Rockfall trajectography: 3D (three dimensional) models predictive capability assessment and coefficients calibration using optimization-based processes

Fantin Raibaut[1], Olivier Ivanez[2], Benjamin Barry[3], Cyril Douthe[4]

[1] IRS, Nice, France
[2] TERZATEC, Nice, France
[3] AEGIS GROUPE, Nice, France
[4] Laboratoire Navier, Champs-sur-Marne, France

SUMMARY: This paper presents a method for assessing the predictive capability of three-dimensional (3D) trajectographic simulation models by back-analysis of real rockfall events. The observed and simulated stop points clouds are seen as probability measures, and are compared with help of the Wasserstein distance, which assesses the average stop point forecast error. The method is illustrated through a case study where the most realistic set of soil restitution coefficients is sought by minimising the Wasserstein distance between the forecasted stop points cloud and the observed one, via a black-box optimisation algorithm.

Keywords: rockfall hazard, 3D trajectographic simulation, predictive capability, coefficients calibration, optimization, Wasserstein distance, Mesh Adaptive Direct Search

Introduction

Trajectographic (2D and 3D) methods, which allow to quantify stop points densities, rock velocities and passing heights at any point in space, are mostly used to assess rockfall hazard. Without prior calibration of the soil parameters, different trajectographic studies might lead to very variable results. The forecast uncertainty is therefore high (C2ROP, 2020). Rockfall models are usually calibrated and validated by comparative analysis with experimental results (laboratory / in situ tests), or with natural rockfall events. Assessing the predictive capability of models, i.e. the accuracy with which simulations approximate real rockfalls, is an issue addressed in various ways.
Unidimensional parameters are often used as model precision indicators: runout distances, energy of impact at the base of rock cliffs, translational and rotational velocities, passing height through virtual evaluation screens. These indicators do not assess the 3D model predictive ability regarding lateral dispersions. Hence, the distribution of stop points within predefined zones, or the dispersion angle of the trajectories can be used to characterize rockfall events. Two dimensional methods to compare the overall distribution of stop points have recently been developed, using goodness-of-fit indices (Žabota, 2020). These indices only depend on the coincidence between real and forecasted stop points, consequently they fail to account for the distance between the simulations and the observations (an observed and a forecasted stop point are treated the same as long they do not coincide, no matter how far apart they are).

The aim of the present paper is thus to propose a methodology to evaluate the predictive capability of trajectographic models and to calibrate their calculation coefficients by comparative analysis with real events. This methodology should be reproducible and easy to implement. It should not require costly trials, and should be based on simple and reliable calibration results, such as stop points clouds. The proposed methodology relies on the use of the Wasserstein distance to assess the predictive capability of the numerical model. This method will be first presented and then used in a case study to calibrate the soil restitution coefficients by reverse analysis, minimising the Wasserstein distance with a black-box optimization algorithm called Mesh Adaptive Direct Search (Audet, 2006).

Theory of the Wasserstein distance applied to rockfalls

The Wasserstein distance is related to the Optimal Transport problem and was already used in many geosciences applications: seismology, sea ice dynamics, meteorology, etc. In the present case, the observed stop points cloud is treated as a probability measure R, and the simulated stop points cloud is seen as a probability measure S. The Wasserstein distance evaluates the similarity between the probability measures R and S:

$$W(R,S) = \inf_{\{\lambda_{i,j}\},i,j} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n'} \lambda_{i,j} \cdot c(\omega_i, \omega'_j) \mid \sum_{i=1}^{n} \lambda_{i,j} = s_j \wedge \sum_{j=1}^{n'} \lambda_{i,j} = r_i \wedge \lambda_{i,j} \geq 0 \right\} \quad (1)$$

Where R is a discrete probability measure on $(\Omega, \mathcal{F})$ and $\mathcal{F}$ is a σ-field on $\Omega = \{\omega_i\}_{i=1}^{n}$. S is a discrete probability measure on $(\Omega', \mathcal{F}')$ where $\mathcal{F}'$ is a σ-field on $\Omega' = \{\omega'_i\}_{i=1}^{n'}$. We write $R(\omega_i) = r_i$ and $S(\omega'_j) = s_j$. Let c be the cost function $c : \Omega \times \Omega' \mapsto \mathbb{R}_+$.
The Wasserstein distance is known in computer sciences as the Earth's Mover Distance. It gives the minimum cost of turning the distribution R into the distribution S. The analogy consists in seeing R as piles and S as holes. The goal is to determine the optimal transport plan $\lambda_{i,j}$ so that all the piles are moved, and all the holes are filled. The optimal transport plan depends on the chosen cost function c. In the presented methodology, piles are observed stop points and holes are forecasted stop points. The volume of each pile (resp. hole) is proportional to the number of observed (resp. simulated) stop points at a given location. The piles might be split up in order to be moved. The Wasserstein distance is the minimum average Euclidean distance (the chosen cost function) by which each observed stop point has to be moved in order to coincide with a simulated stop point area i.e. the average forecast error. The Wasserstein distance is computed with Python OT module (Flamary, 2021).

Case Study

The study site is located in Saint-Sauveur-sur-Tinée, (France, 06420), approximately 1 km South of the village. The fall of an unstable boulder overhanging the RM2265 was man triggered. The roadway had previously been covered with a 1 m-thick mattress of alluvial material (soil B) to protect the asphalt. The talus on either side of the road are rock screes (soil A) with little vegetation, with slopes ranging from 35° to 55°.
A Digital Terrain Model and an orthophotography of the study site were generated after the fall by drone images photogrammetry. A video of the rockfall was recorded. Several areas of vegetation had to be corrected. The stop points of the rocks bigger than a reference rock were identified on the orthophotography, with the help of the video. We chose to pinpoint the rocks in which a 150 mm radius disc could fit (Figure 1). The comparison is made in 2D (projection on a horizontal plane), but it could also be carried out in 3D.
Professional 3D stochastic trajectometric simulation software RocPro3D was used. Probabilistic modelling and hybrid mass options were selected. The starting zone, rocks volume, initial speed and soils positions were defined according to onsite inspections. 10^5 trajectories were simulated (Figure 1). Simulated stop points are then processed in a user-defined 2m · 2m grid (Figure 1). The default soil coefficients for "compact scree" and "loose soil" were chosen from the RocPro3D soil bank (X_0 set of coefficients in the next chapter).

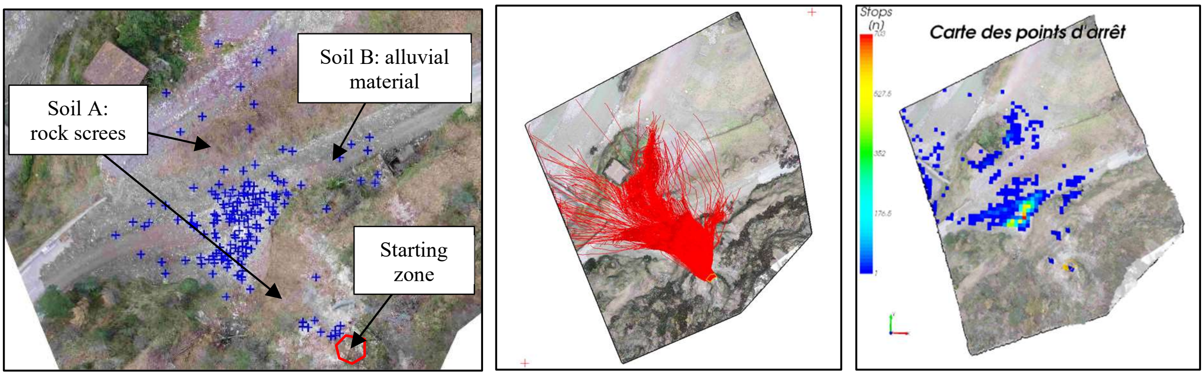

Figure 1. Left – study site: the observed stop points are marked with blue crosses.
Centre – RocPro3D: simulated trajectories – coefficients set sX_0.
Right – RocPro3D: simulated stop points map – coefficients set X_0.

We then processed the results with Python. Each blue cross (resp. red disc) is an observed (resp. simulated) stop point (Figure 2). The radius of each disc is proportional to the number of simulated stop points at a given location. The optimal transport plan $\lambda_{i,j}$ is plotted with Python Matplotlib: the blue lines thickness is proportional to the transported volume between a pile and a hole. Hence, the longer the blue lines, the higher the forecast error. The simulated stop points cloud convex hull was computed with Python Scipy (Figure 2).

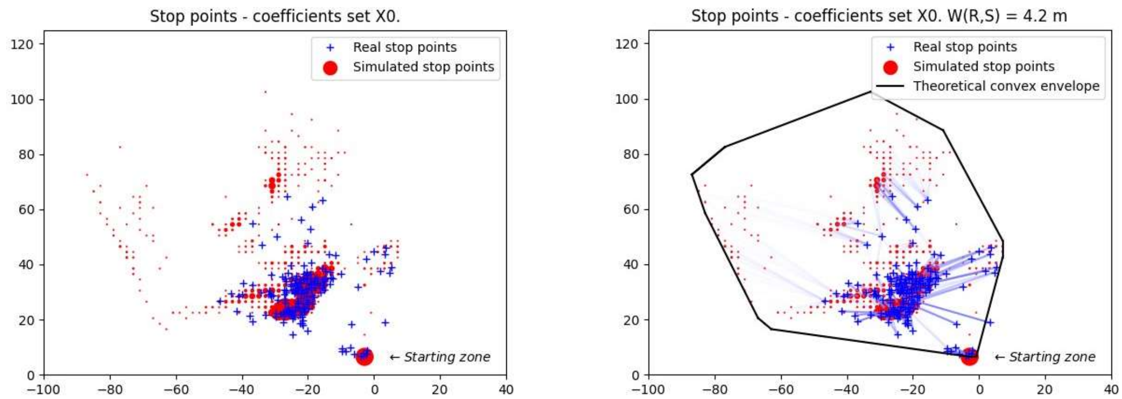

Figure 2. Left - Real and simulated stop points clouds – coefficients set X0.
Right – Optimal transport plan (blue lines) and theoretical convex envelope (black line) – coefficients set X0.

The average predicting error with coefficients set X_0 is $W(R, S) = 4.2m$. A significant proportion of real rocks stopping in the theoretical convex hull means that the simulation is not excessively optimistic.

Calibration of the parameters and conclusion

The calibration methodology consists in determining the coefficients set X* giving the lowest forecast error. We decided to optimize five parameters: the energy dissipation coefficient K – that we assumed to be the same for the two soils -, and the two restitution coefficients R_n and R_t of the soils. The best set of coefficients among the RocPro3D soils bank is X_0.
The RocPro3D simulations processed with Python OT can be seen as a Black Box function: the input is a set of coefficients, the output is a real number, there is no explicit expression nor information on the derivatives. The Black Box is optimized with the Mesh Adaptive Direct Search method implemented in Nomad program (Figure 3) (Audet, 2006). We chose to stop the optimization process after 100 steps. The variation ranges of the coefficients, that have a direct

influence on the speed convergence of the algorithm, are chosen from the literature (extreme soil coefficients: K being bound by 8m/s and 20 m/s, and R_n and R_t ranging respectively from 0.25 to 0.65 and from 0.5 to 1). The same variation ranges were chosen for both the compact soil A and the loose soil B. The optimization result is highly sensitive to the initial set of parameters. Thus, the process was initialized with the best set found by hand, with coefficients from the RocPro3D soils bank: X_0.

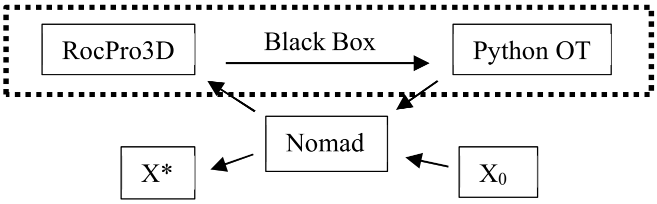

Figure 3. Coefficients calibration via the Black Box optimization process.

The best set of coefficients after 100 optimization steps is X* (average error of 3.1 m):

Table 1. Optimization of the trajectography coefficients with Nomad.

Set	**K *(m/s)***	$R_{n,A}$	$R_{t,A}$	$R_{n,B}$	$R_{t,B}$	**Wasserstein distance *(m)***
X_0	9	0.4	0.85	0.3	0.8	4.2
X*	9.80	0.59	0.73	0.38	0.71	3.1

The forecast error with coefficients set X* is more than 25% lower than the one of the best simulation with coefficients from the soil bank (X_0) (Table 1). Moreover, the optimal normal restitution coefficient is lower than the tangential one for both soils, which is what is commonly admitted in the literature, even though the variation ranges of R_n and R_t are overlapping. Finally, the algorithm automatically converges to higher coefficients for compact soil A than for loose soil B. Thus, without prior constraints on the coefficients (soil A and soil B coefficients have the same variation ranges), X* is physically relevant: the energy loss is lower on hard soils than on loose ones. The explanation is that an increase in soil B coefficients can balance a decrease in soil A ones regarding the average runout distances, but these sets of coefficients lead to irrelevant lateral dispersions, that are penalized by the Wasserstein distance.

Hence, this case study shows that the average forecast error of rock stop points can be assessed objectively with the Wasserstein distance by comparison with real rockfall events. The soil parameters can be calibrated by back analysis via the minimization of the Wasserstein distance with Mesh Adaptive Direct Search based methods (a large database is however crucial here to achieve robust calibrations). The repetition of such studies could help improving initial parameters for simulations in the future.

References

Collectif des participants à l'action A.3.1 du projet C2ROP (2020) Benchmark des approches d'analyse trajectographique par analyse comparative des simulations prédictives et d'essais de terrain. *Revue Française de Géotechnique* 163, 6.

Barbara Žabota & Milan Kobal (2020) A New Methodology for Mapping Past Rockfall Events: From Mobile Crowdsourcing to Rockfall Simulation Validation. *ISPRS International Journal of Geo-Information* 9, 514.

Charles Audet & J.E. Dennis (2006) Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization* 17, 188-217.

Rémi Flamary & al. (2021) POT: Python Optimal Transport. *Journal of Machine Learning Research*. 22, 1-8.