-based simulations and statistical surrogates. *Int J Uncertain Quantif* 5(4):297–325. <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2015011451>
- Bevilacqua A, Patra AK, Bursik MI, Pitman EB, Macías JL, Saucedo R, Hyman D (2019) Probabilistic forecasting of plausible debris flows from Nevado de Colima (Mexico) using data from the Atenuque debris flow, 1955. *Nat Hazards Earth Syst Sci* 19(4):791–820. <https://doi.org/10.5194/nhess-19-791-2019>
- Borstad CP, McClung DM (2009) Sensitivity analyses in snow avalanche dynamics modeling and implications when modeling extreme events. *Can Geotech J* 46(9):1024–1033. <https://doi.org/10.1139/T09-042>
- Bounceur N, Crucifix M, Wilkinson RD (2015) Global sensitivity analysis of the climate-vegetation system to astronomical forcing: an emulator-based approach. *Earth System Dynamics* 6(1):205–224. <https://doi.org/10.5194/esd-6-205-2015>
- Christen M, Kowalski J, Bartelt P (2010) RAMMS: numerical simulation of dense snow avalanches in three-dimensional terrain. *Cold Reg Sci Technol* 63(1–2):1–14. <https://doi.org/10.1016/j.coldregions.2010.04.005>
- Currin C, Mitchell T, Morris M, Ylvisaker D (1991) Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J Am Stat Assoc* 86(416):953–963
- Dalbey K, Patra AK, Pitman EB, Bursik MI, Sheridan MF (2008) Input uncertainty propagation methods and hazard mapping of geophysical mass flows. *Journal of Geophysical Research: Solid Earth* 113(B5):B05203. <https://doi.org/10.1029/2006JB004471>
- Fathani TF, Legono D, Alfath MA (2017) Sensitivity analysis of depth-integrated numerical models for estimating landslide movement. *Journal of disaster research* 12(3):607–616, DOI <https://doi.org/10.20965/jdr.2017.p0607>
- Favreau P, Mangeney A, Lucas A, Crosta G, Bouchut F (2010) Numerical modeling of landslides. *Geophys Res Lett* 37(15). <https://doi.org/10.1029/2010GL043512>
- Fischer JT, Kofler A, Fellin W, Granig M, Kleemayr K (2015) Multivariate parameter optimization for computational snow avalanche simulation. *J Glaciol* 61(229):875–888. <https://doi.org/10.3189/2015JG14J168>
- Fischer JT, Kowalski J, Pudasaini SP (2012) Topographic curvature effects in applied avalanche modeling. *Cold Reg Sci Technol* 74-75:21–30. <https://doi.org/10.1016/j.coldregions.2012.01.005>
- Frank F, McArdell BW, Huggel C, Vieli A (2015) The importance of entrainment and bulking on debris flow runoff modeling: examples from the Swiss Alps. *Nat Hazards Earth Syst Sci* 15(11):2569–2583. <https://doi.org/10.5194/nhess-15-2569-2015>
- Frey H, Huggel C, Bühler Y, Buis D, Burga MD, Choquevilca W, Fernandez F, Hernández JG, Giráldez C, Loarte E, Masias P, Portocarrero C, Vicuña L, Walser M (2016) A robust debris-flow and GLOF risk management strategy for a data-scarce catchment in Santa Teresa, Peru. *Landslides* 13:1493–1507. <https://doi.org/10.1007/s10346-015-0669-z>
- Garres-Díaz J, Fernández-Nieto E, Mangeney A, de Luna T (2021) A weakly non-hydrostatic shallow model for dry granular flows. *J Sci Comput* 86(25). <https://doi.org/10.1007/s10915-020-01377-9>
- Girard S, Mallet V, Korsakissok I, Mathieu A (2016) Emulation and Sobol' sensitivity analysis of an atmospheric dispersion model applied to the Fukushima nuclear accident. *Journal of Geophysical Research: Atmospheres* 121(7):3484–3496. <https://doi.org/10.1002/2015JD023993>
- Gu M, Berger JO (2016) Parallel partial Gaussian process emulation for computer models with massive output. *Annals of Applied Statistics* 10(3):1317–1347. <https://doi.org/10.1214/16-AOAS934>
- Gu M, Palomo J, Berger JO (2019) Robustgasp: robust Gaussian stochastic process emulation in R. *the R Journal* 11(1):112–136, DOI <https://doi.org/10.32614/RJ-2019-011>
- Gu M, Wang X, Berger JO (2018) Robust Gaussian stochastic process emulation. *Ann Stat* 46(6A):3038–3066. <https://doi.org/10.1214/17-AOS1648>
- Heredia MB, Eckert N, Prieur C, Thibert E (2020) Bayesian calibration of an avalanche model from autocorrelated measurements along the flow: application to velocities extracted from photogrammetric images. *J Glaciol* 66(257):373–385. <https://doi.org/10.1017/jog.2020.11>
- Herman J, Usher W (2017) SALib: an open-source Python library for sensitivity analysis. *The journal of open source software* 2(9), DOI <https://doi.org/10.21105/joss.00097>
- Hungr O, McDougall S (2009) Two numerical models for landslide dynamic analysis. *Comput Geosci* 35(5):978–992. <https://doi.org/10.1016/j.cageo.2007.12.003>
- Hussin HY, Quan Luna B, van Westen CJ, Christen M, Malet JP, van Asch TWJ (2012) Parameterization of a numerical 2-D debris flow model with entrainment: a case study of the Faucon catchment, Southern French Alps. *Nat Hazards Earth Syst Sci* 12(10):3075–3090. <https://doi.org/10.5194/nhess-12-3075-2012>
- Janon A, Nodet M, Prieur C (2014) Uncertainties assessment in global sensitivity indices estimation from metamodels. *Int J Uncertain Quantif* 4:21–36. <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2012004291>
- Kelfoun K, Druitt TH (2005) Numerical modelling of the emplacement of Socoma rock avalanche, Chile. *J Geophys Res* 110(B12). <https://doi.org/10.1029/2005JB003758>
- Le Gratiet L, Cannamela C, Iooss B (2014) A Bayesian approach for global sensitivity analysis of (multifidelity) computer codes. *SIAM/ASA Journal on Uncertainty Quantification* 2(1):336–363. <https://doi.org/10.1137/130926869>
- Lee LA, Carslaw KS, Pringle KJ, Mann GW (2012) Mapping the uncertainty in global CCN using emulation. *Atmos Chem Phys* 12(20):9739–9751. <https://doi.org/10.5194/acp-12-9739-2012>
- Lee LA, Carslaw KS, Pringle KJ, Mann GW, Spracklen DV (2011) Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters. *Atmos Chem Phys* 11(23):12253–12273. <https://doi.org/10.5194/acp-11-12253-2011>
- Lucas A, Mangeney A, Ampuero J (2014) Frictional velocity-weakening in landslides on earth and on other planetary bodies. *Nat Commun* 5(3417). <https://doi.org/10.1038/ncomms4417>
- Mahmood A, Wolpert RL, Pitman EB (2015) A physics-based emulator for the simulation of geophysical mass flows. *SIAM/ASA Journal on Uncertainty Quantification* 3(1):562–585. <https://doi.org/10.1137/130909445>
- Mangeney A, Bouchut F, Thomas N, Vilotte JP, Bristeau MO (2007) Numerical modeling of self-channeling granular flows and of their levee-channel deposits. *Journal of Geophysical Research: Earth Surface* 112(F2). <https://doi.org/10.1029/2006JF000469>
- Mangeney-Castelnau A, Vilotte JP, Bristeau MO, Perthame B, Bouchut F, Simeoni C, Yermeni S (2003) Numerical modeling of avalanches based on Saint Venant equations using a kinetic scheme. *Journal of Geophysical Research: Solid Earth* 108(B11). <https://doi.org/10.1029/2002JB002024>
- Marrel A, Iooss B, Laurent B, Roustant O (2009) Calculations of Sobol indices for the Gaussian process metamodel. *Reliab Eng Syst Saf* 94(3):742–751. <https://doi.org/10.1016/j.res.2008.07.008>
- McDougall S (2017) 2014 Canadian geotechnical colloquium: landslide runoff analysis – current practice and challenges. *Can Geotech J* 54(5):605–620. <https://doi.org/10.1139/cgj-2016-0104>
- Mergili M, Fischer JT, Krenn J, Pudasaini SP (2017) r.avaflow v1, an advanced open-source computational framework for the propagation and interaction of two-phase mass flows. *Geosci Model Dev* 10(2):553–569. <https://doi.org/10.5194/gmd-10-553-2017>
- Mergili M, Jaboyedoff M, Pullarello J, Pudasaini SP (2020) Back calculation of the 2017 Piz Cengalo–Bondo landslide cascade with r.avaflow: what we can do and what we can learn. *Nat Hazards Earth Syst Sci* 20(2):505–520. <https://doi.org/10.5194/nhess-20-505-2020>
- Moretti L, Allstadt K, Mangeney A, Capdeville Y, Stutzmann E, Bouchut F (2015) Numerical modeling of the Mount Meager landslide constrained by its force history derived from seismic data. *Journal of Geophysical Research: Solid Earth* 120(4):2579–2599. <https://doi.org/10.1002/2014JB011426>
- Moretti L, Mangeney A, Capdeville Y, Stutzmann E, Huggel C, Schneider D, Bouchut F (2012) Numerical modeling of the Mount Steller landslide flow history and of the generated long period seismic waves. *Geophys Res Lett* 39(16). <https://doi.org/10.1029/2012GL052511>

- Moretti L, Mangeney A, Walter F, Capdeville Y, Bodin T, Stutzmann E, Le Friant A (2020) Constraining landslide characteristics with Bayesian inversion of field and seismic data. *Geophys J Int* 221(2):1341–1348. <https://doi.org/10.1093/gji/ggaa056>
- Naef D, Rickenmann D, Rutschmann P, McArdell BW (2006) Comparison of flow resistance relations for debris flows using a one-dimensional finite element simulation model. *Nat Hazards Earth Syst Sci* 6(1):155–165. <https://doi.org/10.5194/nhess-6-155-2006>
- Navarro M, Le Maître O, Hoteit I, George D, Mandli KT, Knio O (2018) Surrogate-based parameter inference in debris flow model. *Comput Geosci* 22:1447–1463. <https://doi.org/10.1007/s10596-018-9765-1>
- O'Hagan A (1994) Kendall's advanced theory of statistics, Vol. 2B: Bayesian inference. First published by Arnold, a member of the Hodder headline group, co-published by Oxford University press Inc.
- O'Hagan A (2006) Bayesian analysis of computer code outputs: a tutorial. *Reliab Eng Syst Saf* 91(10):1290–1300. <https://doi.org/10.1016/j.res.2005.11.025>
- Oakley JE, O'Hagan A (2004) Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 66(3):751–769. <https://doi.org/10.1111/j.1467-9868.2004.05304.x>
- Pirulli M, Mangeney A (2008) Results of back-analysis of the propagation of rock avalanches as a function of the assumed rheology. *Rock Mech Rock Eng* 41:59–84. <https://doi.org/10.1007/s00603-007-0143-x>
- Pitman E, Nichita C, Patra A, Bauer A, Sheridan M, Bursik M (2003) Computing granular avalanches and landslides. *Phys Fluids* 15(12):3638–3646. <https://doi.org/10.1063/1.1614253>
- Pudasaini SP, Mergili M (2019) A multi-phase mass flow model. *Journal of Geophysical Research: Earth Surface* 124(12):2920–2942. <https://doi.org/10.1029/2019JF005204>
- Quan Luna B, Cepeda J, Stumpf A, van Westen CJ, Remaitre A, Malet J, van Asch TWJ (2013) Analysis and uncertainty quantification of dynamic run-out model parameters for landslides. In: Margottini C, Canuti P, Sassa K (eds) *Landslide science and practice, Spatial analysis and modelling*, vol 3. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 315–318. https://doi.org/10.1007/978-3-642-31310-3_42
- Rauter M, Kofler A, Huber A, Fellin W (2018) faSavageHutterFOAM 1.0: depth-integrated simulation of dense snow avalanches on natural terrain with OpenFOAM. *Geosci Model Dev* 11(7):2923–2939. <https://doi.org/10.5194/gmd-11-2923-2018>
- Razavi S, Tolson BA, Burn DH (2012) Review of surrogate modeling in water resources. *Water Resour Res* 48(7). <https://doi.org/10.1029/2011WR011527>
- Rohmer J, Foerster E (2011) Global sensitivity analysis of large-scale numerical landslide models based on Gaussian-process meta-modeling. *Comput Geosci* 37(7):917–927. <https://doi.org/10.1016/j.cageo.2011.02.020>
- Rougier J (2008) Efficient emulators for multivariate deterministic functions. *J Comput Graph Stat* 17(4):827–843. <https://doi.org/10.1198/106186008X384032>
- Rutarindwa R, Spiller ET, Bevilacqua A, Bursik MI, Patra AK (2019) Dynamic probabilistic hazard mapping in the long valley volcanic region CA: integrating vent opening maps and statistical surrogates of physical models of pyroclastic density currents. *Journal of Geophysical Research: Solid Earth* 124(9):9600–9621. <https://doi.org/10.1029/2019JB017352>
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4(4):409–423. <https://doi.org/10.1214/ss/1177012413>
- Saltelli A (2002) Making best use of model evaluations to compute sensitivity indices. *Comput Phys Commun* 145(2):280–297. [https://doi.org/10.1016/S0010-4655\(02\)00280-1](https://doi.org/10.1016/S0010-4655(02)00280-1)
- Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S (2010) Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput Phys Commun* 181(2):259–270. <https://doi.org/10.1016/j.cpc.2009.09.018>
- Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S (2008) *Variance-based methods*, John Wiley and Sons, Ltd, chap 4, pp 155–182. DOI: <https://doi.org/10.1002/978047025184.ch4>
- Schraml K, Thomschitz B, McArdell BW, Graf C, Kaitna R (2015) Modeling debris-flow runout patterns on two alpine fans with different dynamic simulation models. *Nat Hazards Earth Syst Sci* 15(7):1483–1492. <https://doi.org/10.5194/nhess-15-1483-2015>
- Sobol' I (1993) Sensitivity analysis for nonlinear mathematical models. *Mathematical Modelling and Computational Experiment* 1:407–414
- Sobol' I (2001) Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math Comput Simul* 55(1):271–280. [https://doi.org/10.1016/S0378-4754\(00\)00270-6](https://doi.org/10.1016/S0378-4754(00)00270-6)
- Sosio R, Crosta GB, Hungr O (2008) Complete dynamic modeling calibration for the Thurwieser rock avalanche (Italian Central Alps). *Eng Geol* 100(1):11–26. <https://doi.org/10.1016/j.enggeo.2008.02.012>
- Spiller ET, Bayarri MJ, Berger JO, Calder ES, Patra AK, Pitman EB, Wolpert RL (2014) Automating emulator construction for geophysical hazard maps. *SIAM/ASA Journal on Uncertainty Quantification* 2(1):126–152. <https://doi.org/10.1137/120899285>
- Stefanescu ER, Bursik M, Cordoba G, Dalbey K, Jones MD, Patra AK, Pieri DC, Pitman EB, Sheridan MF (2012) Digital elevation model uncertainty and hazard analysis using a geophysical flow model. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 468(2142):1543–1563. <https://doi.org/10.1098/rspa.2011.0711>
- Sun X, Zeng P, Li T, Wang S, Jimenez R, Feng X, Xu Q (2021) From probabilistic back analyses to probabilistic run-out predictions of landslides: a case study of Heifangtai terrace, Gansu Province, China. *Eng Geol* 280:105950. <https://doi.org/10.1016/j.enggeo.2020.105950>
- Walter F, Amann F, Kos A, Kenner R, Phillips M, de Preux A, Huss M, Tognacca C, Clinton J, Diehl T, Bonanomi Y (2020) Direct observations of a three million cubic meter rock-slope collapse with almost immediate initiation of ensuing debris flows. *Geomorphology* 351:106933. <https://doi.org/10.1016/j.geomorph.2019.106933>
- Zhao H, Kowalski J (2020) Topographic uncertainty quantification for flow-like landslide models via stochastic simulations. *Nat Hazards Earth Syst Sci* 20(5):1441–1461. <https://doi.org/10.5194/nhess-20-1441-2020>

H. Zhao ✉ · **J. Kowalski**

AICES Graduate School,
RWTH Aachen University,
Schinkelstr. 2a, 52062, Aachen, Germany
Email: zhao@aices.rwth-aachen.de

J. Kowalski

e-mail: kowalski@aices.rwth-aachen.de

F. Amann

Chair of Engineering Geology,
RWTH Aachen University,
Lochnerstr. 4-20, 52064, Aachen, Germany
e-mail: amann@ih.rwth-aachen.de

H. Zhao · **J. Kowalski**

Computational Geoscience, Geoscience Centre,
University of Göttingen,
Goldschmidtstr. 1, 37077, Göttingen, Germany